

4-3-2008

Predicting delay reductions from freeway traffic diversion for incident management

Jie Bian

Florida International University

DOI: 10.25148/etd.FI14051149

Follow this and additional works at: <https://digitalcommons.fiu.edu/etd>



Part of the [Civil Engineering Commons](#)

Recommended Citation

Bian, Jie, "Predicting delay reductions from freeway traffic diversion for incident management" (2008). *FIU Electronic Theses and Dissertations*. 1675.

<https://digitalcommons.fiu.edu/etd/1675>

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

PREDICTING DELAY REDUCTIONS FROM FREEWAY TRAFFIC DIVERSION
FOR INCIDENT MANAGEMENT

A dissertation submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

CIVIL ENGINEERING

by

Jie Bian

2008

To: Interim Dean Amir Mirmiran
College of Engineering and Computing

This dissertation, written by Jie Bian, and entitled Predicting Delay Reductions from Freeway Traffic Diversion for Incident Management, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

David L. Shen

Fang Zhao

Zhenmin Chen

Mohammed Hadi, Co-Major Professor

Albert Gan, Co-Major Professor

Date of Defense: April 3, 2008

The dissertation of Jie Bian is approved.

Interim Dean Amir Mirmiran
College of Engineering and Computing

Dean George Walker
University Graduate School

Florida International University, 2008

DEDICATION

I dedicate this dissertation to my parents, Weisheng Qiao and Guochen Bian, who have always supported my efforts first and foremost and who have made me a stronger person than otherwise I could be. To my sister, Jing Bian, who shares all of her life moments with me; to my lovely nephew, Rui Ma, who carries within him the hopes of my family; and to Jorge Vega, whose love has kept and will keep me going.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my sincere appreciation to Dr. Albert Gan, my major advisor, for his dedication in guiding me throughout my entire doctoral study, for setting such a good example of how hard work promotes achievement, for his patience and encouragement, for being considerate of my needs as an international student, and for his willingness to help on issues far beyond this degree. He has impressed upon me both his knowledge and his professionalism while serving as a mentor to me, from which I will benefit throughout my career.

My deepest gratitude is also extended to Dr. Mohammed Hadi, my co-major advisor, for the time and effort he has devoted to my dissertation since the very beginning of my studies.

I am also grateful to the remainder of my committee, Dr. David Shen, Dr. Fang Zhao, and Dr. Zhenmin Chen, for their valuable comments and advice rendered during the development of my dissertation.

Special thanks also go to Mr. Tao Wang, who provided me with help on the MATLAB coding, and Mr. Hongbo Chi, who shared his experience in using DYNASMART with me.

Last, but certainly not the least, I am also thankful to Florida International University for providing me with both a Presidential Fellowship over the course of three years and a superb academic environment during my studies. I would also like to thank all of the faculty and staff in the Department of Civil and Environmental Engineering for making my studies here so enjoyable.

ABSTRACT OF THE DISSERTATION

PREDICTING DELAY REDUCTIONS FROM FREEWAY TRAFFIC DIVERSION

FOR INCIDENT MANAGEMENT

by

Jie Bian

Florida International University, 2008

Miami, Florida

Professor Albert Gan, Co-Major Professor

Professor Mohammed Hadi, Co-Major Professor

Traffic incidents are a major source of traffic congestion on freeways. Freeway traffic diversion using pre-planned alternate routes has been used as a strategy to reduce traffic delays due to major traffic incidents. However, it is not always beneficial to divert traffic when an incident occurs. Route diversion may adversely impact traffic on the alternate routes and may not result in an overall benefit.

This dissertation research attempts to apply Artificial Neural Network (ANN) and Support Vector Regression (SVR) techniques to predict the percent of delay reduction from route diversion to help determine whether traffic should be diverted under given conditions. The DYNASMART-P mesoscopic traffic simulation model was applied to generate simulated data that were used to develop the ANN and SVR models. A sample network that comes with the DYNASMART-P package was used as the base simulation network. A combination of different levels of incident duration, capacity lost, percent of drivers diverted, VMS (variable message sign) messaging duration, and network congestion was simulated to represent different incident scenarios. The resulting percent

of delay reduction, average speed, and queue length from each scenario were extracted from the simulation output. The ANN and SVR models were then calibrated for percent of delay reduction as a function of all of the simulated input and output variables.

The results show that both the calibrated ANN and SVR models, when applied to the same location used to generate the calibration data, were able to predict delay reduction with a relatively high accuracy in terms of mean square error (MSE) and regression correlation. It was also found that the performance of the ANN model was superior to that of the SVR model. Likewise, when the models were applied to a new location, only the ANN model could produce comparatively good delay reduction predictions under high network congestion level.

TABLE OF CONTENTS

CHAPTER	PAGE
1. INTRODUCTION.....	1
1.1 Research Background	1
1.2 Problem Statement	2
1.3 Dissertation Goal, Objectives, and Scope.....	4
1.4 Research Methodology Overview.....	5
1.5 Dissertation Organization	8
2. LITERATURE REVIEW.....	10
2.1 Route diversion Decision Support Models	10
2.2 Intelligent Transportation Systems (ITS).....	11
2.2.1 ITS and National ITS Architecture	11
2.2.2 ITS Applications in Florida	13
2.2.3 ITS Devices and Delivery of Motorist Information.....	15
2.3 Driver's Behavior in Reaction to Traffic Information.....	18
2.4 Traffic Incidents and Incident Management	20
2.5 DYNASMART-P for Route Diversion.....	21
2.6. Data Mining Models and Their Applications in Transportation.....	23
2.6.1 Artificial Neural Networks	24
2.6.2 Support Vector Regression	32
2.6.3 Case-Based Reasoning.....	37
2.7 Summary	41
3. IDENTIFICATION OF POTENTIAL VARIABLES	43
3.1 Introduction.....	43
3.2 Variables Identification.....	43
3.2.1 Restrictive Criteria for Alternate Route Pre-Planning	43
3.2.2 Restrictive Criteria under Consideration at Time of Incidents.	47
3.3 Variable Selection	49
3.3.1 Traffic Conditions.....	50
3.3.2 Incident Conditions.....	50
3.3.3 ITS Device Availability	51
3.3.4 Motorist's Behavior	51
3.4 Summary	51
4. DYNASMART-P SIMULATOR AND INCIDENT CASE DESIGN	53
4.1 Introduction.....	53
4.2 Simulation Network	53
4.3 Simulation Input.....	55
4.3.1 Incident Location, Duration, and Level of Severity.....	55
4.3.2 Alternate Routes Selection.....	58
4.3.3 Variable Message Signs (VMS).....	62
4.3.4 Simulation Scenarios	64

4.3.5 Level of Congestion	66
4.4 Simulation Output	67
4.5 Incident Scenario Design	73
4.6 Data Generation for Training and Testing ANN and SVR Models	73
4.7 Summary	77
5. MODEL DEVELOPMENT	78
5.1 Introduction	78
5.2 Data Preparation	78
5.3 ANN Model Training, Calibration, and Testing	79
5.3.1 Neural Network Architecture	81
5.3.2 Transfer Functions	83
5.3.3 Backpropagation Algorithm	84
5.3.4 Variable Learning Rate	85
5.3.5 Levenberg-Marquardt	86
5.3.6 Principle Component Analysis	87
5.3.7 Number of Hidden Layers and Number of Hidden Neurons	88
5.3.8 Calibration and Selection of ANN Model	90
5.4 SVR Model Training and Model Testing	101
5.4.1 Support Vector Regression Learning	102
5.4.2 RBF Kernel Functions	104
5.4.3 Normalization	104
5.4.4 Parameters for RBF Kernel Functions	105
5.4.5 Training of SVR Models	110
5.5 Summary	114
6. MODEL EVALUATION	115
6.1 Introduction	115
6.2 Data Preparation for Model Application on the Testing Location	115
6.3 Performance of Calibrated ANN Model at the Testing Location	120
6.4 Evaluation of Performance of Selected SVR Model on Testing Location	124
6.5 Comparison between Selected ANN and SVR Models	129
6.5.1 Training Time	129
6.5.2 Calibration of Model Parameters	130
6.5.3 Performance of ANN and SVR Models Based on Preliminary Testing	131
6.5.4 Performance of ANN and SVR Models for the Testing Location	134
6.6 Summary	135
7. SUMMARY, CONCLUSIONS AND RECOMMENDATIONS	136
7.1 Summary of Research Approaches and Results	136
7.2 Conclusions	137
7.3 Recommendations for Future Research	141
REFERENCES	143

APPENDIX.....	149
VITA.....	161

LIST OF TABLES

TABLE	PAGE
Table 2.1 Common Kernel Functions.	34
Table 4.1 Description of INCIDENT.DAT Input File	57
Table 4.2 Sample INCIDEND.DAT File	58
Table 4.3 Link Properties on the Alternate Route for Training Location	60
Table 4.4 Description of VMS.DAT Input File	63
Table 4.5 Sample VMS.DAT File.....	63
Table 4.6 Description of SCENARIO.DAT Input File.....	65
Table 4.7 Sample SCENARIO.DAT File	66
Table 4.8 Description of DEMAND.DAT Input File	66
Table 4.9 Corresponding Average Speed for Different Multiplication Factors.....	67
Table 4.10 Description of OUTPUT_OPTION.DAT Input File	70
Table 4.11 Sample OUTPUT_OPTION.DAT File.....	71
Table 4.12 Attributes for Critical Variables for the Training Location	74
Table 4.13 Impacts of VMS Duration and Level of Congestion on Average Travel Times	76
Table 5.1 Input Variables for Training and Testing of ANN and SVR Models	80
Table 5.2 ANN Model Performance between Training with PCA and without PCA.....	92
Table 5.3 ANN Model Performance Using Different Transfer Functions.....	93
Table 5.4 ANN Model Performance Using Different Training Functions (Transfer Function is logsig)	94
Table 5.5 ANN Model Performance Using Different Training Functions (Transfer Function is tansig)	95
Table 5.6 Preliminary Determination of Number of Neurons in the Hidden Layer	96

Table 5.7 Further Identification of Number of Neurons in the Hidden Layer	97
Table 5.8 Recommended Parameters by Gunn	108
Table 5.9 Recommended Parameters by Vladimir <i>et al.</i>	108
Table 5.10 SVR Models Performance with Exponential RBF Kernel Function	111
Table 5.11 SVR Models Performance with Gaussian RBF Kernel Function	111
Table 6.1 Link Properties on Alternate Route for Testing Location.....	118
Table 6.2 Attributes of Critical Variables for Testing Location	118
Table 6.3 Performance of Selected ANN Model on Testing Location	121
Table 6.4 Performance of Selected SVR Model on Testing Location.	128
Table 6.5 Performance of Selected ANN and SVR Models on Training Location.	131
Table 6.6 Comparison between Performance of Selected ANN and SVR Models.	135

LIST OF FIGURES

FIGURE	PAGE
Figure 1.1. Methodology Overview	7
Figure 2.1 Input and Output for a Route Diversion Decision Support Model	11
Figure 2.2 High-Level Architectural Concept (FDOT, 2005).	14
Figure 2.3 Applying Data Mining to Incident Management (Lee, 2002).	24
Figure 2.4 Tuning Process of ANN Models (Demuth and Beale, 1998).	25
Figure 2.5 Example of MLP Network Topology (Ishak and Trifiro, 2007).	28
Figure 2.6 Positive Slack Variables ζ_i Measuring Points Lying Outside of Tube	35
Figure 2.7 Case-Based Reasoning Cycle (Mitchell, 1997).	39
Figure 4.1 Selected Study Network in DYNASMART-P.....	54
Figure 4.2 Training Location (on Link 48→41)	56
Figure 4.3 Alternate Route for the Training Location	61
Figure 4.4 Link Density of Link 48→41	68
Figure 4.5 Link Density of Link 52→48.....	69
Figure 4.6 Link Density of Link 42→41	69
Figure 4.7 Impact of Different Input Variables on Average Travel Times.....	76
Figure 5.1 A Simple Neural Network with One Neuron and One Layer	82
Figure 5.2 A One-layer Network with Multiple Neurons	82
Figure 5.3 A Multilayer Neural Network Architecture.....	83
Figure 5.4 Graphic Forms of Transfer Functions.....	84
Figure 5.5 Regression Plot of Predicted and Target PDR.....	99
Figure 5.6 Comparison of Predicted and Target Values of ANN Model.....	99

Figure 5.7 Histograms for Target PDR and ANN Model Predicted PDR Values	101
Figure 5.8 Regression of Predicted and Target Values without Normalized Inputs.....	106
Figure 5.9 Regression Relationships of Predicted and Target Values with Normalized Inputs	106
Figure 5.10 Regression Relationships between the Predicted and Target Values with Exponential RBF SVR Models ($\varepsilon=0.25$, $C=3$, $p1=0.82$)	112
Figure 5.11 Comparison of Predicted and Target Values of SVR Model.....	112
Figure 5.12 Histograms for Target PDR and SVR Model Predicted PDR Values	113
Figure 6.1 Testing Location (on Link 39→44)	116
Figure 6.2 Alternate Route for Testing Location	117
Figure 6.3 Link Density of Link 39→44.....	119
Figure 6.4 Link Density of Link 35→39.....	119
Figure 6.5 Link Density of Link 43→44.....	120
Figure 6.6 Regression Plot of ANN Model Predicted and Target Values for Testing Location.....	121
Figure 6.7 Regression Plot of ANN Model Predicted and Target Values for Testing Location under Low Network Congestion Level	123
Figure 6.8 Regression Plot of ANN Model Predicted and Target Values for Testing Location under Moderate Network Congestion Level.....	123
Figure 6.9 Regression Plot of ANN Model Predicted and Target Values for Testing Location under High Network Congestion Level	124
Figure 6.10 Regression Plot of SVR Model Predicted and Target Values for Testing Location.....	125
Figure 6.11 Regression Plot of SVR Model Predicted and Target Values for Testing Location under Low Network Congestion Level	126
Figure 6.12 Regression Plot of SVR Model Predicted and Target Values for Testing Location under Moderate Network Congestion Level	127

Figure 6.13 Regression Plot of SVR Model Predicted and Target Values for Testing Location under High Network Congestion Level	127
Figure 6.14 Histograms of Absolute Difference between Model Predicted PDR and Target PDR for Training Location	132
Figure 6.15 Effect of Capacity Lost on ANN Predicted PDRs	134
Figure 6.16 Effect of Incident Duration on ANN Predicted PDRs	134

CHAPTER 1

INTRODUCTION

1.1 Research Background

Over the past several decades, increasing traffic volume throughout the United States has been causing more and more congestion, and with it more operational and safety concerns. Freeway incidents usually cause major disruptions in traffic flows and can eventually adversely affect the whole traffic network. A traffic “incident” is defined as any non-recurring event that causes a reduction in roadway capacity or an abnormal increase in traffic demand (FHWA, 2000). In most metropolitan areas, incident-related delay accounts for between 40% and 70% of total congestion delay (Schrang, 2007).

Continuous efforts have been made to seek effective measures to reduce incident-related delays. A successful incident management program requires quick detection and verification of the incident, immediate response to the site, and recovery of capacity in a short period of time. In addition, well-planned diversion routes are essential components of any incident management plan. Although route diversion is not always desirable when an incident occurs, on those occasions where it is desirable and appropriately applied, it can greatly facilitate traffic bypassing the incident location and prevent the further queuing up of vehicles.

The deployment of Intelligent Transportation System (ITS) technology has brought about a new era in the management of transportation networks. Route diversion on freeways becomes more conveniently implemented in an ITS environment. Thus, concerns about route diversion are now less concentrated on the application’s capability

but more on under what circumstances it is advantageous to divert traffic from a freeway and onto the surface streets.

1.2 Problem Statement

Is route diversion beneficial given the current situation? For some time, this question has been answered by traffic operators based on their engineering judgment. For small and simple networks, on most occasions, engineering judgment is adequate for making efficient decisions on route diversion. However, along with the escalating size and complexity of roadway networks, the information required for making sound diversion decisions has also escalated to a much higher level. Many factors, such as freeway conditions, incident data, and local network data, need to be considered while making diversion decisions. Such decisions generally require more accurate and informed judgment than any individual engineer possesses. In cases where route diversion may be a solution, decision-making support should be provided to the traffic engineer.

In addition to deficiencies in technical support regarding route diversion decision making, traditional means of conveying route diversion information to drivers need to be upgraded to include more powerful devices. Variable message signs (VMSs) have been commonly recognized as a very effective means of conveying en-route information to drivers nationwide. Hence, VMSs are becoming one of the more important components of ITS and have been put into practical use in many states, including Florida. With the implementation of VMSs, incident information and route diversion suggestions could be posted along roadways. Other Advanced Traveler Information Systems (ATIS), such as

traveler information websites and telephone systems, have also been widely used to deliver traveler information.

As mentioned earlier, the benefits of a route diversion plan are the result of many factors involved in a given transportation system. Extensive research has been performed to assist in decision making regarding route diversion. The range of this research includes the use of dynamic traffic assignment (DTA) in the development and evaluation of diversion strategies (Chauhan, 2003); the examination of real-time freeway origin-destination (O-D) prediction algorithms under the Advanced Traveler Information System (ATIS) environment (Bhattacharjee *et al.*, 1998); drivers' behavior in response to route diversion information (Allen *et al.*, 1990; Khattak *et al.*, 1994; Khattak *et al.*, 1991; Madanat *et al.*, 1995; Pal, 1998; Peeta and Yu, 2002, Shirazi *et al.*, 1988, Ullman *et al.*, 1994, Vaughn, 1995); an analysis of VMS effectiveness under non-recurrent congestion (Srinivasan and Krishnamurthy, 2003); diversion strategies for freeways with ramp meters (Wu *et al.*, 2006); and simulation evaluation of route diversion strategies with different simulation software (Abdel-Rahim, 2002; Chen and Mahmassani, 1999; Cragg and Demetsky, 1995, Duneo *et al.*, 2001; Hawas and Mahmassani, 1995; Rao *et al.*, 1996).

Traffic simulation is a powerful tool for evaluating the potential benefits and impacts of traffic diversion when it comes to considering multiple factors in a network. Most of the factors that should be considered in real-world route diversion situations can be reflected in simulation models. However, running a traffic simulation model is usually time consuming. Real-world traffic conditions may change while the simulation model is run. Therefore, the simulation results may not correctly reflect the real-world

traffic situation, and the route diversion decision based on the simulation results could be inaccurate.

There is a need to develop a model to predict the potential benefits of route diversion to help the traffic engineers determine whether traffic should be diverted under given conditions. This model should be able to function in real time, and it should be compatible with the ITS software and devices implemented in the network. With predictions from the model, traffic operators should be able to make more reliable decisions on when it is beneficial to divert traffic.

1.3 Dissertation Goal, Objectives, and Scope

The goal of this dissertation is to develop a model to predict real-time freeway route diversion system-wide benefits in terms of percent delay reductions using simulated data and data mining methods. The system will integrate the desired information, data, and models for diversion decisions. In addition, it will satisfy the need for real-time operations. The following objectives are aimed at achieving this goal:

- 1) To identify critical factors influencing route diversion decisions through an extensive literature review and by analyzing the relationships between critical factors and average travel time.
- 2) To explore the efficacy of the DYNASMART-P simulator in simulating different incident scenarios and representing the real-world transportation environment.
- 3) To evaluate the feasibility of using different data-mining methods in making real time decisions on whether it would be beneficial to use route diversion plans.

This study is intended to develop a model for freeway traffic diversion. Freeway incidents include traffic incidents, non-traffic incidents, and emergencies. Traffic incidents are those major incidents caused by vehicle crashes or cargo spills, and that cause at least one lane or total road closure. Non-traffic incidents are those incidents rendering sections of a roadway impassible, such as major fires, industrial accidents, and bridge collapses. Emergencies are incidents requiring evacuation and that induce a surge of traffic demand on certain roadway facilities. This surge creates bottlenecks at capacity-constrained locations. Causes may include a severe weather event or other major catastrophe. In this research, freeway incidents are limited to traffic incidents.

Traffic diversion can be categorized into four types: freeway-to-freeway, freeway-to-arterial, arterial-to-arterial, and arterial-to-freeway. This dissertation research considers diversion resulting from freeway incidents; thus, it excludes arterial-to-arterial and arterial-to-freeway cases. Among the remaining two cases, only freeway-to-arterial diversion is considered in this research.

1.4 Research Methodology Overview

Figure 1.1 illustrates the general methodology of this dissertation research. As the figure shows, the first step is to identify the variables that are critical to route diversion decisions. In this step, a comprehensive literature review is performed to identify variables that have been commonly considered in route diversion decisions, including incident characteristics, ITS device availability, driver's behavior towards VMS messages, real-time traffic conditions, etc. These variables are then tested for their sensitivity to average travel times.

After the variables that have a significant impact on average travel times are identified, they are varied systematically to define the different incident scenarios. In this research, these scenarios are simulated in the DYNASMART-P mesoscopic simulator, and a sample network that comes with DYNASMART-P is used both as the model training and the model testing network. Two locations on a freeway in the sample network are selected as the training and testing incident locations, respectively, and the simulated data sets are generated for each. The training data set is used for training and preliminary testing at the training location, and the testing data set is used to test if the developed model can be applied to the testing location. Various incident scenarios are simulated for each of the two locations with VMS messages provided to drivers. After each simulation, the simulation output such as queue length, link speed, and link flow are extracted. Also extracted are the average travel times, which are used later to calculate the percent of delay reductions.

A portion of the incident scenarios are then randomly selected from all of the scenarios and used to train the ANN and SVR models. The remaining scenarios are applied to test the validity of the trained models. The models with the best performance are selected by comparing and evaluating their performances. After the models are selected and their performance is tested at the location they are trained with, they are then tested at the testing location. The purpose of this test is to determine if the models can be used for general locations.

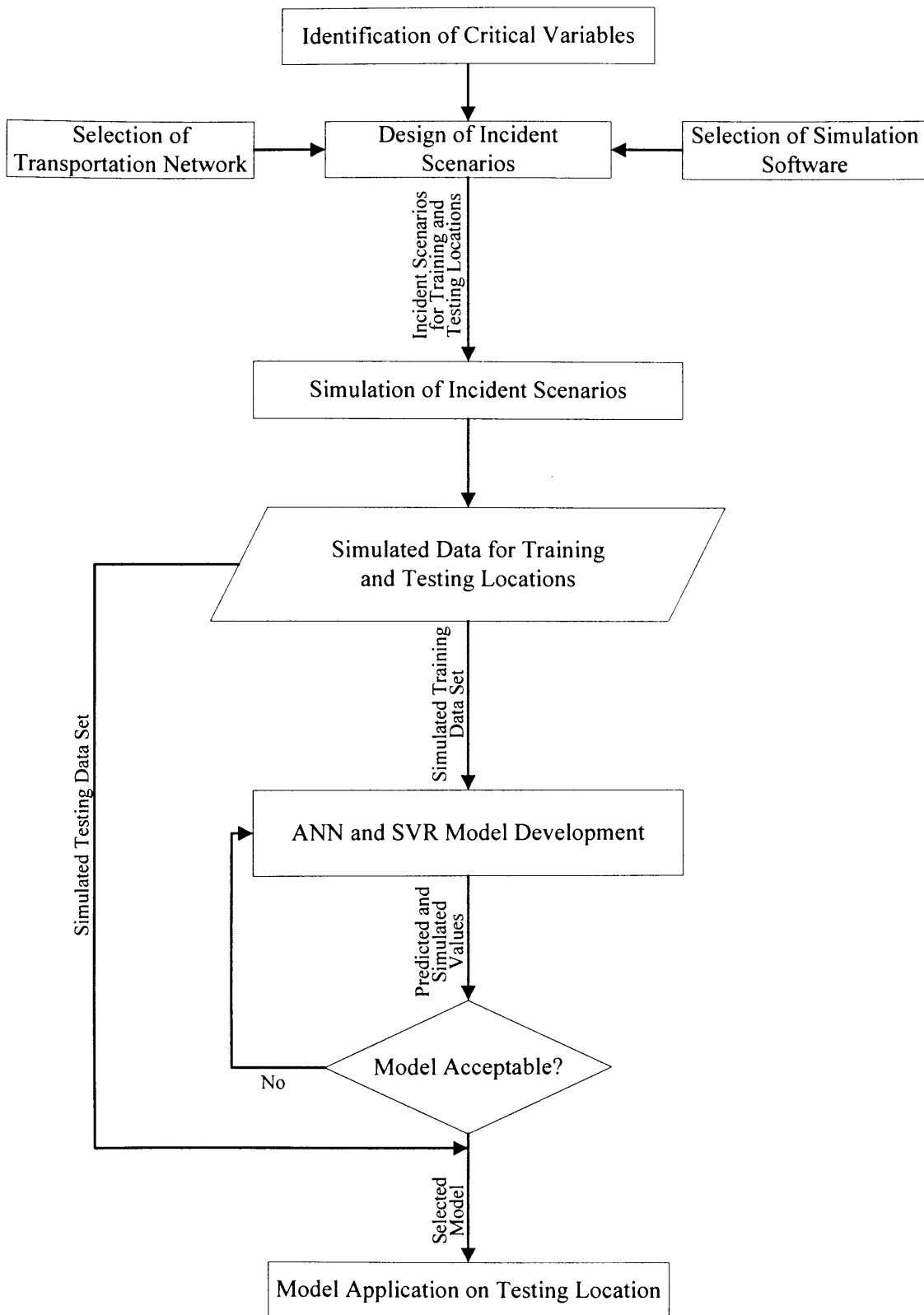


Figure 1.1. Methodology Overview

1.5 Dissertation Organization

This dissertation consists of seven chapters. Chapter 1 introduces the background of the research, puts forward the problem to be solved, and sets the goals and objectives to be achieved.

Chapter 2 reviews the literature. Literature on the current practice in incident management strategies, route diversion related studies, and their findings are described. Data mining techniques, including Artificial Neural Networks (ANN), Support Vector Regression (SVR), and Case-Based Reasoning (CBR), as well as their applications in the transportation field, are also reviewed and summarized in this chapter.

Chapter 3 identifies potential variables that have critical impacts on delay reductions at the time of an incident. These variables are categorized into incident conditions, traffic conditions, ITS device availability, and the drivers' response to the incident.

Chapter 4 introduces DYNASMART-P as the selected simulator for this study. It is used to generate data that are used in the training and testing of ANN and SVR models. Critical variables are further validated with the simulation output from DYNASMART. Different incident scenarios are designed to represent changes in these critical variables.

Chapter 5 describes the development of the ANN and SVR models. Inputs to the ANN models and SVR models are inclusive of previously identified critical variables and real-time traffic conditions extracted from the simulation. Parameters for ANN and SVR models are calibrated for use in this research. All of the models are tested using another 500 randomly selected incident scenarios. The models with the best performance are then selected based on their performances during testing.

Chapter 6 evaluates the selected ANN and SVR models. Both models are evaluated and compared for the same location they are trained with. Furthermore, by applying both models to the testing location, both models are evaluated and compared for their generality based on their performance at the testing location. The performance of ANN and SVR models at both incident locations are also discussed, compared, and summarized.

Chapter 7 summarizes the major results from this research, draws conclusions, and recommends issues for future research.

CHAPTER 2

LITERATURE REVIEW

In this chapter, concepts such as route diversion decision support models, Intelligent Transportation Systems (ITS), ITS architecture, as well as an Advanced Traffic Management System are introduced. The literature related to route diversion and data mining models' application in transportation is reviewed and summarized. The use of DYNASMART-P as a mesoscopic simulation software is also introduced in the chapter.

2.1 Route diversion Decision Support Models

Decision support uses a technique (or a combination of techniques) designed to extract information from data and to use such information as a basis for decision making (Thomason, 2004). A decision support model provides an analytic of framework for optimizing system and process performance and function as the input information is transformed into an output that directly relates to the decision (Grimes, 2001). A decision support model in a real-time scenario does everything described above in real time. It reacts quickly with real-time information and produces timely decisions that assist decision makers (Information Builders, accessed on February 12, 2007).

A real-time decision support model takes the place of the traditional function of engineering judgment and provides traffic operators with route diversion strategies. For the route diversion system to function properly, the cooperation of different organizations

and the integration of data and information are required. Figure 2.1 is a preliminary listing of the input and output information of a route diversion decision support model.

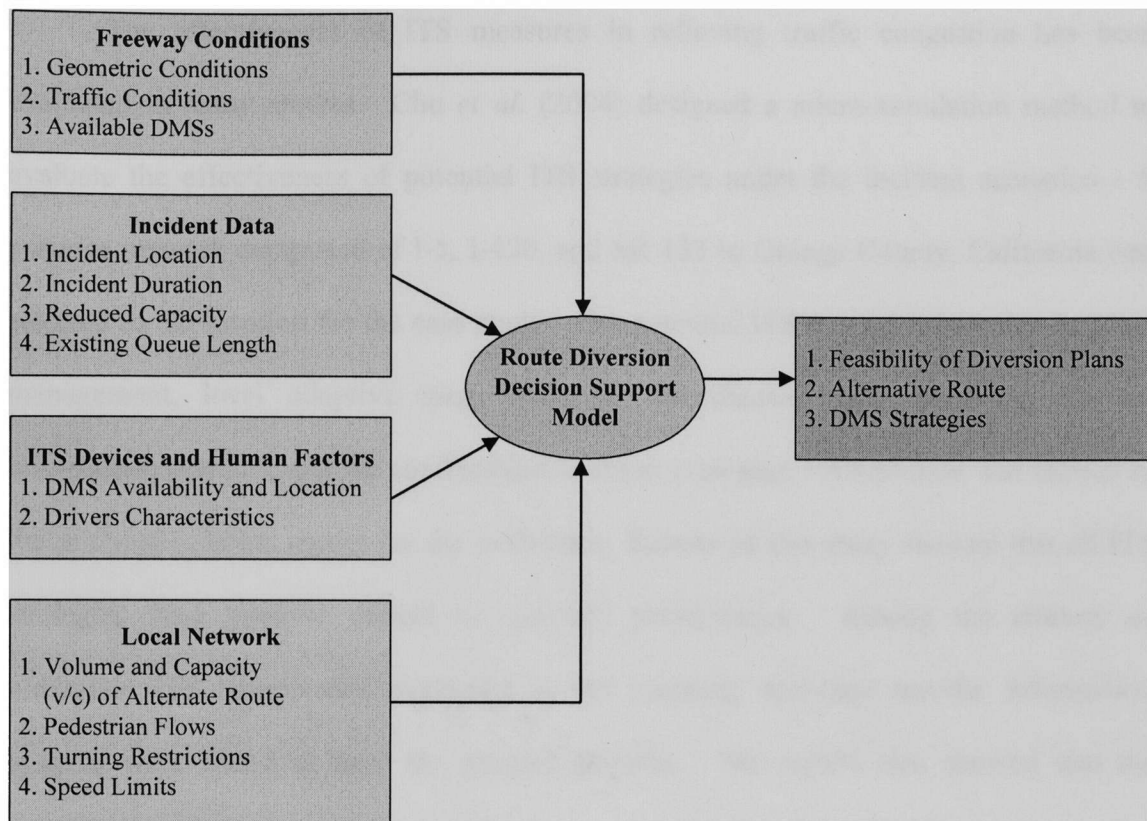


Figure 2.1 Input and Output for a Route Diversion Decision Support Model

2.2 Intelligent Transportation Systems (ITS)

In the following sub sections, the basic concepts of ITS and its components related to this research will be introduced.

2.2.1 ITS and National ITS Architecture

ITS is a system defined as the electronics, communications, and/or information processing, used either singly or integrated, to improve the efficiency and/or safety of

surface transportation (National ITS Architecture Website, accessed on October 22, 2006).

The effectiveness of ITS measures in relieving traffic congestion has been evaluated in many studies. Chu *et al.* (2004) designed a micro-simulation method to evaluate the effectiveness of potential ITS strategies under the incident scenarios. A corridor network composed of I-5, I-450, and SR 133 in Orange County, California was selected as the location for the case study. The potential ITS strategies included incident management, local adaptive ramp metering, coordinated ramp metering, traveler information systems, and the combination of these strategies. PARAMICS was chosen as the micro-simulation model for the evaluation. Results of this study showed that all ITS strategies have positive effects on network performance. Among the entirety of individual ITS components evaluated in this research, real-time traveler information systems were found to have the greatest benefits. The results also showed that the combination of several ITS components could generate even better benefits.

The ITS architecture is an architecture of interrelated systems that work together to deliver transportation services. The ITS architecture defines how systems function and provides for the interconnection of information exchanges that must take place between these systems to provide transportation services. The national ITS architecture is a common, established framework for developing integrated transportation systems.

The Advanced Traffic Management System (ATMS) is one of the service areas in ITS. An ATMS is an advanced transportation management system incorporating the latest communications technologies. A successful ATMS provides controlled-access, real-time correlated traffic information by integrating multiple ITS subsystems (CCTV,

VMS, traffic controllers, and detectors). The ability to integrate traveler websites and provide video feeds to the media is also becoming an integral part of an ATMS. Among the 21 market packages included in ATMS and Advanced Traveler Information Systems (ATIS), the following two are related to this research:

1. Traffic Information Dissemination: This market package provides driver information using roadway equipment such as variable message signs or highway advisory radio. A wide range of information can be disseminated, including traffic and road conditions, closure and detour information, incident information, and emergency alerts and driver advisories.
2. Traffic Incident Management System: This market package manages both unexpected incidents and planned events so that the impact on the transportation network and traveler safety is minimized. This market package supports traffic operations personnel in developing an appropriate response in coordination with emergency management, maintenance and construction management, and other incident response personnel to confirmed incidents. The response may include traffic control strategy modifications or resource coordination between center subsystems.

2.2.2 ITS Applications in Florida

The Florida Department of Transportation (FDOT) has been conducting a program involving the development of SunGuideSM software. SunGuideSM software is a set of ITS software that provides control of roadway devices, as well as information exchange across a variety of transportation agencies. The goal of SunGuideSM software is

to have a common software base that can be deployed throughout the state of Florida. The SunGuideSM software development effort is based on ITS software available from the states of Texas and Maryland. Significant customization of the software is being performed, as is the development of new software modules (FDOT, 2005). Figure 2.2 provides a graphic view of the software to be developed.

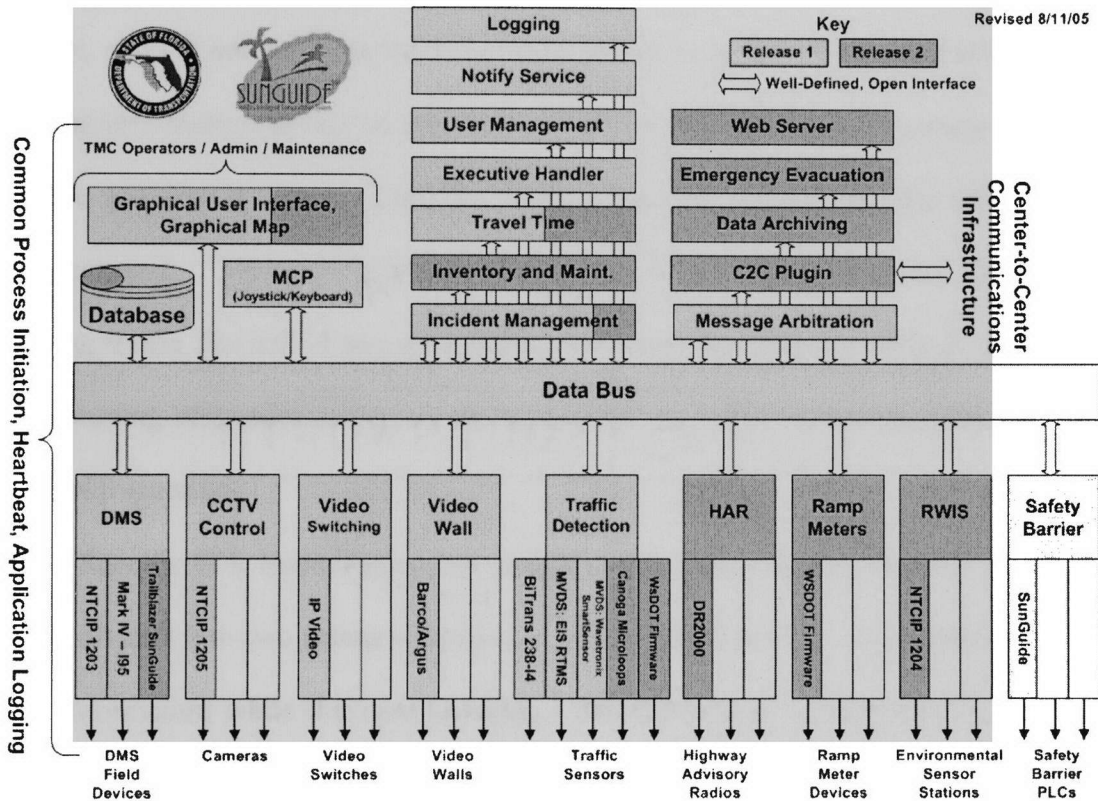


Figure 2.2 High-Level Architectural Concept (FDOT, 2005).

