ARTIFICIAL INTELLIGENCE TECHNIQUES FOR MODELING DYNAMIC TRAFFIC BEHAVIOR AT BOTTLENECKS

A Dissertation

Presented to

The Faculty of the Graduate School

At the University of Missouri

In Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

By

YI HOU

Dr. Carlos Sun and Dr. Praveen Edara, Dissertation Supervisor

DECEMBER 2014

The undersigned, appointed by the dean of the Graduate School,

have examined the dissertation entitled

ARTIFICIAL INTELLIGENCE TECHNIQUES FOR MODELING DYNAMIC TRAFFIC BEHAVIOR AT BOTTLENECKS

Presented by Yi Hou

A candidate for the degree of

Doctor of Philosophy

And hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Carlos Sun

Dr. Praveen K. Edara

Dr. Mark Virkler

Dr. Timothy Matisziw

ACKNOWLEDGEMENTS

I owe my deepest gratitude to my academic advisors, Dr. Carlos Sun and Dr. Praveen Edara, whose encouragement, guidance and support sustained me through the doctoral research in University of Missouri-Columbia. I am heartily thankful for their tremendous efforts, patience, and inspirations during my research. I also would like to express my gratitude to Dr. Mark Virkler and Dr. Timothy Matisziw for serving as my defense committee members.

I would like owe my thanks to my wife, my family, and my friends for companioning and supporting me thought the endeavor of my life. I cannot achieve what I have accomplished without any of you.

I am grateful to all my colleagues in Translab who ever encouraged and supported me during my graduate study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This dissertation applies artificial intelligence (AI) techniques to enhance the models of travel demand and traffic behavior at bottlenecks including natural lane reduction and work zone closure. AI models for accurately forecasting travel demand at work zone bottlenecks in urban areas were developed. Driving behavior models of lane changing at natural lane drops at freeway interchanges were proposed. Real-world datasets were used to develop and test the AI models.

The lane-changing models took into account factors such as gap acceptance in the target lane, vehicle speeds in the target lane, and distance to the end of the merge lane. Bayes classifier, classification tree, genetic fuzzy system, random forest, and AdaBoost were used to model the impact of these factors on driver lane-changing behavior. The models were built using traffic data collected by the Federal Highway Administration (FHWA) on a segment of southbound US Highway 101 in Los Angeles, California. To assess the quality of the models, they were tested on traffic data on Interstate 80 in San Francisco, California. The empirical results demonstrated superior performance of AI models over the conventional binary logit model. Random forest and AdaBoost yielded the highest prediction accuracies of 88.3% and 88.9%. The results also demonstrate that ensemble learning methods, such as random forest and Adaboost, produced even higher prediction accuracy than single classifiers.

Traffic forecast models are classified into two types based on the forecast horizon: daily, and short-term. None of numerous existing traffic flow forecasting models focus on work zone bottlenecks. Work zone bottlenecks create conditions that are different from both normal

operating conditions and incident conditions. Four models were developed for forecasting traffic flow for planned work zone events. Both daily and short-term traffic flow forecasting applications were investigated. Daily forecast involves forecasting 24 hours in advance using historical traffic data, and short-term forecasts involves forecasting 1 hour, 45 minutes, 30 minutes, and 15 minutes in advance using real-time temporal and spatial traffic data. Models were evaluated using data from work zone events on two types of roadways - a freeway, I-270, and a signalized arterial, MO-141, in St. Louis, Missouri. The results showed that the random forest model yielded the most accurate daily and short-term work zone traffic flow forecasts. For freeway data, the most influential variables were the latest interval's look-back traffic flows at the upstream, downstream and current locations. For arterial data, the most influential variables were the traffic flows from the three look-back intervals at the current location only.

CHAPTER 1: INTRODUCTION

1.1. What Is Artificial Intelligence?

Artificial intelligence (AI) attempts to understand and build intelligent entities that think and act rationally like humans for solving problems or making decisions (Russel and Norvig, 2003). According to Zuylen (2012), AI also means computer systems that demonstrate complex living-system like behaviors. For instance, they could mimic insect swarm, ant colony, microbiology, or a neural system.

AI is one of the newer sciences. McCulloch and Pitt (1943) conducted the first work generally known as AI in 1943. They designed a model of artificial neurons based on three sources: "knowledge of basic physiology and function of neurons in brains", "a formal analysis of proportional logic", and "Turing's theory of computation". They showed that a suitably defined network of connected neurons could learn and compute any computable function. It was not until 1956 that the name "artificial intelligence" was given to this new field at a conference held at Dartmouth College. At that time, AI was the only field to build machines that function autonomously in a complex environment. In the 1960s and 1970s, the efforts on AI research were mainly focused on the development of knowledge-based systems. In the 1980s, much of the work was done on improving neural networks. Neural networks were compared with the corresponding methodologies from statistics, pattern recognition, and machine learning. As a result of these developments, a new research field was created by the so-called data mining

techniques. In the 1990s, building AI methods based on rigorous mathematical theorems and solving real-world problems were the focus of the field. (Russel and Norvig, 2003; Sadek, 2007)

After development of more than half a century, AI's current applications span several domains including autonomous control, robotics, language understanding, and computer vision. Applications can be found in a variety of fields such as economics, manufacturing, engineering, and medicine (Zuylen, 2012). Accordingly to Sadek (2007), AI methods include two major categories – symbolic AI and computational intelligence. Symbolic AI concentrates on the development of knowledge-based systems that are capable of making decisions in a particular domain utilizing knowledge from a human expert. Computational intelligence consist of methods such as fuzzy system, neural network, and evolutionary computing. The difference of computational intelligence from symbolic AI is that the output is generated without using knowledge base such as rules, frames, or cases.

