DYNAMICALLY PREDICTING CORRIDOR TRAVEL TIME UNDER INCIDENT CONDITIONS USING A NEURAL NETWORK APPROACH

A Thesis

by

XIAOSI ZENG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2009

Major Subject: Civil Engineering

DYNAMICALLY PREDICTING CORRIDOR TRAVEL TIME UNDER INCIDENT CONDITIONS USING A NEURAL NETWORK APPROACH

A Thesis

by

XIAOSI ZENG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Yunlong Zhang
Committee Members,	Harvey Hawkins
	Michael Sherman
	Praprut Songchitruksa
Head of Department,	John Niedzwecki

December 2009

Major Subject: Civil Engineering

ABSTRACT

Dynamically Predicting Corridor Travel Time under Incident Conditions Using a Neural Network Approach. (December 2009)

Xiaosi Zeng, B.E., South China University of Technology

Chair of Advisory Committee: Dr. Yunlong Zhang

The artificial neural network (ANN) approach has been recognized as a capable technique to model the highly complex and nonlinear problem of travel time prediction. In addition to the nonlinearity, a traffic system is also temporally and spatially dynamic. Addressing the temporal-spatial relationships of a traffic system in the context of neural networks, however, has not received much attention. Furthermore, many of the past studies have not fully explored the inclusion of incident information into the ANN model development, despite that incident might be a major source of prediction degradations. Additionally, directly deriving corridor travel times in a one-step manner raises some intractable problems, such as pairing input-target data, which have not yet been adequately discussed.

In this study, the corridor travel time prediction problem has been divided into two stages with the first stage on prediction of the segment travel time and the second stage on corridor travel time aggregation methodologies of the predicted segmental results. To address the dynamic nature of traffic system that are often under the influence of incidents, time delay neural network (TDNN), state-space neural network (SSNN),

and an extended state-space neural network (ExtSSNN) that incorporates incident inputs are evaluated for travel time prediction along with a traditional back propagation neural network (BP) and compared with baseline methods based on historical data. In the first stage, the empirical results show that the SSNN and ExtSSNN, which are both trained with Bayesian regulated Levenberg Marquardt algorithm, outperform other models. It is also concluded that the incident information is redundant to the travel time prediction problem with speed and volume data as inputs. In the second stage, the evaluations on the applications of the SSNN model to predict snapshot travel times and experienced travel times are made. The outcomes of these evaluations are satisfactory and the method is found to be practically significant in that it (1) explicitly reconstructs the temporal-spatial traffic dynamics in the model, (2) is extendable to arbitrary O-D pairs without complete retraining of the model, and (3) can be used to predict both traveler experiences and system overall conditions.

DEDICATION

This thesis is dedicated to my father, Xianqiang Zeng, and my mother, Meijuan Cai, who have been supporting and motivating me to accomplish the education away from my home country. Also this dedication is extended to Yan Zheng, who had assisted me during the inception of the thesis and who was very significant in my life.

ACKNOWLEDGEMENTS

I am indebted to my committee chair, Dr. Zhang, and my committee members, Dr. Songchitruksa, Dr. Hawkins, and Dr. Sherman for their continuous guidance and helpful recommendations along the full course of this thesis.

I would like to thank Dr. Chu and Mr. De Roche who made significant comments on the topic and some details of my thesis, and helped me gain valuable experiences during my stay in the United States. I also want to extend my gratitude to Mr. Sunkari, Dr. Balke, and other staff in the Texas Transportation Institute, for their honesty, integrity and professionalism that inspire me to pursue an engineering career. Thanks also go to my friends and the faculty in the civil engineering department for making my studies at Texas A&M University a great experience.

Finally, I owe my thanks to my parents for their selflessness, dedication and love to me.

NOMENCLATURE

ANN	Artificial Neural Network
AVI	Automated Vehicle Identification
BRLM	Bayesian Regulated Levenbergh-Marquardt Algorithm
ExtSSNN	Extended State Space Neural Network
FFBP	Feed Forward Back Propagation
GPS	Global Positioning System
HM	Historical Median Travel Time Prediction Method
LM	Levenbergh-Marquardt Training Algorithm
LGD	Learning-Generalization Dilemma
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MNN	Modular Neural Network
MSE	Mean Square Error
NP	Naïve Prediction Method
NRMSE	Normalized Root Mean Squared Error
RMSE	Root Mean Squared Error
SSNN	State Space Neural Network
TDNN	Time Delay Neural Network
TMC	Transportation Management Center

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

Page

Page

LIST OF TABLES

Page

1. INTRODUCTION

Travel time has long been regarded as one of the most important traffic variables. Travel times along different segments of a traffic network provide more meaningful information for travelers than snapshots of other traffic variables at different locations in the network. For this reason, travel time is critical for traveler information. Travel time also helps evaluate the performance of a traffic system and provide a criterion to determine the optimal route choice. It is also a very complex topic that draws a significant amount of research attention.

1.1. Background

Travel has increased steadily on the nation's transportation system, and one can hardly ignore the impacts of traffic congestions due to the expanding population and vehicular ownerships. Among all significant impacts caused by traffic congestions, travel time becomes one of the top concerns in individuals' daily lives. Travel time provides critical information that may affect travelers' decisions to choose routes, travel modes, starting times, or even cancel their trips (*1-3*). To transportation managers, travel time information can help them make better decisions on management strategies and dissemination of the optimal guidance.

Travel time of a trip becomes available only after the trip is realized (*1*); therefore prediction is necessary for better traveler information or traffic management. Nowadays, short-term travel time prediction is an essential component of an advanced traveler information system (ATIS). Traditionally, travel time prediction is generally

---

This thesis follows the style of *Transportation Research Record.*

tackled with one of the two major approaches: a simulation-based modeling approach (*2*), or a mathematical formulation-based approach (*3*). In the past decade, researchers have become more aware of the difficulties of using conventional approaches to accurately predict travel time in a dynamic traffic system. This is especially true with the occurrences of unplanned events, such as incidents, and with the recognition of the dynamic nature of travel time in a real-world environment. To meet the needs of modeling these highly complex issues, data-driven approaches have emerged from the fast development of computer technologies. The Artificial Neural Network (ANN) is one of these approaches that has gained wide recognition.

1.2. Problem Statement

Under free-flow or low flow conditions, travel time prediction can be easily made based on the traffic flow speed. If the traffic state remains stable over time as well as across space even though the volume has increased, travel time is still very predictable given available measurements and relationships of relevant traffic characteristics, such as travel time, speed, volume and occupancy. However, a complex traffic system consists of a variety of time-dependent variables and becomes highly nonlinear when volume levels are close to the design roadway capacity. Under such circumstances, the relationship between dynamic travel time and non-stationary traffic state is by no means proportional. Furthermore, the presence of incidents introduces even more complexities to the traffic system, which lowers the accuracy of travel time predictions. It is argued that the quality (accuracy) of traveler information strongly relates to the credibility of the predicted information. Drivers tend to follow up on the

information to a lesser extent once they have had bad experiences with the provided information (*4*). There is a need to provide reliable travel time information even under the impact of incidents. However, due to many difficulties such as data source, data fusion and modeling problems, very few studies have included the considerations of the presence of traffic incidents on the topic of travel time prediction (*5*). With a handful of research specifically on this topic, many of them predict corridor travel time in a one-step manner – yielding corridor predictions directly from traffic data such as speed and volume. Such methods restrict the practical application of corridor travel time prediction to a certain extent.

This research adopts ANN as a data-driven method to approach the travel time prediction problem under both incident-free and incident-affected conditions. ANN has been proven to be an effective tool in non-linear modeling and predictive type of problems. In addition, the ANN approach can assess the dynamical consequences of an incident towards trip travel time without explicitly investigating its sole impact, which is normally impossible and/or impractical. This research also investigates the application of ANN models to a two-stage process of corridor travel time prediction.

1.3. Research Significance

The principal goal of an advanced traveler information system (ATIS) is the prompt and proper dissemination of traffic information to on-road travelers. Yet in practice, the current strategies of providing travel time information remain unreliable, especially under incident conditions. As mentioned earlier, this usually results in lower

credibility of the information disseminated and further reduces the management efficiency for a traffic system.

The model development process can be used as a demonstration of how, in detail, to predict travel time under various freeway conditions by using the neural network approach. The modeling results may be integrated directly into major TMCs for the purpose of providing accurate, reliable, and real-time travel time information to assist the traffic management and control even under incident-affected conditions. By developing a practical methodology, this thesis can be used as a building block for future development of corridor travel time prediction, such as an implementation guideline, which might help traffic managers to predict corridor traffic conditions and/or traveler experiences using such a powerful, mathematically complicated, yet practical tool.

1.4. Research Objectives

The objective of this research is to develop a well-defined yet practical neural network methodology that can reliably predict corridor travel times under various traffic conditions. Goals to be achieved include the following:

- structurally and mathematically formulizing a set of neural network models that may potentially address the spatiotemporal characteristics under any normal traffic conditions,
- developing a neural network model that can incorporate incident information in an attempt to capture the impacts of unplanned incident events on travel time,

- investigating the prediction performances of these neural models when feeding with and without incident information under abnormal traffic conditions, and
- developing a dynamic prediction method to predict the corridor travel time as is most likely to be experienced by travelers.

2. TRAVEL TIME PREDICTION USING NEURAL NETWORKS

As the main focus of this thesis, this section is organized to provide firstly an overview of artificial neural networks in general. Then a literature review specifically on the topic of neural network application to travel time prediction follows.

2.1. Artificial Neural Networks

The Artificial Neural Network (ANN), also commonly referred to as "neural network," is a massively parallel distributed processor that utilizes experiential knowledge to build up the abstract representation of a system or an object. The concept of ANN stems from the recognition of the structure of the human brain, which is literally a highly complex, nonlinear, and parallel information-processing system (*6*). ANN resembles the sophisticated human brain in two perspectives: (a) the network acquires knowledge from its environment through a learning process, and (b) the interneuron connections that vary in connection strength, known as "synaptic weights" in neurobiology, provide an analogous mechanism that is used to store the acquired knowledge. With the learned knowledge, an ANN can autonomously perform a number of tasks, such as pattern recognitions, pattern associations, function approximations and filtering, to just name a few.

2.1.1. A Neuron Model

Any ANN consists of a number of structural constituents, termed as neurons, which can provide elementary nonlinear computations. Figure 1 exemplifies a neuron model. The computation of a neuron includes two typical processes: (a) receives and

sums input signals; and (b) transforms the summation of the inputs through a transfer function to produce an output of the neuron.

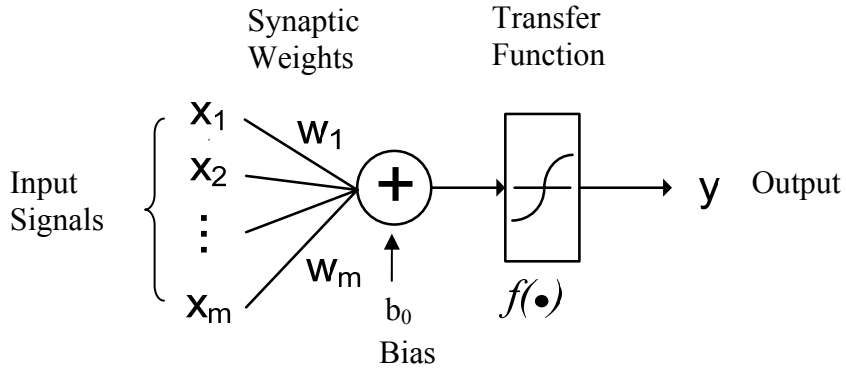


Figure 1: Model of a neuron.

We may describe a neuron in mathematical terms using the following equation:

$$y = f(\sum_{j=1}^{m} w_j x_j + b_0) \tag{1}$$

where x_j is the j-th input data from a total of m inputs; w_1, w_2, …, w_m are the synaptic weights that connect the m input signals to the computation neuron; b_0 is termed the bias and is an external parameter of the neuron that applies an affine transformation to the output of the summing junction in the model of Figure 1; $f(\cdot)$ is the transfer unit, or activation function, which takes the result of the linear combiner as the argument of a differentiable function to produce the final outcome of the neuron model.

Multiple neurons, as building blocks, are correlatively connected in series via synaptic weights and structured in parallel to form a sophisticated interconnected neural network. Therefore, the computational power of such a system has been boosted exponentially and is capable of handling high-dimensional and non-linear problems.

2.1.2. Neuron Learning

The primary significance of a neural network is the ability to learn from its surroundings and to improve its performance through learning (*6*). Since the very inception of ANN development, a set of learning rules have been explored and defined. Error-correction mechanism is a classic feedback control mechanism in the field of signal processing. There are at least two learning modes – supervised and unsupervised. The supervised learning algorithm is learning with known outputs (target), while in the unsupervised mode no targets are used for the networks to be compared with and the networks learn from their current and past behaviors. In the context of neural network prediction on travel time, the input-output paradigms are crucial to achieve accurate prediction ability. Therefore, all the networks to be developed are trained in the supervised mode.

2.1.3. Benefits of Neural Networks

ANN has a number of advantages in terms of capabilities of task-performing. It is generally accepted that proper constructions of ANN architectures theoretically allow approximations of any nonlinear mappings to arbitrary accuracies (*5, 7*). By being randomly presented with unique input data and corresponding response data, a network learns, in supervised modes, to create input-output mappings in a statistical manner without prior assumptions (*6*). Such a process brings to mind the study of nonparametric statistical inference. In addition, the plasticity of a neural network allows its synaptic weights to be adjusted in real time in order to stay adaptive to a non-stationary environment. Last but not least, the parallel distributed computation system possesses a

great engineering characteristic – fault tolerance. This trait minimizes the vulnerability of potential failures due to noisy or even false inputs to a certain extent. All the aforementioned features have popularized the neural networks in many different fields of studies, and the ANNs therefore are considered suitable in studying the dynamics of traffic processes.

2.2. Travel Time Prediction

As mentioned before, a trip travel time would not be available until after the trip is completed. This fact implies the travel time information would be useful for individual travelers only if it is predicted. In the literature, travel time prediction is deemed by most as a highly complex and dynamic problem, as travel times are the result of complex nonlinear interactions of heterogeneous groups of driver-vehicle combinations (*1*). Furthermore, exogenous factors, such as changes in weather and occurrences of abnormal events in or nearby the roadway, worsen the predictability of the nonstationary traffic system.

2.2.1. An Overview of Available Methodologies

To tackle the issue of the prediction of travel time, a large amount of studies have been conducted. Traditionally, developing a travel time prediction model often follows one of the two approaches (*7*): analytical and empirical.

The analytical approach applies the methods of summary and analysis to making predictions of outputs as functions of specified inputs (*8*). Models based on the fundamental theories of traffic flow are often developed to establish relationships between common traffic variables (e.g. speed and volume) and travel time. Well-

developed methodologies often present a good representation of the traffic process and, therefore, yield simple equations and satisfactory results under certain conditions. One major deficiency of this approach often lies in its deterministic and simplistic natures which jointly deteriorate its ability to realize highly complex and dynamic problems. To enhance the ability of analytical approaches, not only one but a set of predefined models can be organized in a way that the outcomes of these models interact internally.

The empirical approach is often based upon a set of field observations obtained on a case-by-case basis and is usually considered as a reverse process of the analytical method. The fundamental part of this approach is that all models and theories are results of observations rather than *a prior* reasoning or purely mathematical deduction (*9*). Such a method is therefore termed as empirical approach. In transportation application, the primary merit of the empirical approach is that the burden of reasonable assumptions on constantly changing traffic conditions has been eased in the process of model development. Regression models and neural network approaches fall into this category.

Additionally, simulation methods have been adopted for travel time prediction in recent years. Macroscopic simulation models often lack the modeling complexity to address dynamic traffic behaviors especially when there are abnormal conditions such as incidents. On the other hand, microscopic simulation models require significant amount of data input and calibration to account for dynamic traffic conditions is always a challenge.

This thesis focuses on traffic prediction using neural networks that can model dynamic, non-linear traffic systems that are too complex to be described by analytical

methods or empirical rules. The following two sections are presented to provide an extensive literature review on travel time prediction using artificial neural networks.

2.2.2. Travel Time Prediction with Neural Networks

Travel time is the product of highly dynamic and nonlinear traffic processes over space and time (*10*). Simplistic abstraction of such a complicated relationship between travel time and other traffic variables is very much a mathematically challenging and assumption-demanding task, if not impossible. Therefore, many researchers turn their focuses onto how they can extract the relationships from observatory data. Driven by field data, the neural network approach is an advanced and intelligent method available for this problem. Table 1 lists recent studies of short-term travel time forecasting based specifically upon the data-driven technique. Various types of neural networks have been developed. It is demonstrated that many of these networks had achieved satisfactory prediction performance under certain conditions.

Comparing the studies in the table, many (*1, 4, 11-13*) did not use the actual travel time captured by probe vehicle devices (e.g. AVI, GPS), but used estimated or simulated travel times. Training a neural network with estimated travel time rather than the real ones implies the network may be learning the wrong mechanism (*1*). Secondly, many of the studies (*4, 5, 13-16*) have not applied static networks to the traffic systems which are inherently dynamic. In this study, all models learn the "ground truth" travel time data collected from Automated Vehicle Identification (AVI) system, and the dynamic structure of the neural network is designed in the model development process.

Table 1: Overview of Major Researches on Travel Time Prediction Using Artificial Neural Network Approach

	Experiment Setups							Analysis	
Year	Research Team	Network Architecture	Data Source	Inputs	Study Period	Time step size	Network Type	Consideration of Incident	Performance Index
1998	Park and Rilett (*16*)	Modular	AVI	Travel Time	6:00-10:00am of 231 weekdays	5 to 25 minutes	Static	Not considered	MAPE<12%
1999	Palacharla and Nelson (*12*)	Fuzzified Feed Forward	Vehicle Detectors	Volume and occupancy	-	15 minutes	Static	Not considered	RMSE<5(sec)
2001	Rilett and Park (*15*)	Spectral-Basis	AVI	Space Mean Speed,	203 weekdays	5 to 25 minutes	Static	Not Considered	MAPE <20%
2002	Krikke (*13*)	Feed Forward Back Prorogation	Vehicle Detectors	Speed, Volume and Length of congestion	24 hours a day during 3 days	1 minute	Static	Not considered	RMSE<1(sec)
2002	Van Lint and Hoogendoorn (*4*)	Recurrent	Simulation	Speed and Volume	14:00-21:00 of 5 days	-	Dynamic	Not considered	RMSE<17(sec) SSE $<1.2*10^5(sec^2)$
2006	Van Lint (*1*)	State-Space	Vehicle Detectors & Estimation	Speed and Volume	14:00-19:45 of 1071 days	1 minute	Dynamic	Accident	MAPE <5.4%
2006	Ran et al. (*5*)	Multi-Layer Perceptron and its variations	-	Incident, Traffic, Weather	15 minutes after incident of 2578 incidents	15 minutes	Static	Accident	RMSE<2.2(min) MAE<1.42(min)
2008	Wei and Lee (*14*)	Feed Forward Back Propagation	GPS, Vehicle Detectors, Incident	Travel Time, Speed, and Occupancy Time and Incident info.	24 hours a day during 9 days	5 minutes	Static	Accident	MAPE<20%
2009	Zou et al. (*11*)	Multi-Topology Network with a Clustering Function	Estimated from a Video-Matching Algorithm	Real-Time and Historical Occupancy, Traffic Counts; Current Time of Day	24hours a day during a 10 week period	-	Dynamic	Not Considered	MAPE<7%

Note: "-" indicates that descriptions are absent in the paper.

Travel time results from the spatiotemporal evolution of prevalent traffic conditions, which is a dynamic process with respect to time and space. Park and Rilett (*16*) incorporated upstream and downstream traffic data during the current time interval as inputs in the prediction; Wei and Lee (*14*) further considered traffic data from a previous time step. Van Lint (*4*) absorbed the considerations of both studies and trained

the network in a dynamic sequence and this network training style better captured the sequential effects of consecutive time steps. Yet Zou et al. (*11*) applied artificial neural network to conduct field experiments on travel time prediction on I-70 in Maryland, and found out that such an approach was accurate but fell shorts on the days affected by incidents. Ran (*5*) and Wei (*14*) both argued that incident inputs are significant pieces of information that affects the performance of prediction under incident conditions. On the other hand, Lint considered that speed and volume serve as the best representations of traffic state where no other inputs are needed (*4*). In light of these efforts, the recurrent network architecture proposed by Van Lint will be modified for explicit investigation the potential impacts of the presence of incident information.

3. MODELING TRAVEL TIME WITH NEURAL NETWORKS

This section initially discusses a two-stage prediction process for corridor travel time and how neural network application can be incorporated. The concepts and the methodologies of segment travel time prediction (i.e. step one of the two-stage prediction) using various neural network models then follow. Finally, based on the results of the first step, the corridor travel time prediction method (i.e. step two of the two –stage process) is developed and described in the last sub-section.

3.1. Two-Stage Prediction Method

Most ATIS implementations calculate route travel time by summing the travel times of all segments along the route (*15*). However, in the context of neural network application to corridor travel time predictions, much of the research work uses various input sources to predict corridor travel time directly (*1, 13, 17*). Direct forecasting using an ANN model is straightforward, yet some issues have to be addressed in order for results to be valid.

First of all, due to the nature of the one-stage prediction process, the corridor travel times are used as the direct targets in ANN trainings. Nonetheless, getting the correct inputs to match with these target outputs is very complicated and sometimes impractical. To explain this difficulty, let us assume that a mean corridor travel time is obtained as t_c at time T. Since all trips are subject to constantly changing traffic conditions, the corridor travel time prediction inevitably involves the issue of how to account for such variants while training the neural networks. Such corridor travel time is a collective result of all the corridor traffic conditions during the immediate past period

of (T-t_c). As far as neural network training are concerned, input data should be relevant with respect to the same period in order for the networks to learn the true relationships. Of course, t_c is a random variable structurally characterized by the prevailing traffic conditions that are constantly changing. This implies that the time intervals to which the input data should be referenced are indefinite. In other words, it is very difficult to set the correct time window to retrieve input data. Conceivably, as the average time to travel through the corridor lengthens (e.g. longer corridor stretch) and/or the variance increases (e.g. more fluctuated traffic conditions), the errors caused by the improper pairing of inputs and outputs aggravate.

In addition to the difficulty in matching correct input and output data, two more problems are identified while applying the one-stage prediction method. First, as a longer corridor is investigated, the sample size of the corridor travel times extracted from AVI stations (by pairing AVI stations at the beginning and the end of the corridor) become smaller. At a certain point, the samples are too small to yield representative measurements of the corridor travel times. To evaluate an extended corridor, GPS-devised vehicles or other methods need to be considered to enable valid extraction of corridor travel times. Secondly, complete retraining of the neural network models is required when the configurations of the corridor are changed (e.g. adding new segments to the corridor). This effort is usually not trivial so that the flexibility of the one-stage prediction method is seriously limited.

Due to these difficulties when applying the one-stage prediction method, we propose a two-stage prediction method to model corridor travel time as shown in Figure

2. The first stage involves the neural network designing and model calibration for segment travel time prediction. Building on the results of the first stage, the second stage puts forward different pre-defined calculation procedures and predicts the corridor travel time accordingly.

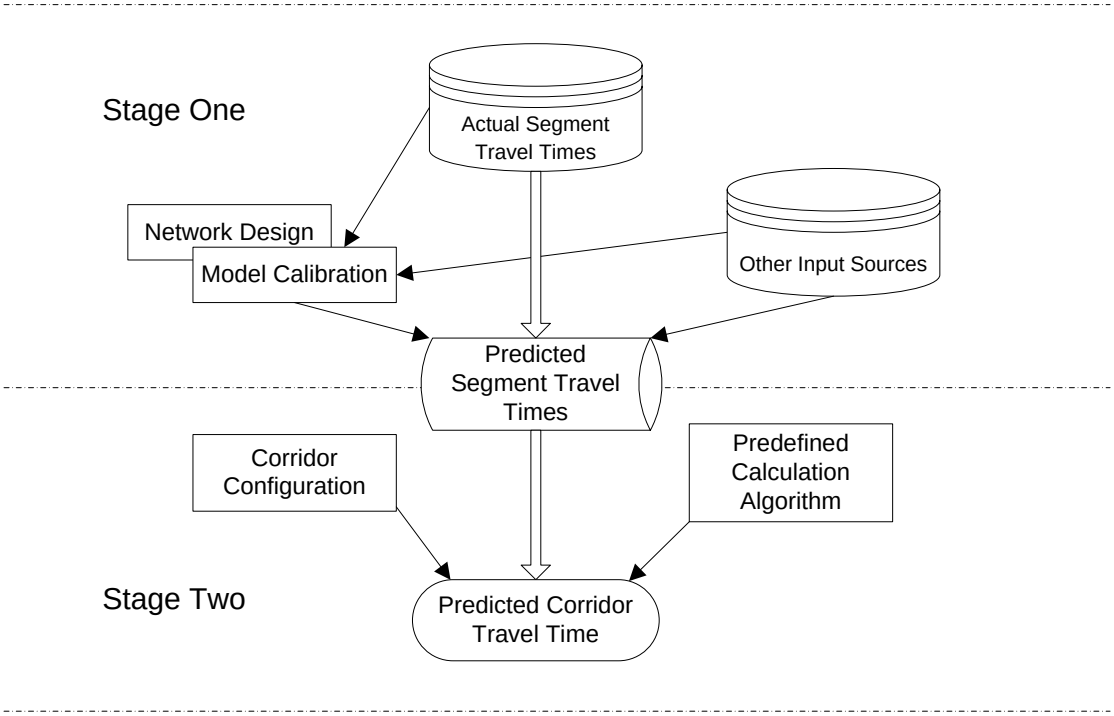


Figure 2: Generic routine of the two-stage prediction method.