The SunGuideSM Transportation Management Centers (TMCs) are the nerve centers of Florida's ITS. The TMCs are physical locations used to monitor traffic conditions, respond to traffic incidents, and coordinate ITS programs. The TMCs are those locations from which information coming in from pavement sensors and cameras can be distributed to motorists in real-time through telephones. Law enforcement and

emergency services will be involved in ITS operation, and this will allow coordinated responses to freeway incidents and rush hour traffic congestion.

2.2.3 ITS Devices and Delivery of Motorist Information

Motorist information includes the dissemination of incident-related information to motorists who are (1) at the scene of the incident, (2) approaching the scene of the incident, and (3) not yet departed from work, home, or another location (FHWA, 2000). Motorist information should be provided as early in the incident management process as possible and should continue until the incident has been cleared and the traffic backup has dissipated. Motorist information supports incident response and clearance by reducing traffic demand at and approaching the scene, by reducing secondary incidents, by improving responder safety on the scene, and by reducing erratic behavior due to motorist frustration.

Specific ITS technologies that can be used to disseminate motorist information can be divided into two primary categories, en-route and pre-trip. Motorists receive en-route information while they are traveling. Pre-trip traveler information is received in non-travel settings, such as the home, office, or shopping areas, and is provided to assist travelers in making travel decisions. This study assumes that the motorists are provided with en-route information. Roadside devices that are commonly deployed to disseminate information to motorists are described below:

1. Variable Message Signs: Variable Message Signs (VMSs) are also known as Dynamic Message Signs (DMSs) or Changeable Message Signs (CMSs). VMSs can be used by operating agencies to disseminate travel information on a near

real-time basis. VMSs are among the most flexible and powerful means of communicating with motorists on the road. Both fixed-location and portable VMSs are used to support incident management functions. The messages presented can be operated on a fixed-time basis, via on-site controls (as in the case of an incident), or remotely, via a connection with a traffic control center. VMSs are most often used to: (1) inform motorists of varying traffic, roadway, and environmental conditions (including variable speed limits in adverse conditions); (2) provide specific information regarding the location and expected duration of incident related delays; (3) suggest alternate routes because of construction or a roadway closure; (4) redirect diverted drivers back onto the freeway; and (5) inform drivers when passage along the shoulder is permissible (e.g., in the event of a major incident, where this can help restore the traffic flow safely).

2. Highway Advisory Radio (HAR): HAR information is communicated to drivers via their vehicles' AM radio receivers. Drivers receive notification to tune into a specific frequency upstream of the transmitter. Message transmission can be controlled either on-site or from a remote location, such as a TOC. The functions served by HAR are similar to those served by VMS. An advantage of HAR, according to research, is that because the information is auditory, the driver experiences less sensory overload than when having to read and process a written message, which also requires that they take their eyes off the road. Another advantage of HAR over VMS is that longer, more complex messages can be communicated. However, HAR has disadvantages, including the following: (1)

Drivers who do not have functioning radios cannot access the information. (2)

Drivers who tune in midway through the message have to listen through another cycle. (3) Many drivers disregard the message to tune in.

3. Automated Highway Advisory Radio (AHAR): One of HAR's primary disadvantages is that it requires specific action on the part of the driver, specifically, that he or she pays attention to the advance signal and then tunes in to the specified station. In practice, many drivers disregard the message to tune in. For this reason, AHAR systems are still under study.
4. Wireless Phone "Hotlines": With the spread of wireless telephone usage, transportation agencies have a new and quite powerful way to communicate with travelers who are already on the road. Wireless phone hotlines can be used by drivers to access construction, congestion, and incident-related travel information on the route or routes in which they are interested in via touch-tone menus.
5. Commercial Radio: In use for decades, commercial radio has long provided traffic information to commuters and travelers. Its clear advantage is that it is well understood and utilized by millions of drivers. However, there are areas for improvement in commercial radio transmissions. In particular, because they are based on programming and scheduling constraints, the information does not always meet the traveler's needs. With technological advances in ITS has also come the recognition for the need for better coordination among all of the organizations involved, including commercial radio stations, that play a part in traffic information dissemination. As such, many transportation agencies recognize the important role commercial radio plays in information dissemination.

To that end, transportation agencies are forging closer alliances with the media, with the goal of improving the quality and timeliness of broadcasts. Some agencies, for example, allow commercial radio staff to work in their Traffic Operations Centers.

2.3 Driver's Behavior in Reaction to Traffic Information

In 1988, Schirazi *et al.* conducted a survey to collect information on commute characteristics, factors affecting route change, and improved traffic information. In total, 400 commuters were surveyed in the Los Angeles area by telephone to identify how various forms of traffic information were used and to assess drivers' attitudes toward route diversion when provided with improved traffic information. Nearly 70% of the surveyed commuters said that they would alter their route if accurate traffic information were provided. The results of the survey also showed that age, gender, travel time, and delay had no significant relationship to the frequency of being open to route change. The most frequently cited factor affecting route change was radio traffic reports, followed by personal experience and having to arrive on time. Note that this study was conducted before the wide use of ITS.

Peeta *et al.* (2000) investigated the effect of different message contents on drivers' response to VMSs. In this research, it was suggested that the contents and the level of detail of relevant information are factors that significantly affect drivers' willingness to divert. Other significant factors include socioeconomic characteristics, network spatial knowledge, and confidence in the displayed information. Results also indicated

differences in the response attitudes of semi-trailer truck drivers compared to other travelers.

Chatterjee *et al.* (2002) used questionnaires to investigate the effectiveness of different messages on drivers' route choices in London. Results showed that about one-fifth of drivers diverted in reaction to message signs. This percentage was further verified by real data collected from another U.K. city.

A report prepared by Houten and Malenfant (2002) for FDOT indicated that most drivers surveyed in their study noticed signs. Furthermore, among those who noticed signs, a large percentage of the drivers selected the proper exit for diversion with the help of signs. This report also indicated that the use of VMSs did not increase the number of crashes.

In 2003, Horowitz *et al.* carried out a similar experiment to determine the degree of alternate route selection from a rural freeway that could occur because of implementation of a traffic-responsive VMS system in a work zone. The message signs gave real-time estimates of the travel time to the end of the work zone. Alternate route selection was measured through extensive volume counts on the freeway and on parallel arterial streets, both before and after implementation of the signage system. In addition, traffic volumes were measured at a variety of locations where alternate route selection was likely to occur and unlikely to occur. An analysis of peak period data found that alternate route selection rates were between 7% and 10% of the freeway traffic, depending on the location and the day of the week.

2.4 Traffic Incidents and Incident Management

As defined previously, an incident is any non-recurrent event that causes the reduction of roadway capacity or an abnormal increase in demand (FHWA, 1991 and 2000). Incident management is the spectrum of activities involved in detecting, responding to, and clearing roadway incidents. It is the coordinated preplanned use of human and technological resources to restore full capacity after an incident occurs, and to provide motorists with information and direction until the incident is cleared (FHWA 1991). The planning and coordination usually involves multiple agencies and jurisdictions, including state and county/municipal departments of transportation (DOT), state and local law enforcement agencies, fire departments, towing and recovery companies, HAZMAT cleanup companies, and other key stakeholder organizations. Therefore, the decision on route diversion is usually politically sensitive and needs to be carried out with extra caution.

In general, the following five major tasks need to be accomplished to enhance the effectiveness of incident management programs (FHWA, 1991):

1. Detection: Determination that an incident of some nature has, indeed, occurred. This information needs to reach that location where a response can be initiated.
2. Verification: Determination of the precise location and nature of the incident, as well as the display, recording, and communication of this information to appropriate agencies.
3. Response: The activation, coordination, and management of the appropriate personnel, equipment, and communication links and motorist information media as soon as there is reasonable certainty that an incident is present.

4. Removal/Restoration of Capacity: The removal of wreckage, debris, etc., from the roadway and the restoration of roadway capacity to its pre-incident condition.
5. Traffic Management: Application of traffic control measures in the area of the incident site including: lane closure and openings, establishing and operating alternate routes, diversions, parking of emergency vehicles, and ensuring the safety of incident victims, motorists, and emergency personnel.

A well-thought traffic diversion plan represents an important response strategy to traffic incidents. It may effectively reduce the incident incurred delay.

2.5 DYNASMART-P for Route Diversion

Simulation software has been found to be a useful tool for modeling complicated traffic conditions. Traffic simulation models can be classified as either microscopic, mesoscopic, or macroscopic (Boxill and Yu, 2000). Microscopic models are models that continuously or discretely predict the state of individual vehicles. Microscopic measures are individual vehicle speeds and locations. Macroscopic models aggregate the description of traffic flow. Macroscopic measures of effectiveness are speed, flow, and density. Mesoscopic models are models that have aspects of both macroscopic and microscopic models.

To simulate a route diversion situation, a traffic simulator should provide the ability to simulate an individual driver's choice in the event of incident-related congestion. In addition, the network used in simulating route diversions should generally be large, with a minimum of one on-ramp and one off-ramp on the freeway and one major arterial route serving as an alternate route at the time when traffic needs to be

diverted off of the freeway. Mesoscopic models are a preferable choice in the route diversion scenario as they provide trade-offs between modeling simplified networks and achieving a sufficient level of detail in modeling drivers' route choices. Among the previous studies on route diversion strategies, many researchers applied simulation software for assistance in using mesoscopic models such as DYNASMART-P and DYNAMIT. In this research, DYNASMART-P will be used as the simulation model.

The name DYNASMART comes from Dynamic Network Assignment Simulation Model for Advanced Road Telematics. DYNASMART-P is the planning version of DYNASMART. As mentioned above, DYNASMART-P is a mesoscopic dynamic network analysis and evaluation tool. It was originally conceived of and developed at the University of Texas at Austin, with the participation of researchers at the University of Maryland, Northwestern University, Purdue University, and the University of California at Irvine. The development of this model began in the early 1990s out of the need to develop a dynamic traffic assignment (DTA) system capable of correctly representing the path choices of drivers given information from various sources, especially from ATIS and in response to ATMS. The main focus of model development was toward the correct representation of used paths in the DTA, as well as the development of methods of determining optimal paths (from a system point of view) toward which drivers should be (re-)directed. Following this problem statement comes the need for the estimation and prediction of time-varying traffic flows on links in the network (Burghout, 2005).

DYNASMART has been calibrated, validated, and tested extensively during the last decade, especially in the DTA project (Burghout, 2005). Part of the test was off-line, using data from multiple networks across the US. In addition, the model has been tested

on-line, but open-loop, meaning that the models' control and information strategies were not actually implemented in real life. Therefore, the exercise was more focused on replicating, and in some cases predicting, traffic situations in the network. DYNAMIT and DYNASMART have been evaluated in an operational context (DYNASMART-X). Both the DYNAMIT and DYNASMART online versions are aimed at actively influencing traffic conditions to obtain a projected system optimum situation. Therefore, their efforts toward the online prediction and estimation of traffic conditions are merely a stage toward that goal. Their focus is not so much on the traffic conditions following incidents as much as a broader replication of general conditions in the network, including ways to influence these conditions by means of "optimal" control and information strategies. Therefore, in this research, although DYNASMART-P can only be applied offline, it is still preferable due to the immaturity of DYNASMART-X. As a result, data mining techniques need to be explored for their capability to predict/search for a feasible route diversion decision.

2.6. Data Mining Models and Their Applications in Transportation

Generally, data mining is the process of analyzing data from different perspectives and summarizing it into useful information (i.e., information that can be used to increase revenue, cut costs, or both). Data mining approaches have been employed to solve transportation problems during the past couple of decades. Such approaches have also been adopted as a popular method in developing traffic diversion plans. Lee *et al.* (2002) explored the opportunities data mining could offer in analyzing

incident situations and deriving better traffic diversion plans. Figure 2.333 depicts how a TMC could apply data mining at each phase of managing an incident situation.

Three data mining techniques, artificial neural networks (ANN), support vector regression (SVR), and case-based reasoning (CBR), will be introduced in the following subsections.

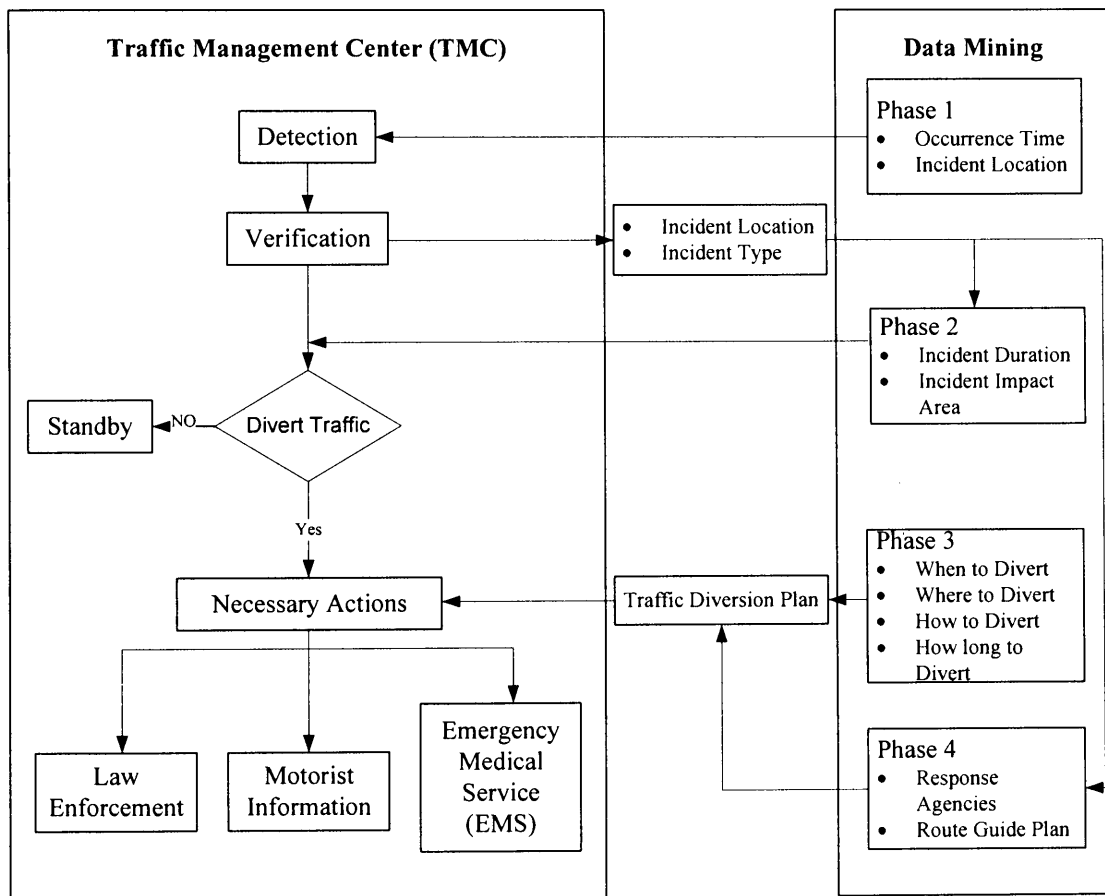


Figure 2.3 Applying Data Mining to Incident Management (Lee, 2002).

2.6.1 Artificial Neural Networks

An Artificial Neural Network (ANN) can be defined as a distributed, adaptive, generally nonlinear learning machine built from different interconnecting processing

elements (PEs) (Ishak and Trifiro, 2007). The functionality of ANN is based on the interconnectivity between the PEs. Each PE receives connections from other PEs and/or itself. The connectivity defines the topology of the ANN and plays a role at least as important as the PEs in the ANN's functionality. The signals transmitted via the connections are controlled by adjustable parameters called weights w_{ij} . An ANN can be trained to perform a particular function by adjusting the values of the connections (weights) between elements. Commonly, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown in Figure 2.4. The network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically, many such input/target pairs are used in this supervised learning to train a network (Demuth and Beale, 1998).

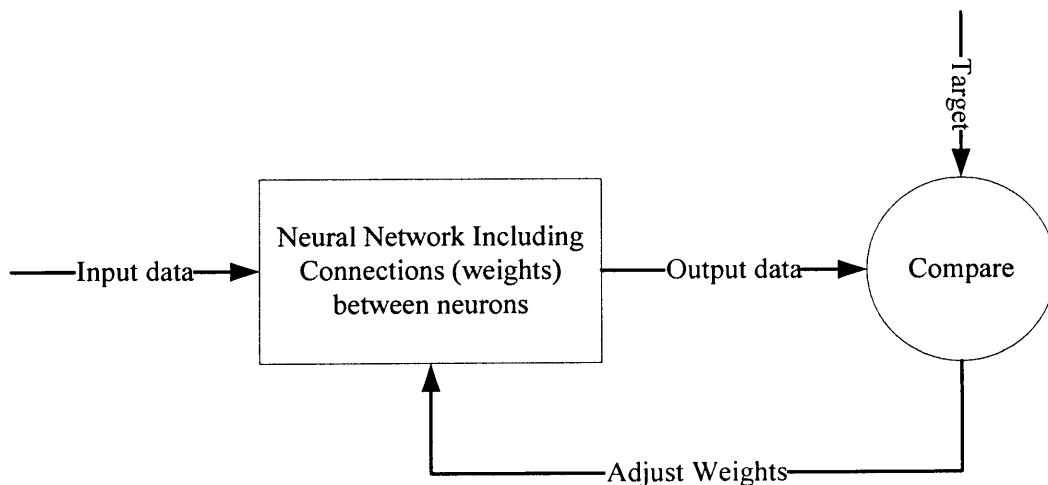


Figure 2.4 Tuning Process of ANN Models (Demuth and Beale, 1998).

The development process of ANN models is typically carried out in two stages: training and testing. During the training stage, an ANN learns from the patterns presented in an existing dataset. The performance of the network is consequently

evaluated using a testing dataset that is composed of patterns that the network was never exposed to before. Because the learned knowledge is extracted from training datasets, an ANN is considered both a model-based and data-driven system. Usually, the learning phase uses an algorithm to adjust the connection weights, based on a given dataset of input-output pairs. Training patterns are presented to the network repeatedly until the error of the overall output is minimized. The presentation of all patterns once to the network is called an epoch. This results in an adjustment of the connection weights such that the network performance is improved. The training stage of an ANN is terminated when the error drops below a pre-specified threshold value or when the number of epochs exceeds a certain pre-specified limit. Another method of controlling the efficiency of the training stage is to monitor the network performance (errors) during the training stage on a cross-validation (CV) dataset, usually smaller than the learning dataset. The role of the CV is to test for the network's generalization capabilities during the training process. If the network is over-trained, a sudden degradation of the network based on the CV data will trigger the training process to stop.

There are different types of ANNs. The most commonly used architecture of ANNs is the multilayer perceptron (MLP). MLP is a static ANN that has been used extensively in many transportation applications due to its simplicity and its ability to perform nonlinear pattern classification and function approximation.

An MLP consists of three types of layers: input, hidden, and output. It has a one-directional flow of information, generally from the input layer, through the hidden layer, and then to the output layer, which then provides the response of the network to the input stimuli. In this type of network, there are generally three distinct types of neurons

organized in layers. The input layer contains as many neurons as the number of input variables. The hidden neurons, which are contained in one or more hidden layers, process the information and encode the knowledge within the network. The hidden layer receives, processes, and passes the input data to the output layer. The selection of the number of hidden layers and the number of neurons within each affects the accuracy and performance of the network. The output layer contains the target output vector.

Figure 2.5 depicts an example of MLP topology. A weight coefficient is associated with each of the connections between any two neurons inside the network. Information processing at the neuron level is performed by an “activation function” that controls the output of each neuron.

An ANN is trained through adaptation of its connection weights based on examples provided in a training set. The training is performed iteratively until the error between the computed and the real output over all training patterns is minimized. Output errors are calculated by comparing the desired output with the actual output. Therefore, it is possible to calculate an error function that is used to propagate the error back to the hidden layer and to the input layer to modify the weights. This iterative procedure is carried out until the error at the output layer is reduced to a pre-specified minimum or for a pre-specified number of epochs. The back-propagation algorithm is most commonly used for training MLPs and is based on minimizing the sum of squared errors between the desired and actual outputs.

The actual validation of an already trained ANN requires testing the network performance on an exclusive set of data, called testing data, which comprises data that were never presented to the network before. If the error obtained in both the training and

testing phases is satisfactory, the ANN is considered adequately developed and, thus, can be used for practical applications.

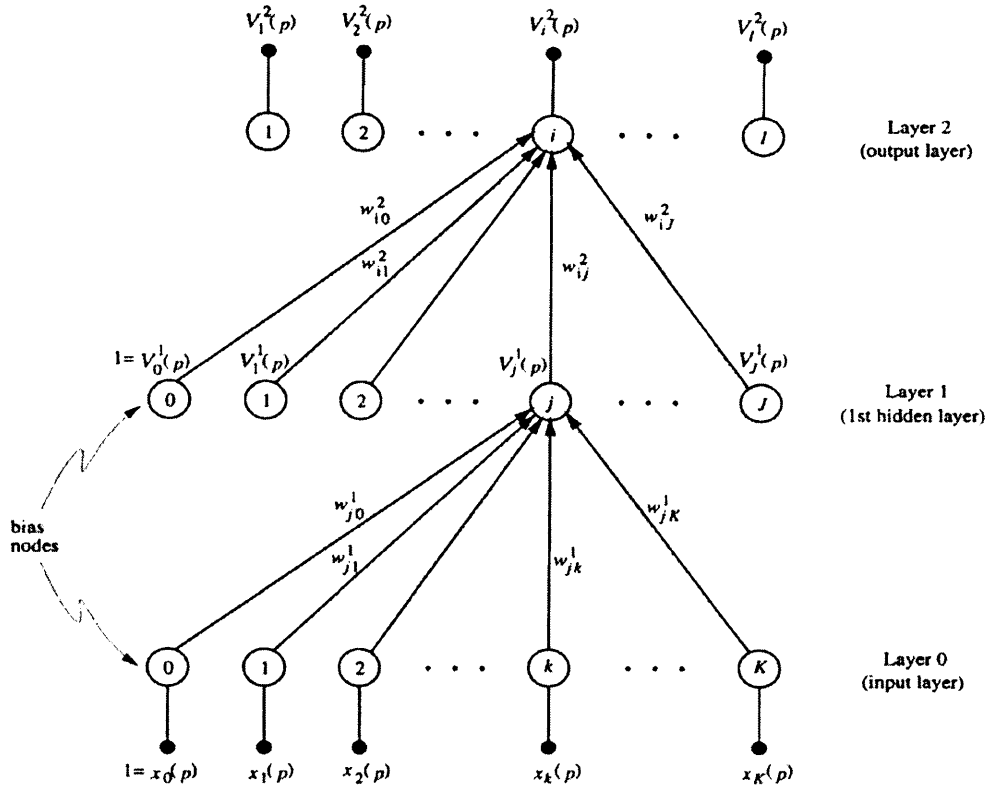


Figure 2.5 Example of MLP Network Topology (Ishak and Trifiro, 2007).

The inherent parallel architecture and fault tolerant nature of ANNs make their use appealing in addressing problems in a variety of application areas. The ANN finds its application in pattern recognition (classification, clustering, feature classification), image compression, image processing, system identification, and prediction. ANN models appear to have great potential for enhancing condition assessments and for performance prediction modeling. One major application of ANNs in transportation is short-term traffic prediction.

The short-term traffic prediction problem, which is concerned with attempting to forecast future traffic volumes, speeds, or travel times, has received increasing attention in the last few years. Huang and Xu (1996) proposed an approach employing ANN and micro-simulation techniques to perform short-term predictions of traffic conditions, which included near-future traffic flow, travel time, and delay time in various locations of a traffic network. The results of the study indicated that an ANN application can deal well with complex situations and dramatically reduce the required computing power. The difference between the predictions by the ANN models and the actual data acquired from the micro-simulation model were within 5%.

Park and Rilett (1999) examined ANNs for forecasting multiple-period freeway link travel times. A fully connected multilayer feedforward ANN, combined with a backpropagation (BP) algorithm, was selected for this study. Four ANN models were designed and tested based on the differences in the type and number of input variables. The prediction results from the proposed ANN showed that, when predicting one and two time periods into the future, the ANN model that used only the preceding travel times from the target link was best. However, when predicting three through five time periods into the future, the ANN models that employed spatial data in the form of link travel times on links immediately upstream and downstream from the target link gave the lowest error. This study also indicated that, based on the disaggregated analysis performed, it might be better to classify the data before applying the model.

Van Lint *et al.* (2005) used state-space neural networks (SSNN) as a tool for freeway travel time prediction. The traditional NN models that were used to predict link travel time have the drawback of being location-specific; therefore, the results obtained

from one location could not be directly applied at a different location due to location-specific variables such as geometry, traffic control, etc. The approach proposed in this research was aimed at overcoming this problem. Another purpose of this study was to test the robustness of the proposed model when there were incidental or structural input failures of traffic data. The results showed that the SSNN model provided satisfactory predictions in a real-time application. The inclusion of corrupted data in the training data sets of the SSNN model improved robustness; however, predictive accuracy was reduced. In 2006, Van Lint further presented a measure for providing the reliability of the prediction given by an ensemble of SSNN models, which was a set of SSNN models with similar mathematical structures and different parameter sets.

Dharia and Adeli (2003) proposed using a counter propagation neural (CPN) network for forecasting freeway link travel times and compared the results from the CPN network with the results calculated using BP models. The results showed that, with similar accuracy, the CPN network demonstrated a great advantage over BP models in terms of calculation efficiency. The CPN network was nearly two orders of magnitude more efficient than the BP network.

In 2007, Singh and Abu-Lebdeh carried out research to develop SSNN models that could predict travel time on arterials under both congested and non-congested conditions. The mean absolute percentage error of modeled travel time was about 9% of observed travel time values for through traffic movement. Results showed that SSNNs can provide a framework for accurate, efficient, and robust travel time prediction.

Tao *et al.*, in their research in 2006, utilized three different types of ANNs to model corridor travel time prediction in the presence of traffic incidents using data

collected from a highway corridor in Northern Virginia. Sensitivity analysis was employed to test the relative importance of the different input channels. Test results revealed that some input channels always played significant roles, while some others were always insignificant. The findings of this study indicated that different types of ANNs might specialize in different portions of the input pattern space. Thus, developing new types of ANNs that can take advantage of the benefits of different types of networks in traffic prediction may be a promising area of study.

In a study carried out by Liu *et al.* in The Netherlands in 2006, a hybrid model for urban arterial travel time prediction, based on an SSNN and the extended Kalman filter (EKF), was presented. In this study, it was proposed that an EKF be used to train the SSNN and improve its effectiveness, as opposed to using some of the more conventional approaches. Instead of treating the neural network as a “black box” model, the design explicitly builds the relationship between neurons and physical relationships. The research team compared the performance of this proposed model with two existing models. The results of the comparisons indicated that this proposed model is capable of dealing with complex nonlinear urban arterial travel time predictions with satisfactory effectiveness, robustness, and reliability.

Adeli (2004) introduced new computational models for estimating the work zone capacity and queue length as a function of a large number of factors impacting the work zone and queue length estimations. Adeli then implemented these models into a user-friendly interactive object-oriented software system for the effective management of traffic in work zones. This research explored the use of several recent computing and information technologies: 1) CBR, 2) ANNs, 3) fuzzy logic, and 4) object-oriented

programming. In Adeli's study, the radial basis function neural network (RBFNN) was used for work zone capacity and queue prediction, and the error was mostly in the range of 0.4% to 11%. However, it was found that the percentage of trucks in the network significantly affected the accuracy of the estimation.

2.6.2 Support Vector Regression

In 1995, Vapnik introduced a regularized kernel-based technique called the Support Vector Machine (SVM). This is a relatively new supervised learning algorithm commonly applied in micro-array analysis. It has been used successfully in a number of applications including speech and letter recognition. Support Vector Regression (SVR) is a particular implementation of SVM. A Support Vector Machine (SVM) is basically a classification method that performs classification tasks by constructing hyperplanes in a multidimensional space. These hyperplanes then separate cases of different class labels (StatSoft Electronic Textbook, accessed on December 10, 2006). The mathematics behind SVM and SVR are quite complex, and so only the basic ideas and concepts will be discussed here.

Consider a set of training data, $\{(x_1, y_1), \dots, (x_l, y_l)\}$, where each $x_i \in R^n$ denotes the input space of the sample and has a corresponding target value $y_i \in R$ for $i=1, \dots, l$, where l corresponds to the size of the training data. The idea of the regression problem is to determine a function that can approximate future values accurately.

The generic SVR estimating function takes the form:

$$f(x) = (w \cdot \Phi(x)) + b \quad (2.1)$$

where $w \in R^n$, $b \in R$, and where Φ denotes a non-linear transformation from R^n to high dimensional space. The goal is to find the value of w and b such that values of x can be determined by minimizing the regression risk:

$$R_{reg}(f) = C \sum_{i=0}^{\ell} \Gamma(f(x_i) - y_i) + \frac{1}{2} \|w\|^2 \quad (2.2)$$

where $\Gamma(\cdot)$ is a cost function, C is a constant, and vector w can be written in terms of data points as

$$w = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) \Phi(x_i) \quad (2.3)$$

By substituting equation (2.3) into equation (2.1), the generic equation can be rewritten as

$$\begin{aligned} f(x) &= \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) (\Phi(x_i) \cdot \Phi(x)) + b \\ &= \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) k(x_i, x) + b \end{aligned} \quad (2.4)$$

In equation (2.4), the dot product can be replaced with function $k(x_i, x)$, known as the kernel function. Kernel functions enable a dot product to be performed in high-dimensional feature space using low-dimensional space data input, without knowledge of the transformation Φ . All kernel functions must satisfy Mercer's condition, which corresponds to the inner product of some feature space. The radial basis function (RBF) is commonly used as the kernel for regression:

$$k(x_i, x) = \exp \left\{ -\gamma |x - x_i|^2 \right\} \quad (2.5)$$

Some common kernels are shown in Table 2.1.

Table 2.1 Common Kernel Functions.

Kernels	Functions
Linear	$x \cdot y$
Polynomial	$[(x * x_i) + 1]^d$
RBF	$\exp\{-\gamma x - x_i ^2\}$

The ε -insensitive loss function is the most widely used cost function. The function is of the form

$$\Gamma(f(x) - y) = \begin{cases} |f(x) - y| - \varepsilon, & \text{for } |f(x) - y| \geq \varepsilon \\ 0 & \text{otherwise} \end{cases} \quad (2.6)$$

By solving the quadratic optimization problem in equation (2.7), the regression risk in equation (2.2) and the ε -insensitive loss function (2.6) can be minimized:

$$\frac{1}{2} \sum_{i,j=1}^{\ell} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)k(x_i, x_j) - \sum_{i=1}^{\ell} \alpha_i^* (y_i - \varepsilon) - \alpha_i (y_i + \varepsilon)$$

subject to

$$\sum_{i=1}^{\ell} \alpha_i - \alpha_i^* = 0, \quad \alpha_i, \alpha_i^* \in [0, C] \quad (2.7)$$

The Lagrange multipliers, α_i and α_i^* , represent solutions to the above quadratic problem that act as forces pushing predictions toward target value y_i . Only the non-zero values of the Lagrange multipliers in equation (2.7) are useful in forecasting the regression line and are known as support vectors. For all points inside the ε -tube, the Lagrange multipliers that equal to zero do not contribute to the regression function. Only if the requirement $|f(x) - y| \geq \varepsilon$ is fulfilled may Lagrange multipliers be non-zero values and be used as support vectors.

The constant C introduced in equation (2.2) determines penalties of estimation errors. A large C assigns higher penalties to errors so that the regression is trained to minimize errors with lower generalization, while a small C assigns fewer penalties to errors. This allows for the minimization of the margin of errors and, thus, higher generalization ability. If C were to become infinitely large, SVR would not allow the occurrence of any error, and the result would be a complex model. When C goes to zero, however, the result would tolerate a large amount of errors, and the model would be less complex.

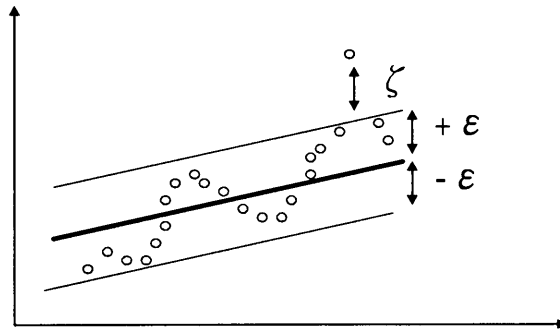


Figure 2.6 Positive Slack Variables ζ_i Measuring Points Lying Outside of Tube

The value of w has been derived in terms of the Lagrange multipliers. The variable b can be computed by applying Karush–Kuhn–Tucker (KKT) conditions which, in this case, implies that the product of the Lagrange multipliers and constraints has to equal zero:

$$\begin{aligned}\alpha_i(\varepsilon + \zeta_i - y_i + (w, x_i) + b) &= 0 \\ \alpha_i^*(\varepsilon + \zeta_i^* + y_i - (w, x_i) - b) &= 0\end{aligned}\tag{2.8}$$

and

$$\begin{aligned}(C - \alpha_i)\zeta_i &= 0 \\ (C - \alpha_i^*)\zeta_i^* &= 0\end{aligned}\tag{2.9}$$

where ζ_i and ζ_i^* are slack variables used to measure errors outside the ε tube. Because $\alpha_i, \alpha_i^* = 0$ and $\zeta_i^* = 0$ for $\alpha_i^* \in (0, C)$, b can be computed as follows:

$$\begin{aligned} b &= y_i - (w, x_i) - \varepsilon \quad \text{for } \alpha_i \in (0, C) \\ b &= y_i - (w, x_i) + \varepsilon \quad \text{for } \alpha_i^* \in (0, C) \end{aligned} \quad (2.10)$$

Having acquired these solutions, SVM and SVR can be used without knowledge of the transformation.

Thus far, SVR has only limited applications within the transportation field. Wu *et al.* (2003) applied SVR to develop a model for travel time prediction. The travel time predicted by the SVR model was compared with predictions derived by other models based on the real highway traffic data in this study. Results showed that the SVR predictor significantly outperformed other prediction models, which demonstrated the capability of SVR in traffic data analysis.

Vanajakshi proposed the use of SVR to predict travel time in his dissertation in 2004. ANN models were also developed as a comparison. Based on the investigation conducted in this dissertation, it was found that SVR is a viable alternative to ANN for short-term prediction, especially when the training data have variations and the amount of training data is not large. In cases where more training data existed, the performance of SVM was comparable to ANN, thus making it an alternative option for prediction problems.

Bhavsar *et al.* (2007) introduced the use of an SVR model in addition to a CBR model in their research on constructing a decision support system for real-time traffic management. In their study, prototype SVR and CBR decision support tools were

developed and used to evaluate the likely impacts of implementing diversion strategies in response to incidents on a highway network in Anderson, South Carolina. The performance of the two models was evaluated based on the results produced by the VISSIM simulation model. The results showed that the predictions from the SVR and CBR were comparable to those from the VISSIM model. In most cases, the SVR outperformed the CBR. In a later study from this research team in 2006, to further illustrate the feasibility of the SVR and CBR models, two different highway networks covering portions of I-85 and I-385 in South Carolina were used to train the SVR and CBR models. These models were then used on a third network covering portions of I-89 in Vermont. In this study, only a 15% difference between the predictions of the SVR and those of the PARAMICS simulation model was found.