1.2. Advantages and Concerns of AI

AI provides numerous advantages in its applications in a wide range of fields. Among the most important of advantages are the following. First, AI is especially suitable for capturing the complex relationship among different variables in an environment of uncertainty. In many cases where uncertainty exists, direct mathematical relationships cannot be established. AI methods can overcome uncertainty by encapsulating the existing knowledge with uncertainty and probability inference theorems. Second, AI provides the advantage of permanency. The knowledge incorporated in an AI framework is valid as long as the problems are relevant or decision circumstances are not changed. AI's capability of learning enables it to further extend

the life span of its application. Third, AI has demonstrated very high reliability in a variety of applications due to its ability to mimic human thinking and behavior. AI provides rational predictions or decisions with higher accuracy than traditional function fitting methods. Fourth, AI provides fast solutions to complex problems. By automating data gathering, processing and the decision-making process, AI supports faster decision making in complex situations. Finally, AI is capable of processing both qualitative and quantitative data. (Chowdhury and Sadek, 2012)

Nevertheless, like any other tools, there are some concerns for AI techniques. Chowdhury and Sadek (2012) discuss two major concerns of AI. First, some AI paradigms lack good interpretations of the models and are usually seen as a "black box" approach that simply represent the relationship between independent and dependent variables based on training data. One solution they propose to alleviate this concern is to build hybrid models by combining multiple AI paradigms or coupling AI with traditional approaches. Second, selecting the best value for parameters in AI models is more of an art than science. For example, researchers using genetic algorithms need to make some important decisions about the population size, the number of generations, mutation and crossover ratio. A trial-and-error procedure is the common approach used by researchers to determine these parameter values.

1.3. Why Use AI in Transportation?

With the increase of travel demand of both passenger and freight in transportation network driven by the fast economic growth, transportation professionals are facing more challenges to meet the goal of facilitating safe, efficient and reliable transportation with minimum impact on environment. Some of the most critical issues are urban congestion, traffic

safety, environmental pollution and wasted energy. The fact that transportation systems are inherently complex systems involving conflicts and interactions among a large number of diverse individuals adds more difficulty for transportation professionals to solving these problems. In recent years, both researchers and practitioners in the transportation field has grown interest in seeking the possibility of utilizing AI tools and methods to address transportation problems that are difficult or impossible to solve using traditional and classical approaches.

Sadek (2007) and Chowdhury and Sadek (2012) contend that transportation problems have some inherent features that make AI methods particularly suitable for solving them. First, in transportation problems, both qualitative and quantitative data are often involved. AI is an obvious choice when dealing with qualitative data like the knowledge provided by a human expert. Human behaviors and interactions in transportation systems are hard to understand and model with traditional approaches, but AI methods such as fuzzy systems and agent-based modeling are suitable to simulate human behaviors and interactions based on knowledge of a human expert or observed data. Second, transportation systems are complex in general and the complex relationships among different variables are difficult to be accurately modeled using traditional methods. AI methods do not require restrictive assumptions about the mathematical forms of the relationships a priori, and they are capable of capturing the non-linear nature of transportation problems. Third, imprecision and uncertainty stem from the human behavior which is fundamental to a transportation system. Traditional methods cannot effectively deal with uncertainties, but AI methods have been shown to make accurate predictions in an environment of uncertainty. Finally, in many intelligent transportation systems, real-time sensing, detection, response and control are important. Thus, fast and reliable decision making are the key

to these applications. AI can provide decision-support tools and advanced surveillance and communication ability for fast and reliable decision making.

AI approaches are currently applied to a wide range of transportation problems. These include:

- Traffic signal timing and optimization: genetic algorithms (Foy et al., 1992; Park et al., 1999; Park et al., 2000; Ceylan and Bell, 2004; Sun et al., 2003; Lee et al., 2005), fuzzy logic control (Pappis and Mamdani, 1977; Chiu and Chand, 1993; Niittymaki, and Pursula, 2000; Zhang, 2005; Trabia et al., 1999; Murat and Gedizlioglu, 2005), artificial neural network control (Nakatsuji and Kaku, 1991; Gilmore and Abe, 1995; Hua and Faghri, 1995; Saito and Fan, 2000; List and Cetin, 2004), reinforcement learning (Choy et al. 2003; Srinivasan and Choy, 2006)

- Short-term traffic and travel time prediction: artificial neural networks (Clark et al., 1993; Vythoulkas, 1993; Smith and Demetsky, 1994; Chang and Su, 1995; Gilmore and Abe, 1995; Dougherty and Cobbet, 1997; Ledoux, 1997; Innamaa, 2000; Florio and Mussone, 1996; Zhang, 2000; Vlahogianni et al. 2005), fuzzy logic (Huisken, 2003; Coufal and Turunen, 2004; Li et al., 2006), Bayesian network (Sun et al., 2006; Sun and Xu, 2011)

- Agent-based modeling and simulation: population-based incremental learning (PBIL) (Sukthankar et al., 1998), artificial neural network (Abbas et al., 2010)

- Travel demand forecasting: fuzzy logic (Teodorović and Kikuchi, 1991; Lotan and Koutsopoulos, 1993; Henn, 2000), genetic algorithm (Sakai et al., 1996; Nakayama and Kitamura, 2000), artificial neural networks (Reggiani, A., and O. Tritapepe,

1998; Nijkamp et al., 1996; Schintler and Olurotimi, 1998; Kim, 2001; Mozolin et al, 2000)

- Transportation safety analysis: Markov chain Monte Carlo (Miaou et al., 2003; Brijs et al., 2007; Miranda–Moreno, 2007), artificial neural networks (Awad and Janson, 1998), Bayesian neural networks (Xie et al., 2007), Classification and regression tree (Chang and Wang, 2006)

1.4. Dissertation Objectives

Due to the increasing passenger and freight demands on urban transportation networks, urban traffic will become more congested at bottlenecks including natural lane reduction and work zone closure, and transportation management centers (TMCs) will be tasked with resolving such challenges in the coming decades. Traffic congestions cause travel time delays, accidents, and air pollution, and they indirectly cause wasted energy as well as economic losses. In the last decade, bottleneck traffic congestion has received increasing attention around the world. For example, work zones with lane drops significantly impact roadway capacity and traffic patterns. According to USDOE (2002), work zones account for approximately 24% of the non-recurring delay. To alleviate bottleneck congestion, TMCs must plan and proactively manage traffic operations.