The two-stage prediction relaxes the requirement on obtaining corridor travel times, and it only requires the segment travel times to be obtained as target outputs. Since freeway segments are typically much shorter, the pairing of input and outputs become less erroneous and the sample size is usually no longer an issue. Furthermore, the first step of the two-stage method can be used as a building block so that the second step of modeling is flexible to be extended to various corridor configurations. As a plus,

different measures for corridor traffic conditions can be derived (see section 3.3.2 for discussion) based on such a method.

3.2. Segment Travel Time Modeling

A traffic system is a time-varying system that is resulted from time-dependant nonlinear interactions of heterogeneous groups of driver-vehicle combinations. However, at short time and distance scales, it is reasonable to assume the system condition remains constant. Therefore, dividing a freeway corridor into an infinite number of infinitesimal segments and taking snapshots of each segment would be the most accurate way to reconstruct traffic conditions. Such reconstructions of traffic conditions are crucial in boosting the performance of travel time predictions. Unfortunately, this method is eventually bounded by the practical constraints, such as data collection and processing costs. The method of partitioning a freeway corridor is not within the scope of this study and has been provided with some insights by Wei and Lee (*14*). In this study, the smallest length of segment from which we can build our model is constrained by the length of the AVI segments. As such, the link or segment travel time being analyzed in the thesis refers to the time needed to travel through a freeway section enclosed by the two AVI stations subsequently located in one direction of the freeway corridor. With segments being identified, in this first step of the two-stage travel time prediction method, we scrutinize various neural network models and associated network training algorithms to identify the model(s) that can appropriately recognize a variety of traffic states, including the conditions of an incident.

3.2.1. Concepts of Neural Network Models

To build a travel time model for each link along a freeway corridor, the key is to reconstruct the spatial-temporal relationships between traffic conditions and segment travel time. Five neural network models are identified as the candidates that could suit this objective. This list includes the classic back propagation neural network, and other networks suitable for predictions in dynamic systems.

3.2.1.1. Back Propagation Neural Network

The Back Propagation Neural Network (BPNN) is the most classic neural network that consists of one input layer, one output layer, one or more hidden layers and connecting weights that correlate these layers. The BPNN is trained with errors that propagate from output layer back to input layer and is therefore named (*6*). The hidden layers give the network the power to model complex problems. Additionally, the model can be easily formulated mathematically and implemented. However, it provides no regards on the relations among temporal attributes (i.e. time series related inputs). Figure 3 illustrates a typical structure of the BPNN which consists of two hidden layers, one input and one output layers.

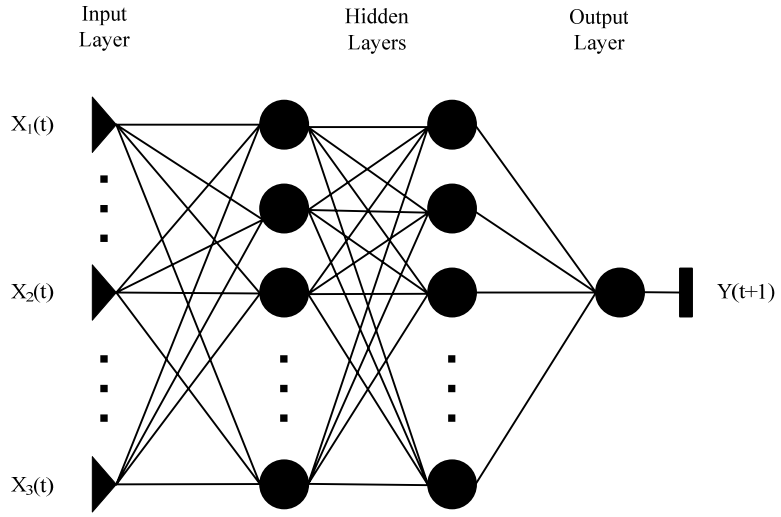


Figure 3: Schematic overview of BPNN.

3.2.1.2. Time Delayed Neural Network

The Time Delayed Neural Network model (TDNN) is essentially a BPNN model with a time delay component that allows the network to obtain inputs from previous time intervals (*6*). It introduces a dynamic characteristic into the typical BPNN model while maintains a low level of model complexity. The white-colored unit in Figure 4 is the delay component that stores inputs from the environment at one or multiple time steps prior. It then feeds the stored information to the network along with the most current inputs. The delay unit acts as a memory function that captures temporal information contained in the input signal and such information is embedded in the synaptic weights $\{w_k(l)\}_{l=1}^{t-p}$ of the delay unit. The output, $y(t)$, of the TDNN in response to the current input x_t, and a number of p past input values, $x(t\text{-}1)$, $x(t\text{-}2)$, … $x(t\text{-}p)$, is given by

$$y(t) = \sum_{k=1}^{N_K} w_k f(\sum_{l=0}^{p} w_k(l)x(t-l) + b_k) + b_0 \quad (2)$$

where the output transfer function is normally assumed to be linear and thus omitted; the synaptic weights of the output neuron $y(t)$ is defined by the set $\{w_k\}_{k=1}^{m}$,which connects N_K hidden neurons; b_k and b_0 are the biases of the hidden neurons and the output neuron. It should be noted that more than one hidden layer is allowed in this type of network although only one is shown Figure 4.

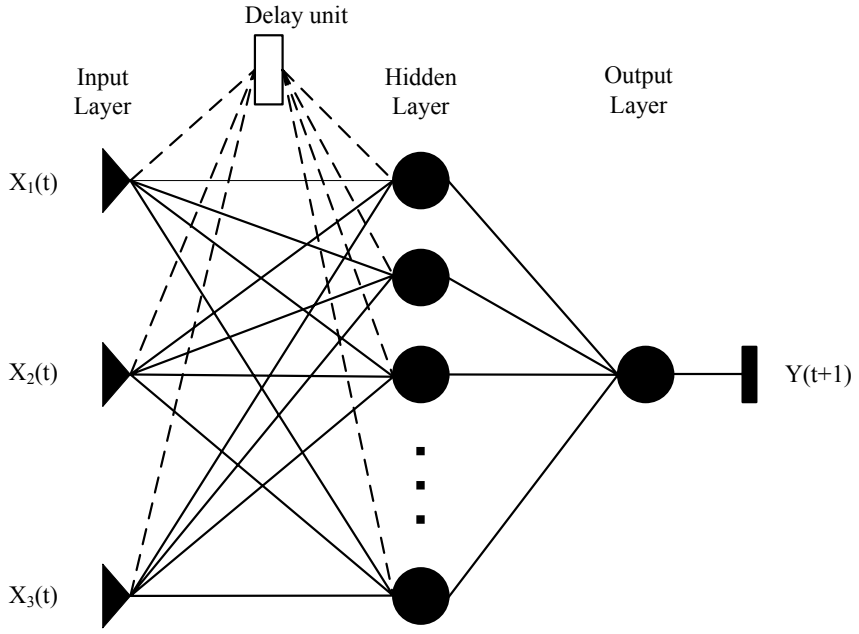


Figure 4: Schematic overview of TDNN.

3.2.1.3. Modular Neural Network

Instead of connecting multiple layers in series, Figure 5 illustrates the Modular Neural Network (MNN) that processes inputs by using several layers connected in parallel. With this architecture, data processing is partitioned within the network, and

each module (division) learns some specific aspects of the problem presented. This theory arises from the concept of "divide-and-conquer" (*5*), where complex problems are difficult to analyze with only one model but may be easier with an ensemble of models.

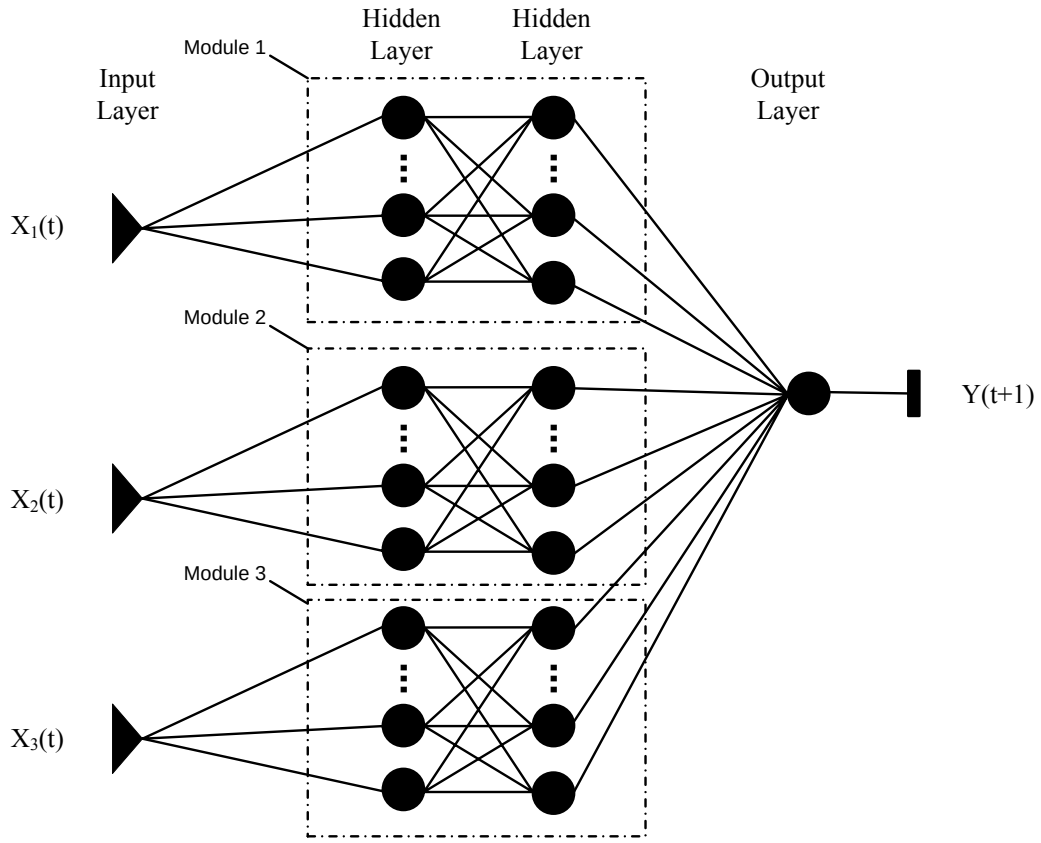


Figure 5: Schematic overview of MNN.

3.2.1.4. State Space Neural Network

Recurrent neural networks are designed to learn sequential or time-varying patterns(*17*), and have been an important focus of research and development in the last twenty years. The State Space Neural Network (SSNN) is a special type of recurrent neural networks and is claimed to be analogous to traffic simulation models in which

traffic states evolve based upon their previous internal states (*1, 5, 7*). According to Haykin, the internal state is defined as a set of quantities that summarizes all the information about the past behavior of the system that is needed to uniquely describe its future behavior, in addition to the purely external effects arising from the applied inputs (*6*).

This type of model is a special type of recurrent network. The model feeds the output of a hidden layer at the current time step to the input layer as an additional input at the next time step. The output of the hidden layer is a rough representation of the internal state of the underlying problem. The feedback process has to firstly store the information of internal state at the last time step in the context layer, which consists of doted connections and white-colored units in Figure 6, then feed the information back to the network for learning at current time step. The dynamic behavior of the model in Figure 6 may be described by a pair of coupled equations in matrix form:

$$x_k(t+1) = f(\sum_{k=1}^{N_K} w_k^i(t)x_k(t) + \sum_{j=1}^{N_J} w_{jk}^c(t)s_j(t) + b_k) \tag{3}$$

$$y(t) = f(\sum_{k=1}^{N_K} w_k^o(t)x_k(t) + b_0) \tag{4}$$

where $s_j(t)$ is the parameter value of context unit j that partially defines the internal states of the network at time t; N_J is the number of context units; w_k^i, w_{jk}^c and w_k^o denotes the weight connections of the k-th hidden neuron to the input, context and output layer respectively; b_k is the bias value for hidden neuron k but that for context unit k is usually omitted; The subtle difference between TDNN and SSNN is appreciated by the fact that

such a mechanism of SSNN is capable of taking into consideration the dynamic evolvement of the network state over time. Another difference is that SSNN may have only one hidden layer while TDNN may be designed to contain more.

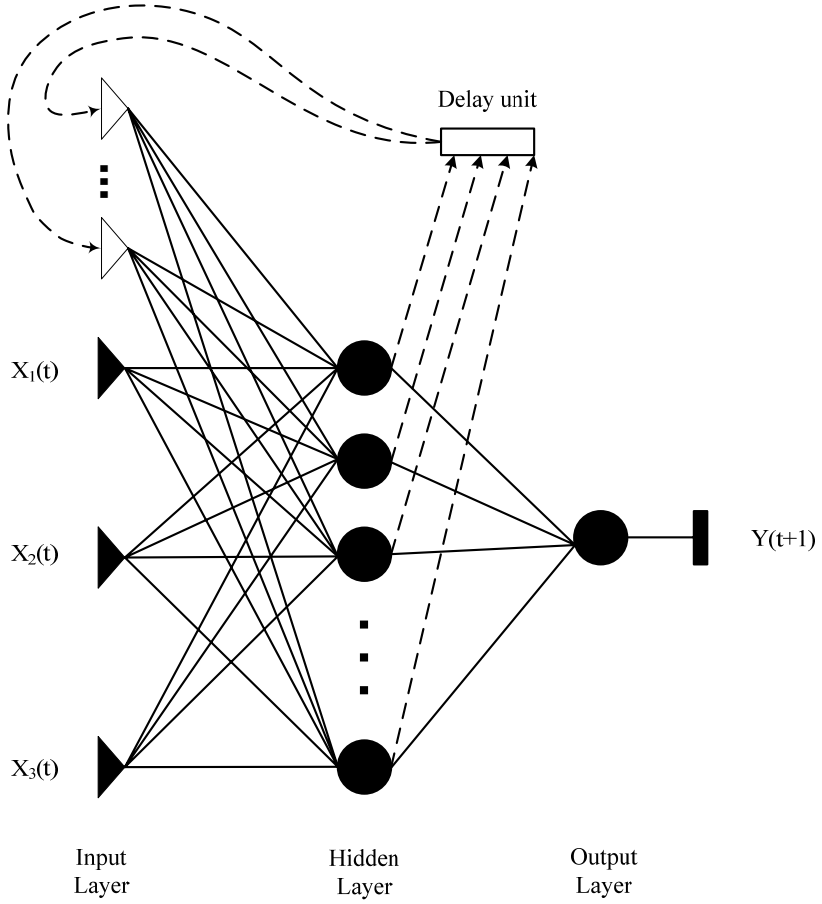


Figure 6: Schematic overview of SSNN.

3.2.1.5. Extended State Space Neural Network

There were observations in the preliminary analyses of this study showing that segment travel time modeling is not adequate in extreme unstable traffic conditions (i.e. incident conditions) when using common types of neural network models. Therefore, the

Extended State Space Neural Network (ExtSSNN) is proposed in this study in an attempt to realize the impact of an incident. The ExtSSNN assumes that the additional travel time or delay caused by incidents is a function of various incident properties. Hence, the incident impact should be learned partially and separately, with sufficient information of the particular incident. In light of the concept of "divide-and-conquer," the ExtSSNN is a SSNN with one additional neural network module that is specialized to capture the additional incident impacts on traffic states while SSNN itself is specialized to learn the evolution of stable traffic states. Based on equation (3) and (4), the network can be modified to incorporate the impacts of incidents by the following equations:

$$x_k^1(t+1) = f\left(\sum_{k=1}^{N_K^1} w_k^{i1}(t)x_k^1(t) + \sum_{j=1}^{N_J^1} w_{jk}^{c1}(t)s_j^1(t) + b_k^1\right) \tag{5}$$

$$x_k^2(t+1) = f\left(\sum_{k=1}^{N_K^2} w_k^{i2}(t)x_k^2(t) + b_k^2\right) \tag{6}$$

$$y(t) = f\left(\sum_{k=1}^{N_K^1} w_k^{o1}(t)x_k^1(t) + \sum_{k=1}^{N_K^2} w_k^{o2}(t)x_k^2(t) + b_0\right) \tag{7}$$

where notations are similar to those defined in equation (3) and (4) except superscripts being added to distinguish module 1 from module 2 as shown in Figure 7; in particular, inputs to module 2, given by the set $\{x_k^2\}_{k=1}^{N_{K2}}$, are the vector of incident inputs determined by the characteristics of the incidents in question; note that there are no recurrent connection in this case and context layer for module 2 is omitted in equation (6) as comparing to (5). This type of neural network assumes the incident input would provide

significant impacts on the outcome of the prediction. However, it is the intention of this study to investigate the significance of this module.

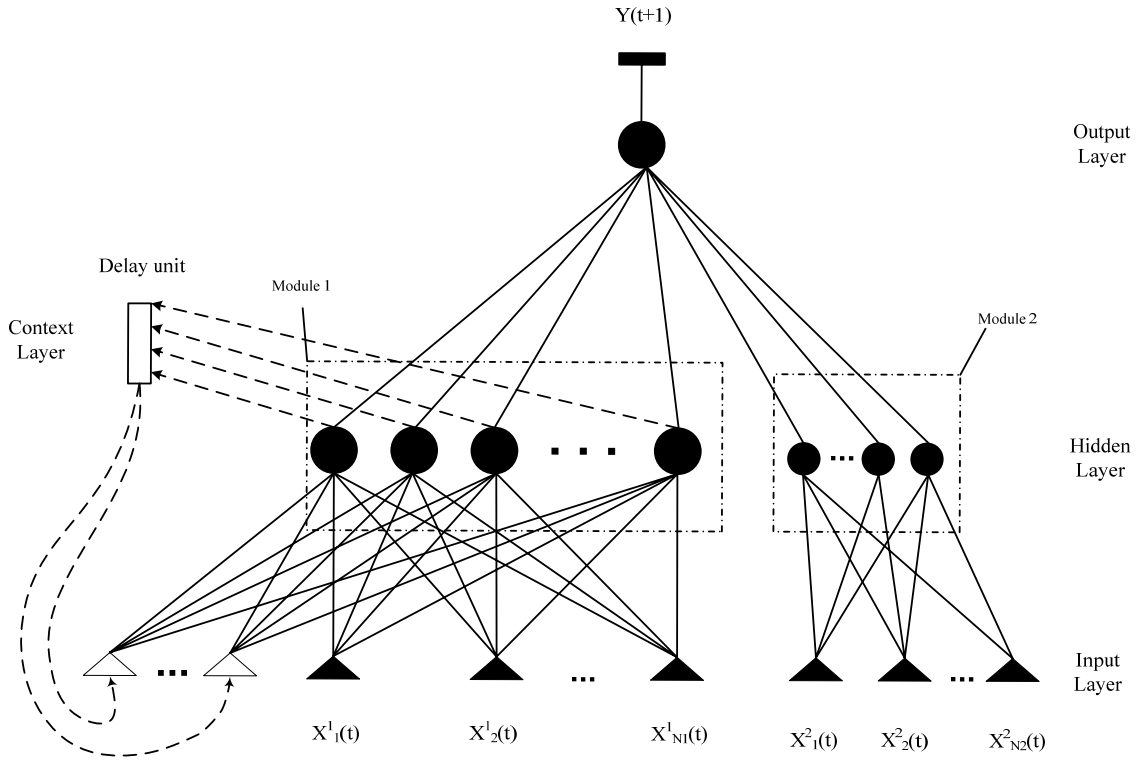


Figure 7: Schematic overview of ExtSSNN.

3.2.2. Classification of Neural Network Structure

Depending on the criteria, neural networks can be categorized into many different classes. This study concerns whether a delay or self-feedback mechanism exists in the structure of a neural network. If a neural network model contains either delay or feedback units in its structure, it is classified as a dynamic neural network; if a neural network includes no such units, it is deemed as a static network. The prototypical use of a static neural network is in structural pattern recognition whereas a dynamic network

recognizes the processing of temporal patterns that evolve over time (*6*). In general, dynamic neural networks are more powerful than static ones in that they have memory about the history of either the input data or the network connection weights and biases (*18*). Such a memory capability enables the realization of time-variant patterns such as that of a typical traffic system. Nonetheless, the primary disadvantage of the dynamic network structure manifests in the model complexity and the training inefficiency. Therefore, simpler structures of static networks can be adopted when the underlying problems, such as free-flow conditions, to be modeled are relatively straightforward.

In the five neural network models proposed in the previous section, FFBP and MNN models are static network while TDNN, SSNN and ExtSSNN are dynamic networks.

3.2.3. Training of the Neural Network

Neural network training can be classified as batch or incremental. In a batch training scenario, all the input and output sample pairs are presented to the network at the same time and the errors between targets and network outputs are derived. Gradients of these errors with respect to network weights can be subsequently computed to find the direction of the minimization of the error function. In an incremental mode, the network parameters are adjusted according to the maximum error gradients on the error surface when, each time, one sample of the input-output pairs is fed to the network. Conceivably, the incremental training mode learns a problem in a more refined yet computationally demanding manner.

In addition to what mode is to be used in training a neural network, the training algorithm is equally crucial in order for the network to learn effectively and efficiently. The following part of this section will focus on two training algorithms: the Levenberg Marquardt (LM) algorithm and the Bayesian-Regulated Levenberg Marquardt (BRLM) algorithm.

3.2.3.1. Levenberg Marquardt Algorithm

In neural network training, the back-propagation algorithm has gained its reputation by its ability to minimize the errors a network would make when learning from desired outputs. Numerical optimization techniques have been attempted to speed up the convergence of back-propagation algorithm, and the Levenberg Marquardt algorithm showcases its outstanding ability to achieve this objective.

The LM algorithm is in fact an approximation to Gauss-Newton's method (*19*). For a detailed formulation of Newton's method in network training, see (*6*). For Gauss-Newton's method, the network parameter update rule reads

$$\Delta w = (\boldsymbol{J}^T \boldsymbol{J})^{-1} \boldsymbol{J}^T \boldsymbol{e} \tag{8}$$

where $\boldsymbol{J}$ is the Jacobian matrix:

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \cdots & \frac{\partial e_1}{\partial w_\psi} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_N}{\partial w_1} & \cdots & \frac{\partial e_N}{\partial w_\psi} \end{bmatrix} \tag{9}$$

and $\boldsymbol{e}$ is the error vector. Marquardt (*20*) modifies the update rule by introducing a hyper-parameter μ that functions as a momentum factor which could be re-parameterized

to adjust the learning rate of the network. The Marquardt-modified update rule is as follows

$$\Delta w = (\boldsymbol{J}^T \boldsymbol{J} + \mu \boldsymbol{I})^{-1} \boldsymbol{J}^T \boldsymbol{e} \tag{10}$$

The symbol $\boldsymbol{I}$ denotes an identity matrix with the same dimension as $\boldsymbol{J}^T\boldsymbol{J}$. With this modification, μ dictates how much the Marquardt-adjustment of network updates is different from the Newton-adjustment. When the performance index (e.g. sums of squared errors) of the training is decreased due to the weight updates, the parameter μ is reduced by a factor η; otherwise, multiply μ by η. The Levenberg-Marquardt algorithm has been proven to be efficient and quick to converge even for large size networks. For relevant implementation issues, see (*17, 21, 22*).

However, the algorithm is not recommended for ANNs with delay components in that LM algorithm only approximates the performance gradient and has large-step update rates (*18*). To compensate the deficiency, Bayesian-regulated LM algorithm is introduced as a more reliable and robust training technique.

3.2.3.2. Bayesian-Regulated Levenberg Marquardt Algorithm

As one of the most encountered issues during the process of neural network training, generalization plays a vital role in determining the robustness of the trained model. Superior generalization ability indicates a better chance that the network circumvents the noisiness in the data and learns the true underlying distribution of the system.

An over-complex model tends to over-fit the data and generalizes poorly (*23*). As illustrated in Figure 8, the best-fit a model can achieve becomes better as the model

complexity (determined by the number of free parameters) increases. However, the generalization performance of the model has an optimal point which does not necessarily increase as the model becomes more complex. The tradeoff point between training precisions and testing precisions are often hard to identify. The phenomenon is therefore termed herein as Learning-Generalization Dilemma (LGD).

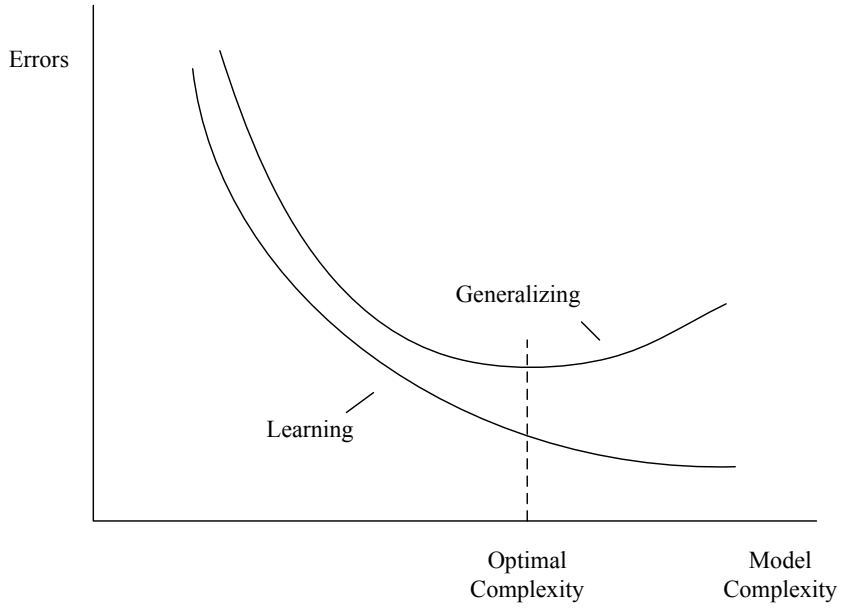


Figure 8: Relationship of over-fitting and model complexity.