2.6.3 Case-Based Reasoning

Case-based reasoning (CBR), broadly construed, is the process of solving new problems based on the solutions of similar past problems. CBR has been applied to problems such as conceptual design of mechanical devices based on a stored library of previous designs, reasoning about new legal cases based on previous rulings, and solving planning and scheduling problems by reusing and combining portions of previous solutions to similar problems (Mitchell, 1997). CBR has been formalized for the purposes of computer reasoning as a four-step process as shown in Figure 2.7 and described below:

- Retrieve: Given a target problem, retrieve cases that are relevant to solving it from memory. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived.
- Reuse: Map the solution from the previous case to the target problem. This may involve adapting the solution as needed to fit the new situation.
- Revise: Having mapped the previous solution to the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise.
- Retain: After the solution has been successfully adapted to the target problem, store the resulting experience as a new case in memory.

At the core of the CBR process is a case base that stores previous instances of problems and their derived solutions. When faced with a new problem, the system first accesses the case base and retrieves the case(s) most similar to the new case. During the reuse process, the solution of the retrieved case is adapted to address the current problem more appropriately. The solution is then tested for success during the revise process, by either directly implementing it in the real world or by having it evaluated by a teacher (e.g., a human expert). If the case is a success, it is retained in the case base for future reuse.

CBR uses specific knowledge of previous situations or cases and allows for incremental, sustained learning because a new experience can be saved each time a problem has been solved. This new case becomes available for future problems.

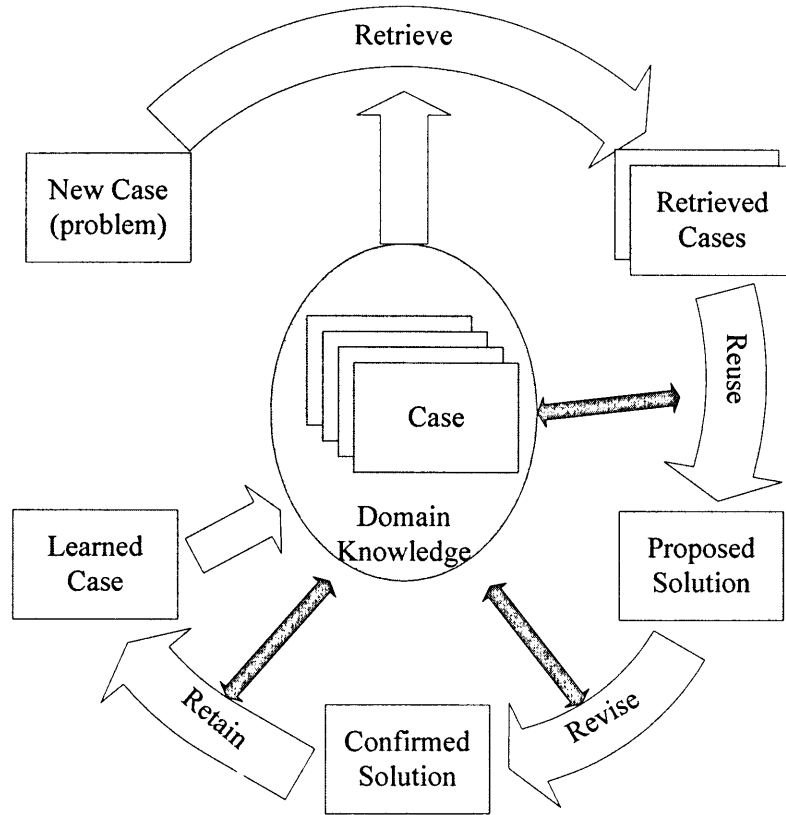


Figure 2.7 Case-Based Reasoning Cycle (Mitchell, 1997).

CBR was introduced to solve transportation problems more than a decade ago. Khattak and Kanafani (1996) explored the ability of CBR to function as a planning tool for the ITS. Sadek *et al.* (1999) explored the potential for using CBR to overcome the limitations of existing approaches to developing real-time routing strategies. To illustrate the feasibility of the approach, the research team developed and evaluated a prototype CBR routing system for the interstate network in Hampton Roads, Virginia. They generated cases for building the system's case base using a heuristic dynamic traffic assignment (DTA) model designed for the region. Using a second set of cases, the research team evaluated the performance of the prototype system by comparing its solutions with those of the DTA model. The research team found that CBR had the

potential to overcome many of the limitations of existing approaches to real-time routing and that a CBR routing system was capable of producing high-quality solutions with a reasonable case base size. In addition, the research team found that real-time traffic flow routing would likely lead to significant user cost savings.

Sadek *et al.* (2003) carried out another study in which they proposed using ANNs in addition to CBR and compared the performance of the two paradigms. For this study, the data-mining models were developed for a test network representing a part of the freeway and arterial system around the city of Hartford, Connecticut, defined by I-91 to the East and Route 99 to the West. The diversion scenario considered in this study involved the use of VMS to divert southbound I-91 traffic onto Route 99, in case of incidents occurring on I-91. A total of 360 different cases were generated, and the DTA model was used to analyze each case. Another data set consisting of 25 cases was generated for the purpose of testing and evaluating the paradigms. The case study illustrated the critical need for including validation and evaluation as a part of the development effort. It also showed the need to tune the design parameters of the data-mining paradigm, which could have a significant impact on performance.

CBR was used as a tool in a study carried out by Dahlgren *et al.* (2004) to help users who wanted to address specific problems and obtain information related to their particular contexts. The CBR tool was used as query tool to search for cases of different locations where different technologies had been implemented once certain criteria were input into the system. The cases were presented in order of their resemblance to the input parameters. The users could access more information on any of these cases by clicking

on a hyperlink. A later study was performed in 2006 by the same research team using CBR to search for the most similar historical ITS implementations and their impacts.

In short, CBR does not identify the mathematic connections between the input data and the result. When faced with a new problem, the CBR system first accesses the case base and searches for the case(s) most similar to the new case. Therefore, CBR is less comparable to the ANN models and SVR models, which, in essence, develop a mathematical relationship between the input data and target data. In this research, it is more reasonable to compare the performance of SVR and ANN models.

2.7 Summary

In this chapter, a comprehensive literature review has been performed to investigate and assess the current techniques, tools, and methods for implementing route diversion plans. The purposes of this review are to determine the current status of the research area, to identify the problems pending to be solved, to determine the objectives, and to form the research framework and tasks for the dissertation. The major findings of the literature review are summarized below:

A route diversion problem involves a complex situation in which variable factors, including freeway geometric and traffic conditions, local network geometric and traffic conditions, incident location and severity, ITS devices availability, and driver's behavior, need to be considered. This requires an understanding of the connection between these factors and the feasibility of a route diversion strategy.

DYNASMART-P was selected to serve the purpose of achieving the correct representation of used paths in dynamic traffic assignment (DTA), as well as the methods

of determining optimal paths (from a system point of view) toward which drivers should be (re-) directed. In this research, DYNASMART-P will be used as the simulation model.

Because DYNASMART-P cannot perform online simulation, methods have to be developed for the design of a real-time route diversion DSS. Data mining techniques have been utilized more and more to solve transportation problems. CBR was first introduced to assist in route diversion decision making. SVR was then introduced a year ago for the same purpose. CBR is more appropriately used when the case base is large enough. When there are limited number of cases in the case base, CBR generally cannot outperform ANN and SVR models. Therefore, in this research, only SVR and ANN models will be explored and evaluated for their performance in assisting with route diversion decisions.

CHAPTER 3

IDENTIFICATION OF POTENTIAL VARIABLES

3.1 Introduction

In this chapter, considerations for the pre-selection of a route diversion alternate route were described. Variables that have critical impacts on freeway route diversion decision making were identified. These factors were explained and evaluated based on: (1) their significance in affecting route diversion decisions, and (2) the possibility of collecting the data or representing the identified factors in a real transportation environment.

3.2 Variables Identification

Freeway traffic diversion is a decision sensitive to a variety of variables. As mentioned in Chapter 2, freeway conditions, incident data, ITS devices, human factors, and local network conditions are all identified as factors that impact route diversion decisions. In addition, weather conditions, time of day and even land use could be determinant factors for route diversion decisions. In this section, these factors will be categorized into factors for pre-planning and factors at the time of incidents

3.2.1 Restrictive Criteria for Alternate Route Pre-Planning

When a major incident occurs that severely limits roadway capacity, motorists will naturally find ways to divert around the incident. Some regions have chosen to formally establish alternate routes to allow the direction of traffic onto those routes that are best suited to handle this increased traffic demand (FHWA, 2000). In this research, it

is assumed that alternate routes are pre-planned. At the time of incidents that justify route diversion decisions, motorists will be advised which alternate route to follow.

If a region chooses to implement alternate routes, it is critical that all agencies affected by the implementation and operation of these routes be involved in every step of planning for them. What appears to be the most logical alternate route to an outside operating agency may, in fact, be unacceptable to local stakeholders. As determined through a literature review (FHWA, 2000; FHWA 2006), the criteria that should be considered when designing an alternate route are summarized and listed below.

- Proximity of alternate route to closed roadway: The distance of the alternate route between the exit point of the freeway and the re-entry point of the freeway will determine the travel time on this alternate route. The longer the distance, the longer the travel time will be, and the less preferable the alternate route will be. In addition, an alternate route far from the freeway will increase the possibility of motorists failing to follow the right route and reduce the reliability of the diversion process.
- Ease of access to/from alternate route: Ease of access from the freeway to an alternate route and from the alternate route back to the freeway affects motorists' willingness to follow diversion instructions. More importantly, difficulty in switching to and from the alternate route could cause safety issues. Incidents are more likely to occur when it is difficult to change lanes or change routes.
- Safety of motorists on alternate route: Safety is always among the primary considerations when making traffic operation decisions. It is not preferable to sacrifice safety for delay reduction. If traffic were to be diverted to an alternate

route that is not as safe as staying on the freeway, it would be advisable to have motorists stay on the freeway.

- Height, weight, width, and turning restrictions on the alternate route (e.g., for commercial and transit vehicles): Normally, when the freeway is not fully closed due to incidents, commercial vehicles or buses can be informed to stay on the freeway if there are height, weight, or width restrictions on the alternate route. However, when the freeway is totally blocked and diversion is the only choice, an alternate route plan considering the restrictions on commercial vehicles and buses should be prepared.
- Percentage of heavy vehicles (e.g., trucks, buses, RVs) on the route from which traffic is to be diverted: Historical data on the percentage of heavy vehicles on the route from which traffic is to be diverted should be taken into consideration when pre-planning alternate routes. This is because these vehicles usually require special accommodations on the alternate route.
- Number of travel lanes on, or capacity of the alternate route: This determines the amount of traffic the alternate route can accommodate. Diverting more traffic than the alternate route can accommodate will cause operational failure on the alternate route and, thus, can deteriorate network performance further. The number of travel lanes or capacity of the freeway should also be considered because it is an indication of the maximum traffic that could be diverted at the time of an incident.
- Congestion induced on alternate route: Similar to the number of travel lanes discussed above, congestion induced on an alternate route affects the system-wide

benefits of diverting traffic to the alternate route. The level of congestion induced on an alternate route cannot be estimated before the occurrence of incidents. It needs to be verified before, as well as when the incident happens.

- Number of signalized intersections, stop signs, and unprotected left turns on alternate route: These affect motorists' travel time and safety on these routes as well. Usually, un-signalized intersections should be avoided when designing the alternate route. An alternate route that requires unprotected left turns for the diverted traffic is not preferable, as it will increase travel times and raise extra safety concerns as well.
- Typical travel time on alternate route (e.g., posted speed limits): Posted speed limits along the alternate route can be used to calculate the minimum travel time for diverted traffic. Also, more consistent speed limits from section to section along the route can provide a more stable traffic flow, especially at times of congestion. The speed limits are usually confined by the geometric condition and surrounding environment.
- Pavement conditions on alternate route: They affect the ease of driving along the way and, therefore, the level of service provided to motorists.
- Type and intensity of residential development and commercial development on alternate route: Usually, areas with intensive residential and commercial development should be avoided when designing alternate routes. The same applies to those areas with high pedestrian flows.

- Existence of schools and hospitals on alternate route: Schools and hospitals make an arterial undesirable as an alternate route. When designing alternate routes, school zones and hospital areas should be avoided.
- Grades, curvature, and turning radii on alternate route
- Potential for deployment of temporary control devices: The alternate route should allow for the deployment of temporary control devices that will be used at the time of diversion.

3.2.2 Restrictive Criteria under Consideration at Time of Incidents.

At the time of incidents, the traffic conditions both on the freeway and on the pre-planned alternate routes are important criteria that should be considered when making the route diversion decision. These criteria are listed below.

- Traffic conditions on the freeway under the impact of incidents: As defined previously, the incident management process starts with incident detection. Following detection is incident verification, which is the confirmation of the occurrence of an incident. Verification includes gathering enough information to dispatch the proper initial response. Traffic conditions that should be taken into account for a route diversion decision include the traffic volume, speed, and queue length when the incident is verified. Also to be taken into consideration is the traffic volume, speed, and queue length upstream of the incident location several minutes before the verification of the incident.
- Traffic conditions on the pre-planned alternate route for the section of freeway under the impact of incidents: Traffic volume on the pre-planned alternate route a

few minutes before the verification of incidents can also impact the route diversion decision.

- Incident conditions on the freeway: Incident conditions such as incident location, number of lanes blocked by the incident, and capacity lost due to the incident are all key information needed for route diversion decisions. In addition to the incident information described above, incident duration should also be predicted as an important input for route diversion decisions. Zhan (2006) proposed using data mining as an approach to predict incident duration in his doctoral dissertation. The historical incident duration data from Florida Department of Transportation District 4 were used, and five models were developed for real-time predictions of incident duration. The results of the study showed that these models could predict incident duration within 20 minutes with satisfactory accuracy. With more effort being put into exploring techniques for predicting incident duration, this information should be made available in real time when an incident is verified.
- ITS devices availability: The application of advanced sensor, computer, electronics, and communication technologies provide a complementary means to maximize the efficiency of the transportation infrastructure. In the event of a freeway incident, Closed Circuit Television (CCTV) can be used to provide the traffic operation centers with timely information about the specific location, nature, and extent of the incident (FHWA, 2000). Although most incidents are reported by wireless calls to 911 numbers given the wide use of cell phones, with the assistance of CCTVs, an incident can be verified much more quickly than it

can via traditional measures. CCTVs can also be used to monitor the whole incident management process until the recovery of the traffic conditions to normal status. Various types of detectors can transmit real-time traffic volume, speed, and even queue length to the traffic operation center. These data provide valuable information at the time of an incident. VMSs and HARs are commonly used to disseminate traffic information. Information displayed on VMSs includes incidents, congestion, work zones, alternative travel time, speed limits, bad weather, or special events. Especially useful under incident conditions, VMSs aim to influence driver routing decisions to enhance network performance. Ramp metering is another important ITS application that can affect route diversion decisions. When an incident occurs, metering rates can be adjusted to control the traffic entering the freeway. Furthermore, motorists knowledgeable about alternate routes will divert to avoid the delay at the meters.

Availability of the aforementioned devices can provide the possibility of quicker detection of, verification of, and response to freeway incidents. Most of the traffic information at the time of incidents is provided to traffic operators through the use of these devices.

3.3 Variable Selection

Route diversion is a decision dependant on many factors. This study is intended to explore the feasibility of incorporating data mining techniques into real-time decision support for freeway traffic diversion in the event of traffic incidents. Therefore, some factors that have minor impacts on the route diversion decision were excluded to simplify

the study. In addition, in this research, only one alternate route was planned for each incident location, therefore, factors such as the length of the alternate route, the volume over capacity ratio on the alternate route, as well as the number of signalized intersections were not used as input to the model. Due to the inability of DYNASMART-P to specify the exact incident location on a link, the distance from the incident scene to its nearest upstream exit ramp was also not used as one of the variables.

3.3.1 Traffic Conditions

Traffic condition measures such as real-time traffic volume, speed, and queue length are considered in this research. Traffic detectors have been widely implemented in the transportation network in this country, which has made obtaining real-time traffic data possible. When an incident occurs, the freeway volume, speed, and queue length patterns under the impact of the incident will be altered. These measures and their changing patterns within a certain time are part of the determinants for route diversion decisions. In addition to freeway traffic conditions, the traffic conditions on the local network, particularly on the alternate route, are components of determinants for route diversion decisions as well. They will also be taken as input variables in this research. The overall network congestion level is another important factor influencing route diversion decisions. In a real transportation environment, the level of congestion of a network is usually highly related to time-of-day and the overall demand of the network.

3.3.2 Incident Conditions

Incident location, level of severity, capacity lost on the freeway, and incident duration are considered important factors impacting route diversion decisions and are,

therefore, selected as variables in this research. Incident duration is defined here as the timeframe when the incident takes place. Capacity lost is the measure of the level of severity of the incident.

3.3.3 ITS Device Availability

In this research, VMSs will be used as the ITS device for delivering information to motorists. It is assumed that VMSs are installed on the immediate upstream links where the pre-planned alternate route diverts from the freeway. The duration of diversion messages shown on the VMSs will be taken as one of the variables for route diversion decisions.

3.3.4 Motorist's Behavior

The percentage of motorists responding to the route diversion information indicates how much more traffic the alternate route needs to accommodate. This is in addition to the route's normal traffic and how much traffic will stay on the freeway.

3.4 Summary

In this chapter, potential variables that may impact route diversion decisions were identified. In summary, the following variables will be considered as the input for the data mining models:

- Relevant link traffic volume, speed, and queue length
- Number of lanes blocked/capacity lost due to incident
- Incident duration
- VMS message duration

- Percentage of motorists responding to route diversion information

These factors will be discussed further in the next chapter, where DYNASMART-P's ability to simulate the incident will be investigated, and where different incident scenarios will be designed. This will allow for the acquisition of both input and output for the data mining ANN and SVR models that will be developed in Chapters 5 and 6.

CHAPTER 4

DYNASMART-P SIMULATOR AND INCIDENT CASE DESIGN

4.1 Introduction

In this chapter, a DYNASMART-P simulator was used to produce the input for the ANN and SVR models. The ability of DYNASMART-P to simulate incident scenarios was explored. The algorithm provided by DYNASMART-P was applied to calculate the optimum path that was considered the pre-planned alternate route for the incident locations designed. Incident scenarios representing different combinations of the variables identified in Chapter 3 were designed. DYNASMART-P was used to simulate each one of these scenarios. A MATLAB program was developed to automate these 3000 simulations, extract the output from the results of the simulations, and prepare for the training and testing of the ANN and SVR models.

4.2 Simulation Network

One of the two demonstration networks in the DYNASMART simulation model, Dallas-Fort Worth 1.0, will be used as the study network. This network was chosen because 1) The size of the network is large enough. It includes at least six on ramps and six off ramps on each side of the freeway, and it possesses at least one alternate route for each exit of the freeway; and 2) The network has been well constructed, calibrated, and is ready for use for the purposes of this study. Figure 4.1 is a screen capture of the network. There are a total of 200 nodes coded in the network, which connects 445 links. The whole network is divided into 13 zones.

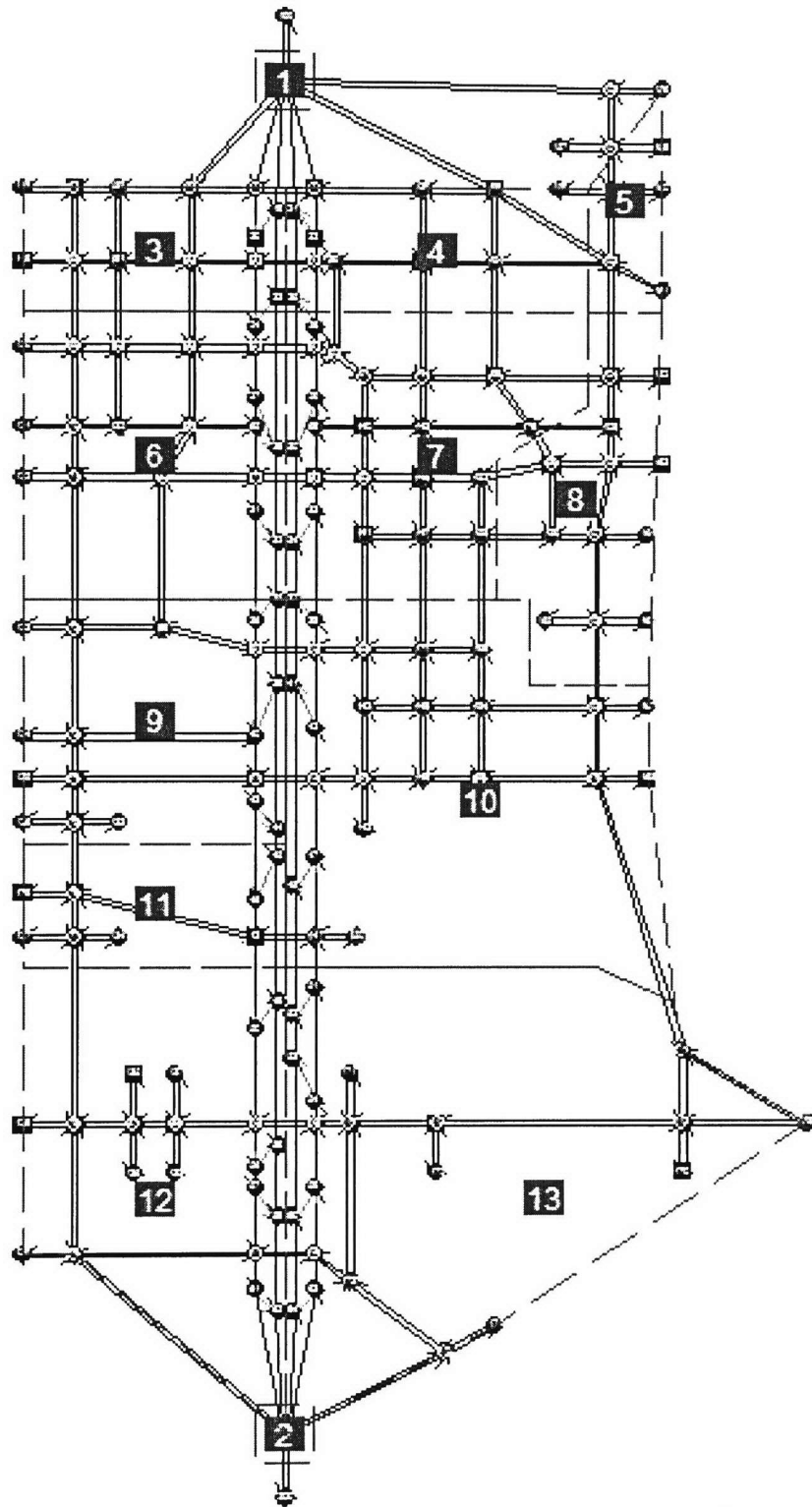


Figure 4.1 Selected Study Network in DYNASmart-P

In Chapter 3, a series of factors were identified as the variables for incident scenarios. They are: relevant link traffic volume, speed, and queue length; number of lanes blocked/capacity lost due to the incident; incident duration; VMS message duration; and the percentage of motorists responding to the route diversion information. These variables need to be further “translated” into the “language” that DYNASMART-P can recognize. Not all of these variables are input for DYNASMART-P. Some of them are inputs to the DYNASMART-P simulator and some are outputs from the simulator. Based on this standard, these variables are divided into two groups: Simulation Input and Simulation Output. These variables will be used as data for training and testing the ANN and SVR models.

4.3 Simulation Input

The simulation inputs include incident location, duration, level of severity, pre-selected alternate route, VMS location, VMS message duration, and the level of congestion in the network. These inputs will be read out from the incident.dat, vms.dat, scenario.dat, and demand.dat files in DYNASMART-P.

4.3.1 Incident Location, Duration, and Level of Severity

Two locations on the freeway were chosen as the training and testing locations, respectively. In DYNASMART-P, when adding an incident to the network, it can only be added to a link. The exact location of the incident on a link cannot be specified in DYNASMART-P.

The training location is coded on a northbound freeway link 48→41. As shown in Figure 4.2, there are five exit ramps upstream of this link and five on ramps downstream of the link.

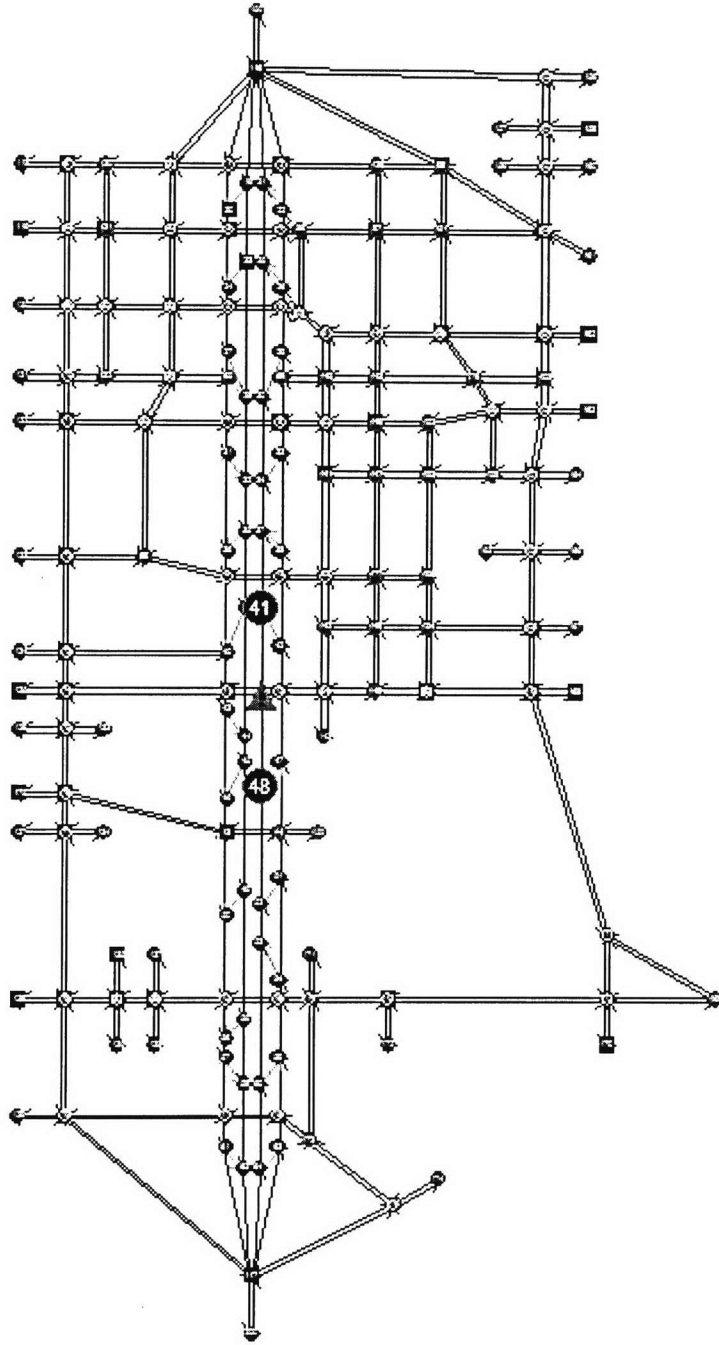


Figure 4.2 Training Location (on Link 48→41)

Incident duration is defined here as the timeframe when the incident takes place. In DYNASMART-P, incident starting and ending time are user input, and the incident duration is the difference of the two. In this research, incident duration is assumed to be an input from the traffic operators upon verification of the incident. Incident level of severity is represented by specifying the capacity lost on the incident impacted link. Usually, the real capacity lost of an incident scene is greater than the proportion of incident blocked lanes to the total number of lanes. The capacity lost is also assumed to be an input based on the judgment of the traffic operators upon verification of the incident.

All the above variables are defined in the incident.dat file in the DYNASMART-P simulator. The description of this file is shown in Table 4.1. In this table, the severity of an incident is defined as the fraction of link capacity lost due to the incident. (The remaining capacity becomes one minus the severity.) Table 4.2 is a sample incident.dat file coded for this study. In this file, a total of one incident is coded on link 48→41. Incident start time is 15 and end time is 40, with a capacity loss of 60%.

Table 4.1 Description of INCIDENT.DAT Input File

Record Type	Field	Description
General	1	Total number of incidents in the network
Incident Description	1	Upstream node of the first incident link
	2	Downstream node of the first incident link
	3	Start time of the first incident (minutes)
	4	End time of the first incident (minutes)
	5	Severity of the first incident
.....
Incident Description	1	Upstream node of the last incident link
	2	Downstream node of the last incident link
	3	Start time of the last incident (minutes)
	4	End time of the last incident (minutes)
	5	Severity of the last incident

Table 4.2 Sample INCIDEND.DAT File

1				
48	41	15	40	0.6

4.3.2 Alternate Routes Selection

A central feature in DYNASMART-P is the notion of multiple user classes (MUCs). DYNASMART-P is capable of simultaneously modeling different classes of users with different choices and assignment rules under different information levels. DYNASMART-P recognizes five different user classes in terms of the availability of Advanced Traffic Information System (ATIS) equipment, driver's knowledge of the network, and driver response to supplied routing information (Mahmassani and Sbayti, 2005). In this regard, DYNASMART-P has the ability to specify a multiple user class (MUC) distribution that is applicable across all vehicle types in the network, or separate MUC distributions for each vehicle type. These five MUCs are (Mahmassani and Sbayti, 2005):

- Class 1 – Unresponsive: This class of users is not responsive to any type of information, and it is used to model pre-trip information. These users receive path assignments at the beginning of simulation and adhere to these paths throughout the entire simulation. This class only responds to VMS type 2. Reference of the different types of VMSs are given in the next section.
- Class 2 – System Optimal (SO): This class of users follows the system optimal (SO) assignment rule, in which travel times are minimized from the system's perspective. The general idea of this assignment rule is to force a small fraction of users to follow sub-optimal routes from their perspective (not user optimal or

user equilibrium) for the benefit of the majority. Such a class is only available if iterative consistent assignment is chosen. Similar to the UE class, this user class is only responsive to VMS types 2 (mandatory detour) and 4 (optional detour) unless the VMS preemption mode is deselected (or unchecked), whereby this user class will also respond to VMS information. The total network-wide travel time resulting from this assignment is less than or equal to that generated from the UE assignment rule.

- Class 3 – User Equilibrium (UE): This class of users follows the UE assignment rule, in which travel times are minimized from the user's (traveler's) perspective. Such a class is only available if iterative consistent assignment is chosen and is used to model travelers who are familiar with the network. This user class is only responsive to VMS types 2 (mandatory detour) and 4 (optional detour) unless the VMS preemption mode is deselected (or unchecked), whereby this user class will also respond to VMS information. Note that this assignment rule provides an upper bound for the network-wide travel time relative to the system optimal assignment rule.
- Class 4 – En-route Info (Boundedly Rational Behavior): This class of users updates its paths at each intersection based on the prevailing shortest path tree. It is designed to reflect en-route information and is based on "boundedly rational behavior." Two criteria are used for route choice, namely, the indifference band and the threshold bound for switching decisions. The indifference band reflects a fraction of travel time improvement below which the user will not switch routes. The threshold bound reflects a time improvement (in minutes) below which the

user will not switch routes. Should any of these two criteria be exceeded, the user will switch routes at the next intersection. This class of users is only responsive to detours (VMS type 2), and generally does not respond to VMS information unless the VMS preemption mode is deselected (or unchecked).

- Class 5 – VMS Responsive: This class of users responds to VMS information. There are four types of VMS information: congestion warnings, optional and mandatory detours, and speed advisories. VMS responsive users receive path assignments at the beginning of simulation, which they adhere to unless they encounter a VMS, and possibly decide to change their paths as a result.

In this research, the DYNASMART-P simulator was used to find the best alternate route by specifying the multiple user class as the 100% system optimum. During the incident duration, more than one path was used as the optimal path. The routes closest to the freeway were chosen as the alternate routes. Table 4.3 presents the properties of the links on the alternate routes planned. Figure 4.3 shows the alternate route for the training location (on link 48→41). The path it follows is 52→51→14→47→12→42→41.

Table 4.3 Link Properties on the Alternate Route for Training Location

Link ID	From Node	To Node	Number of Lanes	Speed Limit	Saturation Rate	Type	Length (mile)
106	52	51	1	40	1,800	Off Ramp	0.28
104	51	14	3	40	1,800	Arterial	0.09
41	14	47	2	40	1,800	Arterial	0.45
98	47	12	4	40	1,800	Arterial	0.16
35	12	42	3	40	1,800	Arterial	0.18
91	42	41	1	40	1,800	On Ramp	0.21

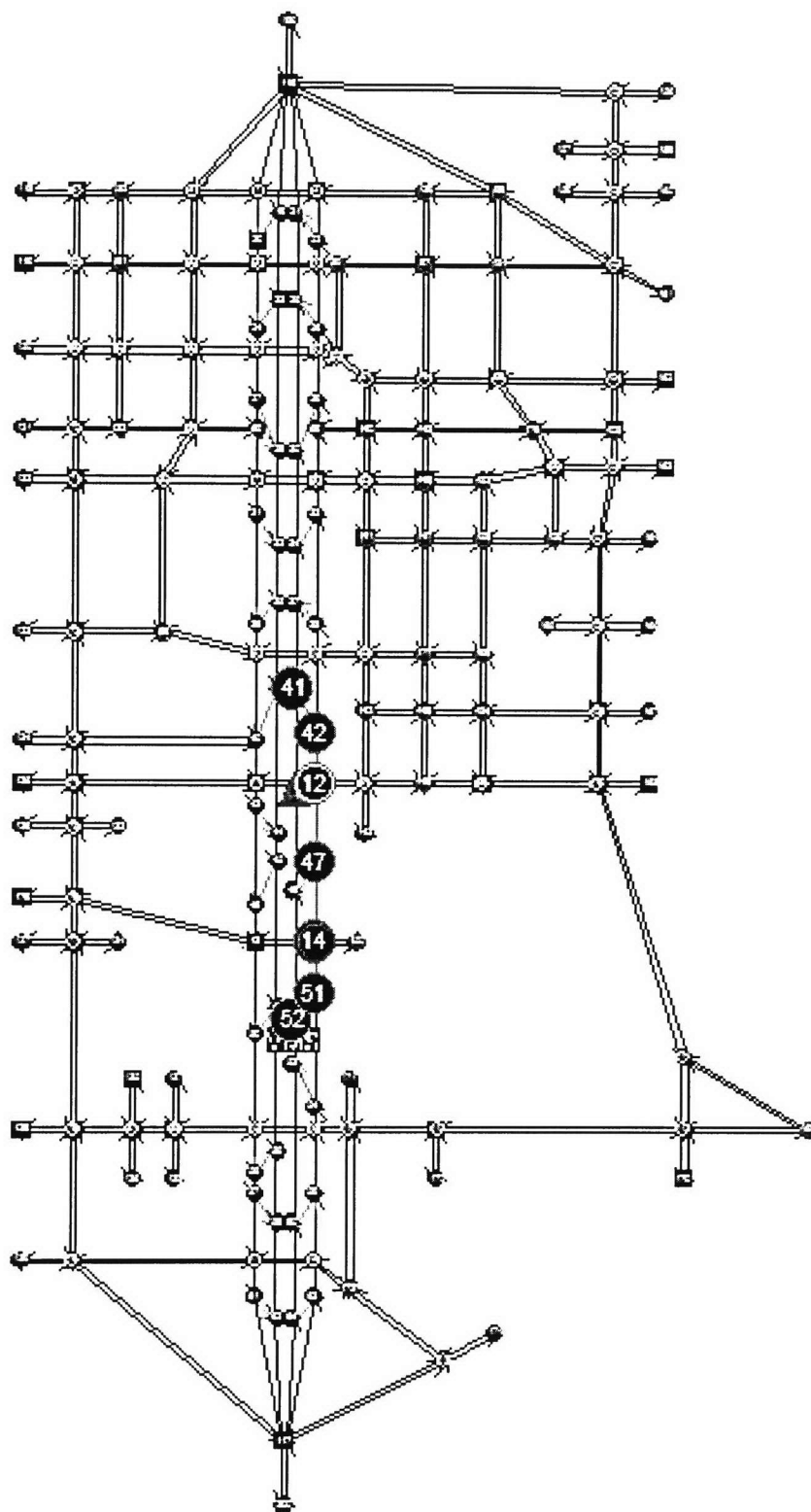


Figure 4.3 Alternate Route for the Training Location

4.3.3 Variable Message Signs (VMS)

In DYNASMART-P, a `vms.dat` file is used to specify the configurations of the VMS. Four types of VMS are supported by DYNASMART-P (Mahmassani and Sbayti, 2005). Type 1 VMS is the speed advisory VMS that allows users to increase/decrease speed by a certain percentage when below/above a certain threshold. Type 2 VMS is the mandatory detour VMS that advises drivers of lane closures and mandates that all vehicles follow some user-specified sub-path in the vicinity. Type 3 VMS is the congestion warning VMS, which allows users to specify percentages of VMS-responsive vehicles (user class 5) to evaluate the VMS information and divert if a better path exists. Finally, type 4 VMS is the optional detour VMS. Similar to type 2, it also advises drivers with lane closure information. However, type 4 gives drivers the option to follow the detour path or keep their original path, based on the boundedly rational decision rule. In this research, type 4 VMS is selected, and the percentage of drivers who will react to this information will be specified in a `scenario.dat` file. This will be discussed later. Two VMSs were selected, one for each incident location.