Due to the complexity of vehicle interaction at bottlenecks, traditional tools based on analytical methods cannot solve the bottleneck congestion problem. Microscopic simulation of roadway traffic is being increasingly used by transportation agencies for evaluations of various traffic control alternatives to determine optimum solutions. Several microscopic traffic

simulation tools have been developed, such as PARAMICS, MITSIM, CORSIM and VISSIM. However, several experiences showed that the current simulation models yield unsatisfactory performance under saturated and oversaturated traffic conditions that occur at bottlenecks. Some studies showed that simulations underestimate bottleneck capacity and over-predict congestion (DYMO, 1999; Abdulhai et al., 1999). The main cause for the weakness of microscopic simulation models is the limited understanding of driver behavior at bottlenecks, especially driver lane-changing behaviors.

Lane changing at bottlenecks is a crucial component of microscopic traffic simulation that involves a high level of interaction between vehicles; therefore, drivers need to react and to make decision based on their assessment of the surrounding traffic environment. This interaction involves a complex decision-making process. The major focus of research in lane-changing behavior has been on gap acceptance modeling. However, they are still not accurate enough to represent driver lane-changing behavior realistically at bottlenecks under congested conditions. Also, their applicability is limited by the difficulty to obtain true values of critical gaps. Thus, more accurate lane-changing models for simulating congested traffic at bottlenecks are needed.

In recent years, intelligent transportation systems (ITS) have been deployed at work zone bottlenecks to improve traffic flow and safety. Traffic Management Centers (TMC) have deployed ITS technologies such as dynamic message signs, variable speed limits, and queue warning systems (Lee and Kim, 2006; Edara et al., 2014). Accurate traffic flow forecasts are necessary for the scheduling and operation of work zones. Scheduling tools use traffic flow during different times of day as input in determining the least impactful times for closing lanes. Daily volume forecasts are usually sufficient for work zone scheduling applications. The use of ITS in work zones for active traffic control, not just traveler information, has increased in recent

years. An example is the use of variable speed limit (VSL) systems that change speed limits based on traffic flow inside and upstream from a work zone. Such a system operates in real-time and relies on traffic flow data measured from traffic sensors. A proactive VSL system that can anticipate traffic flow conditions in the near future and adjust speed limits before the flow deteriorates is more valuable than a reactive system that relies on past flow values alone. Ramp metering, hard shoulder running, and other applications are also being deployed at work zone bottlenecks and can all benefit from accurate short-term traffic flow forecasts. There is an abundance of traffic flow forecasting models from previous research. However, few studies have focused on the impact of dynamic variation of demand and capacity resulting from work zone bottlenecks. The presence of a work zone and its characteristics such as the type of work, number of closed lanes, reduced lane width, and other variables not only affect roadway capacity but also travel demand. In order to avoid work zone bottleneck related congestion by taking proactive traffic management strategies, work zone bottleneck traffic need to be accurately forecasted.

The objective of this dissertation is to explore the applications of AI methods in modeling dynamic traffic behavior at bottlenecks including natural lane reduction and work zone closure in order to solve the aforementioned issues. AI methods have several advantages in the modeling process. The dynamic traffic behavior at bottlenecks involved both qualitative and quantitative data, and AI is suitable to dealing such mixed data. AI methods are useful for modeling human behavior which is essential in lane-changing behavior modeling. Human driving behavior is hard to understand and model with traditional approaches. AI methods are not subject to priori assumptions of conventional models, and can capture the complex non-linear nature of dynamic traffic behavior at bottlenecks. Imprecision and uncertainty are often associated with traffic

behavior at bottlenecks, and AI methods are capable of making accurate predictions under an environment of uncertainty.

Two types of models were developed in this dissertation. One is the lane-changing model that can reflect driver lane-changing behavior at bottlenecks. At any given instance, a driver approaching a natural lane reduction or work zone in the merge lane decides whether to merge or not based on relevant variables. This is essentially a binary classification problem. The output of the model is a class label of merge and non-merge. Five AI classification methods, Bayes classifier, classification tree, genetic fuzzy system, random forest and AdaBoost were used to model driver lane-changing behavior. Detailed trajectory data provided by the Federal Highway Administration's (FHWA) Next Generation Simulation (NGSIM) project (Federal Highway Administration, 2011) were used to train the models.

The other type of model is the traffic demand prediction model that can forecast traffic demand for work zone bottlenecks. The output of this model is a continuous value, therefore, it is a regression problem. Four methods were developed for forecasting traffic flow for planned work zone events. The four methods were: multilayer feedforward neural network, nonparametric regression, regression tree, and random forest. Both daily and hourly traffic flow forecasting models were investigated. Daily prediction allows TMCs to provide a traffic operation plan before a work zone is deployed. The short-term prediction allows real-time traffic control in work zones. While daily forecasts were made 24 hours in advance using historical traffic data, short-term forecasts were made 1 hour, 45 minutes, 30 minutes, and 15 minutes in advance using real-time temporal and spatial traffic data. The models here were evaluated using urban work zone data on two types of roadways - a freeway, I-270, and a signalized arterial, MO-141, in St. Louis, Missouri.

1.5. Dissertation Contributions

The main contributions of this dissertation are listed below. First, the AI lane-changing models were shown to be more accurate than conventional binary logit models. Microscopic traffic simulation using these AI models leads to more realistic traffic behavior at bottlenecks. Second, the lane-changing models presented in here were developed using traffic data collected under challenging traffic conditions: approaching congestion and congested condition. Thus, they enhance the performance of microscopic simulation models at bottlenecks under congested traffic conditions. A more accurate microscopic traffic simulation at bottlenecks leads to a more efficient evaluation of traffic control strategies. Third, the tree structure of classification trees helps traffic engineers to better and more easily understand driver lane-changing behavior at bottlenecks under congested traffic condition. Classification trees reflect human behavior with a few simple rules. They are easy to interpret and program into simulation models. Fourth, the dissertation research produced work zone bottleneck traffic demand prediction models using AI methods. In addition to time-lagged traffic volume data, work zone characteristics including work zone speed limit, work zone type, number of lane drops, work zone length and duration were taken into account in the modeling process. Work zone traffic demand prediction models enable TMCs to utilize proactive traffic management strategies to control work zone traffic in real-time and to mitigate work zone related congestions. Last, unlike some conventional traffic demand forecasting methods, which lack a good interpretation of the model, the prediction process of random forests can be interpreted by estimating predictor importance.