To improve the generalization of a neural network model, regularization is one method that serves this purpose by controlling the model complexity. As such, we adopt the Bayesian framework proposed by Bishop (*21*). The framework is an automated process that determines optimal regularization parameters (i.e. weights and biases) that balances the network learning and generalization. The framework employs the Bayesian regulated Levenbergh-Marquardt (BRLM) algorithm, which is described in a number of

works (*18, 23, 24*). The BRLM algorithm aims at minimize the cost function regularized by the network internal parameters – model error (*E*) and model weights (*W*). The cost function reads

$$C = \beta \cdot E + \alpha \cdot W \tag{11}$$

where *E* and *W* can take various forms to represent the model error and weight values, but Mackay (*23*) was able to show that the energy form of the two terms can lead to easier derivation of α and β without the loss of generality. Hence,

$$E = \frac{1}{2}\sum_{t=1}^{N_T}(e_t)^2 = \frac{1}{2}\sum_{t=1}^{N_T}(u_t - y_t)^2 \tag{12}$$

and

$$W = \frac{1}{2}\sum_{i}^{N_L}\sum_{j}^{Q_i}(w_{ij})^2 \tag{13}$$

are written as sums of model errors and sums of squared weights respectively, where e_t is the error term between the predicted output y_t and target output u_t at time instant t; N_T denotes the number of time instants within the sequence of a training data sample; N_L is the number of layers (both hidden layers and context layers in SSNN) while Q_i is the number of elements in the weight vectors of the i-th layer. In equation (11), α and β are the hyper-parameters that regulates the balance between minimizing *E* and *W*. In addition to the common error term, the introduction of the second term (i.e. sums of squared weights) roots on the notion that larger weights make the model more sensitive and increase the risk of over-fitting the training data (*10*).

MacKay (*23*) argues that minimizing function *C* is equivalent to maximizing the posterior probability distribution of weight vectors given the observed dataset *D* (e.g. a pair of input-output mapping ==x_t ,u_t) and some prior probability of a given model assumption *H*. we can apply Bayes' rule for the posterior probability distribution of weights:

$$P(w \mid D, \alpha, \beta, H) = \frac{P(D \mid w, \beta, H) P(w \mid \alpha, H)}{P(D \mid \alpha, \beta, H)} \tag{14}$$

where $P(D|w,\beta,H)$ denotes likelihood function of such data to be observed given such network parameters and model assumption; $P(w|\alpha,H)$ is the prior probability and $P(D|\alpha,\beta,H)$ is a normalizing constant that represents the experiential evidence for the model in study.

However, neither α nor β is known *a priori* but can be adjusted while minimizing the cost function (11). Mackay (*25*) interprets $1/\alpha$ and $1/\beta$ as the estimations of the variances of the Gaussian distributions from which weights and output errors are drawn. Therefore, the hyper-parameter α regulates how close a model should learn to fit the data, and it thus determines the simplest settings of weight *w* that could fit the data to the desirable degree. This process naturally embodies the Occam's Razor problem, which states a preference for simple models (*23*). Mackay further shows that α and β can be estimated using maximum likelihood method as follows:

$$\alpha = \frac{\gamma}{W} \text{ and } \beta = \frac{N_T - \gamma}{E} \tag{15}$$

where

$$\gamma = \psi - \alpha \cdot trace(A^{-1}) \tag{16}$$

and A can be expressed as the weighted sum of the Hessians of the output errors and the network weights, shown as follows:

$$A = \beta \frac{\partial^2 E}{\partial w^2} + \alpha \frac{\partial^2 W}{\partial w^2} \tag{17}$$

In equation (16), γ denotes the number of good parameter measurements and has value between 0 and ψ (the size of weight vectors). Often, γ implies the simplest network setting that is still warranted by the data D.

To incorporate the Bayesian hyper-parameters into LM algorithm, we can modify equation (10) as

$$\Delta w = (\beta \boldsymbol{J}^T \boldsymbol{J} + (\mu + \alpha)\boldsymbol{I})^{-1} (\boldsymbol{J}^T \boldsymbol{e} + \alpha w) \tag{18}$$

As network weights being adjusted at each batch of N_k inputs and outputs, the Bayesian hyper-parameters and the Marquardt hyper-parameter are simultaneously updated according to rules described above.

Ideally, BRLM has two major advantages over LM training algorithm: (1) no division of input data into “test set” and “validation set” is needed for early stopping (for early stopping criteria, see (*18*)), so the usage rate of input data for network learning is 100%; (2) the number of Model parameter can be reduced based on the needs of the problem. However, both algorithms will be tested on various neural network models, and comparisons will be made.

3.2.4. Prediction Horizon

As one of the common concerns in a typical prediction problem, prediction horizon indicates how far ahead the prediction is to be made. In the context of neural network training, the *k*-step-ahead prediction is done by feeding inputs at current time step and outputs at the next *k* time steps. For example, when making one period ahead prediction, the input-output pairs are formatted as ==x_t, u_{t+1}.

Intuitively, the predictive performance would decrease as the number of time step ahead for prediction increases, especially under incident conditions. However, the practicality of these prediction models depends largely on how much in the future the models can "see" with reasonable confidence. With the longer time step ahead to be predicted at satisfactory level, the traffic mangers and/or individual travelers may better respond to certain situations in a timely manner. In this study, the prediction horizons of 5-minute and 15-minute, corresponding to 1- and 3-step ahead, are investigated.

3.3. Corridor Travel Time Modeling

Segment travel time prediction provides a means to assess useful traveler information in a relatively detailed manner but only for rather short distances. Most of the AVI segments on the freeway network in Houston are within a three-mile range. To forecast travel time in longer distances, such as a freeway corridor or a complete origin-destination path, a corridor travel time modeling approach needs to be developed. In addition, it is acceptable to assume relative stable traffic conditions during a short range of freeway segment, as is the assumption in segment travel time modeling, but that is

not the case for a widely stretched freeway corridor. The methodology to be developed has to account for the traffic dynamics over time and space.

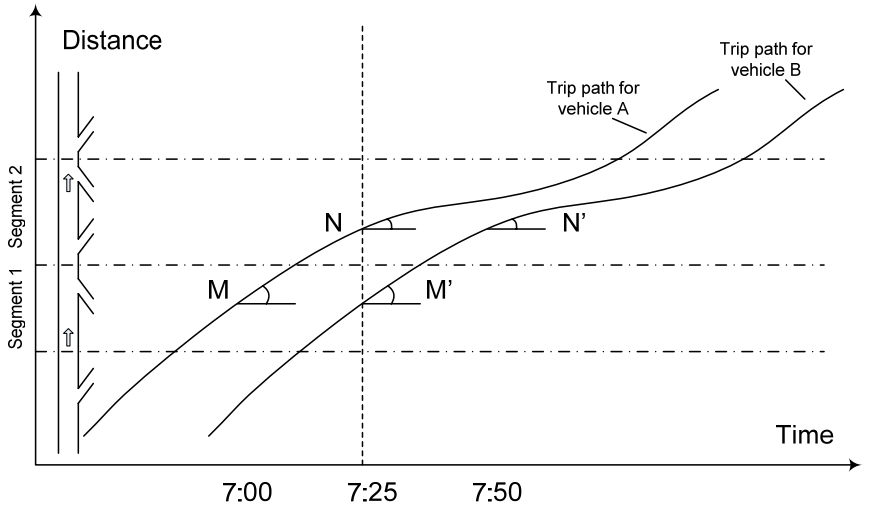
3.3.1. Stationary versus Non-Stationary Traffic Conditions

The temporal and spatial properties of a traffic system bring about the dynamics of traffic flows within the system. Bottleneck situations in traffic flows trigger congestions which propagate several miles upstream of the roadway corridor over time. Nonetheless, to understand how such temporal-spatial dynamics influence the travel time prediction, we can start with a stationary traffic system which assumes that the states of traffic flows at specific locations are homogeneous over time.

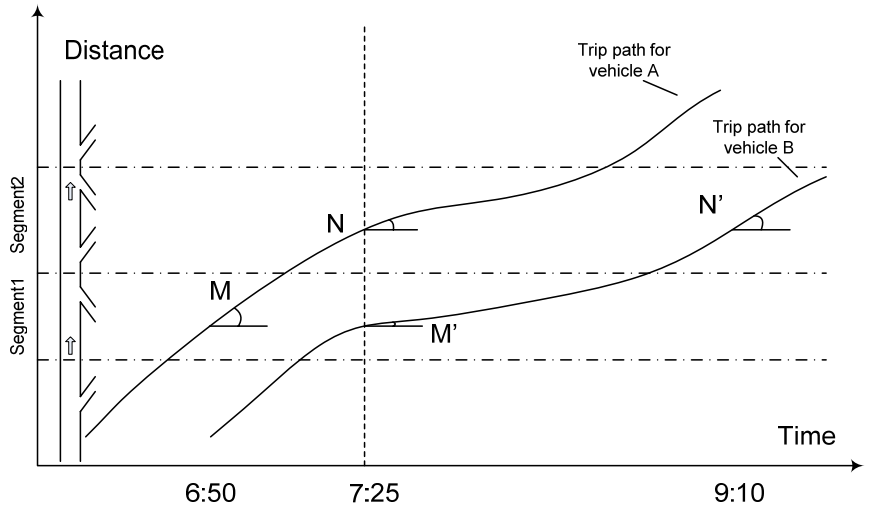
Figure 9-(a) illustrates a stable traffic system in a time-space diagram. Consider the scenario that two virtual trips (trips for vehicle A and B respectively) are made in roughly half an hour apart and both encounter a congested situation close to the end of freeway section 2. Since the traffic system remains stable and balanced in demand and service, the congestion stays and affects the two trips at the same time instant after they enters the section. Due to stationarity, the two vehicles are always half an hour apart during their entire trips. In such a scenario, we can take a snapshot of the traffic conditions along the whole corridor (e.g. section 1 and section 2 combined) at 7:25 in the morning and further assume that vehicle A and B are representatives of the traffic streams in respective sections. Then predictions can be made based on the traffic conditions (i.e. condition at point M' for section 1 and condition at point N for section 2) at 7:25 and be summed up to generate a snapshot corridor prediction. The prediction is not biased and reflects the conditions that will be experienced by vehicle B along the

trip, because current condition for vehicle A at 7:25 (at point N) is simply the projection of the future condition for vehicle B at 7:50 (at point N'). Based on this argument, the corridor travel time prediction may be boiled down to taking a snapshot of the traffic states for each freeway segment at only one time instant.

Dissimilarly, with non-stationary behavior, a traffic system is subject to heavily unstable inflows and outflows that changes across segments and over time. Figure 9-(b) exemplifies the variant traffic system in a time-space diagram. For vehicle A, the trip is relatively pleasant with minor delay caused by light congestion when on segment 2. On the other hand for vehicle B, the trip is seriously delayed due to a surge in demand during morning peak hours which aggravates the early congestion. As the traffic demands continue to rise, the congestion is expected to propagate upstream and affect later trips during an even earlier portion of the trip paths. In cases like such, which is normal, taking a big picture of the traffic conditions at a specific timeline becomes unacceptably inadequate to yield proper representation of the traveler's experience. Consequently, to reproduce traffic conditions a virtual trip might encounter, a dynamic prediction methodology for corridor travel time, which can account for the spatial-temporal characteristic of a traffic system, is warrant. This thesis incorporates a dynamic travel time prediction algorithm – vehicle trajectory method – that is able to reconstruct realistic travel times.



(a) Conceptualized time space diagram of stationary traffic system.



(b) Conceptualized time space diagram of non-stationary traffic system.

Figure 9: Concept of spatial-temporal relationship.

3.3.2. Measures for Corridor Traffic Conditions

In the literature, many performance measures have been developed to assess the traffic conditions of a freeway system – some focus on the evaluations of current

operation efficiency and/or management effectiveness, whereas others survey travelers' satisfactions and their usage frequency of the system. All these measures come down to measuring two primary perspectives of a freeway network: the system perspective and the traveler's perspective.

It is reasonable for a freeway system manager to gather information about the overall conditions of the corridor system while important for an individual traveler to recognize the traffic conditions that they will come across. The former needs snapshots of the system that provides general estimates while the latter requires more detailed and refined estimates of their trips. Accordingly, the following travel time measures are developed to fulfill the respective purposes:

- snapshot corridor travel time
- experienced corridor traveler time

The two travel times evaluate the corridor traffic conditions from different angles. The snapshot travel time gives an instant portrait of the current states of the freeway system, and the values of it reflect the overall conditions averaged across all the segments on the corridor. In contrast, the experienced travel time characterizes individual trips experience by taking into considerations of respective segment experiences at respective time instants. Figure 10 demonstrates the concept of predicting the two travel time measures. For a snapshot travel time to be predicted, one single prediction made at 7:00 for all consecutive segments is sufficient. Nevertheless, multiple predictions at different time horizons are needed to complete the prediction of the experienced corridor travel time.

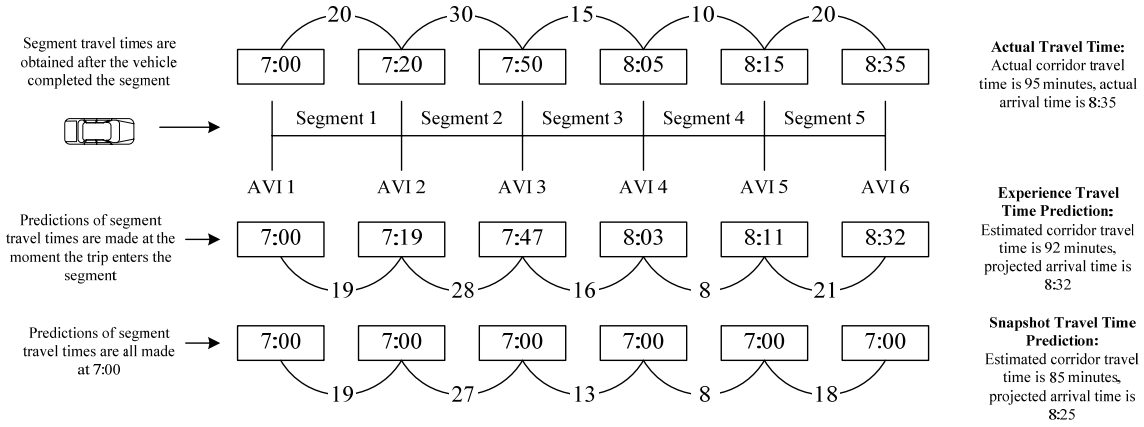


Figure 10: Concepts of snapshot and experienced travel time prediction methods (adapted from (*14*)).

4. EXPERIMENT SETUP

This section is dedicated to describing the necessary steps for the experiments in this study to be reproducible. In the first part, the test bed on which the data are collected is presented along with the description of the sources and the characteristics of these data; the description of procedures and necessary tools for the data extraction and processing then follows; the third part of this section elaborates how specifically the neural network models are designed based on the data collected and processed; and the final subsection focuses on some implementation issues involved in the process of the ANN training and testing.

4.1. Test Bed and Data Sources

The study corridor is located on US-290 southwest of Houston urban area and is one of the busiest commuting routes that connects the downtown commercial districts and the suburb residential areas. The corridor stretches about 20 miles from interstate highway 610 to Spring Cypress Road and extends even further beyond the city limits. To remain focusing on testing the methodology of travel time prediction, we have collected data on the outbound direction (i.e. westbound) of this corridor as shown in Figure 11, where the radar detectors and the AVI stations are densely spaced.

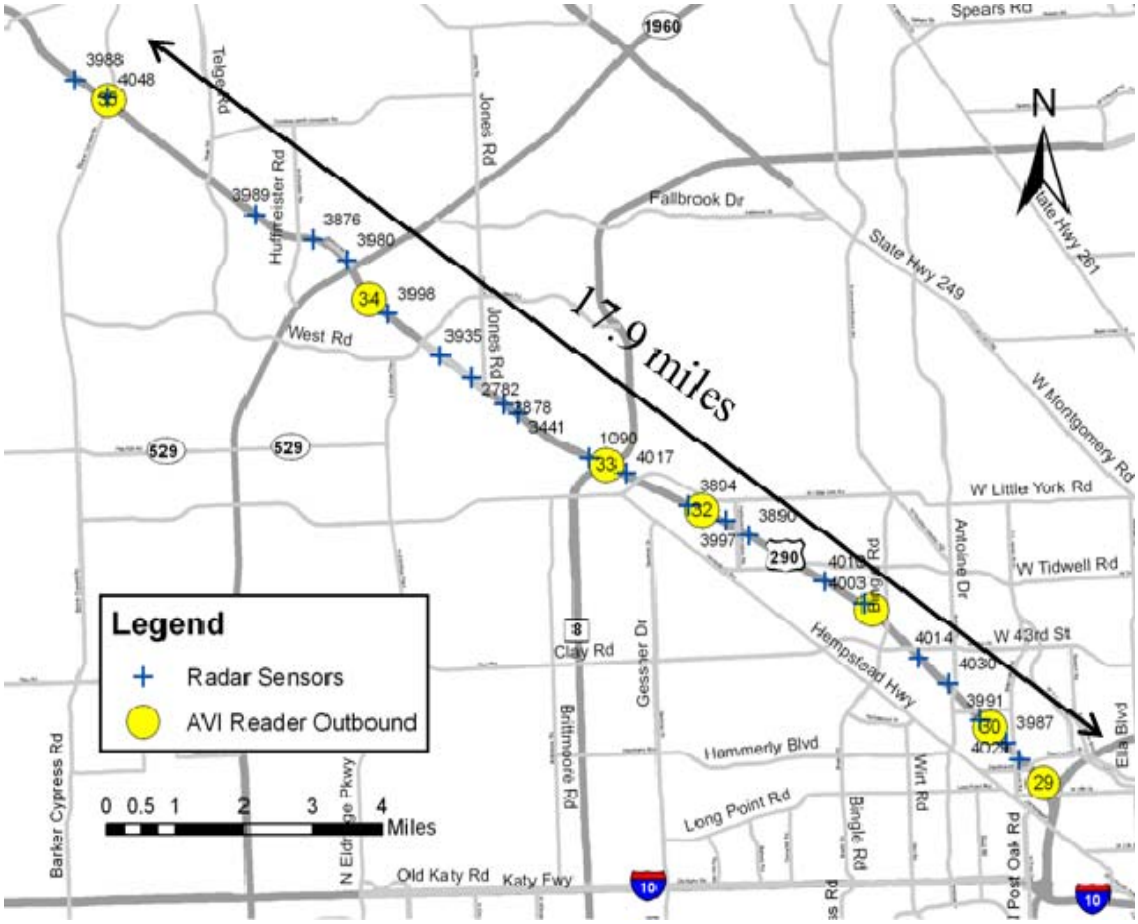


Figure 11: Study corridor.

Three categories of data are required for identifying the relationship between traffic variables and travel time along a freeway corridor: 1). data in relation to traffic operational characteristics such as speed, volume, occupancy and, most importantly, travel time; 2). data in relation to incidents, such as incident type, severity and time after the onset of an incident; and 3). additional relevant information such as weather and holiday information.

The first two types of data sources are collected from Houston TranStar transportation management center. TranStar TMC operates 24/7 and has been archiving AVI travel time and speed data since October 1993, freeway incident data since May 1996 (*26*).

4.1.1. Speed and Volume

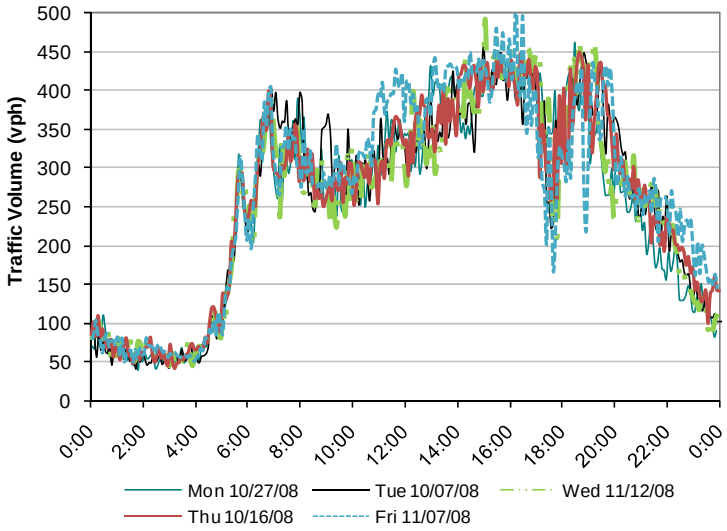
Houston TranStar has installed Wavetronix microwave detection systems at a number of locations. At the time of data collection, the Wavetronix SmartSensor uses a 10.525 GHz frequency modulated continuous wave (FMCW) radar to provide traffic detection. The radar sensors are installed aboveground and collect vehicle volume, occupancy, spot speed, and classification in up to eight lanes of traffic (*26*). Every 30 seconds, the measurements are averaged and archived. Table 2 shows an example of 30-second Wavetronix radar data.

Table 2: Example of 30-Second Wavetronix Data

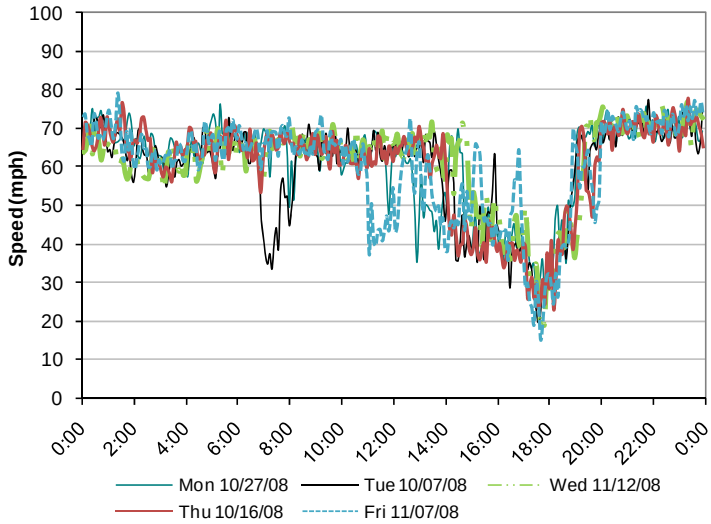
ID	Time Stamp	Lane #	Volume	Speed	Occupancy	Small	Medium	Large
2636	11/12/2008 13:18	1	6	48	3	6	0	0
2636	11/12/2008 13:18	2	10	67	10	7	1	2
2636	11/12/2008 13:18	3	7	63	7	5	1	1
2636	11/12/2008 13:18	4	5	73	2	5	0	0
2636	11/12/2008 13:18	5	5	78	3	4	1	0
2636	11/12/2008 13:18	6	11	70	10	7	3	1
2636	11/12/2008 13:18	7	7	72	7	2	4	1
2636	11/12/2008 13:18	8	13	66	13	2	10	1
2636	11/12/2008 13:18	99	71	66	8	45	20	6
2636	11/12/2008 13:18	1	4	65	3	3	1	0
2636	11/12/2008 13:18	2	11	68	9	7	3	1

On the test bed where the study is conducted, a large proportion of commuters make home-based trips on US 290 during commuting peak hours. Under incident-free

conditions, the rush hours are roughly from 2 pm to 6 pm, resulting in heavy congestions during the afternoon period where travel speeds drop significantly. Figure 12-(a) and Figure 12-(b) show five incident-free weekdays of traffic volume profiles and spot speed profiles respectively.



(a) Traffic Volume Profile



(b) Spot Speed Profile

Figure 12: Traffic data profiles under incident free conditions.

4.1.2. Travel Time

In addition to speed and volume data, the TranStar AVI system also collects vehicle toll tag IDs and the corresponding time stamps when each time vehicles are passing the AVI checkpoints (or AVI stations). An example of raw AVI data is shown in Table 3. Note that actual tag IDs are not displayed here for privacy reasons. These data are used to determine a travel time for each vehicle traveling on the AVI segment.

Table 3: Example of TranStar's Raw AVI Data

Tag ID	Antenna ID	Checkpoint ID	Time Stamp
HCTR00000001	5103	159	11/23/2008 00:00:45
HCTR00000002	8021	216	11/23/2008 00:00:59
HCTR00000003	4111	106	11/23/2008 00:00:59
HCTR00000004	4076	229	11/23/2008 00:03:00
HCTR00000005	8043	219	11/23/2008 00:06:16
HCTR00000006	1200	351	11/23/2008 00:06:31
HCTR00000007	4203	63	11/23/2008 00:07:02
:	:	:	:

As the speed profiles in Figure 12-(b), we can also plot the travel times obtained from AVI stations for corresponding days, as in Figure 13. Under general traffic conditions, speed profiles appear to follow the travel time profiles to a certain extent although the measurements are derived from different data sources. Hence, it can be roughly inferred that speed data alone should serve as a better predictor than traffic volume data. However, we are incorporating both speed and volume data in the prediction of AVI travel time – so called "ground true" travel time.

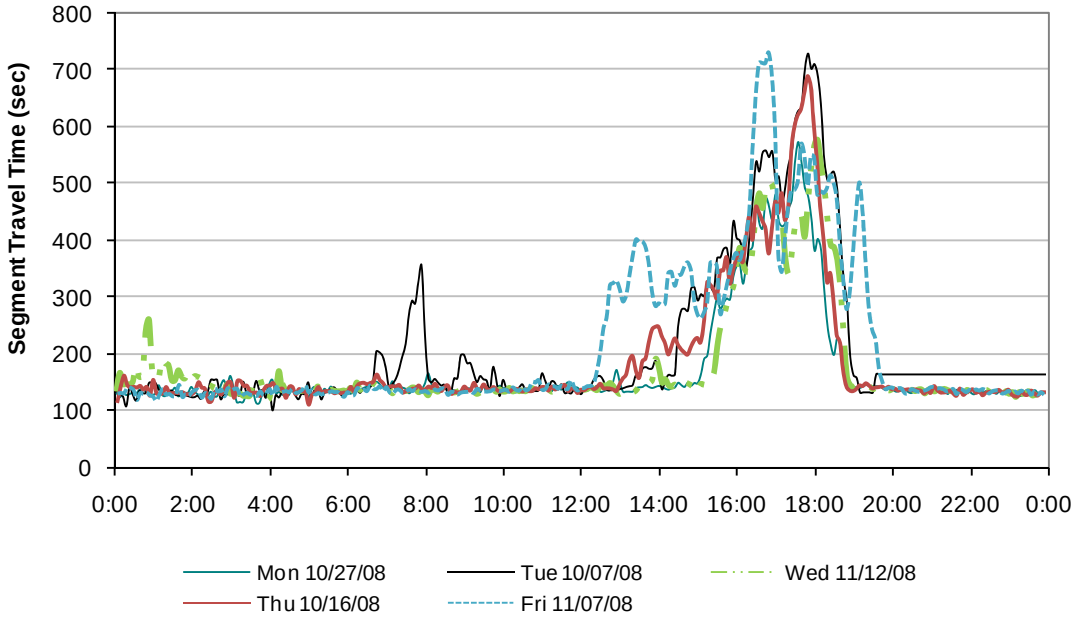


Figure 13: Travel time profiles under incident-free conditions.