VMSs can be specified at any location along the network. In the current version of DYNASMART-P, the diversion behavior is modeled through the response rate, a user-specified parameter that indicates the percentage of VMS drivers who may switch routes due to a VMS. As described in the MUC in DYNASMART, only user class 5 (VMS responsive) drivers will respond to (and evaluate) VMS types 1 and 3, whereas all drivers will respond to VMS types 2 and 4. Table 4.4 describes the attributes of the `vms.dat` file. In this table, field 4, under Sign Description, is 100, which is read but ignored by DYNASMART-P. It is used to keep the same number of fields for VMS types. Table 4.5

presents a sample vms.dat file coded for this research. In this file, a total of one VMS was coded on link 53→52. A type 4 VMS was selected, and the message started at the 18th minute and ended at the 45th minute. There are a total of seven nodes on the alternate route, and they are listed on the last line of the file.

Table 4.4 Description of VMS.DAT Input File

Record Type	Field	Description
Number of Signs	1	Number of Viatiable Message Signs
Sign Description	1	Type of VMS (1-4)
	2	Upstream node of the first VMS link
	3	Downsream node of the first VMS link
	4	Type 4:100
	5	Type 4: number of nodes in detour sub-path
	6	Start time for the first VMS (minutes)
	7	End time for the first first VMS (minutes)
Subpath	1	First node in the detour sequence for the first VMS (if Applicable)

	N	Last node in the detour sequence for the first VMS (if applicable)
.....
Sign Description	1	Type of VMS (1-4)
	2	Upstream node of the first VMS link
	3	Downsream node of the first VMS link
	4	Type 4:100
	5	Type 4: number of nodes in detour sub-path
	6	Start time for the last VMS (minutes)
	7	End time for the first last VMS (minutes)
Subpath	1	First node in the detour sequence for the last VMS (if Applicable)

	N	Last node in the detour sequence for the last VMS (if applicable)

Table 4.5 Sample VMS.DAT File

1							
	4	53	52	100	7	18	45
	52	51	14	47	12	42	41

In this research, a type 4 VMS is selected as the VMS that will disseminate diversion information. The duration of the message shown on the VMS will change accordingly as the incident duration changes. The VMS initiation time should be later than the starting time of the incident to reflect the incident detection and verification time. In this research, it is assumed that the detection and verification time for incidents on a freeway is short. Therefore, a total of three minutes will be used for this time.

4.3.4 Simulation Scenarios

In DYNASMART-P, the information regarding the general settings of the traffic assignment module, the statistics collection period, the proportions of user classes, and the vehicle fleet composition is specified in `scenatios.dat` (Mahmassani and Sbayti, 2005). A detailed description of this file and a sample file coded for this research are provided in Tables 4.6 and 4.7, respectively.

In the sample `scenario.dat` file, the relative difference band 0.2 and the percentage of compliant vehicles 1.00 applies for user class 4 (en-route info) only. In this research, this information is not relevant because all of the drivers are defined as either Unresponsive (Class 1) or VMS responsive (Class 5). The vehicle random number seed was set to 1234, which determines that DYNASMART-P will not reset the seed for each run. The best path was set for all vehicles, and only user class 5 will respond to VMS. The simulation interval was set at six seconds. This number is fixed in the DYNASMART-P 1.2 version. Simulation statistics in the network were set to be collected from 0 to 300 minutes, which comprises the whole simulation process. One hundred percent of the vehicles along the network were coded as passenger cars (PCs) to

simplify the simulation. Therefore, no trucks or HOVs were coded on the network. The demand.dat was selected to load the passenger cars onto the network. The fraction of demand.dat to be loaded as PCs is 1.00. Here 15 percent of the vehicles were defined as Unresponsive, and 85 percent of the vehicles were defined as VMS-responsive.

Table 4.6 Description of SCENARIO.DAT Input File

Record Type	Field	Description
General settings	1	Relative indifference band
	2	Threshold bound for switching decisions (minutes)
	3	Random number (positive) generation seed
	4	Path index (for initial paths) 0: randomly assign a path (from K-paths) to vehicles 1: assign the best path to vehicles
	5	VMS preemption mode 0: Only class 5 (VMS responsive) responds to VMS (default) 1: User classes 2 to 5 respond to VMS.
Compliance	1	Fraction of compliant vehicles
Simulation step	1	The basic simulation time step interval (6 seconds)
KSP	1	No. of simulation intervals in which the KSP calculation algorithm is executed (default = 30)
	2	No. of simulation intervals in which the KSP updating algorithm is updated (default =5)
Statistics collection period	1	Start-up time to collect statistics
	2	End time to collect statistics
No. Veh Types	1	Number of vehicle types (other than bus) modeled in the network. (Range: 1 – 3)
Veh Type /MUC Proportions	1	First vehicle type (must be a PC) in the network (must specify “1”)
	2	Demand mode for PC (must specify “0”: using demand.dat)
	3	Fraction of demand.dat to be loaded as PC
	4	MUC mode for PC (must specify 0)
	5	Proportion of PC to be non-responsive (Class 1)
	6	Proportion of PC to be SO
	7	Proportion of PC to be UE
	8	Proportion of PC to be enroute info
	9	Proportion of PC to be VMS-responsive

Table 4.7 Sample SCENARIO.DAT File

0.20	1.00	1234	1	0						
1.00										
6										
	50	30								
0.00	300.00									
1										
	1	0	1.000	0	0.150	0.000	0.000	0.000	0.850	

4.3.5 Level of Congestion

In DYNASMART-P, the level of congestion can be defined in the demand.dat file. This file is used to load the demand using a time-dependent O-D demand matrix. For this purpose, the number of loading intervals, a demand multiplication factor, the starting time of each loading interval, and the end of the vehicle loading period need to be specified in this file. For each loading interval, an O-D matrix needs to be prepared. A detailed description of this file is presented in Table 4.8 (Mahmassani and Sbayti, 2005).

Table 4.8 Description of DEMAND.DAT Input File

Record Type	Field	Description
General	1	Number of O-D matrices
	2	Multiplication factor
Start-Times	1-(n+1)	Start times for all (n) O-D matrices
Starting time	1	Starting time for first O-D matrix (min)
O-D matrix	1-N	Entries of first O-D matrix (pertaining to first time interval)
.....
Starting time	1	Starting time for last O-D matrix (min)
O-D matrix	1-N	Entries of last O-D matrix (pertaining to last time interval)

The level of congestion is determined by the multiplication factor. The default setting for the network under study is 40. By changing this number, the level of congestion can be changed throughout the network and for the whole simulation time. Table 4.9 presents the corresponding level of congestion and average travel speed in the network as a result of different multiplication factors (Mahmassani and Sbayti, 2005).

Table 4.9 Corresponding Average Speed for Different Multiplication Factors

Multiplication Factors	30.0	40.0	50.0
Network Congestion Level	Low	Moderate	High
Average Speed for All Links	38.7	38.1	37.2
Average Speed for Freeways	65.0	64.7	59.3
Average Speed for Arterials	37.0	36.4	35.8

4.4 Simulation Output

Simulation output comprises the data generated as the result of the simulation. In this section, simulation outputs that are relevant to route diversion decisions were decided upon, specifically, the traffic volume, speed, and queue length of certain links. To determine from which links the data should be collected, visual observations were performed through the Graphical User Interface (GUI) provided by DYNASMART-P. Four Measures of Effectiveness (MOEs) can be displayed in the “Link Information” tab in the Plot Dialog Box, namely, link density, link speed, link queue, and link volume.

All of the links on the two alternate routes and the links upstream of the incident link were examined. Links were selected if they demonstrated different performances with and without the incident. Three critical links were selected for each incident location in this research. For the training location, the selected critical links are Link

48→41, Link 52→48, and Link 42→41. Figures 4.4, 4.5, and 4.6 show the link density comparison of the selected links for the training location. Notice that the incident start time was set at the 15th minute for both incidents.

It was found that critical links on the freeway should be selected if they were very sensitive to the incident start time. Once the incident started, these links began to show higher density. However, for the links on the alternate route, it was found that the on-ramps that lead the traffic back to the freeway were usually bottlenecked and, thus, were more critical than other links. In this research, the link queue and link speed for the incident link and link flow for all identified critical links were observed as relevant to the route diversion decision. A total of five minutes of data before the initiation of the VMS were collected for these selected measures and used as the input for training and testing the ANN and SVR models.

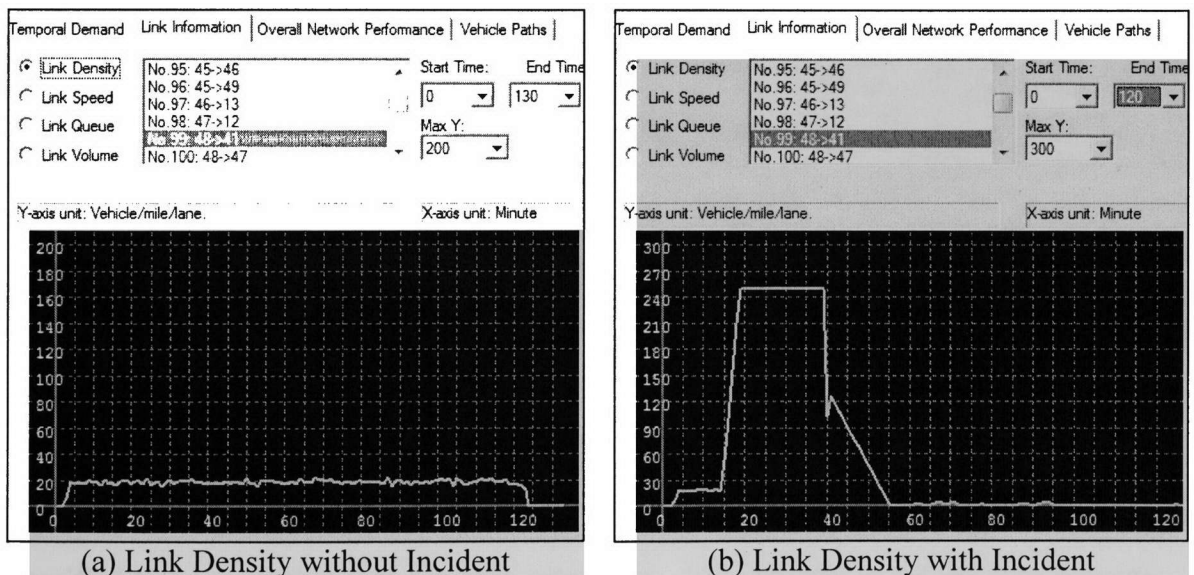


Figure 4.4 Link Density of Link 48→41

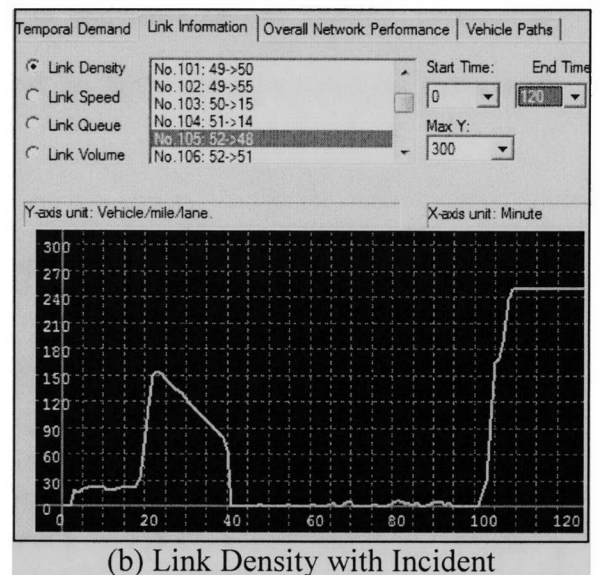
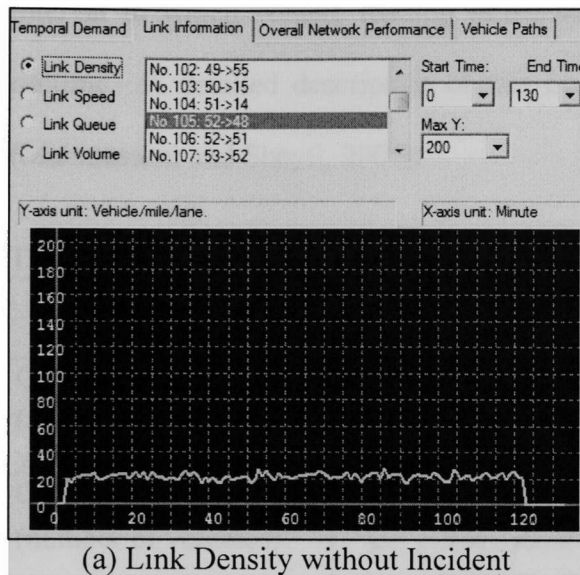


Figure 4.5 Link Density of Link 52→48

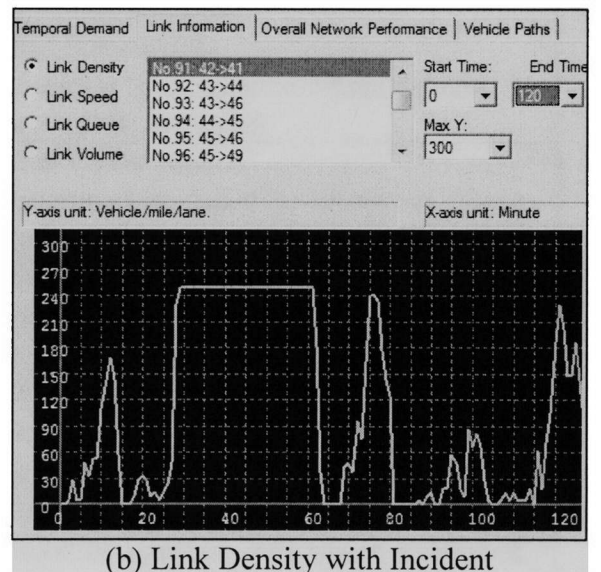
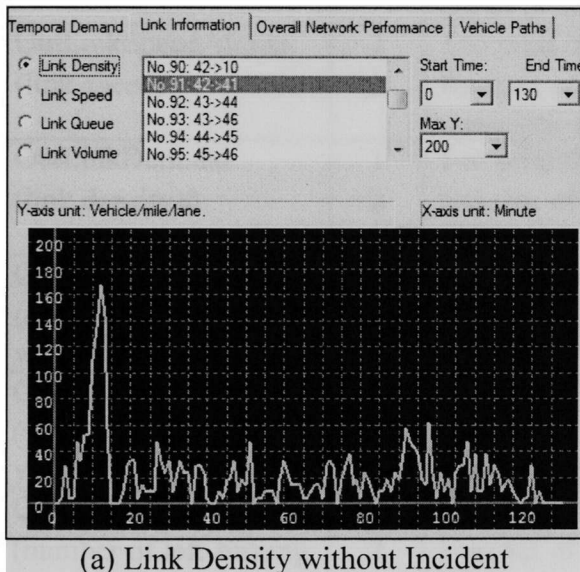


Figure 4.6 Link Density of Link 42→41

In DYNASMART-P, the link performance data will only be generated if they are specified to be generated in the output_option.dat. This file allows the users to indicate whether or not certain output files will be created. The users can also specify the time interval over which the statistics are averaged. In addition, the users can specify the time

interval in which vehicle loading information is displayed on the console window at runtime. A detailed description of this file and its format are provided in Table 4.10 (Mahmassani and Sbayti, 2005).

Table 4.10 Description of OUTPUT_OPTION.DAT Input File

Record Type	Field	Description
OutLinkGen.dat (number of generated vehicles)	1	1 to print the file; 0 otherwise
	2	Number of simulation intervals over which number of vehicles on links will be averaged
OutLinkVeh.dat (number of vehicles)	1	1 to print the file; 0 otherwise
	2	Number of simulation intervals over which number of vehicles on links will be averaged
OutLinkQue.dat (vehicle queue length)	1	1 to print the file; 0 otherwise
	2	Number of simulation intervals over which vehicle queue on links will be averaged
OutLinkSpeedAll.dat (link speed)	1	1 to print the file; 0 otherwise
	2	Number of simulation intervals over which vehicle queue link speed will be averaged
OutLinkDent.dat (link density)0	1	1 to print the file; 0 otherwise
	2	Number of simulation intervals over which link density will be averaged
OutLinkSpeedFree.dat (speed of moving vehicles)	1	1 to print the file; 0 otherwise
	2	Number of simulation intervals over which speed of moving vehicles will be averaged
OutLinkDentFree.dat (density of moving vehicles)	1	1 to print the file; 0 otherwise
	2	Number of simulation intervals over which density of moving vehicles will be averaged
OutLeftFlow.dat (number of left-turning vehicles)	1	1 to print the file; 0 otherwise
	2	Number of simulation intervals over which number of left-turning vehicles will be averaged
OutGreen.dat (green time at intersections)	1	1 to print the file; 0 otherwise
	2	Number of simulation intervals over which green time at intersections will be averaged
OutFlow.dat (number of vehicles crossing intersections)	1	1 to print the file; 0 otherwise
	2	Number of simulation intervals over which number of vehicles crossing intersections will be averaged
GUI updating interval	1	Number of simulation intervals (default 6 seconds) for which the GUI updates vehicles position and loading information on the network

Vehicle queue length (OutLinkQue.dat), the difference between the link speed for moving vehicles and the link speed for all vehicles, and the number of vehicles that have been discharged by the link were selected as the measures for link performance in this research. They were extracted and calculated after each simulation run. The output_option.dat file coded for this study is presented in Table 4.11.

Table 4.11 Sample OUTPUT_OPTION.DAT File

0	10
1	10
1	10
1	10
0	10
1	10
0	10
0	10
0	10
1	10
20	

After the input and output data are collected, they need to be examined to determine the relevance of different variables to delay reductions, which will be used as the MOE in this research. The percent of delay reductions is defined as the total network delay reductions as a percentage of change in incident delay. Therefore, system-wide travel time also needs to be collected after each case is simulated in DYNASMART-P.

The total network percent of delay reductions represents the summation of the individual delay reductions experienced by all vehicles in the network, thus indicating the impact of route diversion on the entire system. It was used in by Bhavsar *et al.* (2007) in

their study to measure the impact of route diversion on the transportation network. The percent of delay reductions (PDR), or percent of incident delay reduced due to route diversion, is calculated using equation 4.1:

$$\begin{aligned}
 PDR &= \frac{\text{Delay Reduction due to Incident with Route Diversion}}{\text{Delay due to Incident without Route Diversion}} \\
 &= \frac{(T_I - T_{NI}) - (T_{RD} - T_{NI})}{T_I - T_{NI}} \\
 &= \frac{T_I - T_{RD}}{T_I - T_{NI}} \tag{4.1}
 \end{aligned}$$

where

PDR = percent of delay reductions,

T_I = travel time under an incident scenario with no route diversion,

T_{NI} = travel time when there is no incident, and

T_{RD} = travel time under an incident when there is route diversion.

This percent of delay reductions (PDR) will be used as the target value for the training and testing of the ANN and SVR models. A positive PDR represents a delay reduction benefit from route diversion, whereas a negative value means it is not beneficial to divert traffic.

DYNASMART-P outputs the summary statistics (both average and total) of travelers' characteristics, including travel time, distance traveled, queues, and stopping durations, through the SummaryStat.dat file. This file provides the following categories of information:

- Network characteristics

- Input parameters
- Vehicle loading and exiting information
- HOT/HOV statistics
- Simulation statistics

Among the summary statistics, the statistic that is of particular interest in this research comprises the “Average Trip Times (including entry queue time) (min).” This statistic needs to be extracted after each incident simulation run to calculate the percent of delay reductions as a result of route diversion. Average travel times when there is no incident on the network and average travel times when there is an incident, but no provision of VMS information, also need to be extracted. These values will be used to calculate the incident delay.

4.5 Incident Scenario Design

In generating the cases for incident scenarios, a variety of parameters of the incident were considered. A total of 1650 scenarios were generated for the training location, including 150 scenarios that were used to generate incidents without the provision of VMS information and another 1500 scenarios with the provision of VMS information. The attributes for critical variables used for incident scenarios designed under route diversion are listed in Table 4.12.

4.6 Data Generation for Training and Testing ANN and SVR Models

As calculated in the previous section, there were a total of 3000 incident scenarios designed. For each case, four input files need to be prepared (incident.dat, vms.dat,

scenario.dat and demand.dat). In addition, the output files, including OutLinkQue.dat, OutLinkSpedFree.dat, OutLinkSpeedAll.dat, and OutFlow.dat files need to be read to extract the critical link performance measures. After each simulation run, the SummaryStat.dat file is read, and the total travel time including the entry queue time is extracted.

Table 4.12 Attributes for Critical Variables for the Training Location

Variables	Values
Capacity lost (Incident Severity)	20%, 40%, 60%, 80%, 100%
Incident Location	Link 48→41
VMS Location	Link 53→52
Incident Detection and Verification	3 minutes
VMS Duration	5 minutes before the incident ends; terminate when the incident ends
Incident Response and Clearance Time	25, 30, 35, 40, 45, 50, 55, 60, 65, 70
Level of Congestion	low (30), moderate (40), high (50)
Drivers Responsive to DMS	15%, 30%, 50%, 75%, 100%

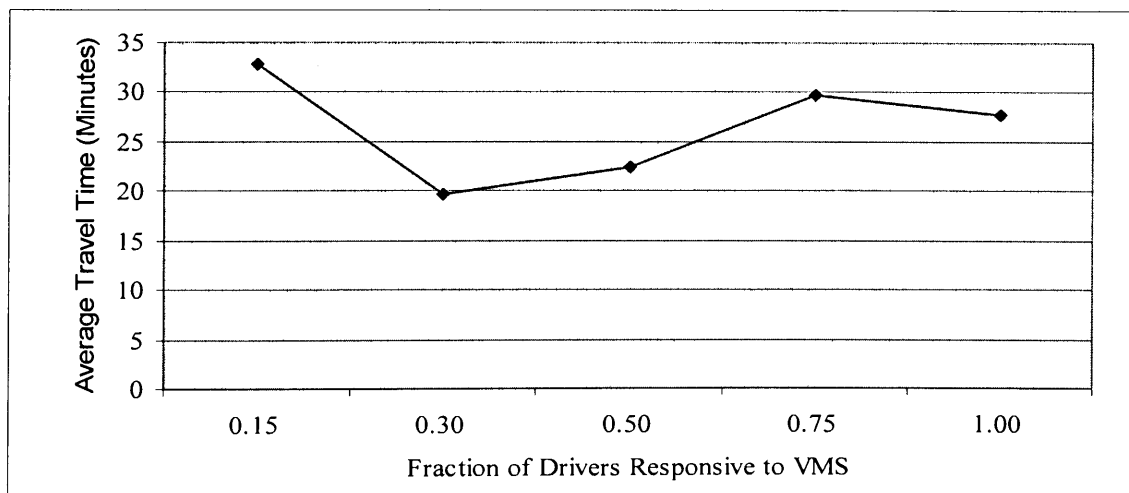
To minimize the labor required for completing the process, a MATLAB program was developed to automate the whole process, inclusive of the automatic reading and writing of input files prepared for each case simulation. The reading and extracting of data resulting from the simulation are also automated. The code for this MATLAB program is attached as Appendix of this dissertation.

After the data were generated from the simulation, relevance between the input variables and the output average travel time was further examined. Figure 4.7 shows the change in average travel times in relation to the change in the fraction of drivers responsive to VMSs, the change in incident severity, and the change in incident duration. Note that in these figures, when the value of one variable changes, the values of the other

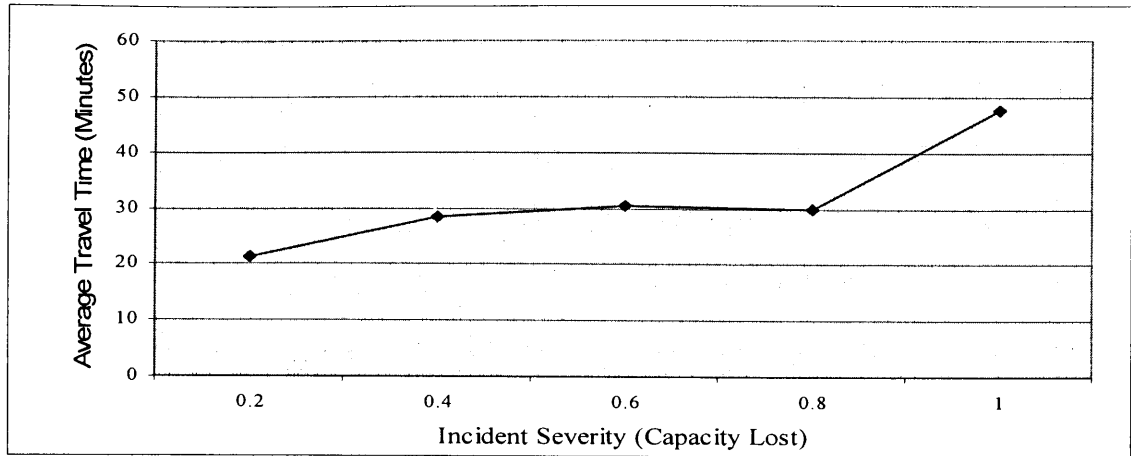
variables are fixed, i.e., the fraction of drivers responsive to VMS is 0.15, incident severity is 0.2, incident duration is 25 minutes, VMS ends 5 minutes before the incident ends, and the demand multiplication factor is 40.

It was found that the average travel time also changed when the VMS duration and level of congestion changed. Due to the limited number of variables, their relevance is presented in Table 4.13.

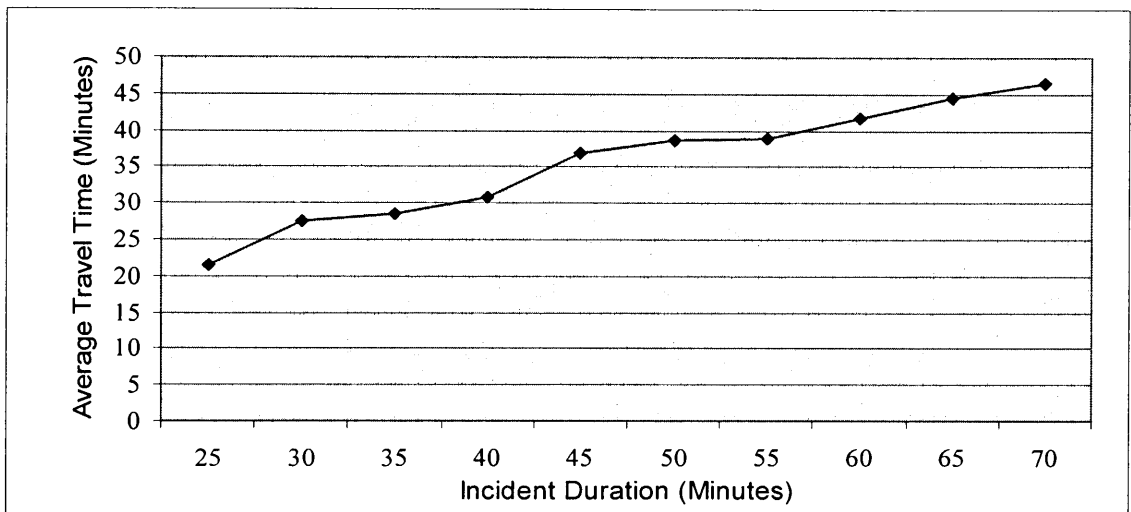
The results from Figure 4.7 indicate that all of the identified input variables are relevant to the output average travel times. However, the level of relevance among these variables varies. For example, average travel time is sensitive to the change in incident duration, while it is not very sensitive to the change in VMS duration. It was also found that the trend of the changes in average travel time does follow the same rule among the variables. It is obvious that average travel time would increase when the level of congestion on the network increased. Nevertheless, the same correlation did not appear between the average travel times and the incident severity level, nor between the incident duration and the fraction of drivers responsive to VMS.



(a) Fraction of Drivers Responsive to VMS



(b) Incident Severity



(c) Incident Duration

Figure 4.7 Impact of Different Input Variables on Average Travel Times

Table 4.13 Impacts of VMS Duration and Level of Congestion on Average Travel Times

Input Variables	Values	Average Travel Time
VMS Duration*	VMS ends 5 minutes earlier than the incident ends	27.90
	VMS ends as the incident ends	28.20
Demand Multiplication Factor	30	6.20
	40	7.00
	50	47.50

4.7 Summary

In this chapter, the relevance of critical variables like incident severity (capacity lost), incident duration, fraction of drivers responsive to VMS, and network congestion level, etc. were further verified. Simulation network came with DYNASMART-P was used as the simulation network. Two incident locations on the freeway from the network were selected as the training and testing locations. Various incident scenarios were designed to represent the change of previous identified critical variables. These incident scenarios were then simulated in DYNASMART-P and useful simulation output were extracted and prepared to be used for ANN and SVR model training and testing.

CHAPTER 5

MODEL DEVELOPMENT

5.1 Introduction

As mentioned in Chapter 1, the goal of this research is to develop a model that can assist traffic operators in making route diversion decisions. The model predicts the expected percent of reduction in incident delay due to route diversion under a given set of conditions. Two data mining models, an Artificial Neural Network (ANN) and Support Vector Regression (SVR), were trained, tested and are described in this chapter.

In this chapter, data were prepared for the training and preliminary testing of ANN and SVR models for the training location. To evaluate the performance of trained ANN and SVR models at both the training and testing locations on a freeway, the data were separated into two sets. The data collected from the training location, which includes 1000 incident scenarios, are used for training these two models. The models were calibrated based on their performance in adapting to training data from the training location.

5.2 Data Preparation

Before the model is trained, the model input data need to be prepared. These data are different from the input data used for simulation as discussed in Chapter 4. A MATLAB program was developed to automate the process of writing all of the input data for training and testing into a file. The average travel times were written into another file, and two additional files were also created. One contained the average travel times for all

incidents without the provision of VMS at different levels of congestion, and the other contained average travel times for the network with no incidents, also at different levels of congestion.

A total of 30 variables were used as inputs for training and testing the ANN and SVR models, while the percent delay reductions (PDR) were used as the target vector. Due to the large amount of input variables involved, only a sample of these data was collected. This is presented in Table 5.1. Note that these data are only collected from the simulation input and output of the training location. For the testing location, the same data were collected via the same procedure. Additionally, for some variables such as queue length, the link speed difference between moving vehicles and all vehicles, and the link flow (taken from different links at different minutes), only the start and end minutes' data for one sample link are listed in Table 5.1.

After these inputs were prepared, they were used as input vectors to train and test the ANN and SVR models. The precision of a predicted PDR was determined by comparing the predicted value and the value extracted from the simulation output.

5.3 ANN Model Training, Calibration, and Testing

In this section, the ANN models are evaluated for their ability to solve route diversion problems. These models were trained and their parameters were calibrated for incident scenarios designed for the training location. The ANN toolbox provided by MATLAB was used to develop the model. A feedforward ANN combined with a backpropagation (BP) algorithm was selected.

Table 5.1 Input Variables for Training and Testing of ANN and SVR Models

Incident Scenario Index	1	2	...	35	...	501	...	1500
Incident Duration	25	25	...	25	...	25	...	70
Fraction of Capacity Lost	0.20	0.20	...	0.20	...	0.20	...	1.00
VMS Responsive Rate	0.15	0.30	...	0.15	...	0.15	...	1.00
VMS Duration	20	20	...	25	...	20	...	70
Level of Congestion	30	30	...	30	...	40	...	50
Queue Length (13 th minute) on Link 48→41	0	0	...	0	...	0	...	0
...
Queue Length (17 th minute) on Link 48→41	71.90	71.90	...	71.90	...	77.20	...	82.00
Link Speed Free- Link Speed All (13 th minute) on Link 48→41	0	0	...	0	...	0	...	0
...
Link Speed Free- Link Speed All (17 th minute) on Link 48→41	18.88	18.88	...	18.88	...	17.8	...	0
Link Flow (13 th minute) on Link 48→41	6.00	6.00	...	6.00	...	7.20	...	10.70
...
Link Flow (17 th minute) on Link 42→41	0.60	0.60	...	0.60	...	0.10	...	0
Average Travel Time with Incident and VMS Provision	15.57	16.02	...	14.04	...	32.87	...	57.10
Average Travel Time With Incident and without VMS Provision	14.27	14.27	...	15.02	...	21.42	...	56.44
Average Travel Time with No Incident	6.20	6.20	...	6.20	...	7.00	...	47.50
Percent of Delay Reductions (PDR)	-16%	-22%	...	11%	...	-79%	...	-7%

BP was created by generalizing the Widrow-Hoff learning rule and applying it to multiple-layer networks and nonlinear differentiable transfer functions (Demuth, 1998). This is one of the most popular and robust tools used in the training of ANNs. BP attempts to associate input vectors with specific output vectors, or to classify input vectors in an appropriate way as defined by the user. Properly trained BP networks tend to give reasonable answers when presented with new inputs. There are generally four steps in the training process: assembling the training data, creating the network object, training the network, and simulating (testing) the network response to new inputs. The first step was completed in section 5.1.

5.3.1 Neural Network Architecture

A multilayer feedforward network is the most commonly used network architecture with the BP algorithm. As shown in Figure 5.1, R is the number of inputs. An appropriate weight W is assigned to each input p . The weighted input was summed up with the bias b and fed into the transform function f . A layer may consist of many neurons. A one-layer network with S neurons is shown in Figure 5.2 (Demuth, 1998). In such a network, each element in the input vector is connected with each neuron in the network through a weight matrix W .

An ANN can also be structured with more than one layer. Such a network is called a multilayer network. In such a multilayer network, the outputs from an intermediate layer are the inputs to the following layer. Therefore, the last layer in the network produces the final outputs and is, thus, called the output layer (i.e., the third layer

in Figure 5.3) (Demuth, 1998). The layers that do not produce the final outputs are called the hidden layers (i.e., the first and second layers in Figure 5.3).