1.6. Dissertation Outline

In Chapter 2, an extensive literature of existing lane-changing models and traffic forecasting models is presented. The relevancy of previous literature to the current models developed in dissertation is discussed. AI methods developed for both lane-changing models and traffic forecasting models, including Bayes classifier, classification and regression tree, random forest, genetic fuzzy system, multilayer feedforward neural network, nearest neighbor nonparametric regression, and AdaBoost, are described in detail in Chapter 3. In Chapter 4, the experimental design and results are presented. First, driver lane-changing behavior is modeled using real-world vehicle trajectory data collected at a natural lane reduction bottleneck. Then, traffic demand forecast models for work zone bottlenecks are developed. In Chapter 5, conclusions and directions for future research are presented.

CHAPTER 2: LITERATURE REVIEW

An extensive literature review of both lane-changing models and traffic demand forecast models is presented in this section. Gap acceptance was the principal focus of research in modeling driver lane-changing behavior. These models were based on the assumption that a driver merges or makes a lane change when both the lead and the lag gaps in the target lane are acceptable. Time series analysis has traditionally been used for forecasting traffic flows. Another major focus of research in traffic flow prediction models was the use of neural network models.

2.1. Lane-Changing Models

Given the assumption on the distribution of critical lead and lag gap lengths, various gap acceptance models were developed in 1960s and 1970s. Herman and Weiss (1961) assumed an exponential distribution for critical gap, Drew et al. (1967) assumed lognormal distribution, and Miller (1972) assumed a normal distribution.

Daganzo (1981) modeled driver's merging from the minor leg of a stop controlled T-intersection to the major leg by proposing a probit model for panel data for estimation of the gap acceptance model parameters. The function form of the critical gap for driver n at time t is assumed as

$$G_n^{cr}(t) = G_n + \epsilon_n^{cr}(t), \qquad (2-1)$$

where G_n is the component of critical gap attributable to driver n and $\epsilon_n^{cr}(t)$ is the random term that varies across different gaps for a given driver as well as across different drivers. Both G_n and $\epsilon_n^{cr}(t)$ are assumed mutually independent and normal distributed. By varying the mean of distribution of G_n other factors that affect driver gap-acceptance behavior were able to be included in the model. However, this model was affected by the problem that non-negative estimated critical gap lengths were not guaranteed.

Mahmassani and Sheffi (1981) solved Daganzo's estimability problem by neglecting panel data formulation. The critical gap was assumed to have a normal distribution. The mean of critical gap were formulated as a function of explanatory variables that affect driver gap-acceptance behavior.

Gipps (1986) designed a lane-changing decision model that was implemented in a microscopic traffic simulator. The model were developed to model driver lane-changing behavior at a variety of traffic conditions such as traffic signals, obstructions, and the presence of heavy vehicles. Necessity, desirability, and safety were included as three major factors in the lance changing decision making process. Drivers may face conflicting goals in some driving condition. The model deterministically prioritized different goals. However, the model did not consider the inconsistency (a driver may behave differently under identical condition at different times) and heterogeneity (different drivers behave differently under identical condition) in driver behavior.

Kita (1993) modeled driver's merging behavior from freeway on-ramp using a logit model to estimate the gap acceptance model. The random utility was formulated as a function of explanatory variables that have impact on driver lane-changing behavior. Other than the gap

length, relative speed of the merging vehicle to the mainline vehicles and the remaining distance to the end of the merging lane were found to affect driver lane-changing behavior.

Yang and Koutsopoulos (1996) established a rule-based lane-changing model that is used in MITSIM. The model only applies for driver lane-changing behavior on freeways. In their model, lane change were classified as either mandatory lane changes (MLC) or discretionary lane changes (DLC). They employed a probabilistic framework instead of prioritizing deterministically when drivers face conflicting goals. For DLC drivers consider two factors, impatience and speed indifference, to determine whether it is necessary to perform DLC. A gap acceptance model was developed for both MLC and DLC. They claimed that only when both lead and lag gaps are acceptable a gap is acceptable. However, they did not propose a formal parameter estimation framework.

Ahmed et al. (1996) developed a general lane-changing model that capture driver lane-changing behavior under both MLC and DLC situation. Driver's decision making process for lane changing was described as a sequence of four steps: decision to make a lane changing, choice of a target lane, gap acceptance, and performing lane-changing maneuver. They employed discrete choice model to model the decision elements. Due to the difficulty of obtaining the indicator to differentiate the first step and fourth step, the utilities capturing these two steps cannot be uniquely identified. The parameters of the model were estimated only for merging from a freeway on-ramp by using the data collected from a site at Interstate 95 northbound near Baltimore Washington Parkway in 1983 (Smith, 1985). Since in this case drivers have already made decision to change to the adjacent lane, the decision process involves only two steps: gap acceptance and performing lance changing maneuver. The model followed the same rule as

stated in Yang and Koutsopoulos's (1996) model, only when both lead and lag gaps are acceptable a gap is acceptable.

Kita (1999) developed a game-theoretic lane-changing model. A two-person non-zero-sum non-cooperative game was developed to model the interaction of drivers in the target lane and the merging lane. In the model, the behavior of a pair of merging and through cars was described at the same time. Both merging and through cars attempted to make decisions that are best for themselves by predicting each other's action. This study concentrated on merging-giveaway conflicts in on-ramp merging section. In the game, the payoffs for either merging and through car are supposed to be based only on its position and speed to the surrounding cars. This model has the advantage of building a simpler model by separating the direct and indirect impacts.

Hidas (2005) used intelligent-agent-based techniques to model driver lane-changing behavior. The model was built using data with microscopic details of merging and weaving maneuvers under congested traffic condition. He also proposed to classify lane-changing maneuvers into free, forced, and cooperative lane changes. The model was implemented in the ARTEMiS traffic simulator and the results demonstrate its capability to reproduce the observed behavior of individual vehicles in both freeways and signalized urban arterial network.

Toledo et al. (2007) developed integrated driving behavior model that captures both lane changing and acceleration behaviors. The framework proposed concepts of short-term goal and short-term plan. The short-term goals were defined by the target lanes, and short-term plans were defined by various gaps in traffic in the target lane. Drivers are assumed to accomplish short-

term goals by conceiving and performing short-term plans by adapting their acceleration behavior to select target lane gap.