4.1.3. Incident Data

Incident detection relies mostly on police dispatch monitoring, MAP calls, commercial traffic services, and CCTV camera scanning. TranStar has an incident detection algorithm that compares and detects changes in segment speeds versus historical speed values. Additionally, operators at TranStar verify incidents using CCTV cameras; then they decide on appropriate responses, such as posting messages on the Dynamic Message Signs (DMS). Incident-related information is then entered into the database through the Regional Incident Management System (RIMS) interface. There are four main time points used to record an evolution of an incident: detected, verified, moved, and cleared. Of these four important timelines, "Detected" refers to the time an operator, including the MAP dispatcher, creates an incident record in the database. This time may or may not coincide with the actual detection time. "Cleared" refers to the time

the appropriate response units clear the incident. With the detection and clearance times, the corresponding incident durations are determined.

Other recorded incident attributes include incident type, severity, weather condition, number of vehicle involved and number of mainlanes blocked, as shown in Table 4. Note that Table 4 is reconsolidated from the original format and shows only the attributes that are important to this study.

Table 4: Example of TranStar's Incident Data

ID	ROADWAY NAME	CROSS STREET NAME	DIRECTION	SEVERITY	TYPE	WEATHER	# OF VEHICLES INVOLVED	# OF MAINLANES BLOCKED	DETECTION TIME	CLEARED TIME	...
683XX	IH-610 NORTH LOOP	US-59 EASTEX	Westbound	Major	Accident	Rain	1	2	1/1/2008 0:47	1/1/2008 1:27	...
683XX	IH-610 NORTH LOOP	SHEPHERD DR	Westbound	Major	Stall	Ice	2	1	1/1/2008 2:40	1/1/2008 3:08	...
683XX	SOUTH SAM HOUSTON TOLLWAY	BLACKHAWK	Eastbound	Major	High Water	Hail	2	1	1/1/2008 3:11	1/1/2008 5:01	...
683XX	IH-10 KATY	BARKER CYPRESS RD	Eastbound	Major	Lost Load	Fog	2	0	1/1/2008 3:14	1/1/2008 4:33	...
683XX	IH-610 EAST LOOP	SH-225	Southbound	Major	Fire	High Wind	1	2	1/1/2008 4:21	1/1/2008 4:26	...
683XX	WEST SAM HOUSTON TOLLWAY	IH-10 KATY	Northbound	Minor	Hazmat	Dust	1	0	1/1/2008 4:34	1/1/2008 4:38	...
683XX	IH-45	FM-1764/JOHNNY PALMER HIGHWAY	Southbound	Major	Accident	Smoke	1	0	1/1/2008 4:48	1/1/2008 5:13	...
683XX	SH-288	OREM	Northbound	Major	Sall	Other	2	1	1/1/2008 5:23	1/1/2008 6:03	...
:	:	:	:	:	:	:	:	:	:	:	

Table 5 summarizes a total number of 71,964 incidents across a five-year period (2004 – 2008). Of these incidents, the "accident" and the "stall" events make up 74% and 18.9% respectively. Because any neural network training requires a large amount of data input, other types of incidents have too few data to be trainable. Therefore, the study will investigate merely the two incident types. However, the two types are not assigned with any index to discriminate one from another, and they are treated as if they are of one type. Because we assume that the significance of one type over another is captured by the recorded severity level, number of lanes blocked and number of vehicles

involved, subjective assignments of type index to distinguish the two types are not necessary.

Table 5: Total Number of Incidents by Year

Incident Type	2004	2005	2006	2007	2008	Total	% of Total
Accident	9713	9426	10,335	12,123	11,628	53,225	74.0%
Stall	2625	3444	2768	2270	2472	13,579	18.9%
Heavy Truck	1293	1560	1590	1534	1393	7,370	10.2%
Construction	762	2020	1203	452	302	4,739	6.6%
Debris	428	580	580	471	402	2,461	3.4%
Vehicle on Fire	286	241	299	275	266	1,367	1.9%
Other	239	261	268	240	277	1,285	1.8%
High Water	126	97	309	149	125	806	1.1%
Bus	150	200	140	78	128	696	1.0%
Hazmat	71	71	103	90	51	386	0.5%
Lost Load	38	49	59	54	61	261	0.4%
Ice	0	0	0	27	33	60	0.1%
All Types	13,105	13,879	14,396	15,467	15,117	71,964	

Note: 1. table updated from (*26*)
2. one incident event may be labeled as one or more incident types

4.2. Data Preprocessing

The raw radar and AVI data acquired from TranStar TMC are not readily available for neural network model development and application. The data reduction and fusion is a necessary step to select study period, aggregate data, filter out false data entries, interpolate missing data and so forth.

4.2.1. Selection of Data Aggregation Interval Size

One of the top concerns while preprocessing raw data is how the data should be aggregated to yield unbiased estimates. “Aggregation interval” refers to the time window at which the data are summarized (*26*). Indeed, smaller aggregation resolution tends to be more favorable. A number of studies attempt to identify the optimal aggregation level

for loop detectors using cross-validated mean square error approach (*27*), and a number of other statistical methods (*28, 29*). These studies have not pointed to a universal consent on what size of the aggregation interval is the optimal, largely because the natures of the data analyses are different. Park et al. (*30*) applied a modified MSE approach to a set of travel time prediction results forecasted by a spectral-basis neural network model (*15*). The study was able to show the differences of optimal aggregation intervals between travel time forecasting and estimation. The result indicated a best choice for travel time prediction would be around the range of 5-10 minute and the degradations were observed thereafter. Based on such a conclusion, we chose the smaller interval size (i.e. 5 minute) to conduct our experiment for prediction results to have a higher resolution.

4.2.2. Speed and Volume Extraction

As mentioned in a previous section, Houston's radar sensor provides a stream of 30-second observations of volume, speed, occupancy, and vehicle classification for each travel lane. This section describes the calculation procedures and routines to derive the station-based (i.e. multiple lanes at a location) measures from the raw radar data.

The current implementation of the speed/volume extraction tool is capable of calculating the following measures: total volume, average speed, average occupancy, and coefficient of variation in speed (CVS). In the travel time prediction concerned in this study, the total volume and the average speed are the only two measures that need extractions.

The total volume per output interval is calculated as:

$$Q_t = \sum_{j=1}^{l} \sum_{i=1}^{n} q_{ij} \tag{19}$$

where q_{ij} is the 30-second volume count of the j-th input interval at lane j, Q_t is the aggregated volume count of the t-th output interval, n is the number of input intervals within the aggregation time window (i.e. 5 minute), and l is the number of lanes in a station.

The weighted average speed per lane is calculated as:

$$\overline{V}_t = \frac{\sum_{j=1}^{l} \sum_{i=1}^{n} q_{ij} v_{ij}}{\sum_{j=1}^{l} \sum_{i=1}^{n} q_{ij}} \tag{20}$$

where v_{ij} is the 30-second weighted average speed of the i-th input interval at lane j, and $\overline{V}_t$ denotes the weighted average speed of the t-th output interval. The weighted average speed has an advantage that better describes the true fluctuation of vehicles' speed over time, particularly during the light traffic volume condition.

Based on equations (19) and (20), speeds and volumes can be derived from a complete set of the traffic raw data. However, it is observed that invalid or missing data entries prevail in the radar dataset collected for the study. For a robust neural network model to be developed, proper handling of these data is critical. In case invalid or missing volume data are present in an interval, the tool re-estimates the total volume by linear extrapolation using the following equation:

$$\hat{\theta}_t = \frac{1}{p} \cdot \theta_t \tag{21}$$

where θ_t is the measure (e.g., volume) calculated for the *t*-th output interval, $\hat{\theta}_t$ is the re-estimated measure extrapolated from θ_t, and *p* denotes the proportion of valid data.

Figure 14 shows a procedural routine implemented in this tool. The routine starts with configuring the lanes for a station. Next, the data are retrieved and processed for every output interval. The algorithm also checks for valid data and performs the adjustment if necessary for the interval. Firstly, the validation process checks to see if there are sufficient valid data for calculation. If the number of valid records for aggregation is more than 50%, the calculation process then continues; otherwise, the module will flag the data for that interval as invalid. This validation module currently examines volume and speed.

4.2.3. Travel Time Extraction

The AVI system consists of a series of tag readers (checkpoints or stations) collecting tag identifications and time stamps for each vehicle passing through the checkpoints on the Houston freeway system. In order to extract travel times from AVI data properly, the following three critical steps are carefully conducted:

- Data matching – Based on the times that a vehicle passes through the origin and the destination of an AVI segment, the corresponding travel time can be valued by finding the difference between the two timelines for the vehicle. In the algorithm, the vehicle toll tag ID is used as the unique identifier that matches subsequent records in the AVI database.

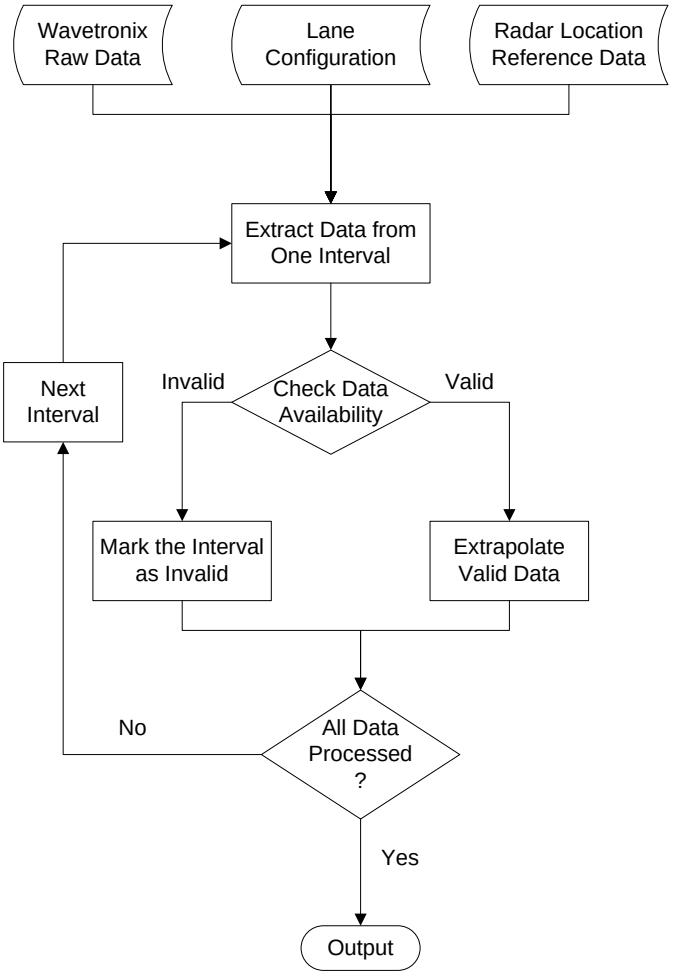


Figure 14: Calculation routine for traffic data processing (*31*).

- Data aggregation – With the valid individual travel times sampled and calculated, the final step of the algorithm is to aggregate (by averaging) the travel time according to predetermined time intervals. The determination of the aggregation interval, however, should abide to a combination of restraints in addition to the factors mentioned in section 4.2.1. The first concern is the sample size (i.e. the number of toll tags being used by Houston commuters).

As we observed, around 3-5 % of the traffic are equipped with toll tags, which is roughly equivalent to 50 to 100 vehicles per five minute period during peak hours. We explicitly assume that the number is significant to represent prevailing traffic conditions. For non-peak hours, the interval of five minute is used to maintain consistency without the loss of representativeness since the traffic conditions during these hours are generally stable. The second concern is the update frequency that is preferred by the training of a neural network. The five minute update rate is acceptable and used in a number of studies of travel time prediction using neural networks (*14, 16*).

- Data validation – The algorithm also addresses the possibility of matching the vehicles that potentially give bias estimates of the prevailing traffic condition, such as vehicles taking exist and re-entering the freeway or temporarily dwelling between the O-D pairs of interest. The below section describes the primary method used to validate the travel time data by addressing these particular problems.

In accordance with the critical steps described above, Figure 15 provides an overview of the travel time extraction algorithm developed in this study.

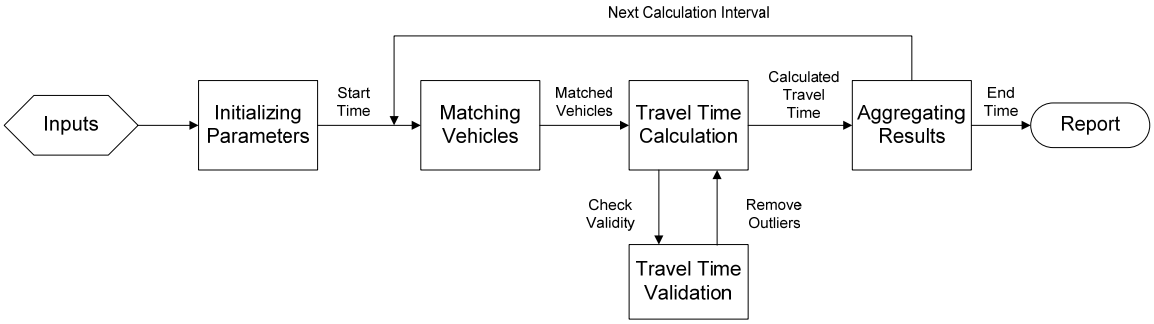


Figure 15: Overview of travel time extraction algorithm (*31*).

The data validation process involves a method of how to define outliers that shall be excluded from the calculation of average travel time. This step is crucial to extract the trip travel times that represent the prevalent traffic conditions.

The algorithm uses the free-flow speed to calculate the free-flow travel time for a specified segment. The free-flow travel time establishes the lower threshold for segment travel time in the algorithm, which is defined as follows:

$$\text{Lower Travel Time Threshold} = \frac{\text{Segment Length}}{\text{Free-Flow Speed}}(1-p) \qquad (22)$$

where p is the adjustment ratio to capture the vehicles traveling faster than the free-flow speeds. A p value of 0.2, for instance, implies an additional 20 percent reduction from the calculated free-flow travel time. As recommended in (*31*), 20 percent is generally sufficient to capture most vehicles traveling faster than the specified free-flow speeds.

Similarly, the algorithm uses congested speeds to calculate the travel time under congested conditions for the segment. The congested travel time defines the upper threshold of the segment travel time, which is calculated by:

$$\text{Upper Travel Time Threshold} = \frac{\text{Segment Length}}{\text{Congested Speed}}(1+p) \tag{23}$$

Given the speeds, origin-destination (O-D) pair and degree of flexibility (i.e. parameter *p*), the upper and lower travel time thresholds are used to construct a time window by which outliers can be prescreened. However, such a process is too rough to identify the trips that were made with a short stop within the segment, a brief detour in a gas station or the like. For a more precise result, two more detailed and advanced options are developed accordingly – error tolerance method and z-score method (see (*31*) for elaboration).

By well defining the methodology to extract and prescreen travel time data from raw AVI database, we expect to find travel times that truly represent predominant traffic states. Nevertheless, it is particularly difficult to obtain enough sample pairs of O-D data records in non-peak hours (e.g. midnight till dawn). Therefore, the treatment for little sample size (*N*) situation is developed as: (1) if $N = 0$, the algorithm will retrieve the average travel time from the previous interval for the current interval; (2) if $N = 1$, the algorithm will use that single value as a travel time for that interval; (3) if $N > 1$, the algorithm will calculate the average travel time and perform the validation process for that interval. Therefore, the data validation process depicted above can be summarized by the flow chart shown in Figure 16.

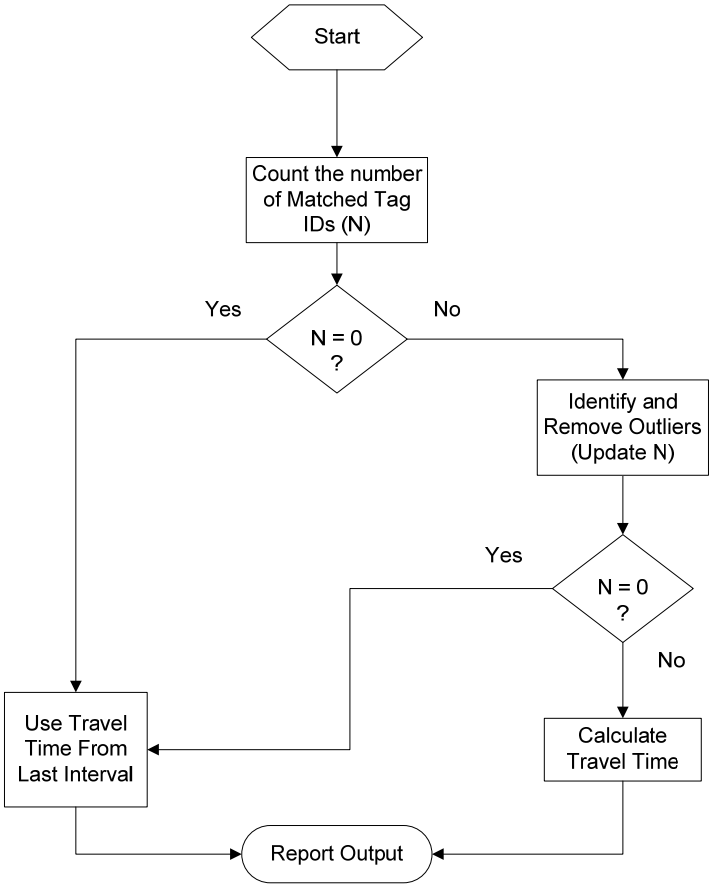


Figure 16: Travel time calculation and validation process (*31*).

4.2.4. Incident Data Reformatting

As Houston incident database record every incident events as one record, the original data format is not suitable in dynamic network training. Therefore, the incident data are reformatted to match with the input format of speed and volume data. Table 6 shows a reformatted incident table for incident 68439 occurred on Jan 3rd, 2008.

Within all the incident information archived, only two pieces of information are selected and directly used: the number of mainlanes blocked and number of vehicles involved. Other indices are calculated from other available incident information.

Table 6: Example of Incident Inputs Reformatted

Time	Time to Onset of Incident	# of Mainlane blokage	# of Vehicles Involved	Severity	Distance to Onset of Corridor	Percent Distance to Onset of Corridor
1/3/2008 6:35	0.0	0	0	0	0.0	0.0%
1/3/2008 6:40	0.0	0	0	0	0.0	0.0%
1/3/2008 6:45	2.8	1	1	5	2.8	22.5%
1/3/2008 6:50	7.8	1	1	5	2.8	22.5%
1/3/2008 6:55	12.8	1	1	5	2.8	22.5%
1/3/2008 7:00	17.8	1	1	5	2.8	22.5%
1/3/2008 7:05	22.8	1	1	5	2.8	22.5%
1/3/2008 7:10	27.8	1	1	5	2.8	22.5%
1/3/2008 7:15	0.0	0	0	0	0.0	0.0%
1/3/2008 7:20	0.0	0	0	0	0.0	0.0%

4.3. Comparison Setups for Segment Travel Time Prediction Models

The segment travel time prediction is the first step of the two-stage process, which involves model development and calibrations. Since the corridor travel time prediction is based on the prediction of segment travel time, neural network models for segment travel time are crucial to the performance of the two-stage prediction process.

To determine the most suitable model for segment travel time prediction under different data inputs, the comparison experiment is setup to compare the performance of 5 ANN models and 2 baseline models under both incident-affected and incident-free condition. Different training algorithms as well as different prediction horizons are compared to refine the model to be developed.

4.3.1. Selected Segment for Model Comparison

An AVI segment from the study corridor is selected for developing the most promising model on which the corridor travel time prediction can be based. The segment is enclosed by the AVI station 31 and station 32 extending a total of 2.9 directional freeway miles, shown in Figure 17. Five radar detectors locate inside the segment at

roughly equal spacing. For the reason of spatial-temporal relationship in a time-variant traffic system (discussed in section 3.3.1), the traffic conditions of the immediate downstream must be accounted for by means of monitoring the downstream speeds and volumes altogether. As a result, data streams extracted from two additional radar detectors in the 32-33 AVI segment are simultaneously fed to neural network models along with the original data inputs.

Within the study segment, there are 141 incidents of all types occurred in all year of 2008, of which 140 are accident and stall types combined.

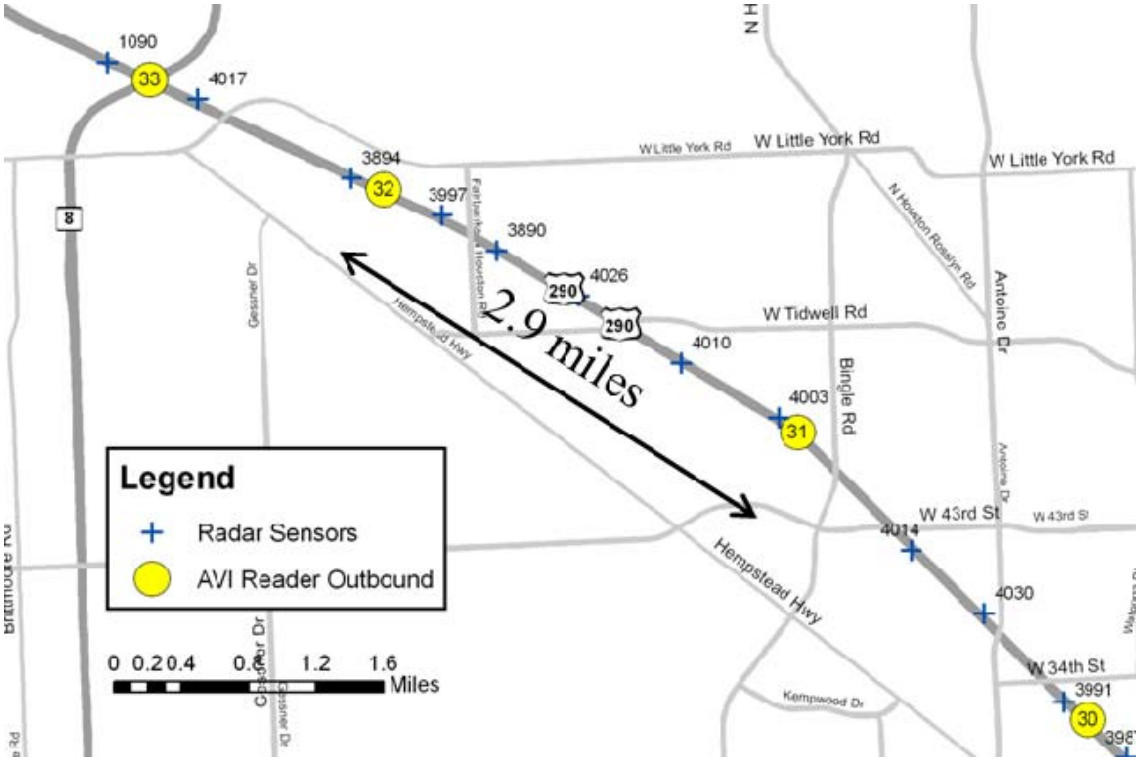


Figure 17: Study segment for segment travel time prediction model development.

4.3.2. Model Design for Segment Travel Time Prediction

The five models proposed in the study (see section 3.2.1 for detail descriptions on the model concepts and structures) are all used in developing segment travel time prediction models for comparison. Since the internal structures and conceptualizations of weight connections across all the proposed models (excluding ExtSSNN) are similar to SSNN model. On this account, this subsection focuses on the descriptions of model development for SSNN and ExtSSNN. In addition, two baseline models are briefly discussed in the end, which are used to compare with the predictive power of the ANN approaches.

4.3.2.1. SSNN Model Development

Given the inputs from a total of five consecutive radar sensors on the current AVI segment in the study and two additional sensors on the downstream segment, the SSNN model structure can be derived accordingly. Based on the schematic diagram of SSNN (i.e. Figure 6), each input node is associated with the one radar detector where two input variables (i.e. speed and volume) are obtained. Then the inputs fully connected to all *M* hidden neurons in the hidden layer resulting in a [*M*x*7*] weight matrix. Figure 18 illustrates a model architecture for SSNN and the associations of its inputs with the physical detectors on a freeway corridor. The context layer consists of *M* delay units with the same number as the hidden neurons. The connections between input nodes and hidden neurons and those between feedback nodes and hidden neurons are trainable connections.

A few *a prior* remarks must be made on the selection of hidden neurons. The hidden layer can be assigned as many hidden neurons as possible, but the designer of the model shall inevitably confront with the dilemma of learning and generalization (such a dilemma is discussed in section 3.2.3.2). As empirically argued in (*10*), the number of hidden neurons should be equal to the number of detectors as input in that each neuron can be conceptually mapped to each detector. However, it is observed in the preliminary analysis that the conceptual mapping is neither certainly nor evidently appreciated in the training process and a larger number of hidden neurons does not necessarily compromise the efficiency of training nor degrade the precision of testing. In addition, there is a better chance for a SSNN model to learn a complex problem with higher number of hidden neurons (*18*). For those reasons, we choose to use the number of hidden neurons the same with the number of physical detectors for the SSNN and the ExtSSNN models and no restrictions for other types of ANN models.