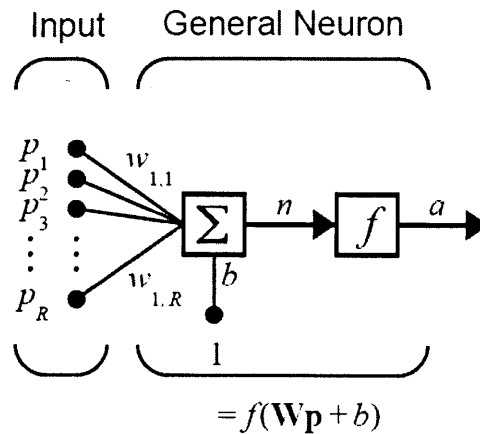


Figure 5.1 A Simple Neural Network with One Neuron and One Layer

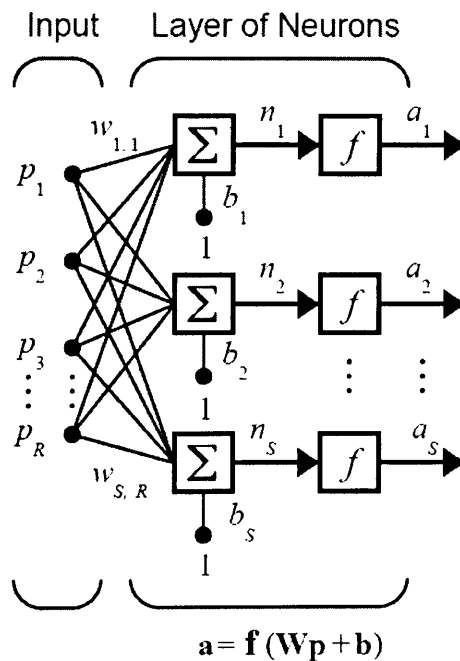


Figure 5.2 A One-layer Network with Multiple Neurons

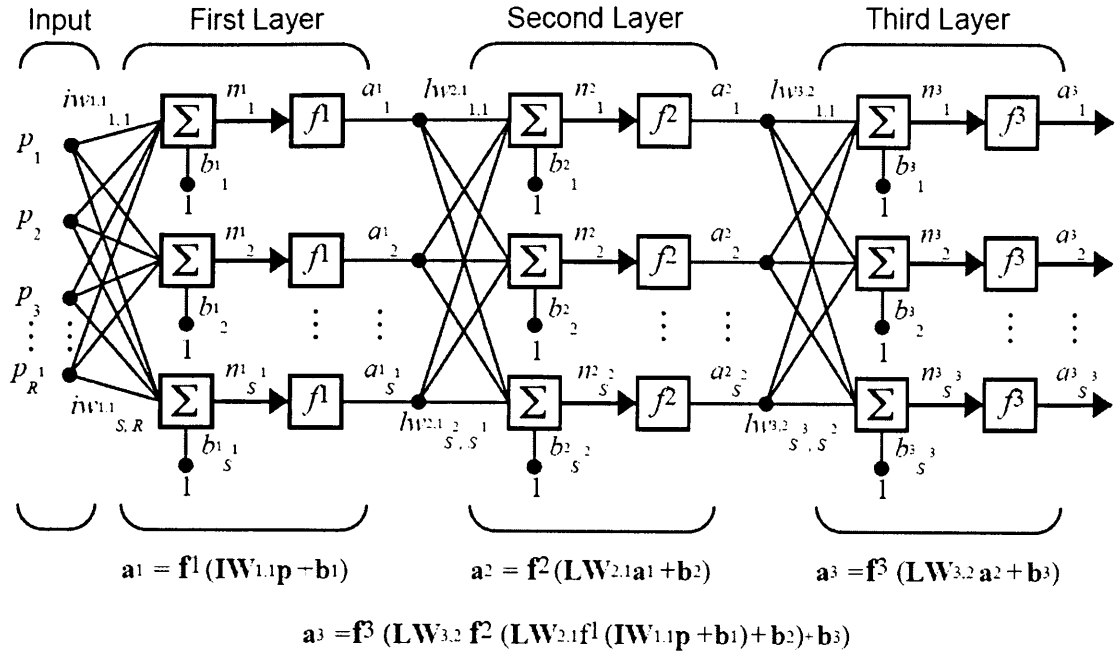


Figure 5.3 A Multilayer Neural Network Architecture

5.3.2 Transfer Functions

The multilayer ANN often uses the log-sigmoid transfer function, or logsig, also known as the logistic function. This function generates outputs between 0 and 1 when the neuron's input has a range of $(-\infty, +\infty)$. It is given by the following function:

$$\Pi(t) = \frac{1}{1 + e^{-\beta t}}$$

where β is a slope parameter. This is called the log-sigmoid because a sigmoid can also be constructed using the hyperbolic tangent function instead of this relation. In this case, it is called a tan-sigmoid (tansig). Tan-sigmoid is named after the hyperbolic tangent, which has the same shape. The function takes one input vector and returns each element of this vector as a squashed number between $(-1, 1)$. Another transfer function that is occasionally used is the linear transfer function, purelin. Usually, if logsig or tansig is used as the transfer function, the output values are limited to a small range, whereas if the

purelin function is used, the output values can have a large range. Figure 5.4 shows the graphic forms of these transfer functions.

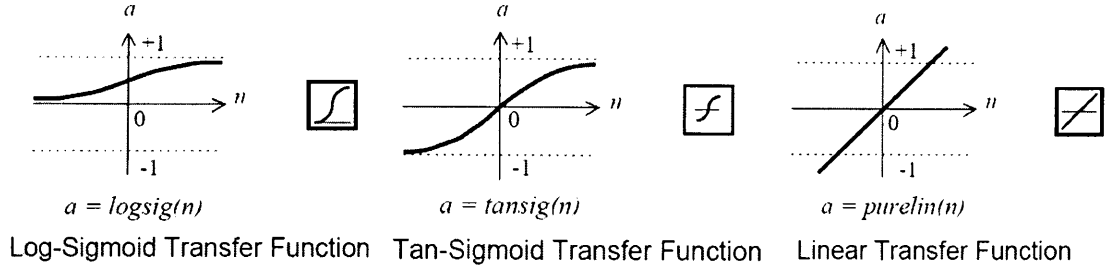


Figure 5.4 Graphic Forms of Transfer Functions

Feedforward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn the linear and nonlinear relationships between the input and output vectors. The linear output layer lets the network produce values outside the range from -1 to $+1$. In this research, the PDR is the target vector, and its value is not limited to a range of $(-1, 1)$. Therefore, a purelin function was selected as the transfer function for the output layer. Both logsig and tansig were tested for their performance as transfer functions for the hidden layer. The tansig function was found to be better than the logsig function. The results are shown later in this chapter.

5.3.3 Backpropagation Algorithm

There are many variations of the BP algorithm. The most common ones are the learning rate and momentum parameters. The simplest BP learning updates the network weights and biases in the direction in which the performance function decreases most rapidly. One iteration of this algorithm can be written as:

$$x_{k+1} = x_k - \alpha_k g_k \quad (5.1)$$

where x_k is a vector of current weights and biases, g_k is the current gradient, and α_k is the current learning rate. There are two different modes in which this gradient descent algorithm can be implemented: the incremental mode and the batch mode. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode, all of the inputs are applied to the network before the weights are updated.

In the batch mode, the weights and biases of the network are updated only after the entire training set has been applied to the network. The gradients calculated at each training example are added together to determine the change in the weights and biases. Under the batch mode, there are several optional high-performance algorithms that can make the training process much faster than the regular algorithms. Among these high-performance algorithms, the heuristic technique variable learning rate and a numerical optimization technique Levenberg-Marquardt were explored.

5.3.4 Variable Learning Rate

The learning rate and momentum are very important parameters in ANN training. Momentum is a technique often used to make it less likely for a BP network to get caught in shallow minima. The learning rate is the rate at which the ANN learns (Anderson and McNeil, 1992).

The performance of an algorithm is largely sensitive to the proper setting of the learning rate and momentum. If the learning rate is set too high, the algorithm may oscillate and become unstable. On the other hand, if the learning rate is set too low, the algorithm may take too long to converge. It is not practical to determine the optimal

setting for the learning rate before training. In fact, the optimal learning rate changes during the training process as the algorithm moves across the performance surface.

The performance of the steepest descent algorithm can be improved if the learning rate is allowed to change during the training process. An adaptive learning rate will attempt to keep the learning step size as large as possible while keeping the learning stable. The learning rate is made responsive to the complexity of the local error surface. When a variable learning rate is used, the initial network output and error are calculated first. At each epoch, new weights and biases are calculated using the current learning rate. New outputs and errors are then calculated. If the new error exceeds the old error by more than a predefined ratio, typically 1.04, the new weights and biases are discarded. In addition, the learning rate is decreased. Otherwise, the new weights, etc., are kept. If the new error is smaller than the old error, the learning rate is increased (typically by multiplying by 1.05). This procedure increases the learning rate, but only to the extent that the network can learn without large error increases. Thus, a near-optimal learning rate is obtained for the local terrain. When a larger learning rate may result in stable learning, the learning rate is increased. When the learning rate is too high to guarantee a decrease in error, it is decreased until stable learning resumes (Demuth, 1998).

5.3.5 Levenberg-Marquardt

Newton's method is another method used for fast optimization. The basic step of Newton's method is:

$$x_{k+1} = x_k - A_k^{-1} g_k \quad (5.1)$$

where A_k is the Hessian matrix (second derivatives) of the performance index at the current weight and bias values and g_k is the current gradient. Newton's method usually converges faster than conjugate gradient methods. Unfortunately, it is complex and expensive to calculate the Hessian matrix for feedforward ANNs. There is a class of algorithms that are based on Newton's method but that do not require the calculation of second derivatives. The Levenberg-Marquardt algorithm is one of them. It uses a Jacobian matrix to approximate the Hessian matrix. The Jacobian matrix contains the first derivatives of the network errors with respect to the weight and biases, and, thus, can be calculated through a standard BP technique. The Levenberg-Marquardt algorithm is considered the fastest training algorithm for networks of moderate size (Demuth, 1998).

Two training functions, a gradient descent with momentum and adaptive learning rate (`traingdx`) and a Levenberg-Marquardt BP were applied in the development of the ANN model. The difference between these two functions is that the `traingdx` function combines an adaptive learning rate with momentum training. It is invoked in the same way as `traingda`, except that it has the momentum coefficient as an additional training parameter. The `traingdx` function was found to outperform the `traingda` function slightly during training attempts. The performance of these two functions is compared later on in this chapter.

5.3.6 Principle Component Analysis

A principle component analysis (PCA) was used in this research in an effort to improve the training efficiency of the BP ANN. A PCA defines the variance-covariance structure of a set of variables through a few linear combinations of these variables. Its

general objectives are data reduction and data interpretation (Johnson and Wichern, 2006). One parameter should be specified in applying the PCA, namely, the minimum fraction variance component to be kept (`min_frac`). This means that PCA eliminates those principal components that contribute less than the minimum fraction to the total variation in the data set.

Before PCA is applied, the input data for training and testing need to be normalized and preprocessed. The input data are adjusted so that their mean equals zero and their standard deviation equals one. After the training process, the predicted values are de-normalized before they are compared to their corresponding target values.

In this research, by applying PCA, the training speed was greatly improved. It reduced by approximately one-half the time used in the training process without applying PCA. In addition, the performance of the trained ANN model exceeds that of the ANN model without PCA. A comparison between the performance with and without PCA is presented in subsection 5.3.7.

5.3.7 Number of Hidden Layers and Number of Hidden Neurons

ANNs with two hidden layers can represent functions with any kind of shape. There is currently no theoretical reason to use an ANN with any more than two hidden layers, and problems that require two hidden layers are rarely encountered (Heaton, 2008). Also note that an ANN without hidden layers is capable of representing linear separable functions or decisions. As mentioned, two hidden layers are rarely needed. For many practical problems, there is no reason to use any more than one hidden layer. Hence, in this research, one hidden layer is used.

After deciding on the number of hidden neuron layers, the number of neurons in each of the hidden layers also needs to be decided upon. This process is covered in the next subsection.

Determining the number of neurons in hidden layers is a very important part of the design of the overall ANN architecture. Using too few neurons in the hidden layers will result in under-fitting, or the inability to adequately detect the signals in a complicated data set. However, using too many neurons in the hidden layers can result in several problems, too. First, too many neurons in the hidden layers may result in over-fitting, which occurs when the ANN has so much information processing capacity that the limited amount of information contained in the training set is not enough to train all of the neurons in the hidden layers. A second problem can occur even when there are sufficient training data. An inordinately large number of neurons in the hidden layers can increase the time it takes to train the network. The amount of training time can increase enough so that it is impossible to adequately train the ANN. Therefore, the number of neurons in the hidden layer should be determined with caution (Heaton, 2008).

There are many rule-of-thumb methods for determining the correct number of neurons to use in the hidden layers. Some of them are summarized as follows (Heaton, 2008):

- The number of hidden neurons should be within the range between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be $\frac{2}{3}$ of the input layer size, plus the size of the output layer.
- The number of hidden neurons should be less than twice the input layer size.

These three rules are only the starting points in selecting the appropriate number of neurons in the hidden layers of an ANN. There are two trial-and-error approaches for determining the number of neurons after the starting point. These are the "forward" and "backward" selection methods. The forward selection method begins by selecting a small number of neurons, usually two. The ANN model is then trained and tested. The number of neurons is then increased, and the process is repeated so long as the overall results of the training and testing improve. The second method, the backward selection method, begins by using a large number of hidden neurons. The neural network is then trained and tested. This process continues until the performance improvement of the ANN is no longer significant (Heaton, 2008).

According to the rule-of-thumb methods described above, the range for the number of neurons is between two and two times the number of input vectors. Therefore, two times the number of input vectors was chosen as the starting point, and the backward selection method was used to further identify the best number of hidden neurons in this research.

5.3.8 Calibration and Selection of ANN Model

In this subsection, the calibration process and preliminary testing results of the ANN model are described. The ANN model with the best performance is selected as the model used in the testing stage.

The mean square error (MSE) was selected as the performance function for the training of the ANN models. MSE is defined as the sum of the square of the difference

between the predicted value and the target value of a model estimate divided by the total number of estimates, as follows:

$$\text{MSE} = \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n} \quad (5.3)$$

where \hat{y} is the predicted value from the model and y is the target value. In this research, the target MSE for training ANN models was set at 10^{-2} . The entire trained ANN model reached an MSE of around 0.05, which is considered acceptable accuracy for the purposes of this research.

Because the MSEs for all of the models are about the same, when comparing the performance of different ANN models, a linear regression between the predictions from the ANN models and the target values was conducted. The slope, y-intercept, and correlation coefficient (R-value) values were used as the MOEs. In a perfect fit (outputs exactly equal to targets), the slope would be 1, and the y-intercept would be 0. The third variable, returned R-value, is a measure of how well the variation in the output is explained by the targets. If this number is equal to 1 (or -1), then there is a perfect correlation between the targets and the model outputs. In addition to these MOEs, the percent of match (POM), defined as the percent of predicted values and target values that have identical signs, was also used to evaluate the ANN. POM measures the ability of the ANN model to produce a correct decision on whether or not traffic should be diverted. Tables 5.2 through 5.6 list the algorithms and variables used in training and calibrating the ANN model, as well as their performance in terms of slope, y-intercept, R-value, and percent of match. These variables include the learning rate (0.05), the learning rate increase/decrease (1.05), and the output layer transfer function (purelin). Also note that

the minimum fraction variance component to be kept in PCA (prepca) is 0.02. Five runs were applied to each set of parameters to eliminate the randomness of the results. The average value of the MOEs from the five runs was used as the evaluation standard.

Table 5.2 presents the difference in model performance between using the PCA and not using it. The result shows that the ANN constructed after training with PCA produces a much better prediction than when the ANN models were trained without PCA. All of the measures indicate a better fit between the target values and the outputs from the ANN. The average slope with PCA is 0.24 higher than that without PCA. The average y-intercept with PCA is 0.05 lower than that without PCA. The average R-value with PCA is 0.15 higher than that without PCA. The average POM with PCA is 0.05 higher than that without PCA. Therefore, PCA is used as one of the default settings in the next round of testing between the transfer functions.

Table 5.2 ANN Model Performance between Training with PCA and without PCA

Index	Slope		Y-intercept	
	With PCA	Without PCA	With PCA	Without PCA
1	0.87	0.48	0.02	0.12
2	0.88	0.99	0.01	-0.01
3	0.86	0.43	0.01	0.06
4	0.85	0.82	0.01	0.06
5	0.90	0.49	0.01	0.06
Average	0.87	0.64	0.01	0.06
Index	R-value		Percent of Match (POM)	
	With PCA	Without PCA	With PCA	Without PCA
1	0.93	0.68	0.76	0.79
2	0.90	0.98	0.79	0.48
3	0.92	0.69	0.79	0.82
4	0.91	0.74	0.8	0.82
5	0.93	0.75	0.82	0.8
Average	0.92	0.77	0.79	0.74

Note: these values were calculated when the transfer function is tansig.

The performance results of the ANN models using different transfer functions are presented in Table 5.3. It can be seen from the results that the logsig transfer function outperformed the tansig transfer function by a slight advantage. The average slope for the logsig function is 0.02 higher than that for the tansig function. The average y-intercept is the same for both functions. The average R-value for logsig is 0.01 higher than that for tansig. The average POM for logsig is 0.03 higher than that for tansig. However, with such a slight difference in performance, it is difficult to decide which transfer function is better. Therefore, both of the transfer functions (logsig and tansig) are passed on to the next round of the training algorithms.

Table 5.3 ANN Model Performance Using Different Transfer Functions

Index	Slope		Y-intercept	
	Logsig	tansig	logsig	Tansig
1	0.93	0.87	0.00	0.02
2	0.87	0.88	0.00	0.01
3	0.90	0.86	0.01	0.01
4	0.90	0.85	0.02	0.01
5	0.87	0.90	0.03	0.01
Average	0.89	0.87	0.01	0.01
Index	R-value		Percent of Match	
	Logsig	tansig	logsig	tansig
1	0.94	0.93	0.80	0.76
2	0.93	0.90	0.84	0.79
3	0.92	0.92	0.82	0.79
4	0.93	0.91	0.82	0.8
5	0.93	0.93	0.82	0.82
Average	0.93	0.92	0.82	0.79

Tables 5.4 and 5.5 compare the performance comparison using the traingdx and trainlm training algorithms. Table 5.4 shows the results when the transfer function for the hidden layers is logsig. The table shows that the adaptive learning rate (traingdx)

algorithm slightly outperformed the Levenberg-Marquardt algorithm. The average of the slope and y-intercept for both training functions are of the same value. The average R-value for traingdx is 0.02 higher than that for the trainlm function. The average POM of traingdx is 0.01 higher than that of trainlm.

Table 5.4 ANN Model Performance Using Different Training Functions (Transfer Function is logsig)

Index	Slope		Y-intercept	
	traingdx	trainlm	traingdx	trainlm
1	0.93	0.84	0.00	0.02
2	0.87	0.91	0.00	0.00
3	0.90	0.89	0.01	0.01
4	0.90	0.95	0.02	0.00
5	0.87	0.86	0.03	0.03
Average	0.89	0.89	0.01	0.01
Index	R-value		Percent of Match	
	Traingdx	trainlm	traingdx	trainlm
1	0.94	0.82	0.80	0.81
2	0.93	0.92	0.84	0.79
3	0.92	0.93	0.82	0.83
4	0.93	0.95	0.82	0.77
5	0.93	0.91	0.82	0.84
Average	0.93	0.91	0.82	0.81

Under the tansig transfer function, as shown in Table 5.5, the model performance using the Levenberg-Marquardt algorithm and the adaptive learning rate (traingdx) are about the same. The average slope and percent of match value using the trainlm function is slightly better than those for traingdx, while the average y-intercept and slope for trainlm is slightly worse than that of traingdx. The performance when logsig was the transfer function and traingdx was the training function is slightly better than the other functions; therefore, they are selected as the functions to be used.

Table 5.5 ANN Model Performance Using Different Training Functions (Transfer Function is tansig)

Index	Slope		Y-intercept	
	traingdx	trainlm	traingdx	trainlm
1	0.87	0.81	0.02	0.03
2	0.88	0.85	0.01	0.03
3	0.86	0.89	0.01	0.01
4	0.85	0.97	0.01	0.01
5	0.90	0.86	0.01	0.02
Average	0.87	0.88	0.01	0.02
Index	R-value		Percent of Match	
	Traingdx	trainlm	traingdx	trainlm
1	0.93	0.93	0.76	0.86
2	0.90	0.91	0.79	0.84
3	0.92	0.92	0.79	0.81
4	0.91	0.94	0.8	0.83
5	0.93	0.85	0.82	0.82
Average	0.92	0.91	0.79	0.83

After the most suitable algorithm and functions were determined through the calibration process, the number of neurons in the hidden layer was calibrated, and the most suitable value was determined.

As mentioned in the preceding subsection, the starting point for the number of neurons in the hidden layer is two times the number of input vectors. After the PCA process during the construction of the ANN, the number of input vectors was reduced to seven. Therefore, the starting point of the number of hidden neurons was set at 14 during the entire calibration process. The next step was to preliminarily determine the most suitable number of neurons in the hidden layer. A range of [2, 14] for the number of hidden neurons was first applied to the ANN models. A logsig transfer function and a training algorithm of variable learning rates (traingdx) were incorporated. Table 5.6 presents the results of this process. After this process, the range was further reduced, and

five runs were performed for each network with a selected number of neurons. The average of the MOEs was used to determine the most appropriate number of neurons.

Table 5.6 Preliminary Determination of Number of Neurons in the Hidden Layer

Number of Neurons	2	3	4	5	6	7	8	9	10	11	12	13	14
Slope	0.45	0.48	0.76	0.84	0.63	0.77	0.79	0.88	0.83	0.85	0.87	0.85	0.88
Y-intercept	0.04	0.05	0.03	0.02	0.04	0.03	0.02	0.01	0.01	0.01	0.01	0.01	0.00
R-value	0.65	0.66	0.85	0.85	0.78	0.87	0.89	0.92	0.90	0.91	0.93	0.93	0.93
Percent of Match	0.75	0.80	0.82	0.80	0.82	0.82	0.89	0.87	0.88	0.87	0.91	0.85	0.88

From the results in Table 5.6, based on the R-value, the best number of neurons in the hidden layer lies between [9, 14]. To further determine the number of neurons to be used, five runs were applied to each ANN with different numbers of neurons. The average of the MOEs was used as the evaluation standard. Table 5.7 presents the results of the testing process.

Based on the training results shown in Table 5.7, it is not evident what number of neurons in the hidden layers should be selected for this research. When the number of neurons in the hidden layer is set at 14, the R-value is slightly higher than in the other cases. When the number of neurons is 12, the slope is 0.88, which is higher than in the other cases. When there are nine neurons, the y-intercept is 0, which is the lowest. Also, when there are nine neurons, the POM value is the highest. Therefore, the number of neurons in the hidden layer should be selected from among 9, 12, and 14.

Table 5.7 Further Identification of Number of Neurons in the Hidden Layer

Number of Neurons	9	10	11	12	13	14
Slope	0.86	0.86	0.93	0.86	0.87	0.91
	0.91	0.78	0.81	0.92	0.89	0.88
	0.88	0.83	0.89	0.96	0.82	0.87
	0.81	0.86	0.86	0.80	0.87	0.88
	0.82	0.87	0.87	0.84	0.84	0.83
Average	0.86	0.84	0.87	0.88	0.86	0.87
Y-intercept	0.02	0.01	0.01	0.01	0.02	0.00
	-0.01	0.03	0.01	0.00	0.01	0.01
	0.01	0.01	0.01	0.00	0.02	0.02
	-0.01	0.02	0.00	0.02	0.02	0.00
	0.01	0.01	0.03	0.01	0.02	0.02
Average	0.00	0.02	0.01	0.01	0.02	0.01
R-value	0.91	0.92	0.94	0.92	0.92	0.94
	0.93	0.90	0.91	0.91	0.92	0.92
	0.92	0.92	0.93	0.93	0.88	0.93
	0.91	0.90	0.91	0.92	0.94	0.92
	0.91	0.92	0.91	0.91	0.93	0.92
Average	0.92	0.91	0.92	0.92	0.92	0.93
Percent of Match	0.83	0.75	0.82	0.82	0.81	0.84
	0.81	0.76	0.80	0.82	0.81	0.81
	0.82	0.81	0.80	0.82	0.84	0.81
	0.86	0.80	0.78	0.81	0.84	0.81
	0.84	0.83	0.78	0.82	0.83	0.81
Average	0.83	0.79	0.80	0.82	0.83	0.82

One problem that may occur during ANN training is over-fitting. Over-fitting occurs when the error in the training set is driven to a very small value. As new data are fed to the network, the error becomes larger. This is an indication that the network has memorized the training examples, but that it has not learned to generalize new situations. One method for improving network generalization is to use a network that is just large enough to provide an adequate fit. The larger the network used, the more complex the

functions that the network can create. If a small enough network is used, it will not have enough power to over-fit the data. However, it is difficult to decide what the smallest number of neurons that can represent a network is. In this research, the smallest number of neurons passed to the preliminary test is nine. Therefore, considering all of the performance measures and the over-fitting problem that may exist, nine neurons were selected. When the number of neurons is determined, the calibration of the ANN is completed. The determined set of parameters is:

- Use of PCA,
- Transfer Function = logsig,
- Training Algorithm = traingdx, and
- Number of neurons = 9.

Figure 5.5 shows a regression plot of the predicted values and the target values. In the figure, the Y-axis plots the predicted values and the X-axis plots the target values. If the predicted values are a perfect fit of the target values, the data would comprise a 45-degree line with a zero y-intercept. This line is drawn as a dashed line in the plot as a reference to the line that fits the real data. As indicated in the plot, the slope is 0.88, the y-intercept is 0.01, and the R-value is 0.92. It can be seen from the figure that an ANN model can satisfactorily simulate the relationships between the input variables and the output targets. Because the size of the testing cases is large (500 scenarios), Figure 5.6 gives a sample (50 scenarios) of the predicted values vs. the target values. It can be seen from the figure that the result from the predicted values shows a larger variance compared to the targets. However, it is also observed that the risk of producing an output with a sign opposite that of the target values arises when the target values are close to

zero. In addition, the trend of the predicted values shows the same trend as the target values. Therefore, it can be concluded that an ANN model can be used to predict the PDR resulting from route diversion when calibrated and applied to the training location.

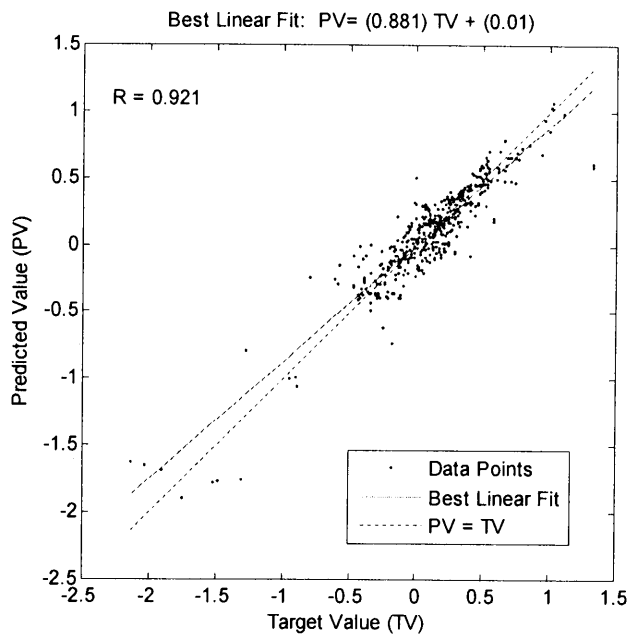


Figure 5.5 Regression Plot of Predicted and Target PDR

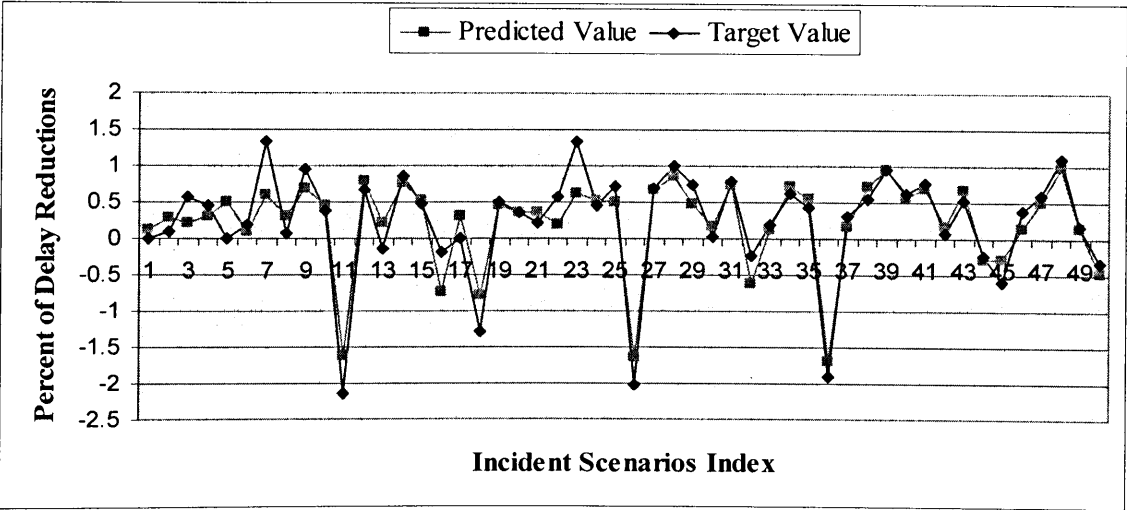
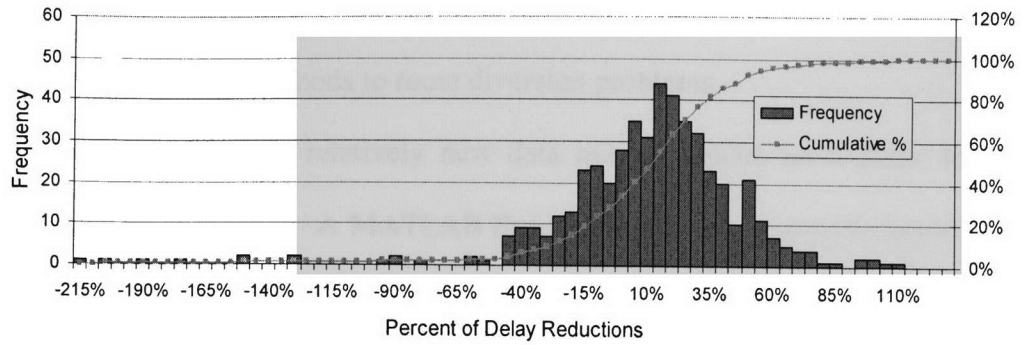


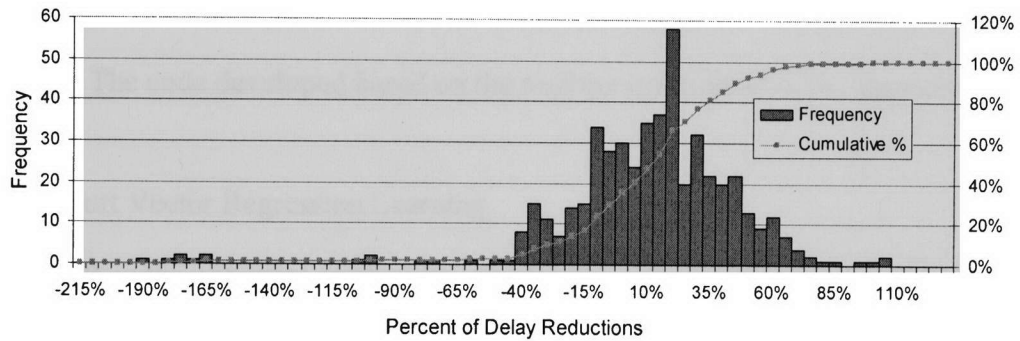
Figure 5.6 Comparison of Predicted and Target Values of ANN Model

In addition to the regression analysis, a frequency distribution analysis was also performed on the target values and the ANN model predicted values. Histograms were used to show the range and frequency of the PDR values as a result of route diversion. Histograms represent the frequency distribution by means of rectangles whose widths represent class intervals and whose areas are proportional to the corresponding frequencies (Merriam-Webster Online Dictionary, accessed March 13, 2008). They permit analysis of extremely large datasets by reducing them to a single graph that can show primary, secondary, and tertiary peaks in data and also provide a visual representation of the statistical significance of those peaks.

The histograms in Figure 5.7 show the number of times each PDR value falls into certain ranges. The X-axis represents the PDR values, with each unit representing a range of 5%. The Y-axis is the frequency or the number of times each delay reduction occurs. The cumulative distributions curve in the figure shows the frequency of delay reductions falling below a certain value. From Figure 5.7 (a), note that, among the 500 incident scenarios randomly selected for ANN model testing, over 90% of the PDR values as a result of diversion falls between -40% and 80%. The highest frequencies are above 40, which occurs when the PDR falls between 10% and 20%. Also, 60% of the PDR values are above 5%. This means that route diversion as an incident management strategy is effective in over half of the occasions considered.



(a) Distribution Frequency for Target PDR Values



(b) Distribution Frequency for ANN Model Predicted PDR Values

Figure 5.7 Histograms for Target PDR and ANN Model Predicted PDR Values

Comparing Figures 5.7 (a) and (b), it was found that the PDR values predicted by the ANN model almost fall into the same range as the target PDR. The most frequently occurring PDRs derived from the ANN model are between 10% and 20%, which coincides with the most frequently occurring PDRs from the simulation outputs. However, the ANN model overestimated the frequency of PDRs in the range of (15%, 20%), whereas it underestimated the frequency of PDRs in the range of (20%, 25%).

5.4 SVR Model Training and Model Testing

In this section, SVR models were evaluated for their ability to solve route diversion problems. These models were trained and their parameters were calibrated for

incident scenarios designed for the training location. As stated, the purpose of this study is to apply data mining methods to route diversion problems.

Because SVR is a relatively new data mining model, no toolbox for SVR is provided in MATLAB 7.0. A MATLAB Support Vector Machines (SVM) toolbox was developed by Gunn in 1998 and published on the Internet for research purposes. The toolbox provides routines for support vector classification and support vector regression. In this research, this toolbox was customized and applied to solve route diversion problems. The code developed based on the toolbox is attached in the Appendix.

5.4.1 Support Vector Regression Learning

SVR is the extended application of SVM in solving regression problems. It was proposed by Vapnik *et al.* in 1996 and has been gaining popularity due to its many attractive features and promising empirical performance.

SVRs and ANNs are both mathematical models. SVR models are actually a close cousin to classical multilayer perceptron ANNs. SVRs embody the Structural Risk Minimization (SRM) principle, as compared to the traditional Empirical Risk Minimization (ERM) principle employed by conventional ANNs. It also minimizes an upper bound on the expected risk. The ERM principle, on the other hand, minimizes the error of the training data (Gunn, 1998). Therefore, SVRs are less likely to encounter over-fitting problems than ANNs.

An SVM finds a nonlinear decision function in the input space by mapping the data into a higher dimensional feature space and separating it there by means of a maximum margin hyperplane. The hyperplane maximizes the minimum distance from

the hyperplane to the closest training point. The maximum margin hyperplane can be represented as a linear combination of training points. Consequently, the decision function for classifying points with respect to the hyperplane only involves dot products between points. Furthermore, the algorithm that finds a separating hyperplane in feature space can be stated entirely in terms of vectors in the input space and dot products in the feature space. Thus, an SVM can locate a separating hyperplane in feature space and classify points in that space without ever representing the space explicitly. The SVM does this simply by defining a function, called a kernel function, that plays the role of the dot product in feature space (Gunn, 1998). This technique avoids the computational burden of explicitly representing the feature vectors.

The selection of an appropriate kernel function is important because the kernel function defines the feature space in which the training set examples will be classified. There are a few kernel functions that have been found to work well in a wide variety of applications. Usually, the default and recommended kernel function is the Radial Basis Function (RBF) (Gunn, 1998). Two types of RBF kernel functions are used in this research: Gaussian RBF and Exponential RBF. The parameters of each of the RBFs were calibrated based on conclusions from previous studies, in which appropriate parameters were selected for RBF kernel functions.