A summary of findings from literature review in lane-changing models is presented as following. First, the primary attention of the research has been on modeling driver gap-acceptance behavior. Second, the majority of driver lane-changing models are developed based on discrete choice models (i.e. probit and logit models) which need priori assumptions about distribution of critical gaps. The estimation for model parameters sometimes is difficult. Third, due to the difficulty to distinguish the decision to make a lane changing and performing lane-changing maneuver, it is hard to capture driver's lane-changing decision process. Only lane-changing maneuver and not the lane-changing decision are models. Forth, few lane-changing models take into the variable of the distance from the merging vehicle to the end of the merge lane in mandatory lane-changing situation, while it is intuitive that driver lane-changing behavior is more aggressive when they are approaching the end of merge lane. Last, most of the lane-changing models are not developed using congested traffic data, thus they cannot be applied to lane-changing situation in a congested traffic.

The lane-changing models proposed in this dissertation built upon the idea of gap acceptance, but other factors, including speed and distance from the merging vehicle to the end of the merge lane, were also considered. Instead of modeling the lane-changing decision-making process, the observable lane-changing maneuver was modeled. In addition, the proposed lane-changing models captured the driver lane-changing behavior in congested traffic using real driver data.

2.2. Traffic Forecasting Models

A large majority of time series approaches are univariate in nature. Univariate time series models use only historical traffic flow data from the location of interest to predict future traffic flow at the same location. The family of autoregressive integrated moving average (ARIMA) models is the most extensively applied time series model form.

Levin and Tsao (1980) evaluated several ARIMA models on 20-, 40- and 60-s interval occupancy and volume data collected during morning peak hours at both local lanes and freeways in Chicago, Illinois. The Box-Jenkins approach was used to estimate the models. They found ARIMA (0, 1, 1) model had the best performance for predicting both volume and occupancy. 60-s forecasting interval was found to be the most effective interval.

Hamed et al. (1995) also conducted Box-Jenkins time-series analyses for forecasting traffic volume in urban arterials. They used 1-min interval volume data collected in five major urban arterials. The same conclusion as Levin and Tsao's (1980) research was drawn from this study. They also found ARIMA (0, 1, 1) turned out to be the most adequate model for forecasting volume.

Williams et al. (1998) applied seasonal time series models to the single-interval traffic flow forecasting for urban freeways. Two time series models seasonal ARIMA and Winters exponential smoothing were built and evaluated on 15-min interval data sets collected from freeways in northern Virginia. They found ARIMA (2, 0, 1)(0, 1, 1)$_{96}$ and ARIMA (1, 0, 1)(0, 1, 1)$_{96}$ had the best fit for the data. The single-step forecasting results showed that the seasonal ARIMA model outperform the nearest neighbor and historical average models.

Other forms of the time series models ranging from non-parametric regression to local linear regression and Kalman filtering were also used. Smith and Demetsky (1997) proposed four models for freeway traffic flow forecasting using northern Virginia freeway volume data. They were historical average, neural network, and nonparametric regression models. The comparison results indicated that nonparametric regression model significantly outperformed other models.

Smith et al. (2002) raised some concerns about the ability to fit and maintain seasonal ARIMA models due to its time-consuming outlier detection and parameter estimation process. In their study, they used data-driven nonparametric regression to compare the model performance of seasonal ARIMA models. They found heuristic forecast generation methods significantly improved the performance of nonparametric regression. The results indicated that the nonparametric regression coupled with heuristic forecast generation method is preferred to using naïve forecasting method, when the implementation requirement of seasonal ARIMA cannot be met.

Clark (2003) presented a nonparametric regression technique described as a *k* nearest neighbor model for traffic prediction. He claimed several advantages of nonparametric regression over alternatives. It is readily understood by practitioners. It does not rely to any parametric assumptions and is likely to be a more robust forecasting tool. Also, the method is simple and quick to implement. The study found that the forecasts of flow and occupancy produced by the method were more accurate than naïve method, but speed forecasts were worse.

Sun et al. (2003) applied the local linear regression model to short-term traffic prediction using 32-day traffic speed data at 5-min intervals collected on US-290 Northwest freeway in Houston. The performance of proposed model was compared with nonparametric approaches

such as *k* nearest neighbor and kernel methods. The results showed the local linear method had better performance than *k* nearest neighbor and the kernel smoothing method. They also found that all model performances decreased as the prediction horizon increased. The accuracy of nonparametric method was highly dependent on the size of database. Large data base was more likely to increase prediction accuracy.

In order to take into account of both temporal and spatial information of traffic flow simultaneously in the prediction model, multivariate time series models were applied. Williams (2001) developed a multivariate forecast model ARIMAX that includes upstream traffic flow data. The results indicated that ARIMAX improved forecast performance over univariate models. However, they also raised some issues of ARIMAX models. These issues included increase of model complexity, model consistency, model robustness in the face of interruptions in the upstream data series, and variability in the cross-correlation between upstream and downstream observations. The most critical issue was the assumption of constant transfer function parameters, but in fact the correlation between upstream and downstream observations varies with traffic conditions.

Stathopoulos and Karlaftis (2003) proposed a multivariate time series state space model that uses traffic flow data from upstream detector and concluded that multivariate state space model results in higher prediction accuracy than univariate time series models. The results of the model suggested that different model specifications are appropriate for different periods.

Kamarianakis and Prastacos (2005) proposed the space–time autoregressive integrated moving average (STARIMA) models to predict traffic flow in an urban area. The STARIMA model incorporated the spatial characteristics of the space-time process by using the weighting

matrices estimated on the basis of the distances among the locations where data were collected. The matrices differentiated space-time approach from the vector autoregressive moving average (VARMA) methodology and enable the user to control the number of parameters to be estimated.

Neural networks ranged from static to dynamic structures and included multilayer feedforward neural networks, radial basis function neural networks, the time-delayed neural networks and the recurrent neural network.

Smith and Demetsky (1994) proposed backpropagation neural network for short-term traffic flow prediction because neural network is able to model undefined, complex nonlinear surfaces. The superiority of neural network was shown in the comparison with other traditional approaches including an historical, data-based algorithm and a time-series model. They found the backpropagation neural network was more responsive to dynamic conditions than other two models.