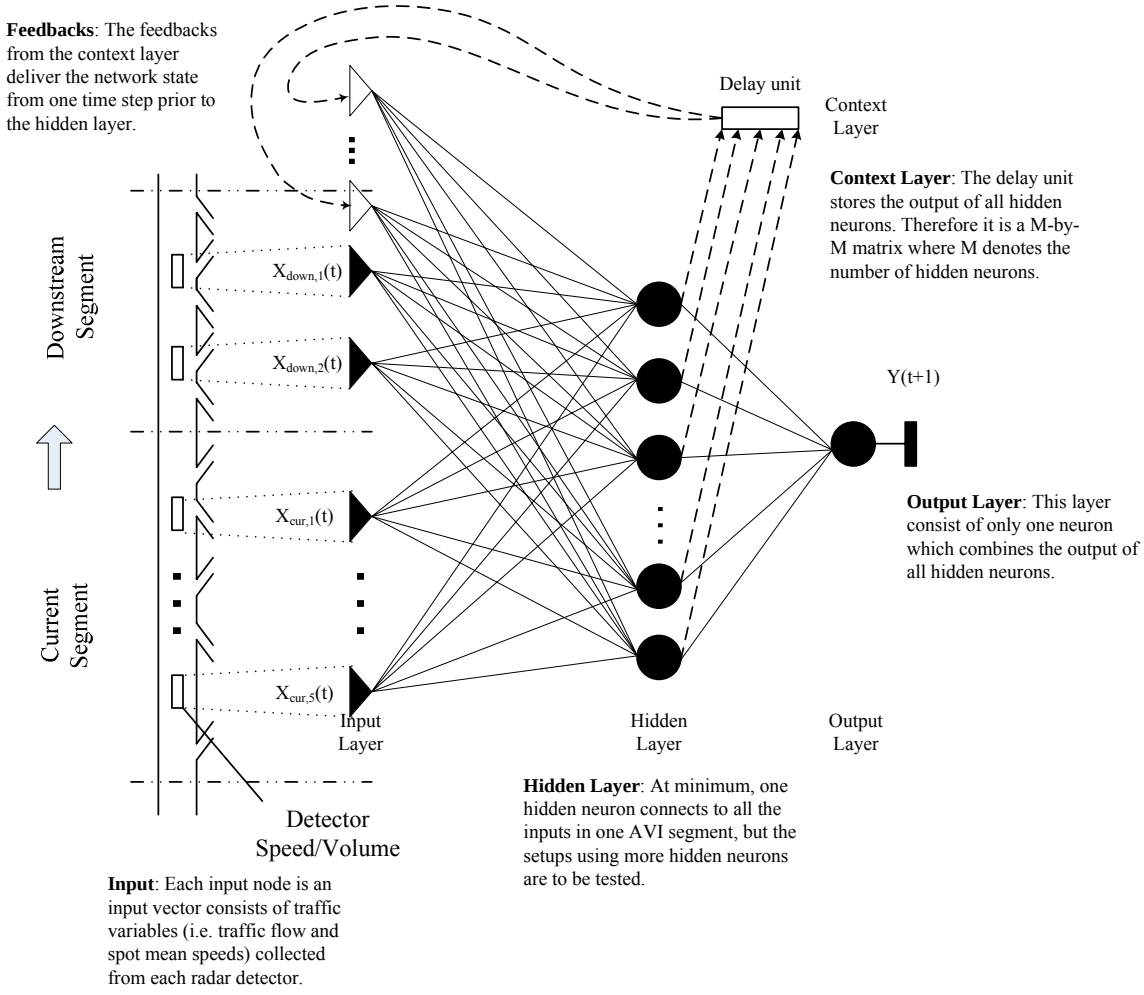


Figure 18: A model design for fully connected SSNN.

4.3.2.2. ExtSSNN Model Development

A traffic system is non-stationary and is especially so during recurrent congestions. Such non-stationarity is naturally captured by the dynamic characteristics implied in the SSNN model. Yet, the additional impacts by unplanned incident events may not be easily accounted for without incident information. The ExtSSNN model proposed in this study is developed in a way that incident information can be pieced

together with speed and volume inputs to model travel time under the impact of incidents.

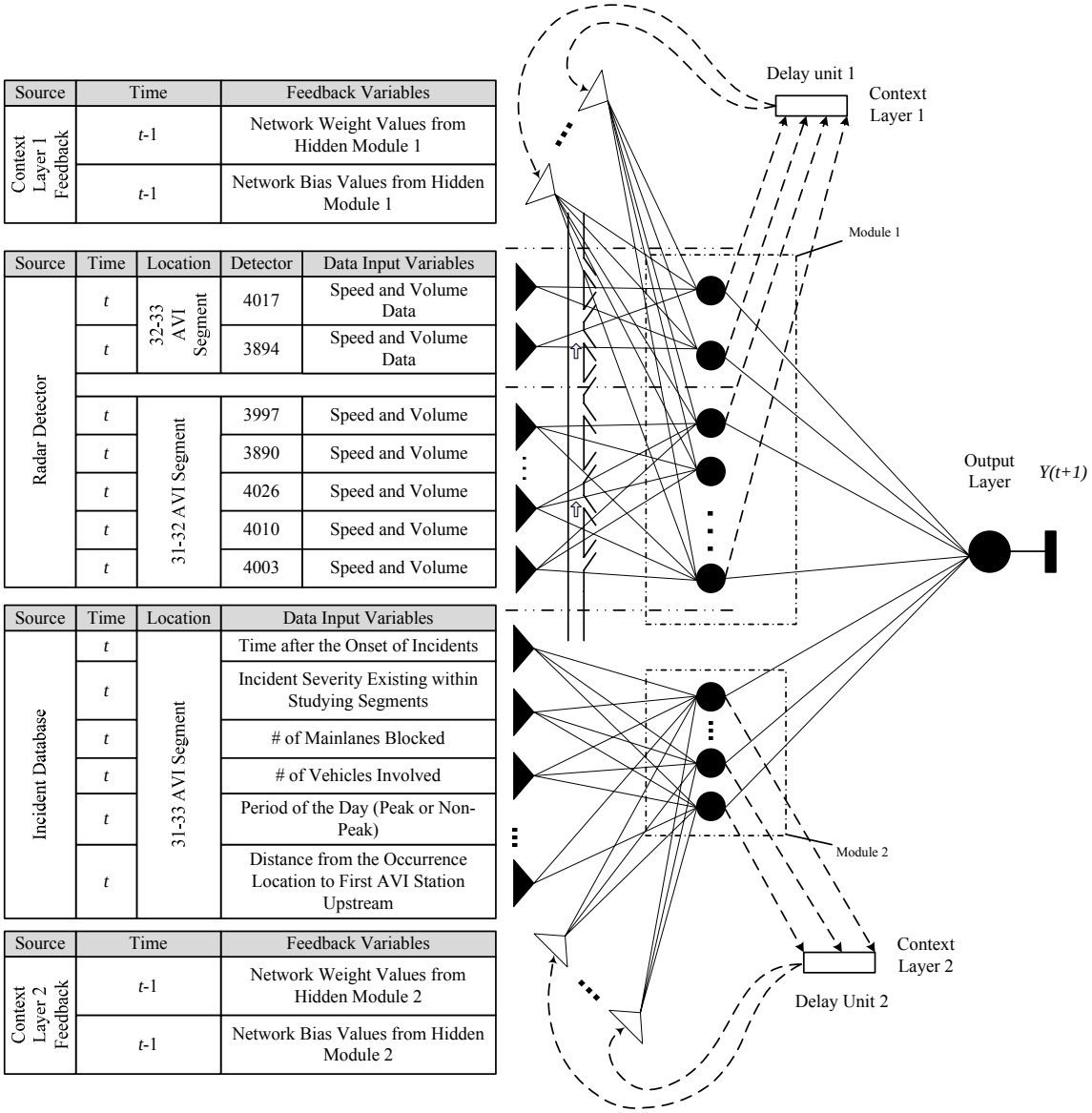


Figure 19: A model design for ExtSSNN.

Figure 19 shows the architecture design as well as detail input descriptions for ExtSSNN. On the basis of state-space model, the ExtSSNN has an "add-on" module that

is characterized by the various information sourced from the TranStar incident database (refer to section 4.2.4 for correct reformatting of incident data to be input in ExtSSNN). The second module (add-on module) is also a basic SSNN model with the delay unit store the evolution information of the incidents.

4.3.2.3. Naïve Prediction Model Development

To be compared against ANN models, a competitive candidate requires certain merits, such as simplicities in terms of computational and procedural demands or accuracies in terms of prediction results. The first baseline model is a naïve prediction model. The model assumes a stable traffic system whose conditions fluctuate little over time. As a result, the approach predicts traffic at next time instants by averaging the travel times of N_V immediate past time instants, which is formulized as

$$TT_{t+1} = \frac{1}{N_V}\sum_{v=1}^{N_V} TT_{t-v+1} \tag{24}$$

In this case, we set N_V to 3 throughout all evaluated scenarios, meaning the travel time data from current and the two previous 5-minute intervals are used to calculate the future travel time value. In effect, this method shifts the actual travel time profile to the negative infinity by a step size between (0, N_v), and the averaging function smoothes out the fluctuation of the profile.

4.3.2.4. Historical Median Travel Time Estimation Model Development

The second baseline model takes into account historical data to estimate the empirical expectations of the normal traffic conditions reflected by travel times. Either the average or the median statistics of historical data at certain time instants can be used

to reveal the central tendencies of the travel time at the same instants. It is not surprising that the average values of travel time over an unfiltered dataset would be unreliable in that the occasional presences of abnormal incident events drive individual travel time values substantially deviant from expected normal conditions. Implementing the average statistic to estimate normal conditions may demand some time-consuming process to filter out the data affected by incidents. Instead, the median statistic addresses the ability to alleviate the problems caused by extreme outliers and still provides satisfactory estimates on the expectation of normal traffic conditions. The method is proven to be efficient and effective to automatically eliminate the effect of the incident affected intervals as long as the samples used to calculate the median statistic is large enough (*26*).The following equation formulizes the method:

$$TT_{t+l} = median(TT_{t+l}^{1}, TT_{t+l}^{2}, ..., TT_{t+l}^{n}) \quad (25)$$

where subscript *t* indicates the current time step and *l* denotes the number of time steps ahead of the prediction (i.e. prediction horizon); superscript *n* is the number of sample days.

The median-based estimation approach for the travel times on a general weekday requires several weekdays selected randomly or from the immediate past from the historical travel time database. Then a series of median values over the analysis period can be constructed and is a travel time profile under incident-free traffic conditions. In this study, a median profile is constructed from Dataset A, and is to be tested against Dataset B-I, B-II, and B-III.

4.3.3. Training and Testing

For comparison purpose, the five ANN models are trained with two batch training algorithms (i.e. LM and BRLM) and are tested on the other datasets with the same characteristics to the training datasets.

The training data (Dataset A) is a dataset that consists of radar and AVI data, measured on the study segment described above (in Figure 17), that totals a number of 125 series (i.e. number of sample days) corresponding 125 weekdays in 2008 regardless of the presence of incidents. Each series is one complete series of 96 time intervals (5 minutes per interval) spanning from 12:00pm to 20:00pm. For a data-driven approach to have a better chance to learn the behavior of the traffic system under both recurrent and non-recurrent congestions, the input data should contain significant amount of data. Restricted by the availability of the incident data, Dataset A contains 56 sample days affected by incidents and 69 sample days free of incidents.

Note that the preservation of time sequences to be fed into the training network is critical to guarantee a stable statistical inference to be drawn from the data. Nevertheless, the sequence can be truncated into smaller portions with different sizes. The need for truncation often depends on the number of time-step-ahead prediction. For example, we truncate one complete time series of a day (288 time sequences) into 287, 286 or 285 time sequences for the purpose of one, two or three time-step-ahead prediction. From the data of the truncated time series, the neural networks are trained in a batch mode to approximate the underlying function between traffic inputs and travel times by

establishing the mappings between the traffic inputs at current time sequence to the travel time at next time sequence (as in the case of one-step-ahead prediction).

The testing of neural network models is to determine the predictive power of the models by means of testing the networks with new inputs that are unseen by the networks before. 84 new general weekdays including 23 new incident-free days and 61 new incident-affected days are assigned to Dataset B-I, Dataset B-II and Dataset B-III for testing. Be advised that the order of the time sequence in testing samples has to coincide with that in training samples for the reason that neural network learns the mapping by partially recognizing the interactions among the sequence – the order dictates these interactions. Similarly, the naïve prediction model and historical median travel time model are tested on the same datasets.

Finally, to determine how well each of the neural network models and the two baseline models capture the nonlinear relationship among traffic inputs, incident inputs and travel time outputs, three error functions are calculated as performance criteria during the testing phases:

- Mean Absolute Error (MAE)
- Mean Absolute Percentage Error (MAPE)
- Root Mean Squared Error (RMSE)
- Normalized Root Mean Squared Error (NRMSE)

The MAE performance measure examines the cumulative deviations of the predicted outputs from the targets, while MAPE translates the deviations into percentage forms by comparing with the absolute target values. The MAPE is a measure of accuracy

in fitted time series due to the ability to compare the errors of fitted time series that differ in level (*32*) , and is hence a particularly useful indicator in benchmarking the model fittings of travel time series. On the other hand, RMSE serves as another important measure of predictive power of a neural model in computational neuroscience. Similarly, NRMSE also transforms RMSE into percentage (similar as MAPE to MAE) by normalizing RMSE with the average of target outputs, where lower values indicate less residual variance.

Simply speaking, all the performance measures are calculated in the following way. We firstly calculate the error of prediction to actual value at one time interval, then all series of errors for all testing days can be calculated similarly. With the pool of errors and the pool of actual values, all the aforementioned measures can be calculated. In the following, the performance functions are listed with their formulas:

$$\mathrm{MAE} = \frac{1}{N_T N_D} \sum_{d=1}^{N_D} \sum_{t=1}^{N_T} \left| u_{t,d} - y_{t,d} \right| \tag{26}$$

$$\mathrm{MAPE} = \frac{1}{N_T N_D} \sum_{d=1}^{N_D} \sum_{t=1}^{N_T} \left| \frac{u_{t,d} - y_{t,d}}{u_{t,d}} \right| \times 100\% \tag{27}$$

$$\mathrm{RMSE} = \frac{1}{N_D} \sum_{d=1}^{N_D} \sqrt{\frac{\sum_{t=1}^{N_T} (u_{t,d} - y_{t,d})^2}{N_T}} \tag{28}$$

$$\mathrm{NRMSE} = \frac{1}{N_T} \sum_{d=1}^{N_D} \frac{\mathrm{RMSE}_d}{\Delta_d} \times 100\% \tag{29}$$

where $\Delta_d = \max(u_{t,d}) - \min(u_{t,d})$; $u_{t,d}$ and $y_{t,d}$ denote the actual travel time and predicted travel time at time instant t on day d; N_T and N_D are the total number of time instants in one sample day and the total number of days to test.

4.3.4. Input Failures

In all detectors, input failures (either missing or corrupted) occur. On average, dataset A and B-I contain corrupted data of 3.54%, and 4.16%, which implies in about three-four time steps (15-20 minutes) of one peak period (8 hours) there is a input failure occurred on either of the 6 detectors along the study segment. Such a statistic has excluded detector 3890 since its data corruption rate is nearly 70% even after initial data correction process.

Input failures may be tolerated during the training phase as long as the failure rates remain low for a single training sample or training sample in general. Yet, ANN models are not advised to be either trained or tested when massive data corruption occurs, as is the case along the segment on Dec, 17, 2008. The data missing rate on this particular date is 33% and the performance of ANN models deteriorate dramatically as shown in Figure 20. In cases like such, the HM method should be used as remedial measure to alleviate the impact of missing data on travel time prediction.

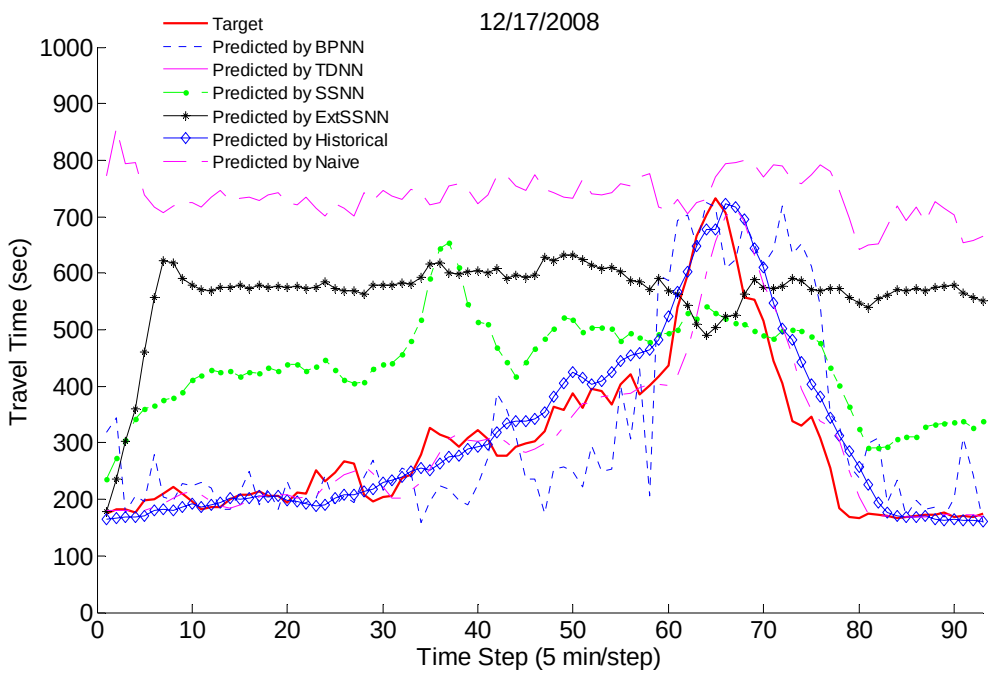


Figure 20: Detector failures causing prediction inability of ANN models.

To ensure days contain massive detector failures would not be presented to any ANN models, we have developed a procedure such that if more than 25% of the time that 1.5 of the 6 radar detectors within the study segment (both current and downstream) did not retrieve valid data, then the day will be deleted from either training or testing.

4.4. Setups for Corridor Travel Time Modeling

In addition to comparing the prediction abilities of various models (i.e. 5 ANN models and 2 baseline models) on segment travel time, we integrate the best ANN model along with the two aforementioned calculation methods to modeling corridor travel time. To implement and compare the concepts of snapshot travel time and experienced travel

time and how the snapshot and vehicle trajectory methods can be applied along with an artificial neural network, we setup the experiment, as described below in this section, to derive both measures of traffic conditions.

4.4.1. Selected Corridor for Prediction Method Comparison

The development of neural networks for corridor travel time is made on the east portion of the US290 freeway corridor, displayed in Figure 21. The selected portion of the corridor, or simply the corridor, stretches 12.5 miles from AVI station 29 to 34. The corridor is composed of 5 consecutive AVI segments with lengths varying from 1.1 miles to 4 miles. A total of 20 radar detectors situate along the corridor. In order to address the spatial dynamics of congestions in the model development, we further consider the three detectors (i.e. detector ID 3980, 3876 and 3989) downstream.

4.4.2. Two Stage Forecasting

As described in Figure 2, the two-stage forecasting method requires the parameter calibration (training) for the selected model to be done in the first place. Then in the second step, we can calculate (in fact predict) corridor travel time based upon the results from the first step. In many current practices, one of two basic methods is used to compute freeway corridor travel times: the “snapshot” method and the vehicle trajectory method (*33*). Therefore, this thesis considers and compares both methods. Note that for both calculations to be feasible, we explicitly assume that the traffic conditions stay constant during any one time interval.

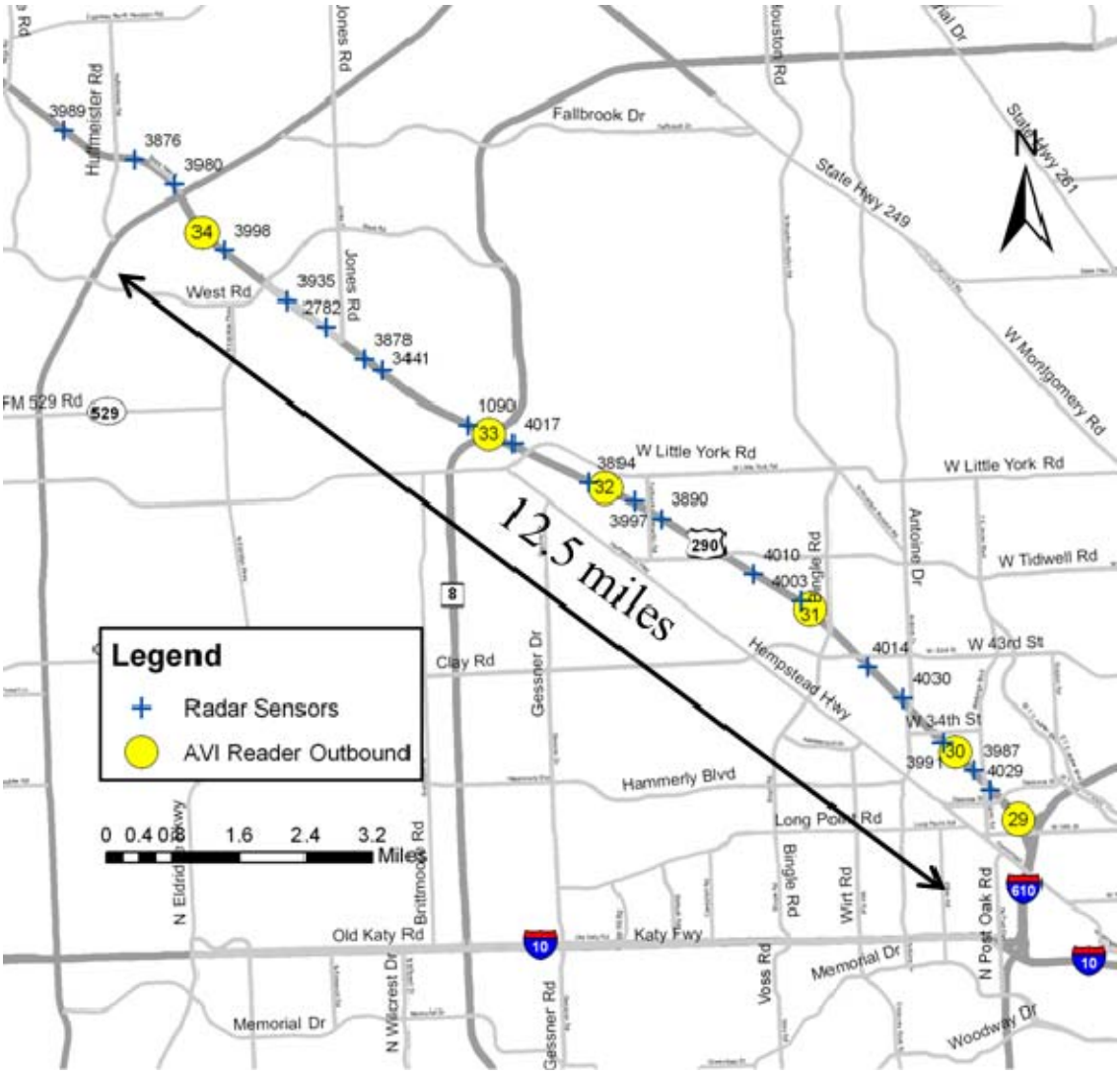


Figure 21: Study corridor for corridor travel time prediction model development.

4.4.2.1. Obtaining Predicted Segment Travel Time Table

For the calculation procedures of corridor travel time to be initiated, tables that are filled with predicted values of segment travel times are to be constructed first. Since there are five AVI segments along the study corridor, five neural network models with similar structure are defined and trained individually. The training dataset for each of

these models consists of speed and volume data on the current and the downstream segments. We select 20 days of samples to form dataset D. These sample days are drawn for training from the pool of incident-free afternoon hours (i.e. 12:00pm – 8:00pm). The rest 32 days (dataset C) are used in the testing phase. After each of these five models being trained at a one-step ahead (i.e. 5 minutes), predictions for the next immediate time step are made using dataset C. Correspondingly, a table filled with predicted segment travel times can be constructed. The columns of the table correspond to each AVI segments and the rows of the table correspond to all the time steps.

It is expected with the performance errors for each individual segments, the aggregated error of corridor travel time prediction might surge dramatically. It is the intention of this study to investigate if the cumulative performance of the collection of ANN models for this study corridor is still within a reasonable level. Therefore, we setup the experiment to evaluate the differences between predicted snapshot travel times and actual snapshot travel times as well as those between predicted experienced travel times and actual experienced travel times.

A brief remark is made here on how many detector inputs ought to be used. Of course, more detector data as inputs can be fed to the network training to account for traffic dynamics either further downstream or upstream. Nonetheless, as observed in preliminary analysis, speed and volume from current and downstream segments are already enough for the neural network to be trained to a satisfactory level, and no further improvement on performance can be made by adding more detector inputs. In fact, the

inclusion of more detector units actually complicates the model structure and thus makes trainings even more resource demanding.

4.4.2.2. Option One: Snapshot Calculation procedure

The second step is to utilize the table of predicted segment travel times to compute the corridor travel time. The first option for the computation task specifically is the snapshot calculation method.

The snapshot method calculates the snapshot corridor travel time. Since the snapshot corridor travel times are representations of the average traffic conditions on the freeway corridor at a specific time, the calculation of it can be straightforwardly carried out by summing the segment travel times predicted at the same moments. Table 7 illustrates the calculation procedure of snapshot calculation method. As an example at time interval 7:00 to 7:05, the snapshot corridor travel time is equal to 2+3+5+6+2 = 18 minutes. It can be seen that the resulting values rise and fall according closely to the conditions across all the five freeway segments.