5.4.2 RBF Kernel Functions

As mentioned, there are two types of commonly used RBF kernel functions: Gaussian and Exponential. The Gaussian RBF is the most commonly used RBF. It takes the form:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (5.4)$$

for all $x, x' \in R^d$, where the $\|x - x'\|^2$ denotes the norm on R^d . Classical techniques utilizing RBFs employ some method of determining a subset of centers. Typically, a method of clustering is first employed to select a subset of centers. An attractive feature of the SVM is that this selection is implicit, with each support vector contributing one local Gaussian function centered at that data point (Gunn, 1998). The Exponential RBF takes the form:

$$K(x, x') = \exp\left(-\frac{\|x, x'\|}{2\sigma^2}\right) \quad (5.5)$$

where $\|x, x'\|$ denotes the inner product of R^d . It produces a piecewise linear solution that can be attractive when discontinuities are acceptable. In this research, both functions were trained, and a comparison of the results shows that the Exponential RBF better suits the input data.

5.4.3 Normalization

Data normalization is required for particular kernels due to their restricted domain, and may also be advantageous for unrestricted kernels. Normalization places the values of numeric attributes on the same scale and prevents attributes with a large original scale

from biasing the solution. Determining if the normalization of the data is necessary requires consideration of the input features. Generally, SVM requires the normalization of the numeric input.

Before it was decided whether or not inputs should be normalized, a test was performed on the SVR model with a Gaussian RBF kernel without normalizing the inputs. A linear regression was performed between the predicted values produced by this model and the target values. The plot of the regression is shown in Figure 5.8. After that, another test was performed with normalized inputs using the same kernel function and the same values for the parameters. The plot of the regression is shown in Figure 5.9. As shown in these two figures, the R value for the first test is 0.29, and that for the second test is 0.88. This indicates that the second simulation produced better outputs. Therefore, based on these results, the inputs for the SVR were all preprocessed with normalization.

5.4.4 Parameters for RBF Kernel Functions

The performance of the SVR models depends on setting meta-parameters C , ε , and the kernel parameters p appropriately. Therefore, after a kernel function is selected, a set of parameters for the kernel function needs to be identified.

The selection of a particular kernel type and kernel function parameter p is usually based on specific application-domain knowledge. It should also reflect the distribution of input (x) values of the training data. Gaussian RBF and Exponential RBF were chosen in this research as the kernel functions due to their general acceptability.

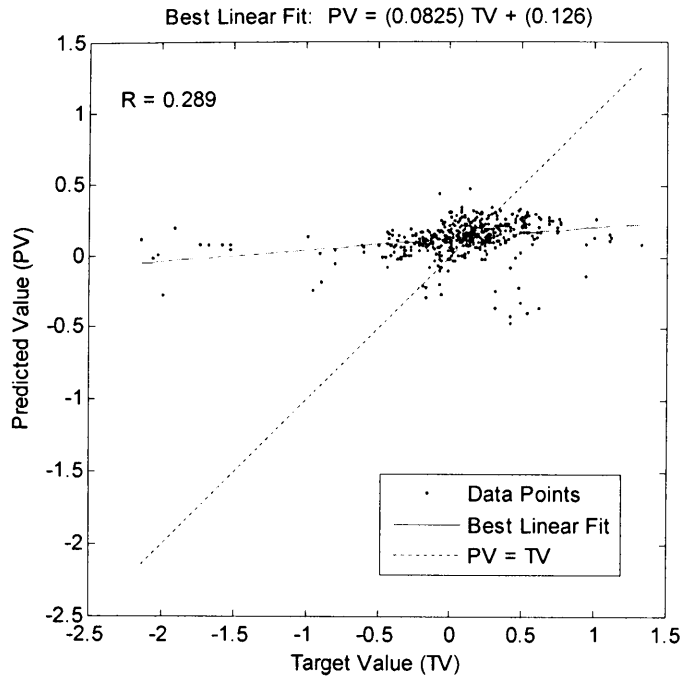


Figure 5.8 Regression of Predicted and Target Values without Normalized Inputs

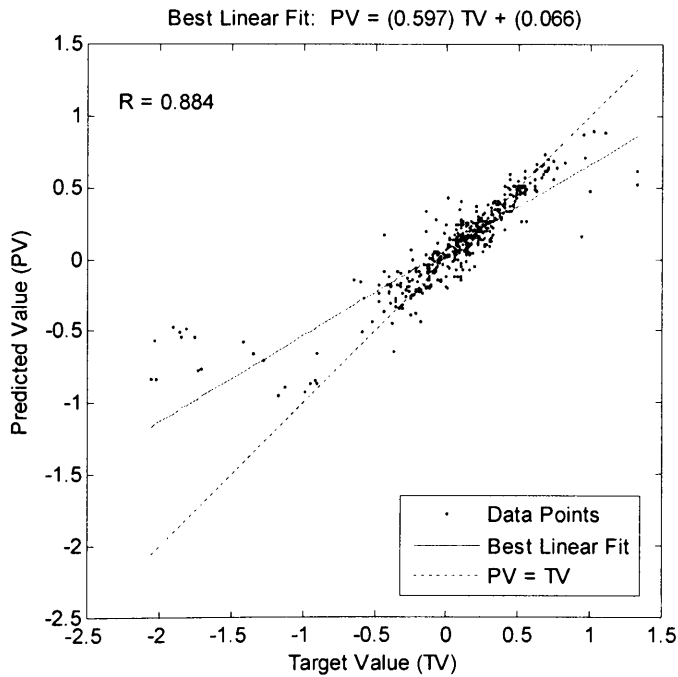


Figure 5.9 Regression Relationships of Predicted and Target Values with Normalized Inputs

The parameter p of the RBF is called the width parameter. It determines the width of the RBF and should reflect the distribution/range of the x -values of the training data. Parameter C determines the trade off between model complexity (flatness) and the degree to which deviations larger than ε are tolerated in optimization formulations. For example, if C is too large (infinity), then the objective is to minimize the empirical risk only, without regard to model complexity in the optimization formulation. Parameter ε controls the width of the ε -insensitive zone, which is used to fit the training data. The value of ε can affect the number of support vectors used to construct the regression function. The larger ε is, the fewer support vectors are selected. On the other hand, larger ε values result in "flatter" estimates. Hence, both C and ε values affect model complexity (Vladimir and Ma, 2004).

In this research, the search for parameters was based on previous studies on the selection of parameters for SVR. When different combinations of parameters were applied to the data, their performance was measured using the same MOEs used by ANN models (i.e., slope, y-intercept, R-value, and percent of match).

Gunn (1998) performed a series of experiments on the kernel parameters for different kernel functions. Both Gaussian RBF and Exponential RBF were included in his experiments. Based on the results of the experiments, a set of parameters given in Table 5.8 were identified. Based on the findings from Gunn's experiments, the parameters listed in Table 5.9 were applied. Their suitability is evaluated in the following section.

Table 5.8 Recommended Parameters by Gunn

Kernel Function	ε	P	C
Gaussian RBF	0.05	0.3	∞
Exponential RBF	0.05	1	∞

Table 5.9 Recommended Parameters by Vladimir *et al.*

Kernel Function	ε	p	C
Gaussian RBF	0.25	$0.2 \times (\max(x) - \min(x))$	3
	0.25	$0.3 \times (\max(x) - \min(x))$	3
	0.25	$0.4 \times (\max(x) - \min(x))$	3
	0.25	$0.5 \times (\max(x) - \min(x))$	3
Exponential RBF	0.25	$0.2 \times (\max(x) - \min(x))$	3
	0.25	$0.3 \times (\max(x) - \min(x))$	3
	0.25	$0.4 \times (\max(x) - \min(x))$	3
	0.25	$0.5 \times (\max(x) - \min(x))$	3

Another methodology for selecting the appropriate parameters for SVR was developed by Vladimir *et al.* (2004). The methodology they followed is briefly described here:

1. Selection of parameter C: The optimal choice of regularization parameter C can be derived from standard parameterization of the SVM solution. A simplified presentation of a kernel function bounded in the input domain assumes the form

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2p^2}\right) \quad (5.6)$$

Where $K(\mathbf{x}_i, \mathbf{x}) \leq 1$. Hence, the upper bound on the SVM regression function is:

$$|f(\mathbf{x})| \leq C \cdot n_{SV} \quad (5.7)$$

The following prescription of a regularization parameter was proposed by Vladimir *et al.*:

$$C = \max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|) \quad (5.8)$$

where \bar{y} is the mean of the training responses (outputs), and σ_y is the standard deviation of the training response values. This prescription can effectively handle outliers in the training data. In practice, the response values of training data are often scaled so that $\bar{y}=0$. The proposed C is, thus, $3\sigma_y$. In this research, the target values were preprocessed with normalization, which made the value of $\sigma_y=1$ and the value of C=3 based on Vladimir *et al.*'s conclusions.

2. Selection of ε : It is well-known that the value of ε should be proportional to the input noise level (i.e., $\varepsilon \propto \sigma$). It was assumed in Vladimir *et al.*'s study that the standard deviation of noise σ is known or can be estimated from the data. The following prescription for choosing ε was suggested:

$$\varepsilon \propto \frac{\sigma}{\sqrt{n}} \quad (5.9)$$

For large values of n, this prescription yields ε values that are too small. Hence, the following dependency was proposed:

$$\varepsilon = \tau\sigma \sqrt{\frac{\ln n}{n}} \quad (5.10)$$

Based on empirical tuning, the constant value $\tau = 3$ gave good performance for various data set sizes, noise levels, and target functions for SVM regression. In this research, the number of inputs is 1000, and, thus, $\varepsilon = 3 \times 1 \times \sqrt{\frac{\ln 1000}{1000}} = 0.25$.

3. Determination of p: The parameter p represents the width of the inputs. In Vladimir *et al.*'s study, a range of p was proposed: $(0.2 \sim 0.5) \times (\text{range of inputs})$. Therefore, p values of 20%, 30%, 40%, and 50% of the training input's width

were applied. Following Vladimir *et al.*'s methodology, a series of parameters were prepared for training the SVR models, inclusive of four sets. These are listed in Table 5.9.

5.4.5 Training of SVR Models

With the parameters ready, the training of the SVR could be attempted. It was found that it generally took 30 minutes to complete the training of an SVR model, which is about 30 times the duration spent on ANN model training.

The MSE for all of the SVR models' predictions were around 0.05. The rest of the performance measures for the SVR model training are presented in Tables 5.10 and 5.11. Table 5.10 lists the performance measures for SVR models when the Exponential RBF kernel functions were used, and Table 5.11 lists those when the Gaussian RBF kernel functions were used. Table 5.11 indicates that the Exponential RBF SVR models generally provide better performance than do Gaussian RBF models. Therefore, the Exponential RBF was considered a more suitable kernel function than the Gaussian RBF. Among the five SVR models with an Exponential RBF kernel function, the second, with an R-value of 0.91, demonstrates better performance than the other models. Based on its performance, this model was selected and applied to the input data from the testing location.

Figure 5.10 shows the regression plots between the predicted values from this selected model and the target values. The first 50 cases were chosen as a sample to demonstrate the difference between the predicted values and the target values. Figure 5.11 shows that the predicted values have a larger variance than that of the target values.

Table 5.10 SVR Models Performance with Exponential RBF Kernel Function

Number of Support Vectors and Parameters Values	MOEs	Values
$\varepsilon = 0.05, C = \infty, p1 = 1, NSV = 826$	Slope	0.60
	Y-intercept	0.07
	R-value	0.88
	Percent of Match	0.90
$\varepsilon = 0.25, C = 3, p1 = 0.2 \times (\text{Max}(X) - \text{Min}(X)) = 0.82, NSV = 487$	Slope	0.66
	Y-intercept	0.03
	R-value	0.91
	Percent of Match	0.88
$\varepsilon = 0.25, C = 3, p1 = 0.3 \times (\text{Max}(X) - \text{Min}(X)) = 1.24, NSV = 505$	Slope	0.60
	Y-intercept	0.05
	R-value	0.87
	Percent of Match	0.90
$\varepsilon = 0.25, C = 3, p1 = 0.4 \times (\text{Max}(X) - \text{Min}(X)) = 1.66, NSV = 500$	Slope	0.60
	Y-intercept	0.06
	R-value	0.87
	Percent of Match	0.87
$\varepsilon = 0.25, C = 3, p1 = 0.5 \times (\text{Max}(X) - \text{Min}(X)) = 2.02, NSV = 476$	Slope	0.58
	Y-intercept	0.04
	R-value	0.85
	Percent of Match	0.87

Table 5.11 SVR Models Performance with Gaussian RBF Kernel Function

Number of Support Vectors and Parameters Values	MOEs	Values
$\varepsilon = 0.05, C = \infty, P1 = 0.3, NSV = 862$	Slope	0.40
	Y-intercept	0.07
	R-value	0.81
	Percent of Match	0.86
$\varepsilon = 0.25, C = 3, P1 = 0.2 \times (\text{Max}(X) - \text{Min}(X)) = 0.83, NSV = 410$	Slope	0.86
	Y-intercept	0.00
	R-value	0.89
	Percent of Match	0.86
$\varepsilon = 0.25, C = 3, P1 = 0.3 \times (\text{Max}(X) - \text{Min}(X)) = 1.24, NSV = 411$	Slope	0.65
	Y-intercept	0.02
	R-value	0.86
	Percent of Match	0.86
$\varepsilon = 0.25, C = 3, P1 = 0.4 \times (\text{Max}(X) - \text{Min}(X)) = 1.68, NSV = 495$	Slope	0.55
	Y-intercept	0.07
	R-value	0.83
	Percent of Match	0.86
$\varepsilon = 0.25, C = 3, P1 = 0.5 \times \text{Max}(X) - \text{Min}(X) = 2.04, NSV = 499$	Slope	0.55
	Y-intercept	0.07
	R-value	0.81
	Percent of Match	0.83

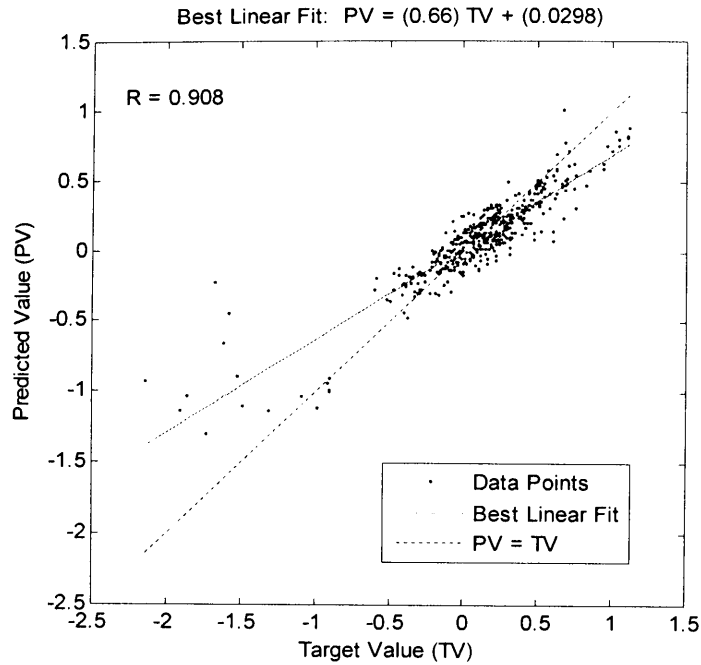


Figure 5.10 Regression Relationships between the Predicted and Target Values with Exponential RBF SVR Models ($\varepsilon=0.25$, $C=3$, $p1=0.82$)

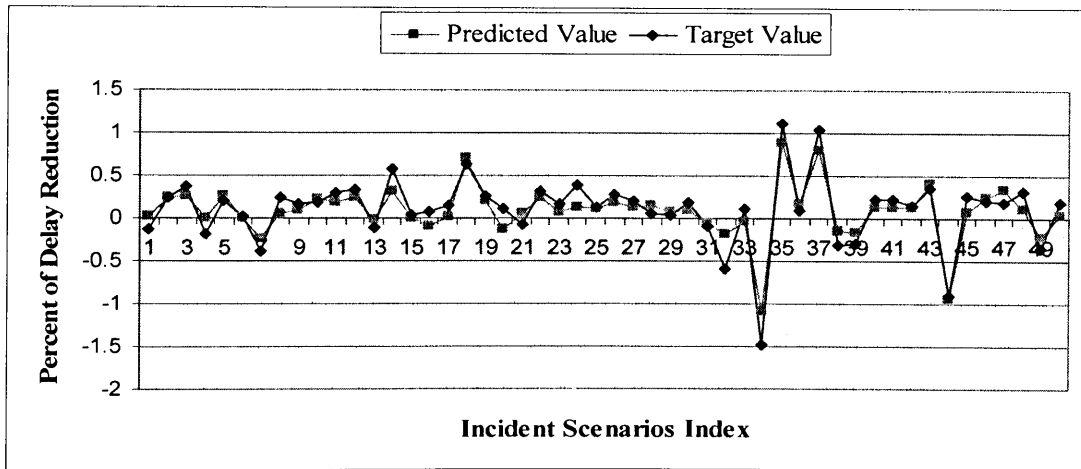
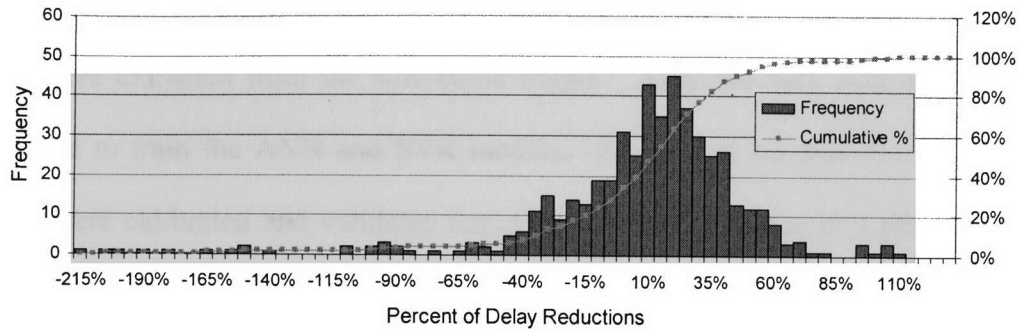


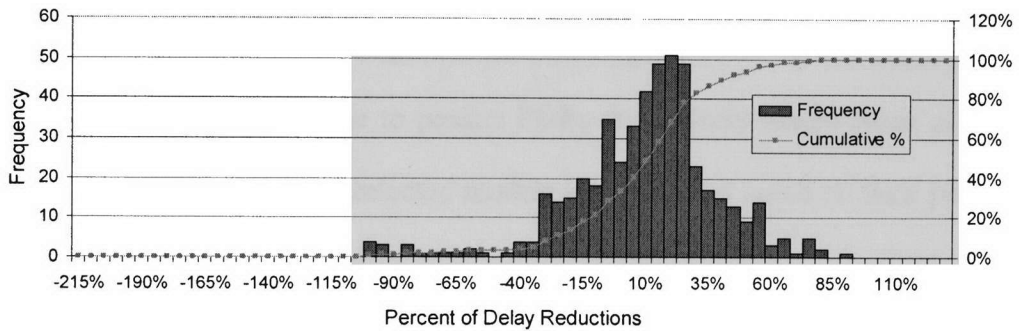
Figure 5.11 Comparison of Predicted and Target Values of SVR Model

A frequency analysis was also performed to evaluate the performance of the selected SVR model. Figures 5.12 (a) and (b) show the frequency distribution of the target PDR and the SVR model predicted PDR, respectively. From Figure 5.13 (a), as

was observed from the distribution of the target PDR for ANN model testing, over 90 percent of the PDR as a result of diversion falls between -40% and 80%.



(a) Distribution Frequency for Target PDR Values



(b) Distribution Frequency for SVR Model Predicted PDR Values

Figure 5.12 Histograms for Target PDR and SVR Model Predicted PDR Values

Comparing Figures 5.12 (a) and (b), it was found that the SVR model predicted PDR falls into a smaller range than the target PDR. Over 90% of the PDR falls into the range of (-30%, 75%). The most frequently occurring PDRs from the SVR model range between 10% and 25%. Because the 500 predictions were distributed to a smaller range, the frequency of the predicted PDRs that occurred in each 5% range is generally higher than that of the target PDRs.

5.5 Summary

In this chapter, different incident scenarios representing different incident conditions were simulated in DYNASMART-P. Critical link flow, speed, and queue length were extracted from the simulation output. After the data were prepared, they were used to train the ANN and SVR models. Parameters for these two data mining models were calibrated and validated based on their performance in a preliminary test. From among all of the ANN and SVR models, those with the best performance were selected as the models for the training location. The predicted values produced from the ANN and SVR models were compared with the target values. The comparison results show that these models are able to predict PDRs that closely match those of the target PDRs. In the next chapter, the selected models are evaluated based on their performance at the testing location.

CHAPTER 6

MODEL EVALUATION

6.1 Introduction

In this chapter, the selected ANN and SVR models that performed best during the preliminary test were applied to the testing location. This location was selected in the opposite direction on the same freeway to evaluate application generality of the models. The same method used for the training location was used to develop the incident scenarios for the testing location. Altogether, 1650 scenarios were designed and simulated in DYNASMART-P. Data generated from the simulation output of these 1650 scenarios were used to calculate PDRs as well as to generate inputs to test the selected ANN and SVR models. The developed ANN and SVR models were evaluated based on their performance at both at the location they were trained and at the testing location.

6.2 Data Preparation for Model Application on the Testing Location

In this section, efforts were made to design incident scenarios for the testing location. The testing location was selected on the same freeway as the training location; however, it was tested on a link taken from the opposite direction. As shown in Figure 6.1, there are five exit ramps upstream of this link and four on ramps downstream of this link. As was done for the training location, a pre-planned alternate route was selected with the assistance of DYNASMART. When an incident occurs at the testing location (on link 39→44), the alternate path that the diverted traffic should follow is

19→20→3→24→5→27→65→7→31 →36→9→40 →11→43→44 (as shown in Figure 6.2). Table 6.1 illustrates the link properties along the alternate route.

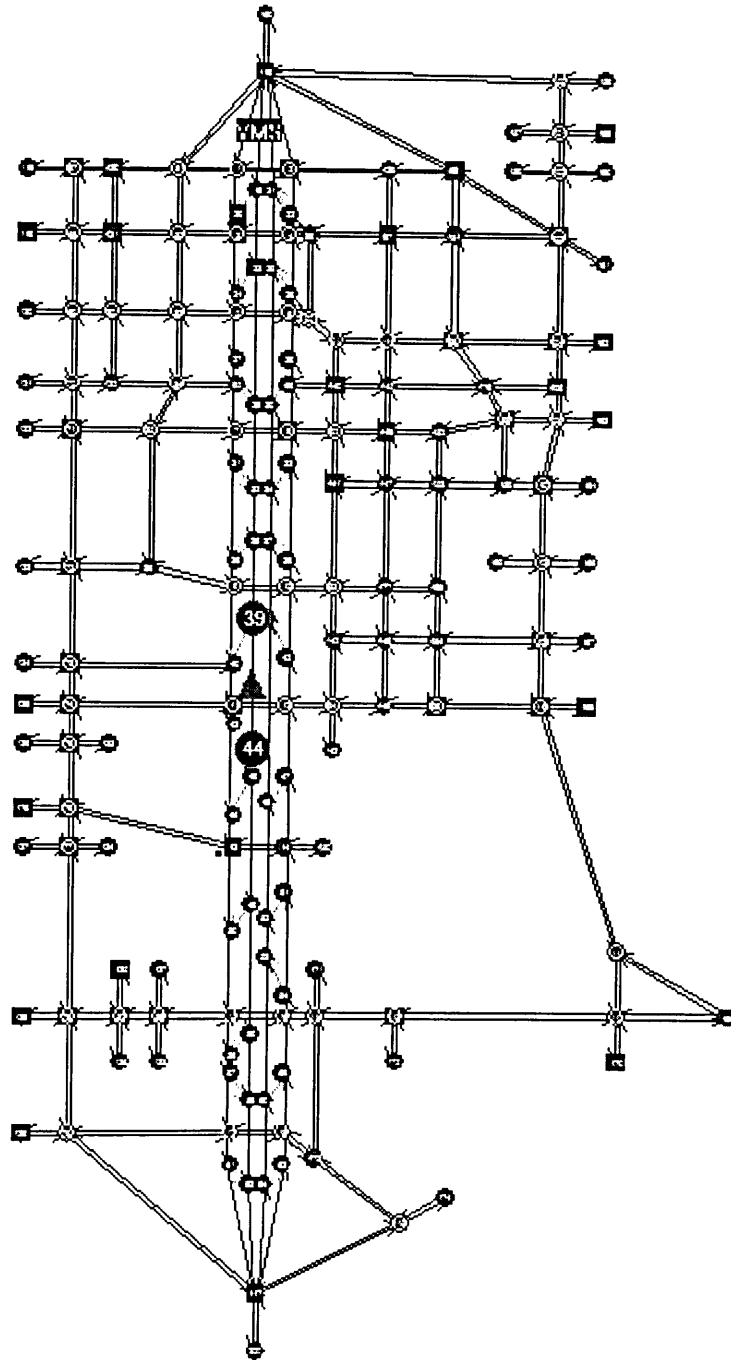


Figure 6.1 Testing Location (on Link 39→44)

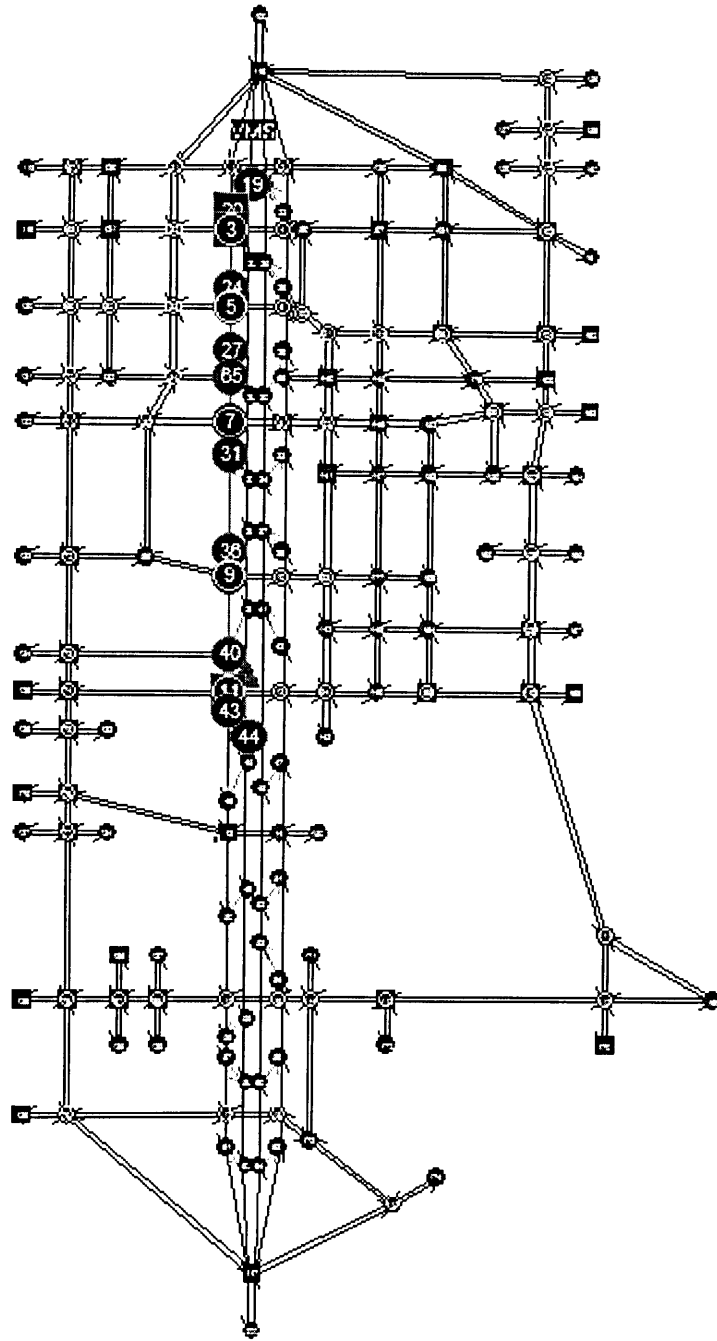


Figure 6.2 Alternate Route for Testing Location

After the testing location and alternate route were decided on, 1500 incident scenarios were created to represent different combinations of attributes of variables as listed in Table 6.2.

Table 6.1 Link Properties on Alternate Route for Testing Location

Link ID	From Node	To Node	Number of Lanes	Speed Limit	Saturation Rate	Type	Length (mile)
55	19	20	1	40	1,800	Off Ramp	0.11
57	20	3	2	40	1,800	Arterial	0.08
8	3	24	2	40	1,800	Arterial	0.26
63	24	5	4	40	1,800	Arterial	0.09
14	5	27	3	40	1,800	Arterial	0.13
68	27	65	3	40	1,800	Arterial	0.13
125	65	7	3	40	1,800	Arterial	0.28
20	7	31	2	40	1,800	Arterial	0.08
74	31	36	2	40	1,800	Arterial	0.71
81	36	9	4	40	1,800	Arterial	0.12
26	9	40	3	40	1,800	Arterial	0.33
87	40	11	4	40	1,800	Arterial	0.16
32	11	43	3	40	1,800	Arterial	0.10
92	43	44	1	40	1,800	On Ramp	0.19

Table 6.2 Attributes of Critical Variables for Testing Location

Variables	Values
Capacity Lost (Incident Severity)	20%, 40%, 60%, 80%, 100%
Incident Location	Link 39→44
VMS Location	Link 116→19
Incident Detection and Verification	3 mins
VMS Duration	5 mins before the incident ends; terminate when the incident ends
Incident Response and Clearance Time	25, 30, 35, 40, 45, 50, 55, 60, 65, 70
Level of Congestion (Multiplication Factor)	low (30), moderate (40), high (50)
Drivers Responsive to DMS	15%, 30%, 50%, 75%, 100%

During the simulation of an incident in DYNASMART-P, all of the links on the alternate route and the links upstream of the incident link were examined. Links were selected if they demonstrated different performances with and without an incident. Three critical links were selected for each incident location. Links 39→44, 35→39, and 43→44

were selected for the training location. Figures 6.3, 6.4, and 6.5 present density comparisons of the selected links for the testing location.

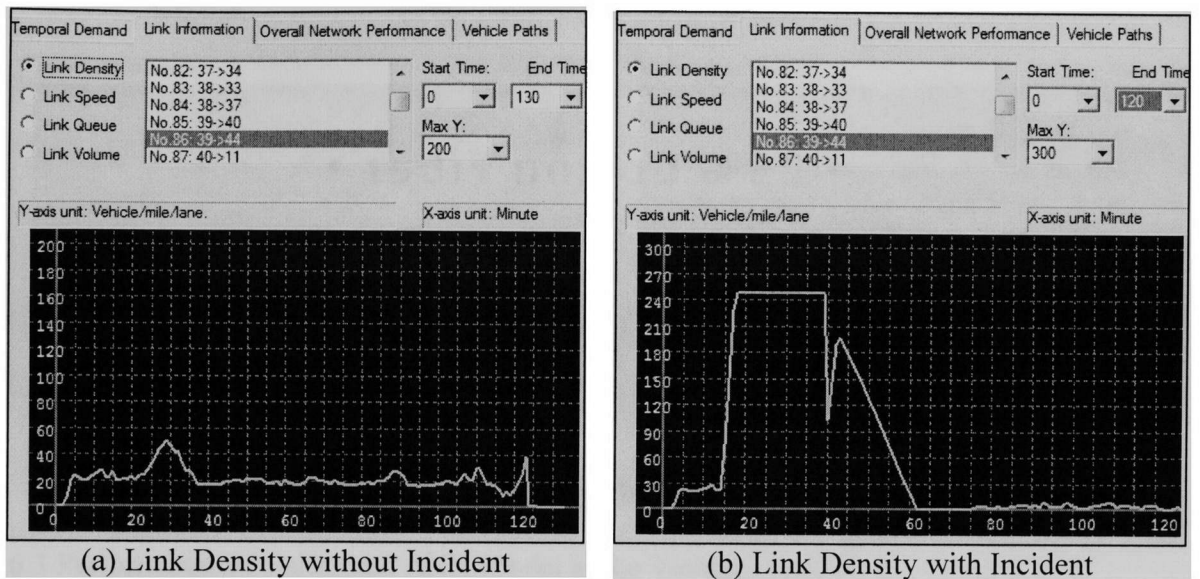


Figure 6.3 Link Density of Link 39→44

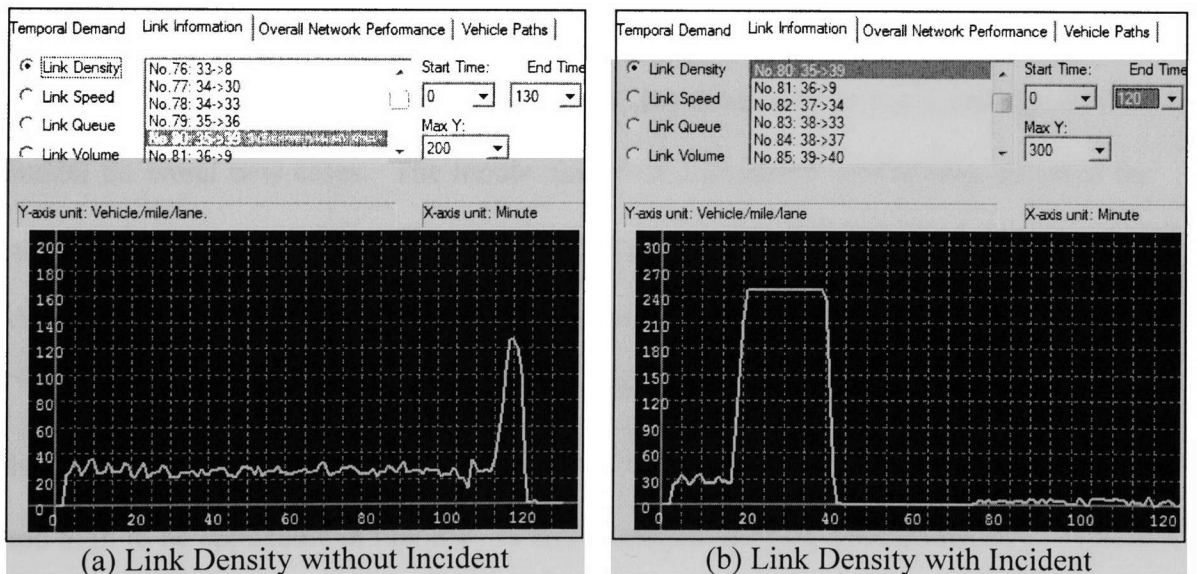


Figure 6.4 Link Density of Link 35→39

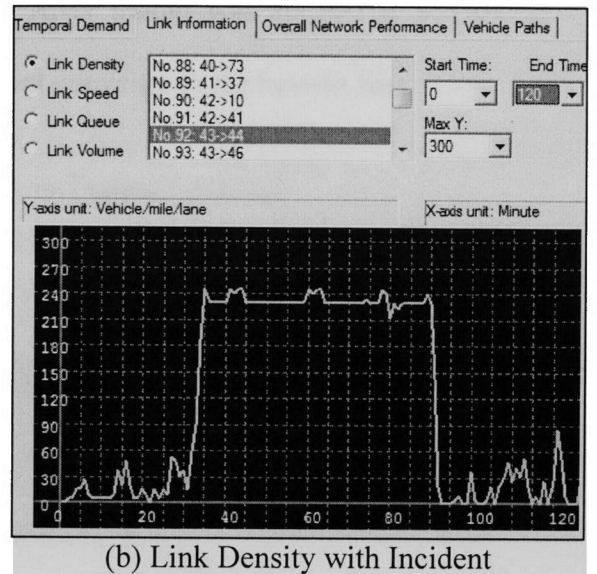
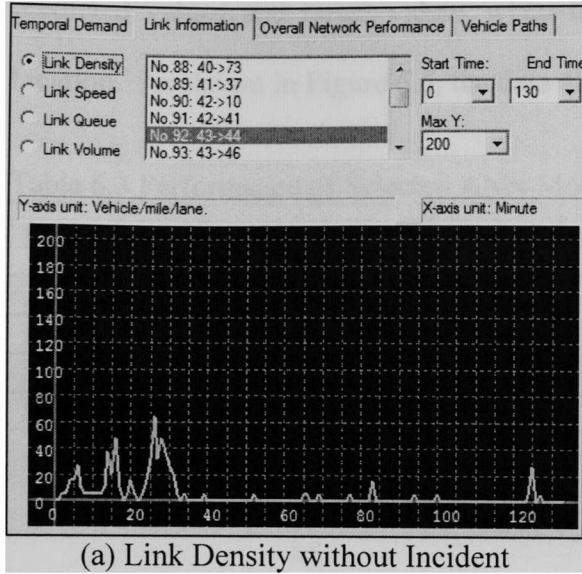


Figure 6.5 Link Density of Link 43→44

6.3 Performance of Calibrated ANN Model at the Testing Location

After all of the incident scenarios were simulated in DYNASMART-P, the required data were prepared for the testing of the ANN and SVR models. The ANN model was tested first. A MATLAB program was developed to test a calibrated ANN model on brand new cases. The inputs were first normalized so that they obtained the same distribution as the training inputs after applying PCA. The predicted values were de-normalized before comparison with the target values.