Dougherty and Cobbet (1997) trained backpropagation neural network to forecast the traffic flow, speed and occupancy in Utrecht/Rotterdam/Hague region of Netherlands. They developed a technique of stepwise reduction of network size by elasticity testing the large neural networks, since the large size made them impractical for implementation. The forecast results showed some promise, but did not outperform naïve predictors. The elasticity testing was found to be useful, not only for reducing network size, but also for interpreting the model.

Ledoux (1997) developed a cooperation based neural network traffic flow model for real time adaptive urban traffic control system using simulation data. There were two steps in the modeling process. The traffic flow was first modeled on a signalized link by a local neural

network. Then the traffic flow over the whole network of junctions was modeled based on the communications between local neural networks. They concluded that one minute ahead predictions of the queue lengths and the output flows produced good accuracy.

Yun et al. (1998) studied the relationship between data characteristics and the forecasting accuracy of different neural network models in forecasting traffic volume. Three different data sets of traffic volume gathered from interstate highways, intercity highways, and urban intersections were used to compare and test the forecasting accuracy of a back-propagation neural network, a finite impulse response (FIR) model, and a time-delayed recurrent neural network. The comparison results showed that time-delayed recurrent neural network outperform others in forecasting very randomly moving data, but for relatively regular periodic data FIR model showed better accuracy than time-delayed recurrent neural network.

Yasdi (1999) employed a neural network approach named recurrent Jordan networks which is popular in the modeling of time series for traffic volume forecasting based on time-series data. In this study, three types of forecasting: weekly, daily, and hourly prediction as long-term, mid-term and short-term prediction were considered. The model performance results were better than the compared methods and the proposed model improved the forecasting accuracy by about 20%.

Abdulhai et al. (1999) proposed a time-delayed neural network (TDNN) model optimized by Genetic Algorithm for short term traffic flow prediction. The model predicted flow and occupancy based on temporal and spatial traffic data. The further investigation of the extent of the “look-back” interval, the extent of prediction in the future, the extent of spatial contribution, the resolution of the input data, and their effect on prediction accuracy were conducted. Both

simulated and real traffic flow data were used for validation of the model's performance. The model showed potential to be superior to the multilayer feed-forward neural network.

Innamaa (2000) built a multilayer perceptron (MLP) network to predict the speed and flow 15 minutes ahead of the observation period in 5-min intervals. The model consisted of two sub-models. One is to forecast the mean speed and the other is to forecast traffic flow. It yielded better results than one single model predicting both variables at the same time. It was found that it was better to increase the number of hidden neurons by reducing input parameters.

Zhang (2000) presented a recursive traffic flow algorithm using neural networks. The parameters of the model were obtained through nonlinear optimization. The study showed that this method can yield reasonable accurate results.

Chen et al. (2001) presented an application of two hybrid neural network approaches on traffic flow application. The first hybrid approach incorporated four Auto-Regressive Integrated Moving Average (ARIMA) models. The second one used two Multi-Layer Perceptron (MLP) models. They found that the ARIMA hybrid approach performed better than all individual ARIMA models. The MLP hybrid approaches outperform all models in this study.

Lingras and Mountford (2001) used genetic algorithms to optimize for selecting connections between input and hidden layers of a time delayed neural network for intercity traffic volume prediction and it reflect a high degree of accuracy.

Dia (2001) developed an object-oriented time-lag recurrent neural network (TLRN) model for short-term traffic prediction. The model predicted speed up to 15 minutes into the future. The results showed a high degree of accuracy (90-94%) for predicting speed up to 5

minutes. Similar models were also successful in predicting travel times up to 15 minutes into the future with a high degree of accuracy (93-95%).

Van Lint et al. (2002) developed recurrent neural networks for freeway travel time prediction. It is capable of dealing with spatiotemporal relationships implicitly. The topology of the recurrent neural network was derived from a state-space formulation of travel time prediction problem. The performance of the model was tested on data from a densely used highway stretch in the Netherlands. The results demonstrated that the neural network was able to predict travel time with error less than 10% of expected travel time.

Ishak et al. (2003) proposed an approach to optimize the short-term traffic prediction performance on freeways using multiple artificial neural network topologies under various network and traffic condition settings. A long memory component was introduced to the input patterns to allow the networks to build internal representation of recurrent conditions, in order to enable the networks to learn from historical information. The study showed that the optimized neural network had better forecasting performance.

Vlahogianni et al. (2005) proposed a genetically optimized neural network for short-term traffic flow prediction. This study extended past neural network models by providing a genetic algorithm based optimization strategy for choosing the proper representation of traffic flow data and the appropriate neural network structure. The model was evaluated by applying it to both univariate and multivariate signalized urban arterial traffic flow data. The results indicated that the simple static neural network with genetically optimized step size, momentum and number of hidden units yielded very satisfactory performance.

Chen and Chen (2007) presented how ensemble learning method such as bagging increased the traffic flow forecasting performance of radial basis function neural network (RBFNN). The effect of the extent of prediction, the "look-back" interval and the time resolution on the prediction accuracy was carefully examined based on the real traffic flow data collected at loop 3 freeway in Beijing, China. The results showed that ensemble learning method such as bagging demonstrated great potential in improving forecasting capability of RBFNN. They also found that the prediction accuracy decreased quickly as the extent of prediction increased. The best time resolution to predict the traffic flow x minutes later was x minutes as well.

Recently, Sun et al. (2006) and Sun and Xu (2011) proposed different Bayesian network approaches to forecast traffic flow using spatial traffic flow data from adjacent road links. Zhang and Ye (2008) proposed a fuzzy logic system to improve traffic flow prediction accuracy. Min and Wynter (2011) developed an extended time-series-based method which takes into account temporal and spatial interactions. Pan et al. (2013) used a stochastic cell transmission framework to predict short-term traffic flow by considering the spatial–temporal correlation. Sun et al. (2012) conducted research on network-scale traffic modeling and forecasting with graphic lasso and neural networks. Huang and Sun (2013) applied kernel regression with sparse metric learning to forecast short-term traffic flow.

The literature review for traffic forecasting models is summarized. First, previous research focused on developing predictive models for normal traffic conditions without the impact of weather, work zone, incidents and special events. Second, the majority of the models were developed using temporal volume data or temporal and spatial data. Last, time series models and artificial neural networks were the two largest family of methods for forecasting traffic under normal conditions.