Table 7: Illustrative Example for Calculating Snapshot Travel Time

Time Interval	Segment Travel Time (min)					Snapshot
	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5	Corridor
7:00 - 7:05	2	3	5	6	2	18
7:05 - 7:10	3	3	6	8	3	23
7:10 - 7:15	4	5	7	10	4	30
7:15 - 7:20	6	6	9	13	5	39
7:20 - 7:25	6	7	10	15	6	44
7:25 - 7:30	7	8	11	17	7	50
7:30 - 7:35	9	10	13	20	9	61
7:35 - 7:40	10	11	13	22	9	65
7:40 - 7:45	8	9	10	23	7	57
7:45 - 7:50	8	9	10	20	7	54
7:50 - 7:55	8	9	10	21	8	56
7:55 - 8:00	8	8	8	17	7	48
8:05 - 8:10	7	7	6	14	6	40
8:10 - 8:15	4	5	7	10	4	30
8:15 - 8:20	4	4	6	9	4	27

* values are fabricated for illustration purposes

4.4.2.3. Option Two: Vehicle Trajectory Calculation Procedure

The second option to compute corridor travel time is the vehicle trajectory method. The experienced travel times serve as better estimation of the temporal-spatial relationships among the heterogeneous groups of traffic flows within a traffic system. Therefore, the aggregation of segment travel time should not be conducted at the same time instant but should adhere to the vehicle trajectories. Abiding this concept, the computation of experienced travel time is the summation of segment travel times predicted along the path of the trip at a series of time sequences.

Table 8: Illustrative Example for Calculating Experienced Travel Time

Time Interval	Segment Travel Time (min)					Experienced
	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5	Corridor
7:00 - 7:05	2	3	5	6	2	
7:05 - 7:10	3	3	6	8	3	
7:10 - 7:15	4	5	7	10	4	
7:15 - 7:20	6	6	9	13	5	
7:20 - 7:25	6	7	10	15	6	
7:25 - 7:30	7	8	11	17	7	27
7:30 - 7:35	9	10	13	20	9	
7:35 - 7:40	10	11	13	22	9	
7:40 - 7:45	8	9	10	23	7	
7:45 - 7:50	8	9	10	20	7	41
7:50 - 7:55	8	9	10	21	8	
7:55 - 8:00	8	8	8	17	7	
8:05 - 8:10	7	7	6	14	6	48
8:10 - 8:15	4	5	7	10	4	
8:15 - 8:20	4	4	6	9	4	53

* values are fabricated for illustration purposes

Table 8 shows an example calculation routine using the vehicle trajectory method. For the same time interval 7:00:00-7:05:00, the 2-minute segment travel time reflects the average time for all the vehicles to travel through the first segment. We assume a vehicle reach the start of the corridor at the midpoint of the interval (referenced as the onset time for calculation for all intervals). Then it takes the vehicle 2 minutes to reach the beginning of the second segment at 7:04:30. Since this time still falls within the interval 7:00-7:05, the vehicle should still experience the same traffic conditions within the interval on segment 2 according to the assumption of constant conditions defined earlier. Then 3 more minutes have been added for the vehicle to complete the first two segments, and the departure time into the third segment is summed to 7:07:30.

Following this trajectory, the virtual vehicle takes 6, 10 and 6 minutes to complete segment 3, 4 and 5 sequentially. Adding these segment travel time up with the initial 2.5 minutes (varies depending on interval length), the vehicle completes the corridor at 7:29:30 using a total of 27 minutes. The same procedure applies to all other trajectories.

As already shown in Table 8, time gaps between calculated experienced travel time outputs are unavoidable, especially under rising traffic demands. This phenomenon in the vehicle trajectory calculation is partially brought about by the discontinuity nature of the initiating time of the procedure. The other unavoidable situation is overlapping of end time windows of multiple vehicle trajectories starting from the same time slot, as commonly observed during the downtrend of traffic congestions. To solve these two moderately encountered cases, a deterministic linear interpolation technique is applied. To simply put, the missing cells are filled by interpolating the most recent available values before and after. The linearization is a simple yet reasonable algorithm to impute missing outputs. Such a technique approximates the dynamics of the changing traffic conditions within a limited time span. For omitted values, the following rule is applied:

$$ETT_i = \frac{ETT_p - ETT_q}{(p-q)} + ETT_p \tag{30}$$

where ETT_i labels experienced travel time value at time step i while p and q designate the previous and the next time step with valid output values respectively.

In contrast to missing outputs, overlapping outputs are also possible. For multiple outputs ($ETT_i^1, ETT_i^2, \ldots ETT_i^k$), the below equation is used:

$$ETT_i = \frac{1}{k}\sum_k ETT_i^k \tag{31}$$

In general, for only about 5% of the time, either an omission or a dual entry is encountered in the case of this study. As one might expect, the longer the length of the corridor is, the more omissions could occur.

4.4.3. Direct Forecasting

The two-stage method above involves the prediction of segment travel times. This approach will be compared with a direct forecasting approach. The direct forecasting method predicts the corridor travel time without the additions of segment travel times. This alternative method follows the procedures below.

To obtain valid corridor travel time using AVI data source, we apply the travel time extraction process on the pair of AVI stations that are the start and the end stations of the corridor. For the corridor travel times are directly extracted from one pair of AVI station, we herein term them *extracted corridor travel times* to distinguish from the *calculated corridor travel times* as in the two-stage forecasting method. In the test corridor in this experiment, AVI station 29 is the start of the corridor and station 34 marks the end. So the corridor travel times extracted by pairing AVI station 29 and 34 are the travel times of all the vehicles that actually completed the entire corridor. This process has led to much fewer samples. It is observed that only there are as few as 3-4 samples even during a 5-minute peak period whereas this number can rise as high as 60-70 for the samples captured in a segment.

Once the corridor travel times are extracted, a neural network model is then defined and trained on dataset C to learn the mappings between traffic conditions (characterized by speed and volume) and corridor travel time. After training, the neural

network model is tested against dataset D. In each day, predictions on corridor travel times for 5 minute ahead are made by using the already trained ANN model and the most current traffic data. No further calculation procedures are needed for direct forecasting.

5. RESULT COMPARISONS AND DISCUSSIONS

This section presents the results and observations of the ANN model predictions on travel time. Based on proper setups of the experiment for segment travel time prediction, we can compare, in the first part, the prediction models in three aspects: (1) training algorithms, (2) prediction horizons, and (3) incident impacts. In the second part, two corridor prediction methods and three measures of corridor conditions are compared and their implications as well as potential applications are discussed.

5.1. Modeling Segment Travel Time

Table 9 shows the average performance of the five ANN models proposed in the study along with a number of accepted MAPEs in other studies of short term travel time prediction. In these literatures, the performances achieved by different research vary. As a rule of thumb, a measure (MAPE) in the range of 5-10% is generally considered "good." Therefore, all the five neural network models are trained to learn the function of the traffic system under study within a satisfactory range. Of course, corrupted data were left out from the analysis in most studies while this study incorporates an automated data interpolation technique, correcting a small portion of missing or corrupted data. When massive data corruption occurs, the performance of network models in this study may degrade significantly. It is expected the errors could be brought down to a lower level if invalid data are deleted either automatically or manually. However, it is the intention of this research to train the models with the ability of handling partial data corruption.

Table 9: Comparison of ANN Model Performance in Respective Studies

Research	Model	MAPE (%)	Data Size
This Study	BPNN	7.3	208 peak periods including incident situations
	TDNN	7.1	
	MNN	7.4	
	SSNN	6.7	
	ExtSSNN	6.5	
	Naive Prediction	10	
	Historical Median	17.5	
Park and Rilett (*16*)	Kalman filter	6.2	231 days including non-peaks
	Spectral Basis FNN	7.2	
	Modular FFNN	8.1	
Wei and Lee (*14*)	FFNN	<20	9 weekdays
Wu (*34*)	Support-Vector Regression	3.9-4.4	5 weeks

Note: FFNN = Feed-Forward Neural Network

5.1.1. Training Algorithm Comparison

The first task is to train the proposed models using Levenberg Marquardt training algorithms with and without Bayesian regulated hyper-parameters and to evaluate their respective responses. All the ANN models are considered in the testing. In order to compare the two algorithms for the networks, we conduct a student t-test to determine if one algorithm yields significant different prediction performance over another. For this reason, in addition to model designs, we prepare the testing as follows.

First, the chances for each training algorithm to learn a problem are equalized. In neural network training, early stopping is one of the countermeasure to avoid over-fitting and thus to achieve better generalization (*6*). As briefly mentioned earlier, in the case of using LM as training algorithm, a validation set is usually employed to signify for early

stopping; while using BRLM, no validation sets are required. To avoid biases caused by the early stop criteria, we take out the validation set from LM training, and set the maximum training epochs 50 as the universal early stopping rule in this comparison. As a result, both algorithms have equal numbers of chances to calibrate both models.

Then, we have to randomize the initial model parameters (i.e. weights and biases) of all models five times before training with both algorithms. Hence, a total of 10 replicas (5 for one algorithm) are trained for each model. It should be noted that the randomization of the initial network parameters is important in that the dependence of model performance on its initial state can be addressed.

With the above conditions, we test the average performances of both ANN models that are achieved by both algorithms. Table 10 shows the two-sided t-test on LM and BRLM algorithms using test dataset B-I, B-II and B-III. The measure of model performance used is MAPE.

In general, the networks trained by BRLM algorithm achieve better generalization abilities because the average values of MAPEs obtained by using BRLM across all scenarios are lower than those achieved by the LM method. However, P-values distinguish the differences made by the two algorithms. For the two static networks, there are 5 out of 6 scenarios that the t-test cannot be concluded with any significant difference between the models trained by the two algorithms. On the other hand, P-values derived from t-test on dynamic ANN models have shown that networks are trained with significantly better performances by BRLM algorithm than those by LM algorithm. The conclusion is made at a 90% confidence level. In fact, the Levenberg-

Marquardt algorithm is a large-step-sized algorithm, the refined approximation of performance gradient is hence very difficult (*18*).

Table 10: T-Test in Comparing Training Algorithms Based on MAPE

Network Class	Net	Test Dataset B-I			Test Dataset B-II			Test Dataset B-III		
		LM	BRLM	P-Value	LM	BRLM	P-Value	LM	BRLM	P-Value
Static	BPNN	7.6%	7.0%	0.146	7.8%	7.1%	0.172	7.5%	6.9%	0.151
	MNN	7.3%	7.1%	0.183	7.7%	7.2%	0.019	7.1%	7.0%	0.536
Dynamic	TDNN	7.4%	6.8%	0.042	7.9%	7.2%	0.044	7.2%	6.6%	0.067
	SSNN	7.4%	6.8%	0.022	7.9%	7.1%	0.038	7.2%	6.7%	0.027
	ExtSSNN	7.8%	6.6%	0.009	8.3%	7.2%	0.007	7.6%	6.4%	0.013

Note: 1. Percentage values are average MAPEs from 5 replicas.

2. P-values are calculated from two-sided T-test, assuming unequal variance.

Note that the stopping criterion unified here is meant to equalize the number of times for each algorithm to adjust model synaptic weights. Yet, the criterion may not be the most suitable ones for the algorithm to achieve optimal performance. Since the scope of this thesis is to identify a more promising training algorithm rather than to investigate the most suitable stopping criteria, we may already have enough evidence to conclude that BRLM significantly improves the prediction performances of dynamic neural networks, and is likely to improve the performance of static networks. Therefore, other comparisons below will base on the superior performance achieved by BRLM.

As an additional benefit, the Bayesian regularized training provides an indicator (i.e. number of effective parameters) of whether an ANN model has been over-designed for a problem. However, the BRLM algorithm consumes generally longer computer times to learn the problem. Table 11 summarizes the training time and the number of effective parameters achieved by each algorithm. For all models except MNN, BRLM

costs on average 126% more training time than LM algorithm. Since MNN has the lowest model complexity (i.e. 161 model parameters in design) and the model has been further divided into several sub-models, it is not surprising that LM algorithm has not gained any advantages in reducing the training time. After BRLM training, the indicator – number of effective parameters – has shown averagely 87% of the parameters have been utilized effectively to learn the underlying problem. The number indicates the models have been designed properly. Of course, too few parameters being used effectively means the model is too complex, and should be redesigned with fewer number of model parameters. On the contrary, all the parameters being used by the model learning may point to the possibility of over-simplicity of model design.

Table 11: Comparison of Training Algorithms

Specifications		LM		BRLM		
Net	# of Design Parameters	Training Time (min)	# of Effective Parameters	Training Time (min)	Minium # of Parameters Attempted	# of Effective Parameters
BPNN	251	0.7	251	1.7	87	210
MNN	161	1.1	161	0.9	42	126
TDNN	491	1.5	491	3.8	51	420
SSNN	313	86.8	313	171.3	132	303
ExtSSNN	383	139.9	383	294.2	63	345

The number of effective parameters will finally converge as the errors converge. This process echoes the Learning-Generalization Dilemma. Figure 22 illustrates the adjustment process of the effective weight parameters by using Bayesian regularized algorithm. However, it should also be noted that the number of effective parameters in a network has no direct associations with the performance superiority, but it is instead a

reference to how many effective parameters might be required in model design to ensure training convergence(*18*).

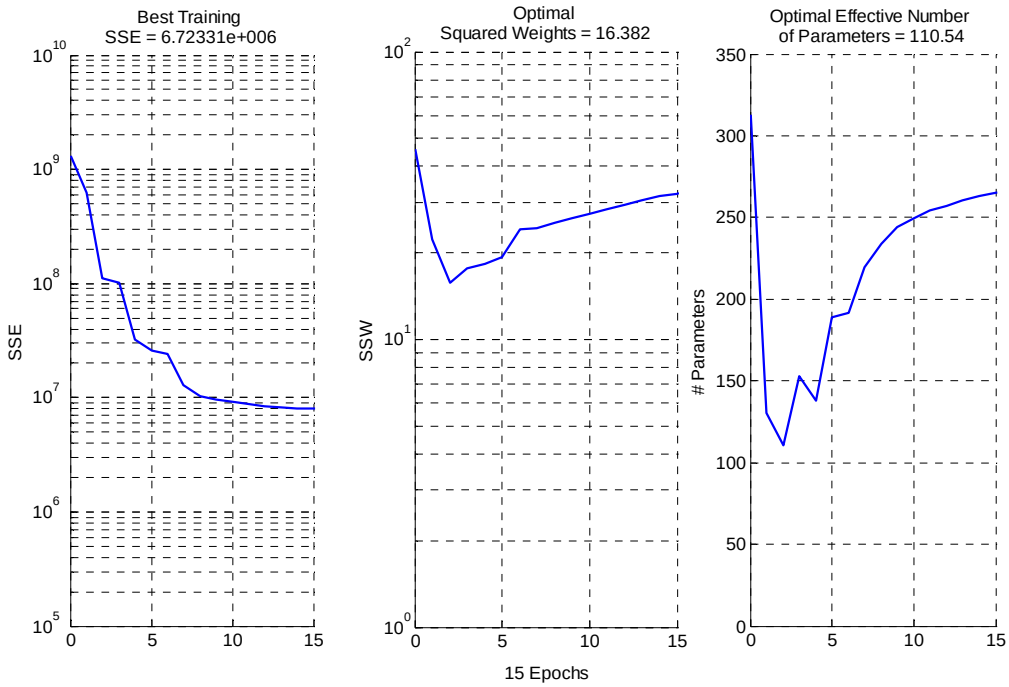


Figure 22: Effective network parameters during the training period by Bayesian Regulated Levenberg-Marquadt algorithm.

A brief note on training time should be brought to readers' attention. Table 11 shows that the computation time has increased dramatically in both algorithms while training the neural network with recurrent structures, such as SSNN and ExtSSNN. It is majorly due to the ongoing adjustment of Bayesian hyper-parameters in a recurrent manner. And the time is expected to be elongated as we increase the number of training samples.

5.1.2. Prediction Ability at Different Temporal Horizons

The second task is to determine if the artificial neural network methodology can still be implemented in a satisfactory manner at a further horizon (i.e. 15 minute in the future) and to find out by what margin the horizon factor degrades the performance of the model prediction. As argued already, the predictability of a model for travel time is a crucial indicator of the model's practicability. Therefore, it is important to investigate the prediction ability at longer horizons as well.

In this task, despite the computational intensity of BRLM algorithm, we adopt the procedure to achieve better model performance for dynamic networks as well as static networks to compare the prediction ability at different time horizons. The following Table 12 summarizes the results of MAE, MAPE, RMSE, and NRMSE of the 5 ANN models and the 2 baseline models for this purpose. Note that, all ANN models have been trained with 5 different initial values to minimize the dependence of neural network training on initial values. Values tabulated in the table are averages of the five outcomes of each model except the two base models, since there are no training mechanisms for the two baseline models to speak of.

According to the overall observations attained by this experiment, the increase in prediction horizon results in 67-76% of additional errors averaged across all models excluding historical median model (i.e. HM). It is not surprising that the historical median method is not affected by the prediction horizon, certainly since the method estimates travel times based on the historical data rather than predicts travel times based on current data inputs. This also implies that HM may show better performance as the

prediction horizon becomes too long for other models to make acceptable prediction accuracy.

Table 12: Model Performance at Different Prediction Horizons

Prediction Horizon		Test - Data B-I					
	Net	**Mean MAE (sec)**	**Mean MAPE (%)**	**Std MAPE (%)**	**Mean RMSE (sec)**	**Mean NRMSE (%)**	**Std NRMSE (%)**
5 minute	BPNN	23.68	6.983	0.192	38.48	6.540	0.585
	TDNN	22.26	6.789	0.191	36.54	6.204	0.296
	MNN	23.37	7.065	0.263	34.52	5.838	0.091
	SSNN	22.57	6.807	0.131	34.29	5.837	0.187
	ExtSSNN	21.65	6.633	0.180	33.25	5.686	0.278
	NP	33.89	10.010	-	51.47	8.710	-
	HM	59.39	17.530	-	83.54	14.190	-
	Net	**Mean MAE (sec)**	**Mean MAPE (%)**	**Std MAPE (%)**	**Mean RMSE (sec)**	**Mean NRMSE (%)**	**Std NRMSE (%)**
15 minute	BPNN	42.1	12.607	0.154	63.5	10.890	0.380
	TDNN	38.6	12.012	0.278	59.1	10.149	0.375
	MNN	42.0	12.683	0.104	60.8	10.428	0.078
	SSNN	38.7	11.697	0.146	55.2	9.471	0.239
	ExtSSNN	38.3	11.674	0.262	55.8	9.625	0.217
	NP	58.7	17.370	-	87.6	14.800	-
	HM	59.4	17.530	-	83.5	14.190	-

Figure 23 and Figure 24 are the plots of the actual travel time profiles and the predicted travel time profiles by various ANN models at both 5 minute and 15 minute horizons. Both days are free of incidents and free of data corruptions along the study segment during the study period. However, differences are as follows. With the smaller horizon, the ANN models track the changes of traffic conditions reasonably well. However, the fittings become more ragged and sometimes fluctuate to opposite directions of the change of travel time values, indicating a less stable behavior of ANN

models at a longer prediction horizon. Fortunately, SSNN model demonstrates a relatively stable behavior in comparison to BPNN, TDNN and MNN models, since it does not fluctuate as much as these models between time step 70 to 80 in Figure 24. Such a stable behavior can be ascribed to the context layer, which endows the network with the ability to learn the evolution of the internal states of the object being learned.

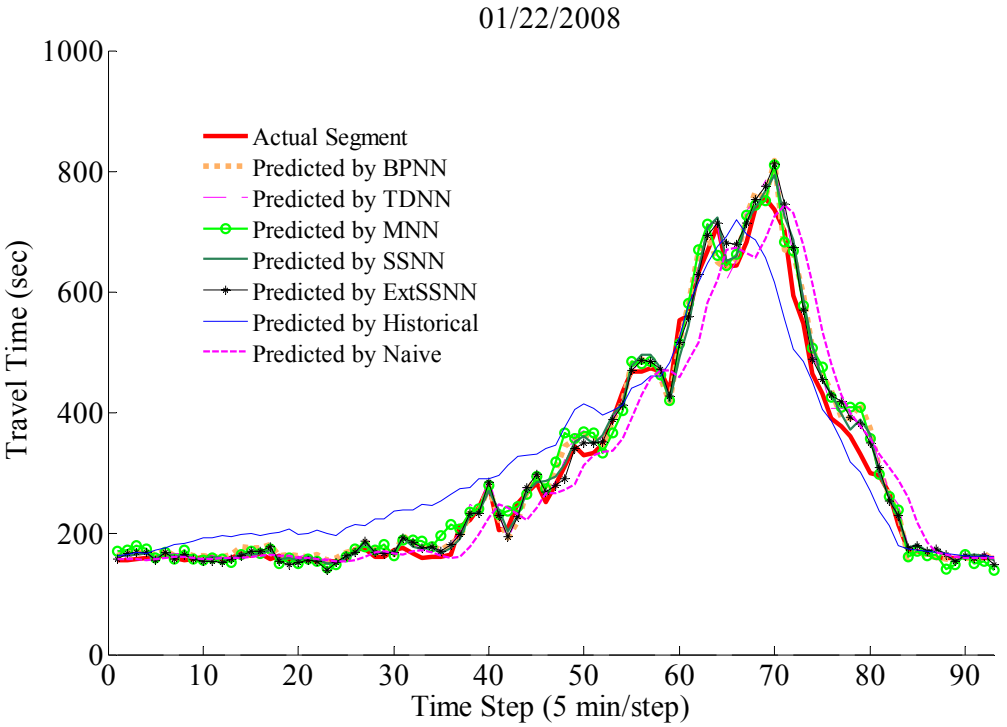


Figure 23: ANN model performance at 5-minute prediction horizon.

The naïve prediction method provides predictions based on previous conditions. Therefore, it demonstrates abilities of accurate predictions under non-congested conditions where the actual travel time profile stays flat. In congested periods, the predictions always lag behind the real-time changes of traffic conditions, and the lags

expand as the prediction horizon increases. The profile derived historical median method remains the same for all the test days, because only one median value is calculated for one time interval.

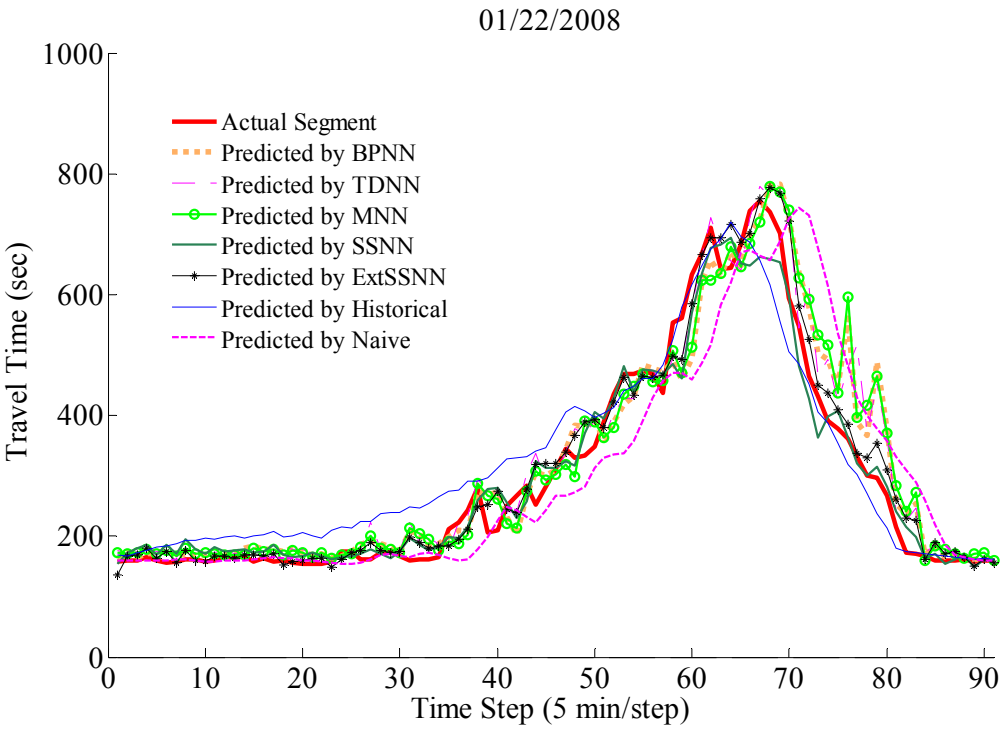


Figure 24: ANN model performance at 15-minute prediction horizon.

Visual inspections of model performances for individual days may reveal certain details that are not otherwise perceived. However, the distinctions of systematic patterns of the predictions to target values on all the test data should also be investigated to uncover model differences at macroscopic scales. Figure 25 and Figure 26 plot the forecasted travel time values against actual travel time values. The plots are similar to residual plots but are with the horizontal axes tilted by 45 degrees. Similar techniques to residual analysis shall apply. But these plots have the advantages of displaying the

values of travel times which are significant indication of whether they fall into peak or non-peak hours. The test dataset B-I is used to for all the plots.

Let us first make comparisons among various prediction models. By comparing the concentration levels of the prediction-versus-actual distributions shown in the plots of Figure 25, artificial neural network models have shown more consistent performance than the other two models. The naïve prediction model appear to track the target travel times better at lower range of values, meaning the model may perform well during non-peak hours. And the error of historical mean method is so widely distributed that its prediction performance is unacceptable.

By comparing model performances across prediction horizons, we can see that the sample points disperse much wider at the 15-minute case than at 5-minute case. This indicates the performances of the models in general are more inconsistent when the predictions are made further ahead in the future. Although MAPEs of naïve prediction and historical mean methods are close in the Table 12, the plots in Figure 26 show naïve prediction is better in terms of performance consistency.