The performance measures when the selected ANN model is applied at the testing location are listed in Table 6.3. The MSE with this prediction was 0.37 overall, which is too high to be applicable in real life. A more detailed analysis was performed to explore the performance of the model at different network congestion levels. The results for this analysis are also presented in Table 6.3. Figure 6.6 shows a comparison between the

predicted values and target values when this ANN model was applied at the testing location. As shown in Figure 6.6, the data did not converge to the best-fit line.

Table 6.3 Performance of Selected ANN Model on Testing Location

Network Congestion Level	Slope	Y-intercept	R-value	MSE	Percent of Match
All Levels	0.87	0.12	0.47	0.37	0.65
Low	0.21	-0.04	0.12	0.16	0.56
Moderate	-0.60	0.30	-0.14	0.67	0.63
High	0.92	0.30	0.71	0.30	0.77

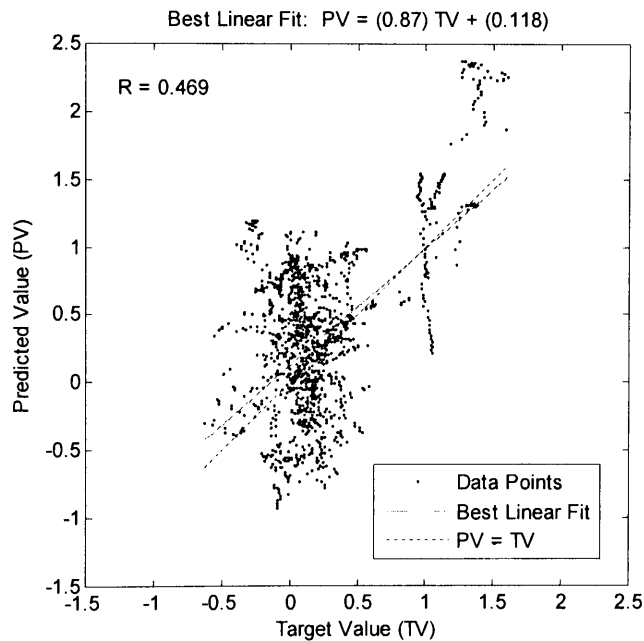


Figure 6.6 Regression Plot of ANN Model Predicted and Target Values for Testing Location

Based on the level of congestion in the network, the testing data for the testing location were divided into three groups with 500 incident scenarios in each group. Figures 6.7, 6.8, and 6.9 present the regression plots for these three groups. As shown in Figure 6.7, when the congestion level in the network is low, the predicted values are very spread out in the plot. However, the range for the predicted values is $(-0.8, 0.6)$, which is

the same as the range for the target value. In addition, the MSE for this group is 0.16, which is the lowest among the three.

In Figure 6.8, when the network congestion level is moderate, the predicted values show very different trend than do the target values. The range for the predicted values is also much larger than that of the target values. The MSE for this group is 0.67, which is the highest among the three groups. All of the evaluation standards show that the ANN model performed worst when the congestion level in the network is moderate.

When the network congestion level is high, however, as Figure 6.9 shows, the regression plot is different from the other two. The predicted values show a similar trend to that of the target values ($R\text{-value}=0.71$). The slope for the best linear fit line is 0.92, which means that this line is very close to 45 degrees. However, the y-intercept value for the data is 0.3, which is much greater than the situation wherein the network experiences low congestion. With regard to accuracy, the MSEs for these three groups are 0.16, 0.66, and 0.30, respectively, and the POM values are 0.56, 0.63, and 0.77, respectively.

Based on a detailed analysis of the ANN model's performance for different network congestion levels, it was concluded that the ANN model worked best when the level of congestion in the network was high. Whereas when the level of congestion in the network was moderate, the ANN model could not produce reliable results.

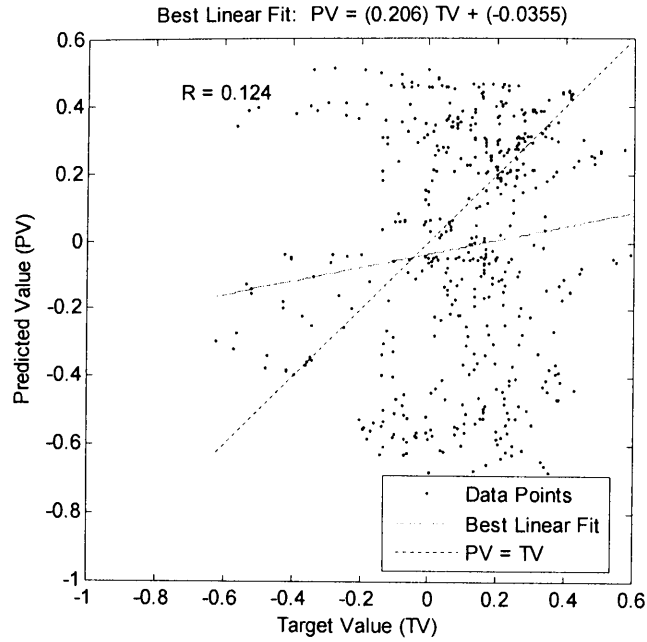


Figure 6.7 Regression Plot of ANN Model Predicted and Target Values for Testing Location under Low Network Congestion Level

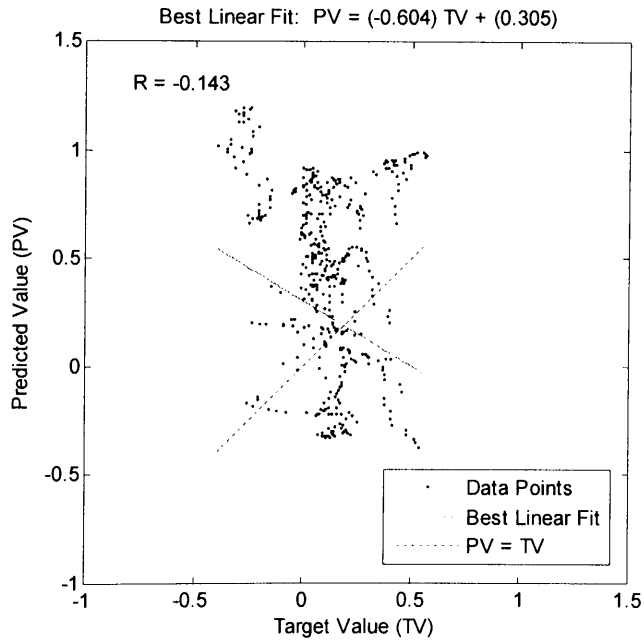


Figure 6.8 Regression Plot of ANN Model Predicted and Target Values for Testing Location under Moderate Network Congestion Level

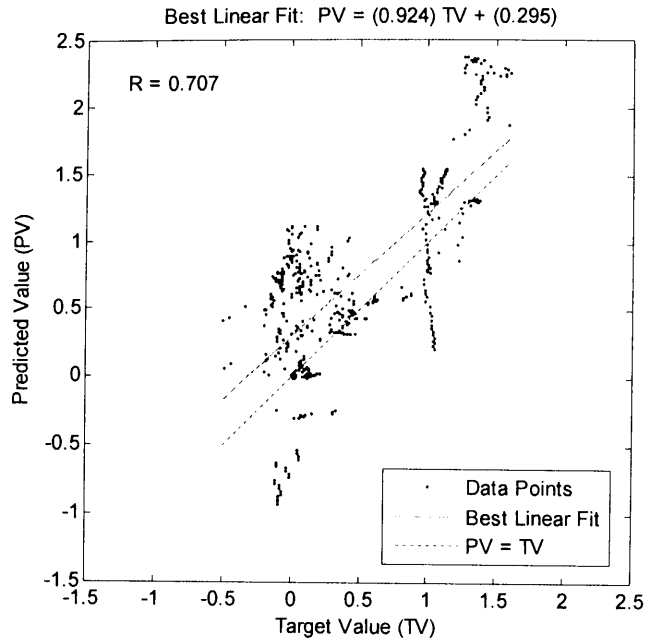


Figure 6.9 Regression Plot of ANN Model Predicted and Target Values for Testing Location under High Network Congestion Level

In conclusion, the ANN model developed in this research produced a comparatively reliable prediction for the location it is trained for. However, when applied to the testing location, the model could not produce a prediction reliable enough for real-world applications. Adjustments can be made to improve the prediction reliability when the network exhibits high congestion.

6.4 Evaluation of Performance of Selected SVR Model on Testing Location

After the selected SVR model was applied to the location it was trained for, it was applied to the testing location to assess its performance as a more general model. The testing procedure was very similar to the test of the ANN models. A total of 1500 cases generated for testing location were used as the testing cases. A MATLAB program was developed to test the calibrated SVR model on new cases. The inputs were first

normalized by applying the mean and standard deviation of the training inputs of the training location. After preprocessing, these data were then fed into the SVR model to produce the predicted values. The predicted values were also de-normalized before comparison with the target values.

After a comparison of the performance of the selected SVR model at both the training and testing locations, it became evident that the model performed much better at the location it was trained for. The overall MSE for the prediction at the testing location is 0.13. This indicates that the SVR model gave a more accurate prediction than the ANN model when each was applied to the testing location. A comparison between the predicted values and the target values is shown in Figure 6.10. As the figure shows, the slope for the best fitted line is much smaller than 45 degrees. This means that the model produced predictions with a greater variance.

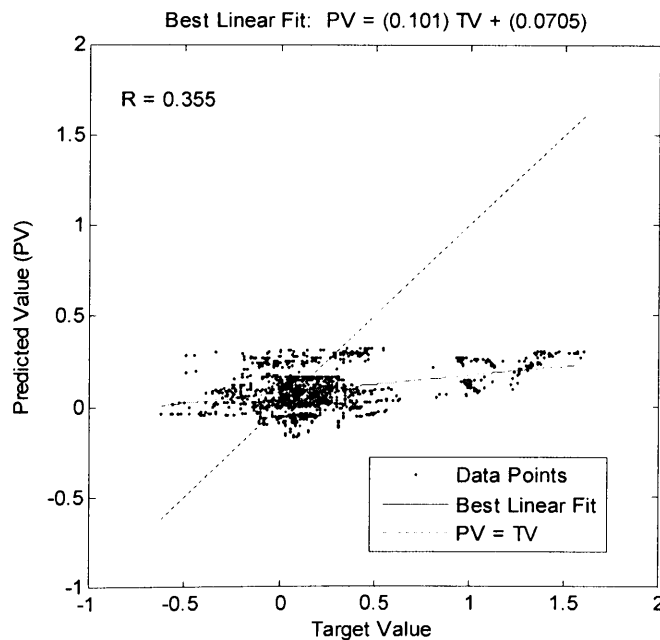


Figure 6.10 Regression Plot of SVR Model Predicted and Target Values for Testing Location

To evaluate the performance of the model for different network congestion levels, the whole dataset was divided into three groups. Each group comprised 500 incident scenarios representing a certain network congestion level. Regression analysis was carried out to evaluate the performance of the SVR model on these three groups of data. The regression plots for the three groups are shown in Figures 6.11, 6.12, and 6.13, respectively. Figure 6.12 shows the plots when the level of congestion in the network was low. The R-value was 0.46 and the slope of the best linear fit line was 0.14. Both values are the highest among the three groups. When the congestion level was moderate, the R-value was -0.31. This represents a negative relation between the predicted and target values. When the level of congestion in the network was high, the R-value was 0.22 and the slope of the best fitted line was 0.05.

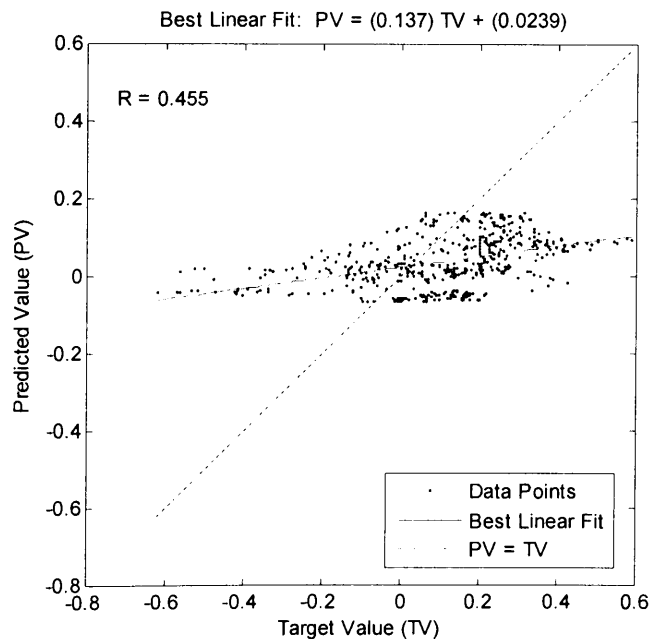


Figure 6.11 Regression Plot of SVR Model Predicted and Target Values for Testing Location under Low Network Congestion Level

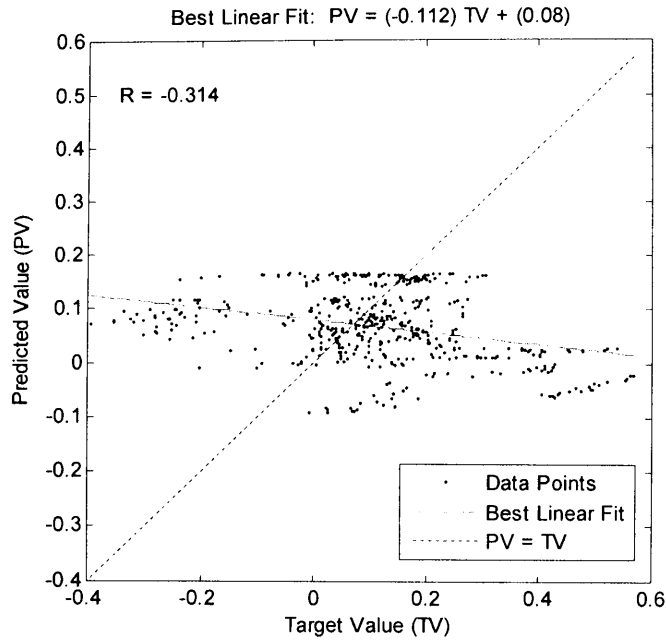


Figure 6.12 Regression Plot of SVR Model Predicted and Target Values for Testing Location under Moderate Network Congestion Level

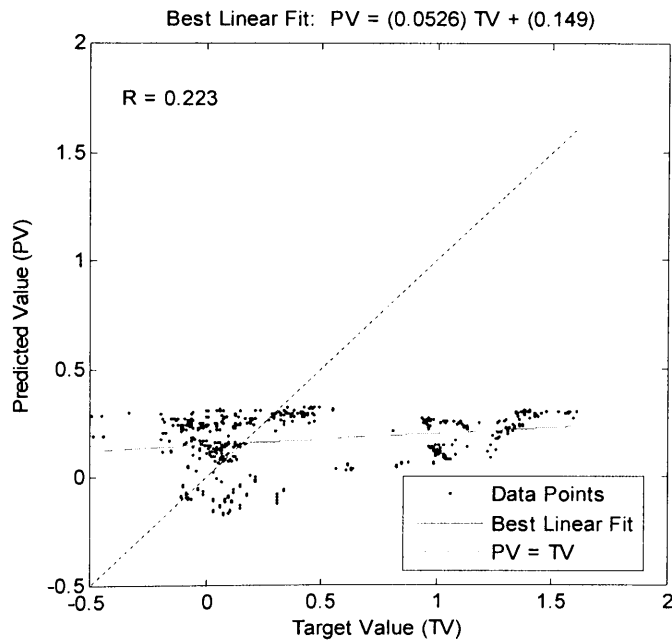


Figure 6.13 Regression Plot of SVR Model Predicted and Target Values for Testing Location under High Network Congestion Level

In terms of accuracy, the MSEs for these three groups were 0.04, 0.05, and 0.31, respectively, and the POM values were 0.69, 0.71, and 0.74, respectively. The performance measures for the entire dataset, as well as for the three groups, are listed in Table 6.4.

Table 6.4 Performance of Selected SVR Model on Testing Location.

Network Congestion Level	Slope	Y-intercept	R-value	MSE	Percent of Match
All Levels	0.05	0.08	0.31	0.13	0.76
Low	0.14	0.02	0.46	0.04	0.69
Moderate	-0.11	0.08	-0.31	0.05	0.71
High	0.05	0.15	0.22	0.31	0.74

In conclusion, when the SVR model was applied at the testing location, it was evident that SVR produced predictions with a low MSE. However, when the regression correlation was considered, the SVR model could not be regarded as a better model than the ANN model.

The reduced performance of the ANN and SVR models at the testing location was likely a result of the fact that there were variables that were not accounted for in the inputs for training the models. These variables may include alternate route length in comparison to the primary route, capacity (or volume to capacity ratio) on the alternate route, number of signalized intersections and left turns on the alternate route, differences in congestion along the primary and alternate routes, on-ramp and off-ramp geometry, and possibly other factors that may impact the average travel time. To account for these variables, a fairly large number of locations must be selected for model training so that

the changes in all potential critical variables influencing route diversion decisions may be reflected in the training data set.

6.5 Comparison between Selected ANN and SVR Models

In this section, the selected ANN and SVR models are compared. The comparison includes training time, the calibration of parameters, an evaluation of performance based on preliminary testing of the ANN and SVR models at the training location, and an evaluation of their performance based on testing the ANN and SVR models at the testing location. The advantages and disadvantages of these two models as applied to the route diversion problems this research intends to solve are discussed and summarized below.

6.5.1 Training Time

The training times for the ANN and SVR models differ greatly in the experiments. It took about one minute to train an ANN model when the variable learning rate algorithms were used, whereas it took a minimum of 30 minutes to train the SVR model. This was mentioned in the section on SVR training. One reason for this difference was that PCA was introduced into the ANN models, reducing the training size of the inputs. Another reason was that the variable learning rate was a model specializing in rapid training. However, although it was a fast training model as well, the Levenberg-Marquardt algorithm did not achieve the same training speed as the variable learning rate models (traingdx and traingda). It took about three minutes to train a Levenberg-Marquardt ANN model. Nevertheless, when compared to the training time for the SVR models, the Levenberg-Marquardt ANN models could still be considered fast. The

average duration for training the SVR models ranged from 30 to 100 minutes, depending on the parameter values used in the models. When the value for ε was set to 0, the training time of the SVR models was as long as 150 minutes.

The time used for testing the ANN and SVR models was different, too. The ANN model could calculate at a faster rate than the SVR model. However, both models were able to produce a prediction within one second, which is fast enough for real-world applications.

In conclusion, as far as the training and testing times required for the models are concerned, the ANN models outperformed the SVR models. Further study needs to be carried out to explore the possibility of reducing the training time for SVR models.

6.5.2 Calibration of Model Parameters

In this research, the ANN models experienced a more complicated calibration process than the SVR models, as more parameters had to be calibrated for the ANN models. The parameters exclusive to ANN include transfer functions, BP algorithms (variable learning rate and Levenberg-Marquardt), learning rate and momentum, the number of hidden layers, and the number of hidden neurons in these layers. The calibration of SVR models involved processes that included kernel function selection (Gaussian RBF or Exponential RBF) and determination of C , ε , and p .

As mentioned, there were more parameters to be determined for the ANN models than for the SVR models. In addition, the calibration process of the SVR models was made easier by utilizing the findings from previous studies. Despite that the selection of the ANN models parameters required more steps, the calibration process was shorter for

the ANN models than for the SVR models. This is because a shorter training time was required for the ANN model.

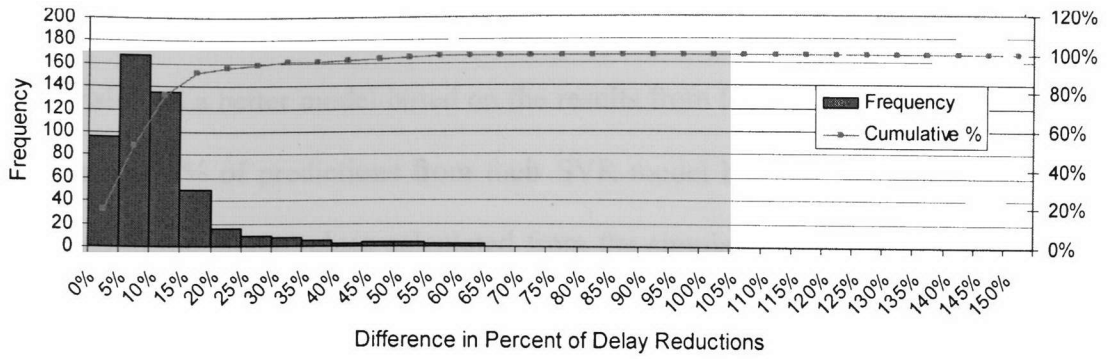
6.5.3 Performance of ANN and SVR Models Based on Preliminary Testing

In this research, both the ANN and SVR models demonstrated the ability to produce acceptable predictions on the percent of delay reductions to be gained from traffic diversion. The performance measures for both selected models are listed in Table 6.5. It can be concluded from the table that the overall performance of the ANN model is better than that of the SVR model.

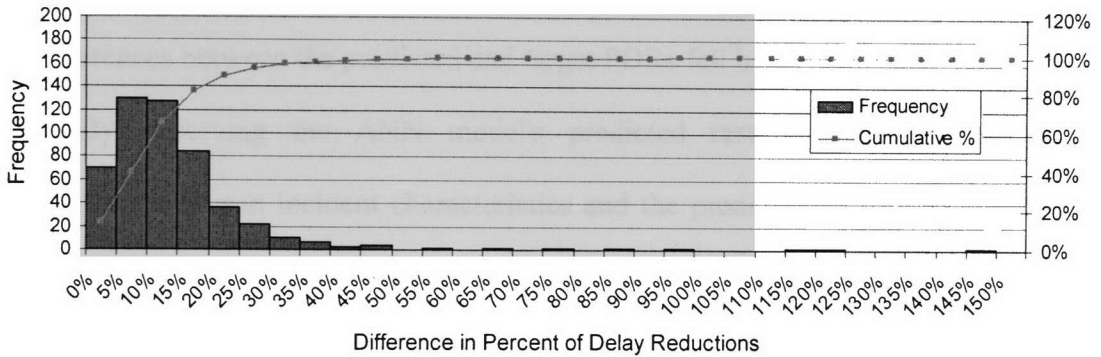
Table 6.5 Performance of Selected ANN and SVR Models on Training Location.

Data Mining Models	Slope	Y-intercept	R-Value	MSE	Percent of Match
ANN Model	0.88	0.01	0.92	0.02	0.87
SVR Model	0.66	0.03	0.91	0.04	0.88

To further show that the ANN model outperformed the SVR model, histograms were used. These histograms show the frequency distribution of the absolute difference of the predicted PDR from the models and the target PDRs calculated from the simulation outputs. Figure 6.14 (a) shows the frequency distribution of the absolute difference between the predicted PDR from the ANN model and the target PDRs, whereas Figure 6.14 (b) shows that of the predicted PDR from the SVR model and the target PDRs.



(a) Difference between Predicted PDRs from ANN Model and Target PDRs



(b) Difference between Predicted PDRs from SVR Model and Target PDRs

Figure 6.14 Histograms of Absolute Difference between Model Predicted PDR and Target PDR for Training Location

Based on a comparison of the frequency distribution of the absolute difference between the model predicted PDRs and the target PDRs, the ANN model demonstrates a stronger ability to produce more accurate predictions. For more than 95% of all 500 testing incident scenarios, the absolute difference between the ANN model predicted PDRs and the target PDRs are below 20%. Only 90% of the PDR predictions from the SVR model fell below a 20% difference when compared to the target PDRs.

Chowdhury *et al.* (2005) attempted to use SVR and CBR models to predict the PDRs in their study, as mentioned in the literature review in Chapter 2. In their study,

they used 35 incident scenarios to test the SVR and CBR models they developed. SVR was considered a better model based on the results from the test. Among all of the cases they tested, 91% of predictions from their SVR model had a difference of below 20% compared to the target values calculated from the simulation. Assuming that 35 cases was a large enough sample size to test the precision of the model, the ANN model developed in this research is still considered a better model than the SVR model Chowdhury *et al.* developed. This is based on the result from this research that 95% of the differences between the predicted and target PDRs fell below 20%.

By observing the ANN model's predicted results, some features of the relationships between incident characteristics and the predicted PDRs were determined. Figure 6.15 presents the relationships between incident durations and ANN predicted PDR values, and Figure 6.16 presents the relationships between the capacity lost (incident severity) and ANN predicted PDR values. Notice that these figures show a situation wherein 50% to 75% of the drivers are VMS responsive. Each figure contains three curves representing three different network congestion levels. In Figure 6.15, the data are not available when the capacity lost is 1 and the level of congestion in the network is high. In Figure 6.16, there are also some data missing when the incident duration is longer than 55 minutes. From Figure 6.15, it is observed that the capacity lost and the PDR are almost negatively related when the network is highly congested and are nearly positively related when the network congestion level is low. Figure 6.16 shows that when the level of congestion in the network is low, the PDR values are negative for most of the incident duration; however, when the level of congestion level is either moderate or high, the PDR values are all greater than zero. In Figure 6.16, the curve for a

high congestion level shows some significant ups and downs. This is due to the difference in the percent of drivers responsive to the VMS. These two figures also indicate the possibility of developing a better performance model for each congestion level because different characteristics are exhibited when network congestion levels are different.

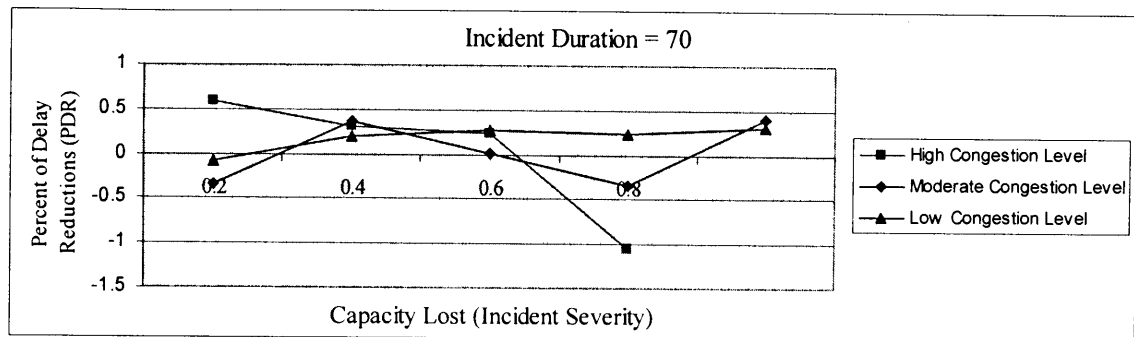


Figure 6.15 Effect of Capacity Lost on ANN Predicted PDRs

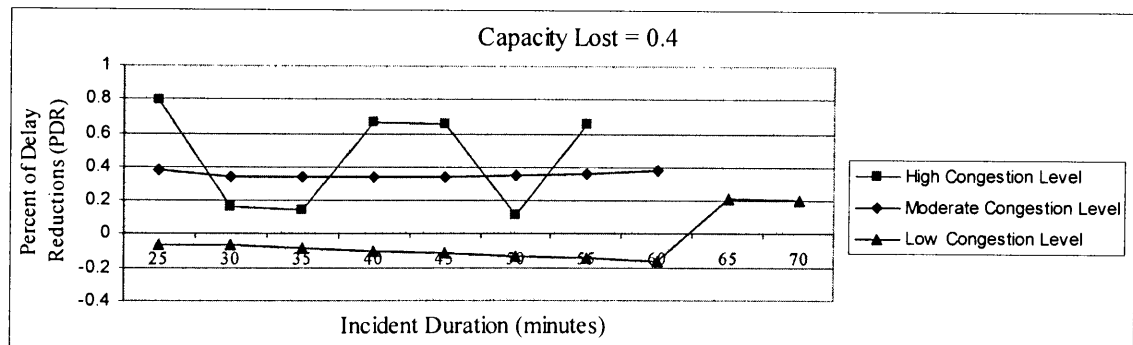


Figure 6.16 Effect of Incident Duration on ANN Predicted PDRs

6.5.4 Performance of ANN and SVR Models for the Testing Location

When the performances of the ANN and SVR models at the testing location were compared, the SVR model was found to be superior to the ANN model in terms of MSE. However, the predicted PDRs from the ANN model were more highly correlated to the PDRs. Note from Table 6.6 that the R-value for the ANN model is closer to 1 than it is for the SVR model, while the POM for the SVR model is higher than it is for the ANN

model. Also note that the MSE of the SVR prediction is much lower than that of the ANN model when the network is less congested; however, it is slightly higher than that of the ANN model when the network is more congested.

Table 6.6 Comparison between Performance of Selected ANN and SVR Models.

MOEs	ANN Model	SVR Model
Slope	0.87	0.05
Y-intercept	0.12	0.08
R-value	0.47	0.31
MSE	0.37	0.13
Percent of Match	0.65	0.76

6.6 Summary

In summary, a comparison between the selected ANN and SVR models leads to the conclusion that the performance of the ANN model was better than that of the SVR model when each model was applied to the location it was trained for. However, both models were able to produce acceptable PDR predictions. When applied to the testing location, predictions from both models were significantly less accurate than they were when applied at the location they were trained for. Their predictions at the testing location, overall, are not adequate for real-world applications. However, when the network congestion level is high, the ANN model's prediction can be improved to produce comparatively accurate results.

CHAPTER 7

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

7.1 Summary of Research Approaches and Results

This dissertation study has accomplished several major research tasks. First, a comprehensive literature review was conducted in this research to investigate and assess the current techniques, tools, and methods for implementing route diversion plans. The primary results of this review are summarized as follows: (1) Factors that have critical impacts on route diversion decisions were preliminarily identified. These include freeway geometric and traffic conditions, local network geometric and traffic conditions, and incident location and severity. (2) ITS technologies were reviewed, as was their significance in facilitating freeway route diversions. (3) The application of DYNASMART-P as a simulator was reviewed. (4) Data mining algorithms, including Artificial Neural Networks (ANN), Support Vector Regression (SVR), and Case-Based Reasoning (CBR) were reviewed, as were their applications in solving transportation related problems.

Based on preliminary identification of critical variables, this research further evaluated these variables according to: (1) their significance in affecting route diversion decisions, (2) the availability of these data in a real transportation environment, and (3) the complexity and feasibility of including these variables in this research.

DYNASMART-P was selected as the simulator to represent the real-world transportation environment for freeway incidents and route diversion. The algorithm provided in DYNASMART-P was employed to calculate the optimum path to be used as

the pre-planned alternate route for the incident locations designed. A total of 3000 incident scenarios representing different combinations of variables identified previously at two incident locations were designed. The DYNASMART-P simulator was then used to simulate each of these scenarios.

Data were prepared for the training and testing of the ANN and SVR models and were separated based on the location of the incident. A simulation output for 1000 incident scenarios randomly selected for the training location was extracted for use in training the ANN and SVR models. A simulation output for the remaining 500 scenarios was extracted and used to test the ANN and SVR models. Both the ANN and SVR models were calibrated for their best performance through specific procedures. The selected ANN and SVR models with the best performance were then tested with data from scenarios designed for the testing location. A detailed analysis was performed to evaluate both models at different network congestion levels.

7.2 Conclusions

The primary task of this research was to develop ANN and SVR models to predict the benefits of route diversion at the time of an incident. Because of the lack of real-world traffic data related to traffic diversion, DYNASMART-P was selected as the simulator to represent the real-world transportation environment for freeway incidents and route diversion. It was found that, as a mesoscopic simulation model, DYNASMART-P was able to conveniently simulate different incident and route diversion scenarios in an ITS environment. The incident input file for DYNASMART-P (i.e., incident.dat) includes all of the incident related information, and DYNASMART-P

allows this file to be changed externally by MATLAB. This is convenient and efficient when multiple incident scenarios need to be simulated. The diversion information dissemination using Variable Message Signs (VMS) is controlled by the vms.dat file in DYNASMART-P. The information as to where the sign is located, what type of VMS is used, and which alternate route the VMS should direct drivers to are all coded in this file. MATLAB can easily change the settings of the vms.dat file externally. In addition to the convenience when changing the inputs for DYNASMART-P, DYNASMART-P also provides user friendly output options. These options enable users to selectively collect link volume, queue length, and speed information for each simulation interval they specify. These features greatly facilitated the simulation and data preparation needed in this research. However, despite all of the conveniences of using DYNASMART-P in route diversion simulation, there were also some problems associated with this simulator, including its inability to simulate the exact location of the incident on a link.

Percent of delay reductions (PDR) was used as the Measure of Effectiveness (MOE) of the route diversion strategy and used as the target value for model training. It was found through the calculation of PDR that, in over 60% of occasions, route diversion as an incident management strategy was able to bring over 5% of PDR to the whole system. This suggests that route diversion could potentially be an effective way of reducing incident delay.

Two locations were selected and used as the training and testing locations, respectively. Data collected from two-thirds of the incident scenarios at the training location were used as model training and calibration data, and the remaining one-third of

incident scenarios from this location were used as model testing data. The data from the testing location were used for model testing purposes only.

During the calibration of the ANN and SVR models, a series of parameters were calibrated for optimum performance of these models. When preliminarily tested with data from the training location, an ANN model was chosen for its best performance. The parameters of this ANN model included: a log-sigmoid transfer function for the input layer, a linear transfer function for the output layer, a variable learning rate algorithm, and nine neurons in the hidden layer. Findings in two previous studies, carried out by Vladimir *et al.* (2004) and Gunn (1998), respectively, were employed to reduce the range of appropriate SVR parameters for this research. Parameters with values within a certain range were then used to calibrate the SVR model having the best performance. An SVR model with an exponential RBF kernel function and parameters set at $\varepsilon=0.25$, $C=3$, and $p1=0.82$ produced the best performance during the preliminary testing of these models. During the training of the ANN and SVR models, it was observed that the training time for SVR was much longer than that for the ANN models. It took less than five minutes to train ANN models with any given set of parameters. In contrast, it took 30 to 150 minutes to train the SVR models, depending on the parameters used for the model. For example, when ε was set at zero, the SVR model training took as long as 150 minutes.

As part of the model evaluation effort, the ANN and SVR models were tested using the data from the location they were trained for. A group of performance measures was applied to evaluate these models: the slope and the y-intercept for the best linear fit line, and correlation coefficient (R-value) for regression analysis; mean square error (MSE) and percent of match (POM) for precision evaluation; and histograms for

frequency distribution analysis. Both models, when tested at the locations they were trained for, provided acceptable accuracy in their predictions and are ready for use in real-world applications. It was also found that the ANN model outperformed the SVR model. In addition, a comparison between the results from a previous study carried out by Chowdhury *et al.* (2005) indicated that the ANN model developed in this research has better performance than the SVR model developed in the previous study. In this research, 95% of the difference between PDR predictions from the ANN model and the target PDR are below 20%.