Traffic prediction models in this dissertation were aimed at predicting traffic demand for work zone bottlenecks. Models considered the impact of work zones on traffic demand by taking into account work zone characteristic variables such as work zone speed limit, work zone type, number of lane drops, work zone length and duration. Temporal and spatial volume data were used to develop the short-term traffic prediction model. Previous traffic prediction models for normal traffic – nearest neighbor non-parametric regression and feedforward neural network were also implemented here for work zone traffic prediction.

CHAPTER 3: METHODOLOGY

The methodology of this dissertation consists of seven AI models including Bayes classifier, classification and regression tree, random forest, genetic fuzzy system, multilayer feedforward neural network, nearest neighbor nonparametric regression, and AdaBoost. Bayes classifier, classification tree, genetic fuzzy system, random forest and AdaBoost were used for lane-changing modeling. Regression tree, random forest, neural network, and nearest neighbor nonparametric regression were used for traffic demand forecasting.

3.1. Bayes Classifier

3.1.1 Bayes Decision Theory

For a binary classification problem, let y_1, y_2 denote two classes. According to the Bayesian classification rule (Theodoridis and Koutroumbas, 2006),

$$P(y_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y_i)P(y_i)}{p(\boldsymbol{x})}, i = 1,2, \qquad (3-1)$$

where $\boldsymbol{x}$ is the input vector, $P(.)$is the probability, and $p(.)$ is the probability density function. The Bayes classification rule (Hastie et al., 2001) is stated as follows:

- If $P(y_1|\boldsymbol{x}) > P(y_2|\boldsymbol{x})$, $\boldsymbol{x}$ is classified to y_1.
- If $P(y_1|\boldsymbol{x}) < P(y_2|\boldsymbol{x})$, $\boldsymbol{x}$ is classified to y_2.
- If $P(y_1|\boldsymbol{x}) = P(y_2|\boldsymbol{x})$, $\boldsymbol{x}$ can be assigned to either y_1 or y_2.

Using (3-1), the classification decision is equivalently based on the inequalities

$$p(\boldsymbol{x}|y_1)P(y_1) > (<)p(\boldsymbol{x}|y_2)P(y_2), \qquad (3-2)$$

3.1.2 Risk of Misclassification

Risk considers both the likelihood of misclassification and the cost of the misclassification. A penalty term λ_{ki} denotes the cost of misclassifying $\boldsymbol{x}$ to a wrong class y_i while belonging to class y_k (Theodoridis and Koutroumbas, 2006). In order to minimize the average risk, the classification decision inequalities (3-2) become

$$(\lambda_{12}-\lambda_{11})p(\boldsymbol{x}|y_1)P(y_1) > (<)(\lambda_{21}-\lambda_{22})p(\boldsymbol{x}|y_2)P(y_2), \qquad (3-3)$$

Adopting the assumption that $\lambda_{ij} > \lambda_{ii}$ and $\lambda_{ii} = 0$, the Bayes classification rule becomes

$$\boldsymbol{x} \text{ belongs to } y_1(y_2) \text{ if } l_{12} = \frac{p(\boldsymbol{x}|y_1)}{p(\boldsymbol{x}|y_2)} > (<)\frac{P(y_2)\lambda_{21}}{P(y_1)\lambda_{12}}, \qquad (3-4)$$

where l_{12} is likelihood ratio.

3.1.3 Density Estimation

The class label of $\boldsymbol{x}$ can be predicted using the class-conditional probability density function, $p(\boldsymbol{x}|y_i)$. One of the most commonly used probability density functions in practice is the Gaussian or normally probability density function. The main reason for its popularity is its computational tractability. However, this is not the most common case. In many problems, the underlying probability density functions need to be estimated from the available data. There are

various ways to approach this problem. They can be categorized into parametric methods and nonparametric methods. Parametric methods include maximum likelihood parameter estimation, maximum *a posteriori* probability estimation, Bayesian inference and maximum entropy estimation. Nonparametric methods include Parzen windows, k nearest neighbor (kNN) density estimation.

In this study, the nonparametric method kNN density estimation method (Buturovic, 1993) was used to estimate the class-conditional probability density functions. The kNN estimation method was chosen because similar to kernel estimation, it is a nonparametric method; thus, there is no need to assume a distributional form unlike maximum likelihood. Also, the kNN estimation method is computationally simple and easy to implement. By using this method, the class-conditional probability density functions is estimated as

$$p(\boldsymbol{x}|y_i) = \frac{k}{N_i V_i}, i = 1,2, \qquad (3-5)$$

where N_i is the total number of training samples in class y_i, and V_i is the volume of the multi-dimensional hypersphere (i.e. input data space) centered at $\boldsymbol{x}$ that contains k points from class y_i.

$P(y_i)$ is easily estimated from observations as follows:

$$P(y_i) = \frac{N_i}{N}, i = 1,2, \qquad (3-6)$$

where N_i is the total number of training samples in class y_i, and N is the total number of training samples.

By substituting equation (3-5) and (3-6) into equation (3-4), Bayes classification rule is equivalent to

$$\boldsymbol{x} \text{ belongs to } y_1 \ (y_2) \text{ if } l_{12} = \frac{V_2}{V_1} > (<) \frac{\lambda_{21}}{\lambda_{12}}, \qquad (3-7)$$