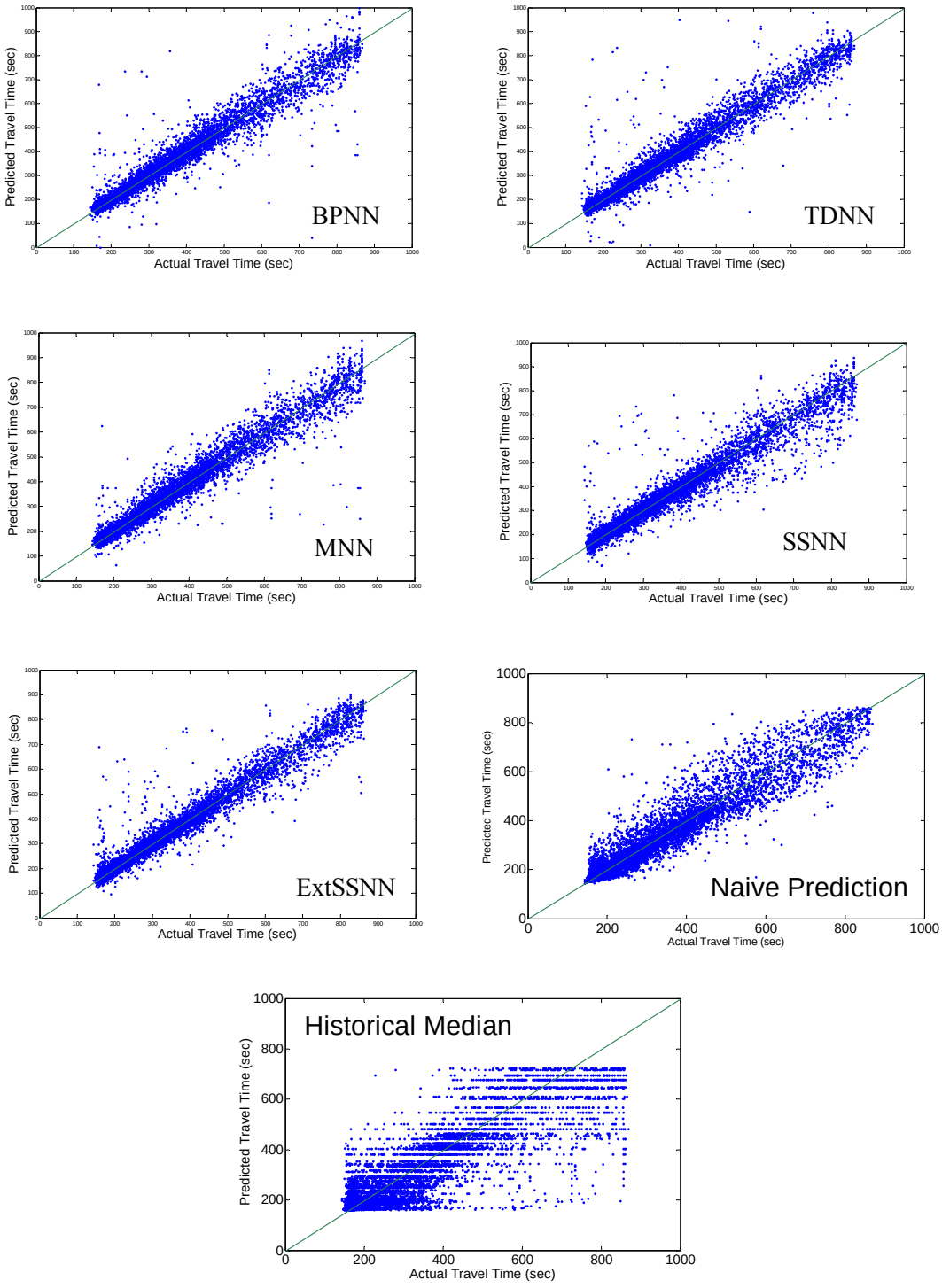


Figure 25: Plots for predicted and actual travel times at 5-minute horizon.

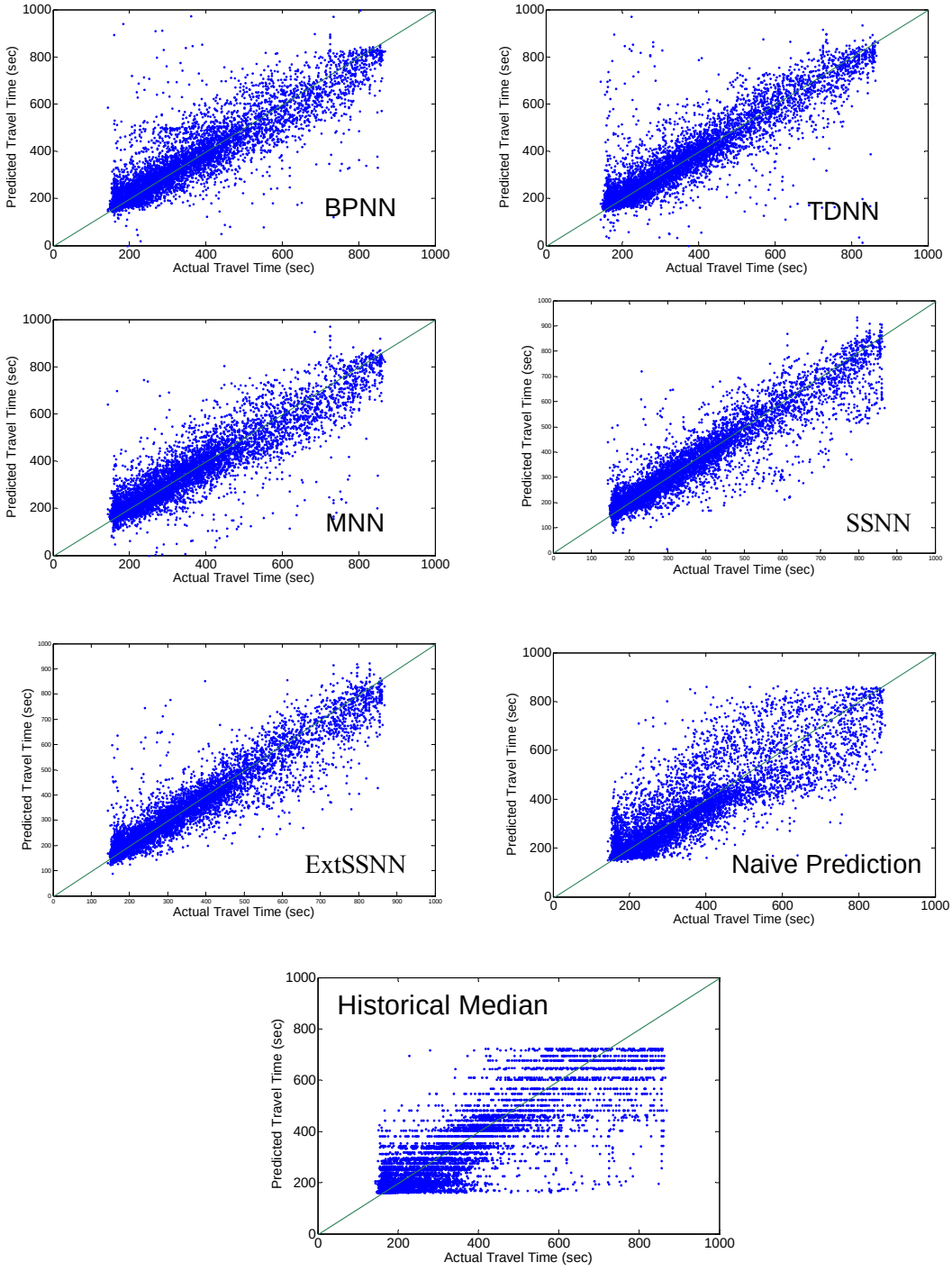


Figure 26: Plots for predicted and actual travel times at 15-minute horizon.

It needs to be noted that modeling segment travel time is the first step of the two stage process to develop a reliable corridor travel time prediction methodology. Instead of taking additional time to repeat the process, we therefore focus on only the predictions at 5 minutes into the future for both segment and corridor travel time modeling where the problem formulations and calculation procedures can be replicated.

5.1.3. Incident Impact Modeling

Another primary task of this study is to investigate the incident impacts on prediction performance as well as the possibility and the methodology to incorporate this piece of information into the neural network modeling process. The measures of performance for various models being tested are tabulated in Table 13. The test Dataset B-II and B-III are incident affected data and incident free data as described in detail in earlier sections.

Table 13: Performance of Prediction Models on Dataset B-II and B-III

Prediction Horizon	Model	Test - Data B-II				Test - Data B-III			
		Mean MAE (sec)	**Mean MAPE (%)**	**Mean RMSE (sec)**	**Mean NRMSE (%)**	**Mean MAE (sec)**	**Mean MAPE (%)**	**Mean RMSE (sec)**	**Mean NRMSE (%)**
5 minute	BPNN	25.6	7.134	42.3	7.139	23.0	6.929	37.1	6.324
	TDNN	25.0	7.249	43.7	7.361	21.3	6.624	34.0	5.786
	MNN	25.2	7.150	39.6	6.645	22.7	7.035	32.7	5.548
	SSNN	24.1	7.130	38.6	6.565	22.0	6.691	32.7	5.575
	ExtSSNN	24.1	7.199	39.6	6.785	20.8	6.429	31.0	5.290
	NP	37.3	10.810	57.1	9.490	32.6	9.700	49.3	8.410
	HM	61.7	17.310	86.7	14.380	58.5	17.620	82.3	14.120
15 minute	BPNN	50.1	14.215	78.6	13.278	39.2	12.027	58.0	10.029
	TDNN	44.9	13.278	71.4	12.121	36.3	11.555	54.7	9.438
	MNN	48.8	13.961	74.3	12.542	39.5	12.222	55.9	9.665
	SSNN	44.3	13.026	66.9	11.298	36.6	11.217	51.0	8.813
	ExtSSNN	45.5	13.513	71.5	12.227	35.7	11.010	50.1	8.687
	NP	63.9	18.570	94.2	15.680	56.7	16.920	85.1	14.470
	HM	61.7	17.310	86.7	14.380	58.5	17.620	82.3	14.120

It is clearly shown that all ANN models as well as the NP and HM models are affected by the presence of incidents. At 5 minute prediction horizon, the MAPEs and the NRMSEs of all the model outputs have been raised by averagely 6% and 17% (6.5% and 21% for only ANN models) from test Data B-III to B-II. The increase has been enlarged to 13% and 24% (17% and 31% for ANN models) at 15 minute prediction horizon. By comparing all error measures across the two test datasets, we may find that the increase in errors due to the presence of incident is consistent. This result shows that the presence of incidents generally degrades the prediction performances of all these models except HM method. And the impacts become more severe when the prediction is made more time steps into the future. However the impacts are rather marginal, especially for MAPEs, a closer examination is therefore needed. A two-sided student t-test with unequal variance assumption is conducted, and shown in Table 14, for this identifying the significance of incident impacts based on MAPE values.

Table 14: T-Test of Incident Impacts on Prediction Performance

Network Class	Net	5 minute			15 minute		
		Data B-II	Data B-III	P-Value	Data B-II	Data B-III	P-Value
Static	BPNN	7.1%	6.9%	0.144	14.2%	12.0%	2.94E-04
	MNN	7.2%	7.0%	0.532	14.0%	12.2%	2.13E-06
Dynamic	TDNN	7.2%	6.6%	0.026	13.3%	11.6%	1.64E-05
	SSNN	7.1%	6.7%	0.026	13.0%	11.2%	8.02E-06
	ExtSSNN	7.2%	6.4%	0.001	13.5%	11.0%	5.92E-06

Note: H_0: Data B-I = Data B-II; H_a: Data B-I ≠ Data B-II. Unequal variance is assumed.

At 5 minute prediction cases, incidents have increased the prediction errors for all models. For all three dynamic networks, P-values suggest the presence of incident

significantly affects the model performances. Nevertheless, for static networks, the test cannot reject the null hypothesis that the models perform equally under both incident and non-incident conditions. This difference in conclusion may be attributed to the difference in the existence of delay mechanism. A dynamic neural network have incorporated the ability to "remember" either the inputs or the states of the network in the past time steps. Thus, they tend to produce a smoother travel time profile. Subjecting to incidents however, the actual travel times may be increased dramatically from one step to another. Static networks, which have no considerations of temporal dynamics, tend to fit the sudden spikes in actual travel time profile better.

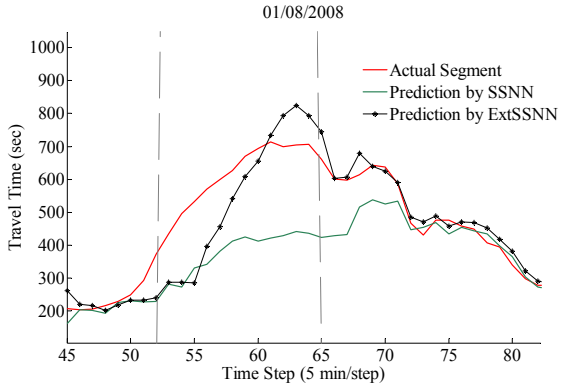
At 15 minute prediction cases, the t-test has confirmed for all the networks the significance of the negative impacts of incidents. Based on the results, we don't recommend the predictions be made for incident days at 15 minute prediction horizon and onward. But ANN models may be fine tuned to achieve better prediction performance for incident-free days at the same horizon.

As a highlight, the ExtSSNN model was originally developed with an additional module that attempts to account for the incident impacts on segment travel times. Nevertheless, the network architecture of this extended version of SSNN yields no improvements for the incident-affected dataset. According to the performance measures, the ExtSSNN, which has incident inputs, performs slightly worse than the SSNN, which has no incident inputs, at both 5 minute and 15 minute horizons. As according to visual inspections, the inclusion of incident information seems to have random impacts on the

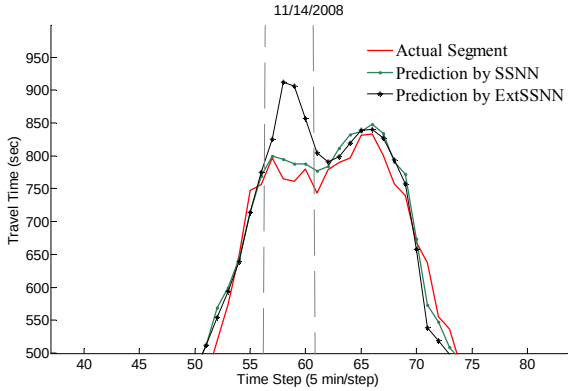
predictions. The actual and the predicted travel time profiles by ExtSSNN and SSNN are plotted in Figure 27 to facilitate the visual examination. And discussions are as below.

In most cases, we observe very minor differences of predictions made by the two models, similar to Figure 27-(c). Random effects of incident inputs appear in all other cases. Four causes might contribute to this result. First, there are simply too few data points for the network to learn well. Of all the incidents during the full course of 2008, only 72 events occurred at the study segment within the hours (12-8pm) of a day under study. And only 60% of these incident events are used in training while the rest are used in testing. Secondly, the speed and the volume of a traffic stream at a specific time *t* intrinsically reflect the conditions of the system already. Hence, the incident impacts may be roughly perceived by the detector data used as general inputs. Thirdly, the nature of a dynamic training procedure presented in this study requires input data with time-variant characteristics. However, the incident events are logged as individual records in Houston incident database so that incident states are not updated on a continuous basis. The incident inputs usually provide constant information with which the network learning is not enhanced. Lastly, the quality of incident information obtained from the TMC is not guaranteed since the practice of manual recording incident information involves high degrees of subjectivities and non-standardization. For example, the perceptions of different operators recording the information vary significantly in some cases. In sum, although the architecture of ExtSSNN is designed gracefully and the incident inputs are interpolated deliberately, the extra network module adding to a base SSNN module supplies no consistent enhancements on predicting travel times under an

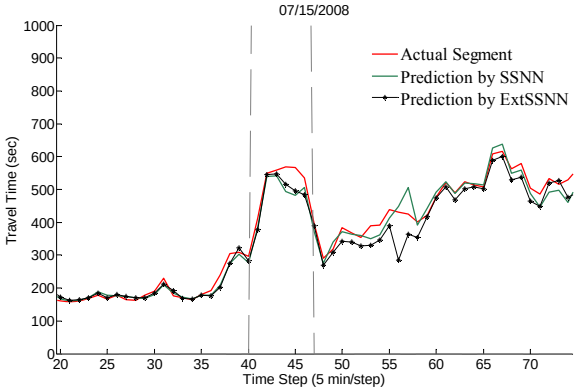
incident condition. And, SSNN model alone yields satisfactory prediction accuracies under both incident and incident-free conditions.



(a) Positive impact of incident information on performance.



(b) Negative impact of incident information on performance.



(c) Unobservable impact of incident information on performance.

Figure 27: Various impacts of incident information on performance.

5.1.4. Model Comparisons

With the above major aspects compared, we can identify a segment travel time prediction model that can be applied in corridor travel time prediction. Table 15 summarizes the means and standard deviations of all the performance measures. For each ANN model, 5 replications are trained with different initial values. The trainings are performed by BRLM algorithm at 5 minute horizon and the trained replicas are tested with incident-free weekdays at 5 minute horizon. All the scenarios conducted in this experiment are listed in appendix Table A-1 and Table A-2.

Table 15: Performance of Prediction Models on Dataset B-III

Model	Test - Data B-III							
	Mean MAE (sec)	**Std MAE (sec)**	**Mean MAPE (%)**	**Std MAPE (%)**	**Mean RMSE (sec)**	**Std RMSE (sec)**	**Mean NRMSE (%)**	**Std NRMSE (%)**
BPNN	23.00	0.778	6.929%	0.24%	37.11	3.795	6.324%	0.65%
TDNN	21.45	0.330	6.692%	0.21%	34.43	0.808	5.864%	0.16%
MNN	22.71	0.535	7.035%	0.26%	32.70	0.456	5.548%	0.09%
SSNN	22.01	0.786	6.691%	0.13%	32.73	1.490	5.575%	0.25%
ExtSSNN	20.76	0.802	6.429%	0.18%	30.96	1.658	5.290%	0.30%
NP	32.62	-	9.700%	-	49.34	-	8.410%	-
HM	58.51	-	17.620%	-	82.34	-	14.120%	-

Note: 5 replica for each ANN model, and are trained by BRLM algorithm and tested at 5 minute prediction horizon
Color scheme is scaled according to cell values in the respective columns

Across all the prediction models, the ExtSSNN and the SSNN are the two relatively better performed models in the experiment, jointly according to mean of all performance measure. MNN and TDNN appear to be the next promising models in 5 and 15 minute horizon respectively. Of course, ranking the models by 1 or 2% of margins is not very meaningful, especially since other models are observed to outperform SSNN

and ExtSSNN models at some individual days. In fact, a student t-test also cannot consistently conclude the significant difference between any two of all the models (see Appendix Table A-3 and Table A-4). However, the point here to make is that using SSNN model and the models based on SSNN structure can explicitly incorporate the temporal and spatial dynamics of a traffic system into the network design. The gain of using SSNN type of models is actually more than just a few percent of increase in prediction performances. Therefore, we promote to adopt SSNN based neural networks to develop travel time prediction models when speeds and volumes are used as inputs.

5.2. Predicting Corridor Travel Time

Based on the argument above, the SSNN model is used to accomplish the task of computing (in fact predicting) corridor travel time. With exactly the same structures, five SSNN models for the five respective segments are trained in batch mode to learn the dynamic relationships of traffic input data (i.e. speed and volume) and travel time. A total of 20 detectors are considered along the study corridor (i.e. from AVI station 29 to 34), of which 19 are valid. Again, the training dataset is dataset C. All models are trained on dataset C using BRLM training algorithm for a total of 5 times. After completing each training, the predictions are made on testing data D at one time step ahead, and the measures MAE and MAPE are calculated to estimate the ANN performances of predicting the corridor travel times. For comparison, the direct forecasting method is also trained on dataset C and tested using dataset D, and the prediction errors are then calculated.

First of all, it should be noted that the errors are not calculated against one but three respective bases. The reason for comparing on separate grounds is that these methods are measuring different perspectives of a traffic system, as discussed in detail in previous sections. The aim here is to illustrate the ANN performances in predicting these different perspectives. Therefore, the predicted snapshot (calculated by snapshot method) and experienced travel time (calculated by vehicle trajectory method) are compared with actual snapshot and experienced corridor travel times respectively; whereas the predicted corridor travel time (calculated by direct forecasting method) is compared with extracted corridor travel time.

Based either on the predicted values of segment travel times or the directly predicted corridor travel times, we then compare these with actual values, yielding MAPEs of 15.36%, 15.20% and 33.45%, as tabulated in Table 16.

Table 16: ANN Performances of Predicting Corridor Travel Time

Prediction Methods	Measures	Mean MAE (sec)	Mean MAPE (%)	Std MAPE (%)
Two Stage -- Snapshot	Snapshot Corridor Travel Time	189.2	15.36	2.15
Two Stage -- Vehicle Trajectory	Experienced Corridor Travel Time	186.0	15.20	2.08
Direct Forecasting	Extracted Corridor Travel Time	398.9	33.45	7.77

The performance levels of the first two methods merely fall in the acceptable range. The reason for this can be ascribed to the fact that fitting errors are generated at each of the segment, and the summation process of the two-stage methods inevitably accumulates these errors. Despite the accumulation of errors, the first two methods

clearly outperform the direct forecasting method by a large margin. As argued before, the neural network training requires correct input-output mappings. For a corridor stretching 12.25 miles, any free-flow traffic may take as few as 10 minutes or as long as 40 minutes to complete the corridor. It implies that the speed and volume inputs need to be obtained from a varying time interval to reflect a varying corridor travel time. Since this is very difficult to implement, the experiment in this case adopts a fix interval (i.e. 5 minute). Therefore, by using the direct forecasting technique, the neural networks inherently learn the wrong mechanism if the pairings of input-target are not done dynamically according to the varying corridor travel time.

To visualize the SSNN performance in such applications, Figure 28 illustrates the fittings of two-stage predictions on actual calculated corridor travel times in a typical example day (i.e. Jan 21th, 2008). Overall speaking, the five SSNN models provides acceptable accuracy on predicting both measures of calculated corridor travel times, even though the additions of segment travel times along the corridor slightly increases the errors of predictions at the peak period around time step 70 and thereafter. Based on the performance measures and visual observations, we argue that the application of ANN approach on both methods of calculating corridor travel time is possible and satisfactory in this case.

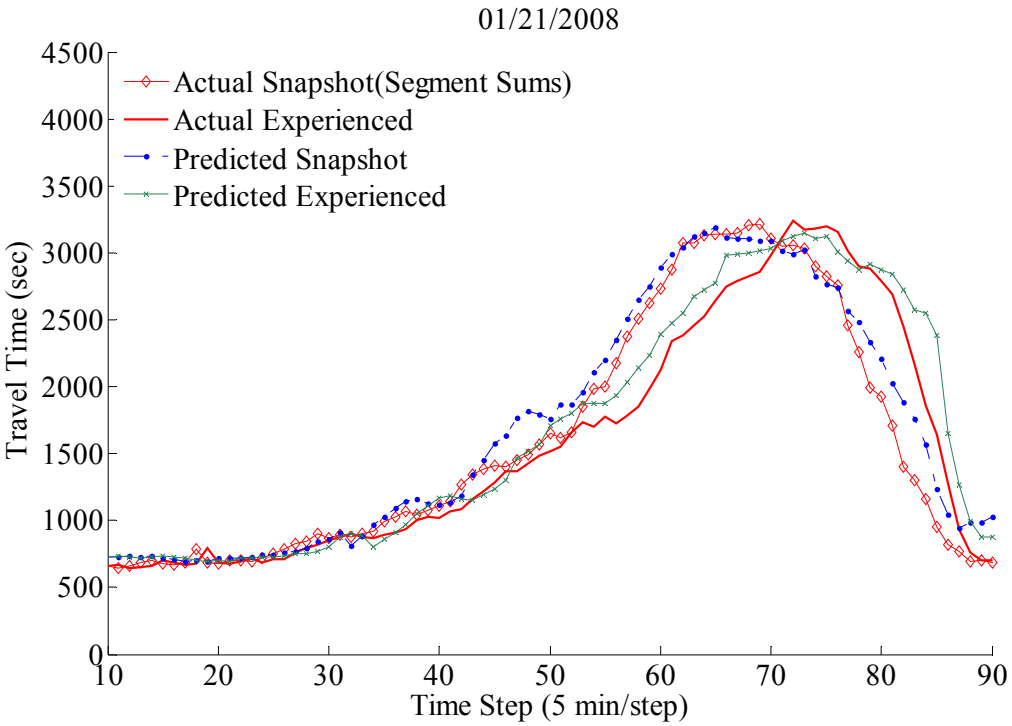


Figure 28: SSNN performance on two-stage forecasting.

Figure 29 display the plot of actual observed corridor travel time and predicted corridor travel time. As a typical example, the graph clearly shows that the direct forecasting provides adequate performance to reconstruct the profile of observed corridor travel time. In this particular case, the prediction underestimates the actual corridor travel time around time step 45 but overestimates between 60 and 75. In general observation, the behavior of the prediction using direct forecasting method is not stable. The predictions deviate from the actual values in most of the days whereas there are also some good fits observed in a few other days. Additionally, we found out that the performance of direct forecasting method is relatively susceptible to detector failure. For the case in the figure, there are at least three detectors failed during time step 34 through

41. We suspect this might be one of the reasons why the prediction values become negative during this period.

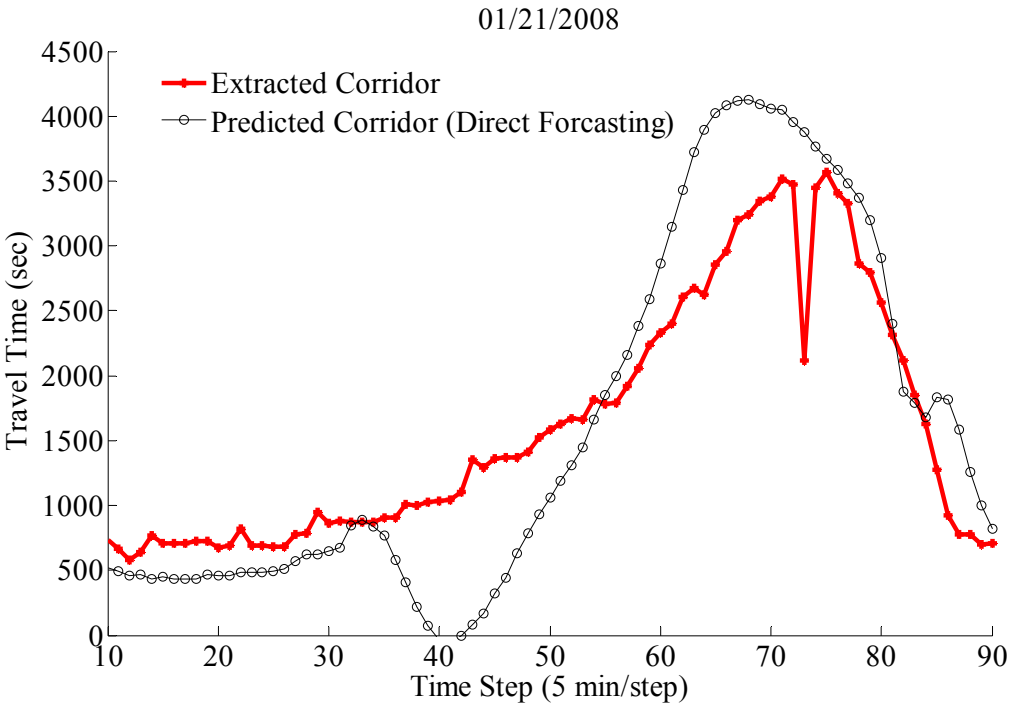


Figure 29: SSNN performance on direct forecasting.