The selected ANN and SVR models were further tested with data from incident scenarios designed for the testing location. The purpose of this test was to evaluate the models' performance in providing an accurate prediction of PDR regarding route diversion for a more generalized location. The performance of the selected ANN and SVR models was relatively poor when applied to the training location. The predictions from both models were separated according to the level of congestion in the network in order to determine if the performance of these models differed at different network congestion levels. The results show that, when the network congestion level was high, the ANN model produced better predictions for the testing location than it did when congestion was low or moderate.

7.3 Recommendations for Future Research

Although this research has achieved the proposed goals and objectives, much remains to be further studied. Some areas for future research include the following:

1. Identify additional critical variables that may impact route diversion decisions. It was found that the selected ANN and SVR models did not demonstrate consistent performance between the training and testing locations. The performance of the models at the training location far exceeded that at the testing location. One probable explanation for this significant performance difference is that, due to the limited incident locations selected for this study, many changes among the alternate route features were not reflected in the training input for the models. These features may represent additional critical variables that should be included in the training data. Therefore, future efforts should be made to identify additional critical variables.
2. Design a hypothetical network to serve as the model training and testing network. The ability to quickly change the geometric and traffic conditions of this hypothetical network is paramount. The design of such a network can greatly facilitate simulating a variety of incident scenarios at different locations in a network and at different locations in different networks. Through such a network, many non-relevant network features can be simplified or eliminated, and the simulation duration for each incident scenario can be shortened. This advantage is especially important in solving route diversion problems when a great number of scenarios at various different locations are required.

3. Explore the possibility of developing ANN and SVR models for different network congestion levels. It was found in this research that the ANN model produced better predictions for incident scenarios when the level of congestion in the network was high. With proper adjustment, the ANN model should be applicable in the real world. This finding presents the possibility of developing different models for different levels of congestion to improve the performance of these models when used for more general locations.
4. Explore more methods to improve the performance of ANN and SVR models. As mentioned in this dissertation, over-fitting is a common problem associated with ANN models. This over-fitting might have partially degraded the performance of the selected ANN model at the testing location. More research should be carried out in the future to find better solutions to this problem. For SVR models, efforts should be made toward performing other studies that recommend methods of choosing appropriate parameters for the model.

REFERENCES

- Abdel-Rahim, A., "Real-Time Estimation of Freeway Diversion Route Plans in Integrated Incident Management System", *Proceedings of the Eighth International Conference on Application of Advanced Technologies in Transportation Engineering*, ASCE, pp. 159-166, 2002.
- Adeli, H., *An Intelligent Decision Support System for Work Zone Traffic Management and Planning*, Ohio Dept. of Transportation, Columbus, OH, 2004.
- Allen, R D., D. Ziedman, T. J. Rosenthal, A. C. Stein, J. F. Torres, and A. Halati, "Laboratory Assessment of Driver Route Diversion in Response to In-Vehicle Navigation and Motorist Information Systems", *Transportation Research Record 1306*, Transportation Research Board, Washington, DC, pp. 82-91, 1990.
- Anderson, D, G. McNeil, *Artificial Neural Network Technology*, Contract Number F30602-89-C-0082, Prepared for Rome Laboratory, Griffiss Business Park, Rome, NY, August, 1992.
- Berenji, H. R., and Y. Wang, "Case-Based Reasoning for Fault Diagnosis and Prognosis", *2006 IEEE International Conference on Fuzzy Systems*, pp.1316-1321, 2006.
- Bhattacharjee, D., K. C. Sinha, and J. V. Krogmeier, "Real-Time Freeway O-D Prediction Algorithm Under ATIS Environment", *Proceedings of the 5th International Conference on Application of Advanced Technologies in Transportation*, ASCE, pp. 147-159, 1998.
- Bhavsar, P., M. Chowdhury, A. Sadek, W. Sarasua and J. Ogle, "A Decision Support System for Predicting Traffic Diversion Impacts across Transportation Networks using Support Vector Regression", *Transportation Research Board 86th Annual Meeting CD-ROM*, 2007.
- Boxill, S. A. and L. Yu, *An Evaluation of Traffic Simulation Models for Supporting ITS Development*, Report No. SWUTC/00/167602-1, Center for Transportation Training and Research, Texas Southern University, 2000.
- Burghout, W., *Mesoscopic Simulation Models for Short-Term Prediction*, Centre for Traffic Research, Institute of Technology in Linköpings University, 2005.
- Chatterjee, K., N. B. Hounsell, P. E. Firmin, and P. W. Bonsall, "Driver Response to Variable Message Sign Information in London", *Transportation Research. Part C: Emerging Technologies*, 10(2), pp. 149-169, 2002.

- Chauhan, A. S., "Development and Evaluation of Diversion Strategies under Incident Response using Dynamic Traffic Assignment System", *Master's Thesis*, Submitted to Department of Civil and Environmental Engineering, Massachusetts Institute of Technology (MIT), 2003.
- Chen, S. T. and H. S. Mahmassani, *Dynamics of Urban Commuter Behavior Under Real-Time Traffic Information*, Report No. SWUTC/99/472840-00066-1, Texas Department of Transportation (TxDOT), 1999.
- Cherkassky, V. and Y. Ma, "Practical selection of SVM Parameters and Noise Estimation for SVM Regression", *Neural Networks*, Volume 17, Issue 1, January 2004.
- Chu, L., H. X. Liu, and W. Recher, "Evaluation of Potential ITS Strategies under Non-recurrent Congestion Using Microscopic Simulation", *Transportation Research Board 83th Annual Meeting CD-ROM*, Paper #04-5201, 2004.
- Cragg, C. A. and M. J. Demetsky, *Simulation Analysis of Route Diversion Strategies for Freeway Incident Management*, Report No. VTRC 95-R11, Virginia Department of Transportation (VDOT), 1995.
- Dahlgren, J., A. Khattak, P. McDonough, I. Banerjee, P. Orrick, and A. Sharafsaleh, *ITS Decision Enhancements: Developing Case-Based Reasoning and Expert Systems and Incorporating New Material*, California PATH Research Report, UCB-ITS-PRR-2004-41, 2004.
- Demuth, H. and M. Beale, *Neural Network Toolbox Matlab User's Manual 3.0*, 1998.
- Dharia, A. and H. Adeli, "Neural Network Model for Rapid Forecasting of Freeway Link Travel Time", *Engineering Application of Artificial Intelligence*, 16, pp. 607-613, 2003.
- Duneo, D., M. Jha, and M. Ben-Akiva, "An Evaluation of Integrated Freeway Traffic Control and Route Diversion using Microscopic Simulation", <http://web.mit.edu/its/papers/TCONTROL.PDF>, 2001, (Accessed on Dec. 7, 2006).
- Federal Highway Administration (FHWA), *Alternate Route Handbook*, Washington, DC, 2006.
- Federal Highway Administration (FHWA), *Traffic Incident Management Handbook*, Washington, DC, 2000.
- Federal Highway Administration (FHWA), *Freeway Incident Management Handbook*, Report No. FHWA-SA-91-056, July 1991.
- Florida Department of Transportation (FDOT), *SunGuideSM: Software Users Manual SunGuide-SUM-2.0.0*, Tallahassee, FL, 2005.

Gunn, S. R., *Support Vector Machine for Classification and Regression*, Technical Report, School of Electronics and Computer Science, University of Southampton, May 1998.

Haefner, L. E. and L. V. Dickinson, "Preliminary Analysis of Disaggregate Modeling in Route Diversion", *Transportation Research Record 0316-1981*, Transportation Research Board, Washington, DC, pp. 66-75, 1974.

Hawas, Y. E., and H. S. Mahmassani, "A Decentralized Scheme for Real-Time Route Guidance in Vehicular Traffic", *Proceedings of the Second World Congress on Intelligent Transportation Systems*, Yokohama, Japan, pp. 1956-1963, 1995.

Heaton, J., "Introduction to Neural Networks with Java", <http://www.heatonresearch.com/book/programming-neural-networks-java.html>, (Accessed on Jan. 9, 2008).

Horowitz, A. J., I. Weisser, and T. Notbohm, "Diversion from a Rural Work Zone Owing to a Traffic-Responsive Variable Message Signage System", *Transportation Research Record 1824*, Transportation Research Board, Washington, DC, pp. 23-28, 2003.

Houten, R. V. and J. E. L. Malenfant, *Evaluation of Changeable Message Signs (CMS) on I-4 at Exits 30A and 30B to Assign Ramp Traffic and at Princeton St. To Sign for Cultural Events*, Final Report for Contract Number: BC 354 RPWO #40, Florida Department of Transportation (FDOT), 2002.

Huang, X. P. and G. Xu, "A Neural Network Based Approach for Predicting Network Traffic Conditions", *Proceedings of the 1996 Annual Meeting of ITS America*, Houston, TX, pp. 507-516, 1996.

Information Builder website, <http://www.informationbuilders.com/decision-support-systems-dss.html> (Accessed on February 12, 2007).

Ishak, S., and F. Trifiro, "Neural Networks", *Transportation Research Circular Number E-C113*, Transportation Research Board, Washington, DC, pp. 17-32, 2007.

Johnson, R. A., D.W. Wichern, *Applied Multivariate Statistical Analysis, Fifth Edition*, Pearson Education, Inc., Pearson Prentice Hall, One Lake Street, Upper Saddle River, NJ 07485 USA and Dorling Kindersley Ltd., 80 Strand, London WC2R ORL, UK, pp. 427-461, 2006.

Khattak, A. and A. Kanafani, "Case-Based Reasoning: A Planning Tool for Intelligent Transportation Systems", *Transportation Research. Part C*, pp. 267-288, 1996.

Khattak, A., A. Kanafani, and E. L. Colletter, "Stated and Reported Route Diversion Behavior: Implications of Benefits of Advanced Traveler Information System",

Transportation Research Record 1464, Transportation Research Board, Washington, DC, pp. 28-35, 1994.

Khattak, A. J., J. L. Schofer, and F. S. Koppelman, "Factor Affecting Commuters' En Route Diversion Behavior in Response to Delay", *Transportation Research Record 1318*, Transportation Research Board, Washington, DC, pp. 125-136, 1991.

Lee, D-H., S-T. Jeng, and P. Chandrasekar, "Application of Data Mining in Developing Traffic Diversion Plans", *Proceedings of the 9th World Congress on Intelligent Transport Systems*, Chicago, IL, 2002.

Liu, H., H. V. Zuylen, H. V. Lint, and M. Salomons, "Urban Arterial Travel Time Prediction with State-Space Neural Networks and Kalman Filters", *Transportation Research Board 85th Annual Meeting CD-ROM*, 2006.

Mahmassani, H. S. and H. Sbayti, *DYNASMART-P Version 1.2 User's Guide*, Maryland Transportation Initiative, University of Maryland, 2005.

Madanat, S. M., C. Y. Yang, and Y. M. Yen, "Analysis of Stated Route Diversion Intentions Under Advanced Traveler Information System Using Latent Variable Modeling", *Transportation Research Record 1485*, Transportation Research Board, Washington, DC, pp. 10-17, 1995.

Mitchell, T. M., *Machine Learning*, The McGraw-Hill Companies, Inc., 1997.

National ITS Architecture website, <http://www.iteris.com/itsarch/html/glossary/glossary.htm> (Accessed on October 22, 2006).

Pal, R., "Investigation on Latent Factors Affecting Route Diversion Intentions", *ASCE Journal of Transportation Engineering*, ASCE, 124(4), pp. 362-367, 1998.

Park, D. and L. R. Rilett, "Forecasting Freeway Link Travel Times with a Multilayer Feedforward Neural Network", *Computer Aided Civil and Infrastructure Engineering*, 14, ASCE, pp. 356-367, 1999.

Peeta, S. and J. W. Yu, "Data-Consistent Fuzzy Approach for Online Driver Behavior under Information Provision", *Transportation Research Record 1803*, Transportation Research Board, Washington, DC, pp. 76-86, 2002.

Peeta, S., J. L. Ramos, and R. Pasupathy, "Content of Variable Message Signs and On-line Driver Behavior", *Transportation Research Board 79th Annual Meeting CD-ROM*, 2000.

- Rao, K. S., S. P. Vengular, R. Sarma, and V. Bogarapu, *Real-Time Assessment and Use of Arterial Street Capacity for Freeway Diversion Routing*, Report No. TX-97/2978-1F, Texas Department of Transportation (TxDOT), 1996.
- Sadek, A. W., M. J. Demetsky, and B. L. Smith, "Case-Based Reasoning for Real-Time Traffic Flow Management", *Computer-Aided Civil and Infrastructure Engineering*, 14, ASCE, pp. 347-356, 1999.
- Sadek, A. W., G. Spring, and B. L. Smith, "Toward More Effective Transportation Applications of Computational Intelligence Paradigms", *Transportation Research Board 82th Annual Meeting CD-ROM*, 2003.
- Schirazi, E., S. Anderson, and J. Stesney, "Commuters' Attitudes Toward Traffic Information System and Route Diversion", *Transportation Research Record 1168*, Transportation Research Board, Washington, DC, pp. 9-15, 1988.
- Schrank, D., T. Lomax, *The 2007 Urban Mobility Report*, Texas Transportation Institute, The Texas A&M University System, 2007.
- Singh, A. K. and G. Abu-Lebdeh, *State Space Neural Networks for Travel Time Predictions in Signalized Networks*, *Transportation Research Board 86th Annual Meeting CD-ROM*, 2007.
- Srinivasan, K. K. and A. Krishnamurthy, "Roles of Spatial and Temporal Factors in Variable Message Sign Effectiveness Under Nonrecurrent Congestion", *Transportation Research Record 1854*, No. 03-4483, Transportation Research Board, Washington, DC, pp. 124-134, 2003.
- StatSoft Electronic Textbook, <http://www.statsoft.com/textbook/stathome.html?stsvm.html&l>, (Accessed on December 10, 2006)
- Tao, Y., F. Yang, Z. Qiu, and B. Ran, "Travel Time Prediction in the Presence of Traffic Incidents Using Different Types of Neural Networks", *Transportation Research Board 85th Annual Meeting, CD-ROM*, 2006.
- Thomason, A., <http://www.csulb.edu/~athomaso/> (Accessed on February 10, 2007).
- Ullman, G., C. L. Dudek, and K. N. Balke, "Effect of Freeway Corridor Attributes on Motorist Diversion Responses to Travel Time Information", *Transportation Research Record 1464*, Transportation Research Board, Washington, DC, pp. 19-27, 1994.
- U.S. Department of Transportation, Intelligent Transportation Systems Joint Program Office, *Developing Traffic Signal Control Systems Using the National ITS Architecture*, Washington, DC, 1998.

U. S. Department of Transportation Intelligent Transportation System Website, <http://www.itsdocs.fhwa.dot.gov/edldocs/13480/appm.pdf>, (Accessed on April, 2007).

Vanajakshi, L. D., “Estimation and Prediction of Travel Time from Loop Detector Data for Intelligent Transportation System”, *Ph.D. Dissertation*, Texas A&M University, 2004.

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

Van Lint, J. W. C., S. P. Hoogendoorn, and H. J. Van Zuylen, “Accurate Freeway Travel Time Prediction with State-Space Neural Network under Missing Data”, *Transportation Research Part C*, 13, pp. 347-369, 2005.

Vaughn, K., R. Kitamura, and P. P. Jovanis, “Experimental Analysis and Modeling of Advice Compliance: Results from Advanced Traveler Information System Simulation Experiments”, *Transportation Research Record 1485*, Transportation Research Board, Washington, DC, pp. 18-26, 1995.

Vladimir, C. and Y. Ma, “Practical selection of SVM parameters and noise estimation for SVM regression”, *Neural Networks*, Volume 17, Issue No. 1, pp. 113-126, January 2004.

Wu, J. C., A. J. Horowitz, M. S. Hardy, and J. Corbin, “An Experimental Measurement of Diversion Owing to Ramp Meters”, *Transportation Research Board 2006 Annual Meeting CD-ROM*, 2006.

Wu, C-H., C-C. Wei, D-C. Su, M-H. Chang, and J-M. Ho, “Travel Time Prediction with Support Vector Regression”, *IEEE Transactions on Intelligent Transportation Systems*, Volume 5, No. 4, pp. 276 – 281, 2003.

Zhan, C., “Development of Prediction Models for Freeway Incident Durations using Data Mining Techniques”, *Ph.D. Dissertation*, Department of Civil and Environmental Engineering, Florida International University, 2006.

APPENDIX: MATLAB PROGRAM FOR ANN AND SVR MODEL TRAINING AND TESTING

1. Work Files Configuration

```
global WORK_PATH DATA_PATH DYNASMART_EXE INPUT_FILE  
OUTPUT_FILE ANN_FILE RESULT_FILE OUTLINK_FILE_1 OUTLINK_FILE_2
```

```
%path for M-files, input.xls, output.xls and network.m, last '\' is
```

```
%required, there must no space characters
```

```
WORK_PATH = 'E:\work\';
```

```
%path for planning.exe and files necessary to run planning.exe, last '\' is
```

```
%required, there must be no space characters
```

```
DATA_PATH = 'E:\Fort_Worth_Data_Set_1.2_in_use\';
```

```
%name of DYNASMART exe file
```

```
DYNASMART_EXE = 'planning.exe';
```

```
%name of input file
```

```
INPUT_FILE = 'input.xls';
```

```
%name of output file
```

```
OUTPUT_FILE = 'output.xls';
```

```
%name of ANN file
```

```
ANN_FILE = 'ann.mat';
```

```
%name of final result file
```

```
RESULT_FILE = 'ANN_result.xls';
```

```
%name of outlink file
```

```
OUTLINK_FILE = 'outlink.xls';
```

```
SVR_FILE = 'svr.mat';
```

```
SVR_RESULT_FILE = 'SVR_result.xls';
```

2. Automatically Running of DYNASMART-P for Different Incident Scenarios

```
%run DynaSmart to generate travel times
```

```
% %global setting
```

```

config;

%read input cofigure file
run_cmd = ['cd ' WORK_PATH];
eval(run_cmd);

input_v = xlsread(INPUT_FILE_BASECASE);

%for each input record:
%1. write input file (incident.dat, scenario.dat, VMS.dat)
%2. run planning.exe
%3. read output file (SummaryStat.dat)
travel_times = [];
outlink = [];
[row_input_v,n] = size(input_v);
for i = 1:row_input_v

    input_current_running = i
    total_inputs = row_input_v

    %write incident.dat
    fid = fopen(strcat(DATA_PATH, 'incident.dat'),'w');
    fprintf(fid,'%d\r\n%d\t%d\t%.1f\t%.1f\t%.1f\r\n',input_v(i,1),...
        input_v(i,2), input_v(i,3), input_v(i,4), input_v(i,5), ...
        input_v(i,6));
    fclose(fid);

    %write scenario.dat
    fid = fopen(strcat(DATA_PATH, 'scenario.dat'),'w');
    fprintf(fid,['%.2f %.2f\t%d   %d   %d\r\n' ...
        '%.2f\r\n' ...
        ' %d\r\n' ...
        ' %d %d\r\n' ...
        '%.2f %.2f\r\n' ...
        ' %d\r\n' ...
        ' %d %d %.3f %d %.3f %.3f %.3f %.3f %.3f\r\n'], ...
        input_v(i,7), input_v(i,8), input_v(i,9), input_v(i,10), input_v(i,11), ...
        input_v(i,12), input_v(i,13), input_v(i,14), input_v(i,15), input_v(i,16), ...
        input_v(i,17), input_v(i,18), input_v(i,19), input_v(i,20), input_v(i,21), ...
        input_v(i,22), input_v(i,23), input_v(i,24), input_v(i,25), input_v(i,26),
        input_v(i,27));
    fclose(fid);

    %write VMS.dat
    fid = fopen(strcat(DATA_PATH, 'VMS.dat'),'w');

```

```

fprintf(fid,['%d\r\n' ...
    '\t%d\t%d\t%d\t%d\t%d\t%d\t%d\r\n'], ...
    input_v(i,28), input_v(i,29), input_v(i,30), ...
    input_v(i,31), input_v(i,32), input_v(i,33), input_v(i,34), input_v(i,35));
fclose(fid);

fid = fopen(strcat(DATA_PATH, 'VMS.dat'),'a');
for m = 36: (36 + input_v(i, 33) - 1)
    fprintf(fid, '\t%d', input_v(i,m));
end
fprintf(fid, '\r\n');
fclose(fid);

%run planning.exe
run_cmd = ['cd ' DATA_PATH];
eval(run_cmd);
run_cmd = strcat('!', DYNASMART_EXE);
eval(run_cmd);

%read output file (SummaryStat.dat)
fid = fopen(strcat(DATA_PATH, 'SummaryStat.dat'), 'r');
while feof(fid) == 0
    tline = fgetl(fid);
    matches = strfind(tline, 'AVERAGE TRIP TIMES (INCLUDING ENTRY QUEUE
TIME) (MINS)');
    num = length(matches);
    if num > 0
        break;
    end
end
average_trip_time_line = fgetl(fid);
average_trip_time = sscanf(average_trip_time_line, '%*s %*s %f');
travel_times(i, 1) = average_trip_time;

%For OutLinkQue.dat, OutLinkSpeedAll.dat, OutLinkSpeedFree.dat, OutFlow.dat
%Please take the following numbers from table 16-25
%the 99th,105th,36th,364th number.
%Note: Use the difference of outLinkSpeedFree and OutLinkSpeedAll as the input
(OutLinkSpeedFree-OutLinkSpeedAll)
%2. For OutLeftFlow.dat, please take the following numbers from table 16-25
%the 36th and the 364th.

%Read OutLink files
run_cmd = ['cd ' WORK_PATH];
eval(run_cmd);

```

```

%queue = ReadOutlinkFile(strcat(DATA_PATH, 'OutLinkQue.dat'), [16:25], [99 105
36 364]);
%speed_all = ReadOutlinkFile(strcat(DATA_PATH, 'OutLinkSpeedAll.dat'), [16:25],
[99 105 36 364]);
%speed_free = ReadOutlinkFile(strcat(DATA_PATH, 'OutLinkSpeedFree.dat'),
[16:25], [99 105 36 364]);
%out_flow = ReadOutlinkFile(strcat(DATA_PATH, 'OutFlow.dat'), [16:25], [99 105
36 364]);
%out_left_flow = ReadOutlinkFile(strcat(DATA_PATH, 'OutLeftFlow.dat'), [16:25],
[36 364]);
queue = ReadOutlinkFile(strcat(DATA_PATH, 'OutLinkQue.dat'), [13:17], [86]);
speed_all = ReadOutlinkFile(strcat(DATA_PATH, 'OutLinkSpeedAll.dat'), [13:17],
[86]);
speed_free = ReadOutlinkFile(strcat(DATA_PATH, 'OutLinkSpeedFree.dat'), [13:17],
[86]);
out_flow = ReadOutlinkFile(strcat(DATA_PATH, 'OutFlow.dat'), [13:17], [86 80
92]);

```

```

%outlink = [outlink; queue, speed_free - speed_all, out_flow, out_left_flow];
outlink = [outlink; queue, speed_free - speed_all, out_flow];

```

```

%write output result file
run_cmd = ['cd ' WORK_PATH];
eval(run_cmd);
xlswrite(OUTPUT_FILE, travel_times);

```

```

%write outlink file
run_cmd = ['cd ' WORK_PATH];
eval(run_cmd);
[num_obs, num_outlink_var] = size(outlink);
num_sheets = ceil(num_outlink_var/256) + 1;

```

```

run_cmd = ['delete ' OUTLINK_FILE];
eval(run_cmd);
xlswrite(OUTLINK_FILE, num_sheets, 1, 'A1'); %record the number of sheets used
for i = 0: (num_sheets -2)
    if(i == num_sheets -2)
        xlswrite(OUTLINK_FILE, outlink(:, (1 + i*256):end ), i+2);
    else
        xlswrite(OUTLINK_FILE, outlink(:, (1 + i*256):(256 + i*256) ), i+2);
    end
end
end

```

end

3. Reading Output File from the DYNASMART-P Simulation

```
function value = ReadOutlinkFile(file, matrix_ids, value_ids)

num_matrix_row = 44;

%sort matrix_ids
sort(matrix_ids, 2);

value = [];
[r,num_matrix] = size(matrix_ids);

fid = fopen(file, 'r');

for( i = 1:num_matrix)

    mid = matrix_ids(i);

    %create the string used to locate the matrix_id-th matrix
    if(mid < 10)
        midstr = [' ', int2str(mid), '.0'];
    elseif(mid >= 10 && mid < 100)
        midstr = [' ', int2str(mid), '.0'];
    elseif(mid >= 100 && mid < 1000)
        midstr = [' ', int2str(mid), '.0'];
    end

    while feof(fid) == 0
        tline = fgetl(fid);

        if(1 == strcmp(tline, midstr)) % a matrix located
            matrix = [];
            for( j = 1 : num_matrix_row)
                tline = fgetl(fid);
                row = sscanf(tline, '%f%f%f%f%f%f%f%f%f');
                matrix = [matrix, row'];
            end
            tline = fgetl(fid);
            row = sscanf(tline, '%f%f%f%f%f');
            matrix = [matrix, row'];

            value = [value, matrix(value_ids)];
```



```

        break;

    end %if

end %while

end %for

fclose(fid);

end

```

4. Data Preparation

```

function [train_input, train_target, test_input, test_target] = prepare_data()
%prepare_data prepares the data for train and test input and target
%

%global setting
config;

%Read input and output data
run_cmd = ['cd ' WORK_PATH];
eval(run_cmd);

input = xlsread(INPUT_FILE);
output = xlsread(OUTPUT_FILE);

outlink = [];
num_sheets = xlsread(OUTLINK_FILE, 1); %record the number of sheets used
for i = 2: (num_sheets)
    outlink = [outlink, xlsread(OUTLINK_FILE, i)];
end

%get input and target for training and testing
input_size = size(input);
num_input_v = input_size(1, 1);
num_total_input = input_size(1, 2);

%input = [outlink(1:num_total_input,:), input(:, 5) - input(:,4), input(:, 6), input(:, 27),
input(:, 35) - input(:,4)]';
input = [outlink(1:num_total_input,:), input(:, 5) - input(:,4), input(:, 6), input(:, 27),
input(:, 35) - input(:, 4)]';

```

```
output = output(1:num_total_input,:);
```

```
%generate the training and testing matrixes randomly  
num_train = num_total_input*2/3;  
rand_nums = randperm(num_total_input);  
train_input = input(:, rand_nums(1:num_train));  
train_target = output(:, rand_nums(1:num_train));  
test_input = input(:, rand_nums(num_train + 1 : end));  
test_target = output(:, rand_nums(num_train + 1 : end));  
  
end
```

5. Training ANN

```
%-----simulate with ANN-----%
```

```
%global setting  
config;
```

```
%prepare data for train and test input and target  
[train_input, train_target, test_input, test_target] = prepare_data();  
train_target_size = size(train_target);  
num_output = train_target_size(1,1);
```

```
%principal component analysis %%%%%%%%%%need further study  
[pn,meanp,stdp,tn,meant,stdt] = prestd(train_input,train_target);  
[ptrans,transMat] = prepca(pn,0.02);
```

```
input_range = minmax(ptrans);  
input_range_size = size(input_range);  
input_num = input_range_size(1,1);
```

```
%create and initialize net  
net = newff(input_range, [9, num_output], {'logsig','purelin'},'traingdx');  
net = init(net);
```

```
%train net  
net.performFcn = 'sse';  
net.trainParam.show = 50;  
net.trainParam.lr = 0.05;
```

```

net.trainParam.lr_inc = 1.05;
net.trainParam.epochs = 10000;
net.trainParam.goal = 1e-2;

[net,tr]=train(net,ptrans,tn);

save(ANN_FILE, 'net');

%Post-Training Analysis

%sim with test data
[pnewn] = trstd(test_input,meanp,stdp);
[pnewtrans] = trapca(pnewn,transMat);
a = sim(net,pnewtrans);

%post standard deviation
[anew] = poststd(a,meant,stdt);

%measure net's performance
[m,b,r] = postreg(anew,test_target)
e=a-test_target
perf = sse(e)
%save results
results = [m; b; r];
result_cmp = [test_target', a', test_input'];
xlswrite(RESULT_FILE, results, 'Sheet1');
xlswrite(RESULT_FILE, result_cmp, 'Sheet2');

```

6. Testing ANN

```

%-----Calculate with ANN-----%

%global setting
config;
%load ANN trained
run_cmd = ['cd ' WORK_PATH];
eval(run_cmd);
load(ANN_FILE);

%prepare data for train and test input and target
[train_input, train_target, test_input, test_target] = prepare_data();
test_input = [train_input, test_input];
test_target = [train_target, test_target];

```

```

%sim with test data
%a = sim(net,test_input);
[pnewn] = trstd(test_input,meanp,stdp);
[pnewtrans] = trapca(pnewn,transMat);
a = sim(net,pnewtrans);

%measure net's performance
[m,b,r] = postreg(a,test_target)

%save results
results = [m; b; r];
result_cmp = [test_target', a', test_input'];
xlswrite(RESULT_FILE, results, 'Sheet1');
xlswrite(RESULT_FILE, result_cmp, 'Sheet2');

```

7. Training ANN without CPA

```

%-----simulate with ANN-----%

%global setting
config;

%prepare data for train and test input and target
[train_input, train_target, test_input, test_target] = prepare_data();
train_target_size = size(train_target);
num_output = train_target_size(1,1);

%principal component analysis %%%%%%%%%%need further study
%[pn,meanp,stdp,tn,meant,stdt] = prestd(train_input,train_target);
%[ptrans,transMat] = prepca(pn,0.02);

input_range = minmax(train_input);
input_range_size = size(input_range);
input_num = input_range_size(1,1);

%create and initialize net
net = newff(input_range, [input_num * 2 + 1, num_output], {'tansig','purelin'}, 'traingdx');
net = init(net);

%train net
net.trainParam.show = 50;

```

```

net.trainParam.lr = 0.05;
net.trainParam.lr_inc = 1.05;
net.trainParam.epochs = 10000;
net.trainParam.goal = 1e-2;

[net,tr]=train(net,train_input,train_target);

save(ANN_FILE, 'net');

%Post-Training Analysis

%sim with test data
%[pnewn] = trstd(test_input,meanp,stdp);
%[pnewtrans] = trapca(pnewn,transMat);
a = sim(net,test_input);

%post standard deviation
%[anew] = poststd(a,meant,stdt);

%measure net's performance
[m,b,r] = postreg(a,test_target)

%save results
results = [m; b; r];
result_cmp = [test_target', a', test_input'];
xlswrite(RESULT_FILE, results, 'Sheet1');
xlswrite(RESULT_FILE, result_cmp, 'Sheet2');

```

8. Training SVR

```

%-----simulate with SVC-----%

%global setting
config;
% Values for kernel function:
%      'linear' -
%      'poly'  - p1 is degree of polynomial
%      'rbf'   - p1 is width of rbf (sigma)
%      'sigmoid' - p1 is scale, p2 is offset
%      'spline' -
%      'bspline' - p1 is degree of bspline
%      'fourier' - p1 is degree
%      'erfb'  - p1 is width of rbf (sigma)
%      'anova' - p1 is max order of terms

```

```

KER_FUNC = 'erbf';
p1 = 1;
percent_match = 0;
%prepare data for train and test input and target
[train_input, train_target, test_input, test_target] = prepare_data();
train_target_size = size(train_target);
num_output = train_target_size(1,1);

train_input = train_input';
train_target = train_target';
test_input = test_input';
test_target = test_target';
ker = KER_FUNC;
%C =
[nsv beta bias] = svr(train_input,train_target,ker);
%err = svrerror(train_input,train_target,test_input,test_target,ker,beta,bias);
tstY = svroutput(train_input,test_input,ker,beta,bias);
[num_test, y] = size(test_target);
num_match = 0;
for i = 1:num_test
    if ((test_target(i, 1) > 0 && tstY(i, 1) > 0) || (test_target(i, 1) < 0 && tstY(i, 1) < 0))
        num_match = num_match + 1;
    end
end
if num_match/num_test > percent_match
    percent_match = num_match/num_test;
%svplot(train_input,train_target,ker,alpha,bias,0,1,2,test_input)
save('svr');

result_cmp = [test_target, tstY, test_input];
xlswrite(SVR_RESULT_FILE, result_cmp, 'Sheet1');
backupSVR;
end

```

9. Testing SVR

```

%-----Calculate with SVR-----%

%global setting
config;

%load SVR trained
run_cmd = ['cd ' WORK_PATH];

```

```

eval(run_cmd);
load(SVR_FILE);

%prepare data for train and test input and target
[train_input, train_target, test_input, test_target] = prepare_data();
test_input = [train_input, test_input];
test_target = [train_target, test_target];
test_input = test_input';
test_target = test_target';

%test
tstY = svroutput(train_input',test_input,ker,beta,bias);
[num_test, y] = size(test_target);
num_match = 0;
for i = 1:num_test
    if ((test_target(i, 1) > 0 && tstY(i, 1) > 0) || (test_target(i, 1) < 0 && tstY(i, 1) < 0))
        num_match = num_match + 1;
    end
end
percent_match = num_match/num_test

%save results
result_cmp = [test_target, tstY, test_input];
xlswrite(SVR_RESULT_FILE, result_cmp, 'Sheet1');

```

VITA

JIE BIAN

Mar. 1977 Born, Hebei Province, P. R. China

EDUCATION

Jul. 1999 B.S., Industrial Engineering and Management
University of Petroleum
Beijing, P. R. China

Jan. 2003 M.S., Industrial Engineering and Management
University of Petroleum
Beijing, P. R. China

Mar. 2008 Doctoral Candidate, Civil Engineering
Florida International University
Miami, FL

EMPLOYMENT

2000 - 2003 Teaching Assistant
University of Petroleum
Beijing, P. R. China

2004 - 2005 Graduate Teaching Assistant and Instructor
Florida International University
Miami, FL

2006 - 2007 Graduate Research Assistant
Florida International University
Miami, FL

2007 - 2008 Transportation Analyst
Cambridge Systematics
Ft. Lauderdale, FL

AFFILIATIONS

- Student member, Chi Epsilon Honor Society, 2003–Present
- Student member, Institute of Transportation Engineers, 2003–Present

PUBLICATIONS AND PRESENTATIONS

- Bian, J., “A Decision Support System for Real-Time Freeway Traffic Diversion,” Accepted for presentation at the 2008 Freeway and Tollway Operations Conference, Ft. Lauderdale, Florida, June 15-19, 2008.
- Zhan, C. J., P. Zhu, J. Bian, A. Gan, W. Y. Wu, and J. Rodriguez, “A CORSIM Simulation Testbed for Freeway Ramp Metering Algorithms,” Accepted for

Presentation at the 10th International Conference on Application of Advanced Technologies in Transportation, ASCE, Athens, Greece, May 27-31, 2008.

- Bian, J., P. Zhu, A. Gan, and K. Y. Liu, “Analysis of Traffic Crashes on Five Freeways in Miami-Dade County,” Final Report Submitted to District 6 ITS Office, Florida DOT, April 2007.
- Gan, A., C. J. Zhan, P. Zhu, and J. Bian, “Simulation Analysis of I-95 Ramp Metering Project in Miami-Dade County: CORSIM Models,” Final Report Submitted to District 6 ITS Office, Florida DOT, February 2007.

HONORS AND AWARDS

- Best Student Chapter Presentation Award (with Mr. Peng Zhu), District 10 Institute of Transportation Engineers, 2006
- Presidential Fellowship, Florida International University, 2003-2006
- First Place, ITE Student Chapter Presentation Competition, Fall 2006
- Engineer in Training, 2005
- Best Student Chapter Award, International Institute Transportation Engineers, while serving as the Chapter’s Membership Chairperson, 2003-2004