There are some other interesting observations that are also worthwhile mentioning. First is the consistent pattern of snapshot travel times in comparison to experienced travel time. Close to or during non-peak hours, both measures of travel time are relatively equal at any time steps due to stable traffic conditions. Yet, the distinctions start to emerge along with the growth of traffic demands when the freeway system becomes congested and volatile. The differences are the largest as the congestion peaks. As always the case, the snapshot travel time profiles start to rise and fall earlier than the experienced travel time profiles (visually, snapshot is always to the left of the

experienced). This is not a surprising case since the experienced travel time can be obtained only after the trip has been completed, which implies these values are relatively obsolete to the current traffic conditions. As such, we conclude that experienced travel time values structurally underestimate the current traffic conditions during the time periods when congestions start to build up, while they over-estimate when congestions is dissipating.

Second, the application of neural networks to predict corridor travel time is not a trivial excise if not performed properly. To adequately train a neural network in either batch or incremental manner requires extensive computational resources as well as correct input-output mappings. While undertake direct prediction method using sources data like AVI (segment based travel time), we need to extract new corridor travel times measured by trips completed the new origin-destination (OD) pair. For each new OD pair, complete retraining has to be conducted for the new pair of input and outputs. Both data extraction and new model trainings are painstaking efforts that consume a lot of computer resources.

In addition, a practical note should also be made. For evaluation purposes, the experienced travel time may provide an alternative to quantify the traveler experiences in the past. However, with the predictive ability of the SSNN model, the experienced corridor travel time can be projected back to present time after being accurately predicted. The projection procedure is explained in arrow flows in Table 17. Using the projection made at each time step, the travelers can expect the future conditions instead of those in the immediate past, so they can plan the trip properly. For the purpose of this,

this piece of information ought to be disseminated at the starting point of the corridor and meet the individual travelers' needs of "knowing" the future of the freeway system.

Table 17: Present Projection of Experienced Travel Time

	Predicted Segment Travel Time (sec)					Predicted Corridor Travel Time (sec)		
Time	AVI 29 to 30	AVI 30 to 31	AVI 31 to 32	AVI 32 to 33	AVI 33 to 34	Snapshot	Experience	Projected Experience
:	:	:	:	:	:	:	:	:
16:30:00	103	380	401	426	910	2220	1889	2441
16:35:00	122	396	423	430	975	2346	2031	2487
16:40:00	107	381	441	434	1034	2397	2174	2428
16:45:00	111	386	471	433	1072	2473	2280	2409
16:50:00	93	427	508	421	1101	2550	2305	2410
16:55:00	95	493	545	354	1107	2593	2352	2426
17:00:00	124	609	590	279	1116	2719	2398	2672
17:05:00	127	681	650	241	1124	2823	2419	2845
17:10:00	121	723	732	162	1105	2843	2441	2931
17:15:00	127	739	794	114	1086	2860	2487	2914
17:20:00	132	744	824	102	1085	2887	2428	2804
17:25:00	143	780	797	101	1104	2925	2409	2567
:	:	:	:	:	:	:	:	:

6. CONCLUSIONS

6.1. Summary

In this study, a two-stage prediction methodology is developed to predict corridor travel time from spot speed and traffic volume data. In the first stage, data from three sources – AVI, radar and incident management database – are considered and five artificial neural network models are developed accordingly. Also the ANN models are compared against two baseline models. A total of 208 days of peak-hour traffic data are experimented on a stretch along the US-290 northwest corridor, which is bounded by AVI station 29 and 34.

In the first stage, the dynamic neural networks (i.e. TDNN, SSNN and ExtSSNN) are developed to incorporate the temporal-spatial traffic dynamics explicitly, and the static networks (i.e. BPNN and MNN) are designed to incorporate the spatial traffic dynamics only. Different aspects in the model development process are examined. Firstly, the impact of incidents on segment travel times and the usefulness of incident information in realizing these impacts are researched. Additionally, the differences between LM and BRLM training algorithms and the impacts of two prediction horizons (i.e. 5 minute and 15 minute) on the performances of all the ANN models proposed are studied. Certain implementation issues are discussed.

In the second stage, we investigated two measures of corridor traffic conditions (i.e. snapshot travel time and experienced travel time) and corresponding corridor travel time calculation methods (i.e. snapshot method and vehicle trajectory method). Then we developed the prediction procedure based on the SSNN model to compute the two

measures. Comparisons and discussions on their respective implications as well as potential applications are made.

6.2. Findings

The empirical studies in this thesis research have shown satisfactory prediction performance of all five neural network models. These ANN models significantly outperform the naïve prediction method and the historical median prediction method. In general, both the SSNN and the ExtSSNN models demonstrate slightly better performance in term of all error measures at both 5-minute and 15-minute horizons. Other findings are listed below:

- The Bayesian Regulated Levenberg Marquardt algorithm is found to improve the prediction abilities significantly for the dynamic networks as comparing to the Levenberg-Marquardt algorithm. However, BRLM also consumes much more computer times when it is applied to recurrent neural networks, such as SSNN and ExtSSNN in this thesis.
- Although the performance of the State-Space neural network does not consistently show significant difference from other networks, its unique structure provides insights on how to use such a "black box" to describe the temporal-spatial characteristics of a traffic system explicitly. The model is promising and deserves more research attention on the topic of short-term travel time prediction.
- By researching on the incident information as input data, we concluded that the incident information is redundant since speed and volume data have

already captured the inherent variations of traffic dynamics. And the incident information recorded in Houston TMC is not updated according to the change of incident conditions. Therefore, the information is not useful in dynamic training environment.

- The State-Space neural network model with snapshot travel time calculation method can predict prevailing corridor traffic conditions at next time instant with satisfactory result; while combining with the vehicle trajectory travel time calculation procedure, the SSNN can predict traveler experiences as well.
- By comparing the snapshot travel time and the experienced travel time, we found that the experienced travel time obtained at time interval *T* structurally underestimates the instant corridor conditions if the level of congestion is rising but overestimate when the congestion start to dissipate. On the other hand, the snapshot travel time can hardly represent traveler experiences under non-stationary traffic conditions. And both methods can be used jointly to enhance the prediction of different perspectives (i.e. the system perspective and the traveler's perspective) of a traffic system.

Finally, a brief note on practicality of this research should be made. Although the two-stage prediction method for corridor travel time performs merely satisfactory, the practical gain of this method is large. The method firstly provides a means for the corridor travel time modeling to be extended to even longer corridors without trapping in the pitfall of insufficient samples. Additionally, with the vehicle trajectory method

employed, the predicted experience travel time can be disseminated to travelers when they enter the corridor as an estimate of the trip experience. This piece of information is more favorable because it implicitly predicts all the traffic conditions that would be encountered by individual travelers.

6.3. Future Research

Notwithstanding the satisfactory performances of the neural network models developed in this study, they were all trained offline with historical data. For a method to be used by traffic managers on a daily basis, the models should be trained online adapting constantly to changes in either the underlying traffic process or the monitoring system, which collects the models' input and output data (*10*). Due to the ability to recognize the temporal pattern (*6*), the dynamic networks developed in this study can potentially achieve such complicated recognition by training at an incremental mode (online). Further attempts can be made to realize the incorporation of incremental training algorithms into the two-stage prediction method.

Additionally, as observed in the model training phase, dynamic networks with recurrent structures (e.g. SSNN) require a significant amount of computational effort, which naturally lowers the practicality of the model. By developing a partially connected SSNN model may help ease the intensity of training (or model calibration) process. However, the prediction performance might be affected by removing some connections in the model, and future research should not ignore this possibility.

After the analysis, the author becomes aware that a single prediction value may not be confident enough for traffic managers or individual drivers to use. Since the

artificial neural network are often performed with noise, a reliability measure of the prediction may yields better confidence thus better practicality of the two stage prediction method. By randomly partitioning the training dataset into several sub-samples, it is possible to develop an ensemble of models that procedure a set of predictions, and thus a prediction interval for a single time step.

REFERENCES

1. Van Lint, J. W. C. Reliable Real-Time Framework for Short-Term Freeway Travel Time Prediction. *Journal of Transportation Engineering*, Vol. 132, No. 12, 2006, pp. 921-932.

2. Ben-Akiva, M., M. Bierlaire, D. Burton, H. N. Koutsopoulos, and R. Mishalani. Network State Estimation and Prediction for Real-Time Transportation Management Applications. *Proceedings of Transportation Research Board 85th Annual Meeting*, Washington D.C., 2002.

3. Zhang, X., and J. A. Rice. Short-Term Travel Time Prediction. *Transportation Research Part C: Emerging Technologies*, Vol. 11, No. 3, 2003, pp. 187-210.

4. Van Lint, J. W. C., S. P. Hoogendoorn, and H. J. Van Zuylen. Freeway Travel Time Prediction with State-Space Neural Networks: Modeling State-Space Dynamics with Recurrent Neural Networks. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1811, Transportation Research Board of the National Academies, Washington, D.C., 2002, pp. 30-39.

5. Ran, B., F. Yang, Y. Tao, and Z. Qiu. Travel Time Prediction in Presence of Traffic Incidents Using Different Types of Neural Networks. *Proceedings of Transportation Research Board 85th Annual Meeting*, Washington, D.C., 2006.

6. Haykin, S. *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, Inc., New Jersey, 1998.

7. Wang, R., and H. Nakamura. Short Term Prediction Work in Traffic Engineering: The State-of-The-Art. *Proceedings of 9th World Congress on Intelligent Transport Systems*, Chicago, 2002.

8. May, A. D. *Traffic Flow Fundamental*, Prentice-Hall, Inc., New Jersey, 1990.

9. Wikipedia. Empiricism. http://en.wikipedia.org/wiki/Empiricism, Accessed August 1st, 2009.

10. Van Lint, J. W. C. Online Learning Solutions for Freeway Travel Time Prediction. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 9, No. 1, 2008, pp. 38-47.

11. Zou, N., J. Wang, G. L. Chang, and J. Paracha. ATIS Application: Field Test of A Travel Time Prediction System with Widely Spaced Detectors. *Proceedings of Transportation Research Board 88th Annual Meeting*, Washington, D.C., 2008.

12. Palacharla, P. V., and P. C. Nelson. Application of Fuzzy Logic and Neural Networks for Dynamic Travel Time Estimation. *International Transactions in Operational Research*, Vol. 6, No. 1, 1999, pp. 145-160.

13. Krikke, R. Short-Range Travel Time Prediction Using an Artificial Neural Network. *Proceedings of 9th World Congress on Intelligent Transport Systems*, Chicago, 2002.

14. Wei, C. H., and Y. Lee. Development of Freeway Travel Time Forecasting Models by Integrating Different Sources of Traffic Data. *IEEE transactions on vehicular Technology*, Vol. 56, 2007, pp. pp 3682-3694.

15. Rilett, L. R., and D. Park. Direct Forecasting of Freeway Corridor Travel Times Using Spectral Basis Neural Networks. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1752, Transportation Research Board of the National Academies, Washington, D.C., 2001, pp. 140-147.

16. Park, D., and L. R. Rilett. Forecasting Multiple-Period Freeway Link Travel Times Using Modular Neural Networks. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1617, Transportation

Research Board of the National Academies, Washington, D.C., 1998, pp. 163-170.

17. Medsker, L. R., and L. C. Jain. *Recurrent Neural Networks: Design and Applications*, CRC Press LLC, Boca Raton, Florida, 2000.

18. *Neural Network Toolbox 6: User Guide*. The MathWorks, Inc. 2008.

19. Hagan, M. T., and M. B. Menhaj. Training Feedforward Networks with Marquardt Algorithm. *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, 1994, pp. 989-993.

20. Marquardt, D. W. An Algorithm for Least Squares Estimation of Non-Linear Parameters. *Journal of Society for Industrial and Applied Mathematics*, Vol. 11, No. 2, 1963, pp. 431-441.

21. Bishop, C. M. *Neural Networks for Pattern Rcognition*, Oxford University Press, Oxford, U.K., 1995.

22. Williams, R. J., and D. Zipser. *Gradient-Based Learning Algorithms for Recurrent Networks and Their Computational Complexity*, Erlbaum Associates, Hillsdale, NJ, 1995.

23. Mackay, D. J. C. Probable Networks and Plausible Predictions: A Review of Practical Bayesian Methods for Supervised Neural Networks. *Network Computation in Neural System* Vol. 6, No. 3, 1995, pp. 469-505.

24. Van Hinsbergen, D. J. J., and J. W. C. Van Lint. Bayesian Training and Committees of State Space Neural Networks for Online Travel Time Prediction. *Proceedings of Transportation Research Board 88th Annual Meeting*, Washington D.C., 2008.

25. Mackay, D. J. C. Bayesian Interpolation. *Neural Computation*, Vol. 4, No. 3, 1992, pp. 415-447.

26. Songchitruksa, P., K. Balke, X. Zeng, C. L. Chu, and Y. Zhang. A Guidebook for Effective Use of Incident Data at Texas Transportation Management Centers. *FHWA/TX-09/0-5485-P2*, Texas Transportation Institute, College Station, Texas, February, 2009.

27. Gajewski, B., S. Turner, B. Eisele, and C. Spiegelman. ITS Data Archiving: Statistical Techniques for Determining Optimal Aggregation Widths for Inductance Loop Detectors. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1719, Transportation Research Board of the National Academies, Washington, D.C., 2000, pp. 85-93.

28. Qiao, F., X. Wang, and L. Yu. Optimizing Aggregation Level for Intelligent Transportation System Data Based on Wavelet Decomposition. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1840, Transportation Research Board of the National Academies, Washington, D.C., 2003, pp. 10-20.

29. Qiao, F., L. Yu, and X. Wang. Double-Sided Determination of Aggregation Level for Intelligent Transportation System Data. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1879, Transportation Research Board of the National Academies, Washington, D.C., 2004, pp. 80-88.

30. Park, D., L. R. Rilett, B. J. Gajewski, C. H. Spiegelman, and C. Choi. Identifying Optimal Data Aggregation Interval Sizes for Link and Corridor Travel Time Estimation and Forecasting. *Transportation*, Vol. 36, No. 1, 2009, pp. 77-95.

31. Songchitruksa, P., K. Balke, X. Zeng, C. L. Chu, and Y. Zhang. Evaluating and Improving Incident Management Using Historical Incident Data: Case Studies at

Texas Transportation Management Centers. *FHWA/TX-09/0-5485-1*, Texas Transportation Institute, College Station, August, 2009.

32. Wikipedia. Mean Absolute Percentage Error. http://en.wikipedia.org/wiki/MAPE, Accessed August 24th, 2009.

33. Margiotta, R., T. Lomax, M. Hallenbeck, S. Turner, A. Skabardonis, C. Ferrell, and B. Eisele. Guide to Effective Freeway Performance Measurement: Final Report and Guidebook. *NCHRP Web-Only Document 97*, National Cooperative Highway Research Program, TRB, http://onlinepubs.trb.org/onlinepubs/nchrp/nchrp_w97.pdf, Washington, D. C., 2006.

34. Wu, C. H., J. M. Ho, and D. T. Lee. Travel Time Prediction with Support Vector Regression. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 5, No. 4, 2004, pp. 276-281.

APPENDIX

Table A-1: Model Comparison at 5 Minute Horizon

Model	Test - Data B-I							
	Mean MAE (sec)	**Std MAE (sec)**	**Mean MAPE (%)**	**Std MAPE (%)**	**Mean RMSE (sec)**	**Std RMSE (sec)**	**Mean NRMSE (%)**	**Std NRMSE (%)**
BPNN	23.68	0.718	6.983%	0.19%	38.48	3.454	6.540%	0.58%
TDNN	22.33	0.484	6.827%	0.22%	36.33	1.059	6.179%	0.17%
MNN	23.37	0.549	7.065%	0.26%	34.52	0.521	5.838%	0.09%
SSNN	22.57	0.574	6.807%	0.13%	34.29	1.093	5.837%	0.19%
ExtSSNN	21.65	0.852	6.633%	0.18%	33.25	1.537	5.686%	0.28%
NP	33.89	-	10.010%	-	51.47	-	8.710%	-
HM	59.39	-	17.530%	-	83.54	-	14.190%	-
Model	Test - Data B-II							
	Mean MAE (sec)	**Std MAE (sec)**	**Mean MAPE (%)**	**Std MAPE (%)**	**Mean RMSE (sec)**	**Std RMSE (sec)**	**Mean NRMSE (%)**	**Std NRMSE (%)**
BPNN	25.57	0.708	7.134%	0.14%	42.27	2.667	7.139%	0.44%
TDNN	24.77	1.061	7.203%	0.32%	41.60	2.967	7.052%	0.47%
MNN	25.22	0.738	7.150%	0.30%	39.55	1.214	6.645%	0.19%
SSNN	24.13	1.151	7.130%	0.30%	38.59	2.555	6.565%	0.39%
ExtSSNN	24.11	1.445	7.199%	0.28%	39.59	2.204	6.785%	0.42%
NP	37.26	-	10.810%	-	57.12	-	9.490%	-
HM	61.74	-	17.310%	-	86.73	-	14.380%	-
Model	Test - Data B-III							
	Mean MAE (sec)	**Std MAE (sec)**	**Mean MAPE (%)**	**Std MAPE (%)**	**Mean RMSE (sec)**	**Std RMSE (sec)**	**Mean NRMSE (%)**	**Std NRMSE (%)**
BPNN	23.00	0.778	6.929%	0.24%	37.11	3.795	6.324%	0.65%
TDNN	21.45	0.330	6.692%	0.21%	34.43	0.808	5.864%	0.16%
MNN	22.71	0.535	7.035%	0.26%	32.70	0.456	5.548%	0.09%
SSNN	22.01	0.786	6.691%	0.13%	32.73	1.490	5.575%	0.25%
ExtSSNN	20.76	0.802	6.429%	0.18%	30.96	1.658	5.290%	0.30%
NP	32.62	-	9.700%	-	49.34	-	8.410%	-
HM	58.51	-	17.620%	-	82.34	-	14.120%	-

Note: 5 replica for each ANN model, and are trained by BRLM algorithm and tested at 5 minute prediction horizon
Color scheme is scaled according to cell values in the respective columns

Table A-2: Model Comparison at 15 Minute Horizon

Model	Test - Data B-I							
	Mean MAE (sec)	**Std MAE (sec)**	**Mean MAPE (%)**	**Std MAPE (%)**	**Mean RMSE (sec)**	**Std RMSE (sec)**	**Mean NRMSE (%)**	**Std NRMSE (%)**
BPNN	42.08	0.479	12.607%	0.15%	63.49	2.076	10.890%	0.38%
TDNN	38.60	0.891	12.012%	0.28%	59.14	2.309	10.149%	0.37%
MNN	41.97	0.213	12.683%	0.10%	60.80	0.407	10.428%	0.08%
SSNN	38.67	0.923	11.697%	0.15%	55.24	1.490	9.471%	0.24%
ExtSSNN	38.29	0.996	11.674%	0.26%	55.80	1.342	9.625%	0.22%
NP	58.69	-	17.370%	-	87.59	-	14.800%	-
HM	59.39	-	17.530%	-	83.54	-	14.190%	-
Model	Test - Data B-II							
	Mean MAE (sec)	**Std MAE (sec)**	**Mean MAPE (%)**	**Std MAPE (%)**	**Mean RMSE (sec)**	**Std RMSE (sec)**	**Mean NRMSE (%)**	**Std NRMSE (%)**
BPNN	50.12	1.595	14.215%	0.47%	78.59	4.762	13.278%	0.86%
TDNN	44.86	0.965	13.278%	0.27%	71.35	2.220	12.121%	0.32%
MNN	48.84	0.592	13.961%	0.21%	74.26	0.825	12.542%	0.14%
SSNN	44.33	1.201	13.026%	0.30%	66.93	1.231	11.298%	0.24%
ExtSSNN	45.53	1.610	13.513%	0.40%	71.54	3.385	12.227%	0.61%
NP	63.90	-	18.570%	-	94.22	-	15.680%	-
HM	61.74	-	17.310%	-	86.73	-	14.380%	-
Model	Test - Data B-III							
	Mean MAE (sec)	**Std MAE (sec)**	**Mean MAPE (%)**	**Std MAPE (%)**	**Mean RMSE (sec)**	**Std RMSE (sec)**	**Mean NRMSE (%)**	**Std NRMSE (%)**
BPNN	39.19	0.188	12.027%	0.11%	58.05	1.393	10.029%	0.26%
TDNN	36.34	0.875	11.555%	0.31%	54.74	2.369	9.438%	0.40%
MNN	39.50	0.277	12.222%	0.12%	55.95	0.778	9.665%	0.14%
SSNN	36.63	1.288	11.217%	0.25%	51.02	2.108	8.813%	0.34%
ExtSSNN	35.68	1.108	11.010%	0.31%	50.13	1.322	8.687%	0.21%
NP	56.73	-	16.920%	-	85.09	-	14.470%	-
HM	58.51	-	17.620%	-	82.34	-	14.120%	-

Note: 5 replica for each ANN model, and are trained by BRLM algorithm and tested at 15 minute prediction horizon
Color scheme is scaled according to cell values in the respective columns

Table A-3: Model Comparison Using T-Test at 5 Minute Horizon

5 minute prediction horizon							
Test Data B-I							
	BPNN	TDNN	MNN	SSNN	ExtSSNN	NP	HM
BPNN		0.148	0.589	0.133	0.018	3.9E-06	2.6E-08
TDNN			0.098	0.869	0.220	3.0E-06	2.4E-08
MNN				0.098	0.019	1.5E-05	9.6E-08
SSNN					0.122	6.8E-07	5.4E-09
ExtSSNN						1.9E-06	1.8E-08
NP							-
HM							
Test Data B-II							
	BPNN	TDNN	MNN	SSNN	ExtSSNN	NP	HM
BPNN		0.589	0.915	0.984	0.656	5.4E-07	9.2E-09
TDNN			0.682	0.624	0.832	4.8E-05	7.6E-07
MNN				0.919	0.796	1.1E-05	1.8E-07
SSNN					0.716	1.0E-05	1.8E-07
ExtSSNN						8.6E-06	1.4E-07
NP							-
HM							
Test Data B-III							
	BPNN	TDNN	MNN	SSNN	ExtSSNN	NP	HM
BPNN		0.061	0.516	0.093	0.006	1.2E-05	5.7E-08
TDNN			0.024	0.556	0.148	4.7E-06	2.9E-08
MNN				0.037	0.003	2.0E-05	8.2E-08
SSNN					0.030	7.6E-07	4.4E-09
ExtSSNN						2.0E-06	1.5E-08
NP							-
HM							

Note: H_0: model 1 = model 2; H_a: model 1≠ model 2.

light color indicates t-test is significant, whereas dark color indicates insignificant test

Table A-4: Model Comparison Using T-Test at 15 Minute Horizon

15 minute prediction horizon							
Test Data B-I							
	BPNN	TDNN	MNN	SSNN	ExtSSNN	NP	HM
BPNN		0.005	0.390	0.000	0.000	2.6E-07	2.3E-07
TDNN			0.004	0.066	0.084	1.7E-06	1.5E-06
MNN				0.000	0.000	5.8E-08	5.1E-08
SSNN					0.870	1.0E-07	9.3E-08
ExtSSNN						1.1E-06	9.5E-07
NP							-
HM							
Test Data B-II							
	BPNN	TDNN	MNN	SSNN	ExtSSNN	NP	HM
BPNN		0.007	0.312	0.002	0.034	3.1E-05	1.2E-04
TDNN			0.002	0.201	0.307	1.6E-06	4.8E-06
MNN				0.001	0.067	1.1E-06	3.8E-06
SSNN					0.062	2.0E-06	5.7E-06
ExtSSNN						8.9E-06	2.8E-05
NP							-
HM							
Test Data B-III							
	BPNN	TDNN	MNN	SSNN	ExtSSNN	NP	HM
BPNN		0.024	0.031	0.001	0.001	6.4E-08	3.7E-08
TDNN			0.006	0.097	0.024	2.7E-06	1.7E-06
MNN				0.000	0.000	1.1E-07	6.6E-08
SSNN					0.276	8.7E-07	5.5E-07
ExtSSNN						1.7E-06	1.1E-06
NP							-
HM							

Note: H_0: model 1 = model 2; H_a: model 1≠ model 2.

light color indicates t-test is significant, whereas dark color indicates insignificant test

VITA

Name: Xiaosi Zeng

Address:
Department of Civil Engineering
Texas A&M University
3136 TAMU
College Station, Texas 77843-3136
c/o Yunlong Zhang

Email Address:
naoh.zeng@gmail.com/david_zeng@neo.tamu.edu

Education:
B.E., Traffic Engineering, South China University of Technology, 2006
M.S., Civil Engineering, Texas A&M University, 2009

Work Experience:
Texas Transportation Institute, 2007-present
South China University of Technology, 2006-2007
Parsons BrinkerHoff – China, 2006