Let r_i denote the radius of the hypersphere centered at $\boldsymbol{x}$ that contains k points from class y_i. Since hypersphere dimension in this study is five (the total number of input variables), the likelihood ratio can be computed as

$$l_{12} = \frac{V_2}{V_1} = \left(\frac{r_2}{r_1}\right)^5, \qquad (3-8)$$

3.1.4. Distance Measurement

The hypersphere radius r_i can be easily obtained by searching for the kth nearest distance from all the training vectors of class y_i. Various distance measures can be used. The commonly used Euclidean and Mahalanobis distances, while simple computationally, imply that the input space is isotropic or homogeneous. However, the assumption for isotropic is invalid and undesirable in many practical applications. There are several alternatives for determining the weights for input space, such as analytical hierarchy process and expert opinion. In this study, the weighted distance metric proposed by Domeniconi et al. (2005) was adopted to calculate hypersphere radius, because the decision function constructed by support vector machines (SVMs) can be used to determine the most discriminant direction in a neighborhood around the query. Such a direction provides a local feature weighting scheme. It relaxes the assumption of isotropic space. Let $\boldsymbol{x_j}$ and $\boldsymbol{x_k}$ denote two vectors of l features. The weighted distance is:

$$D(\boldsymbol{x_j}, \boldsymbol{x_k}) = \sqrt{\sum_{i=1}^{l} w_i (x_{ji} - x_{ki})^2}, \qquad (3-9)$$

where w_i are the weights associated with features. Let $\boldsymbol{q}$ be the query point whose class label is to be predicted. SVMs classifier gives decision hyperplane $g(\boldsymbol{x})$. Let $\boldsymbol{p}$ be the point with the closest Euclidean distance to $\boldsymbol{q}$ on decision hyperplane $g(\boldsymbol{x})$**.** $R(\boldsymbol{q})_j$ is defined as

$$R(\boldsymbol{q})_j = |\boldsymbol{e}_j^T \nabla g(\boldsymbol{p})|, \qquad (3-10)$$

where $\boldsymbol{e}_j$ denote the canonical unit vector along input feature j. The weights are given by

$$w(\boldsymbol{q})_j = \frac{(R(\boldsymbol{q})_j)^t}{\sum_{i=1}^{l}(R(\boldsymbol{q})_i)^t}, \qquad (3-11)$$

where t is a positive integer. In this study, t values ranging from 1 to 4 were applied and $t = 2$ produced the best model performance. In this case, the SVMs decision hyperplane is in linear form $g(\boldsymbol{x}) = \boldsymbol{b}^T\boldsymbol{x} + b_0 = 0$, Thus, $R(\boldsymbol{q})_j \equiv b_j$.

3.1.5. Advantages and Limitations of Bayes Classifier

Bayes classifier is derived from probability theory, thus, it is simple and easy to understand. Bayes classifier is fast to train, and is not sensitive to irrelevant features. It can handle both continuous and discrete data. Another important feature of Bayes classifier is that it is able to take into account the cost of misclassification. However, there is a drawback associated with high-dimensional feature space. For high dimensional feature space, obtain good estimate of posteriori probability demand a large number of training points. The number grows exponentially with the dimensionality. If kNN density estimation method is used, Brute-force searching amounts to operations proportional to $O(kN)$. It demands large amount of processing time and high computational requirement.

3.2. Classification and Regression Tree (CART)

CART has emerged as one of the most popular method for both classification and regression. It performs binary split on input variables. Classifying patterns or outputting regression results may only takes a few tests. Also, it can simultaneously treat a mixture of numeric and categorical variables. Moreover, it is easily understandable due to its simple model structure.

3.2.1 Classification Tree

A classification tree, or decision tree, achieves a classification decision by performing a sequence of tests on feature vectors along a path of nodes as shown in **FIGURE 3-1** (Russel and Norvig, 2003). Each internal node in the tree provides a question, "Is feature $x_i \geq a$?", where a is a threshold value. The binary answer to the question corresponds to a descendant node. At the end, each terminal node returns a class label y_i. There may be two or more terminal nodes with the same class label. Classification tree is constructed by a sequence of binary splits of training set $\boldsymbol{X}$ into terminal nodes. The entire tree construction process revolves around three components:

- The selection of the splitting rules.
- The criterion to stop splitting and declare a terminal node.
- The assignment of each terminal node to a class.

The key to the problem is how to determine the splits, terminal nodes and their assignments using training data.

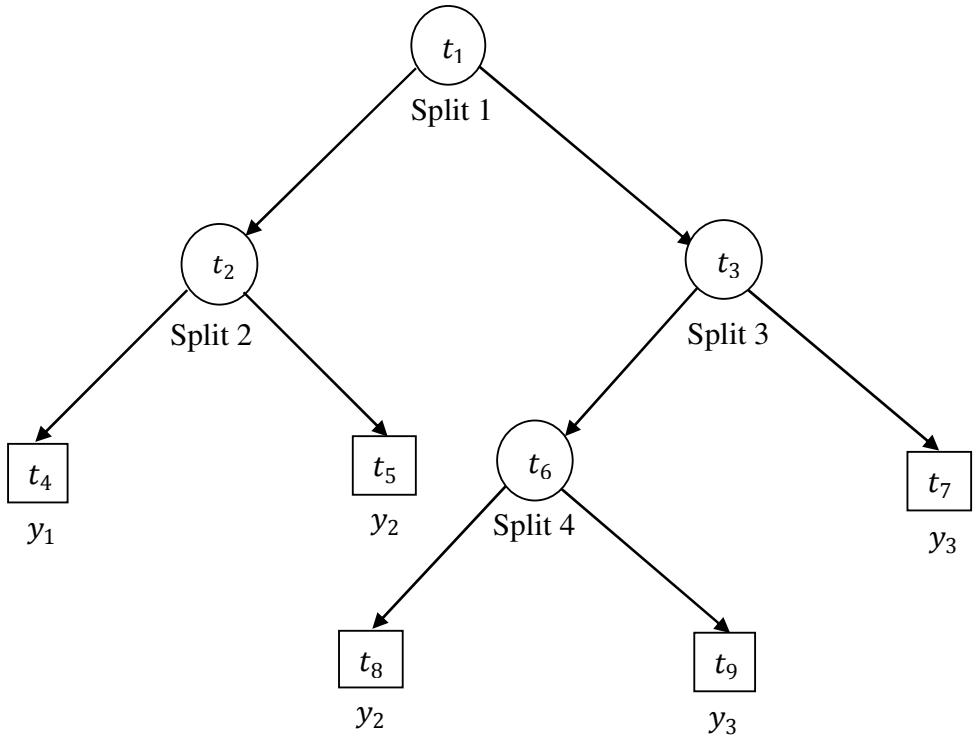


FIGURE 3-1 Classification tree structure.

3.2.1.1. Node Splitting

In order to construct a classification tree, the set of splits at tree nodes are to be determined. The root node is assigned with the entire training set $\boldsymbol{X}$. The goal of the binary split at each node is to produce subsets at descendant nodes that are more class homogeneous or purer than the parent subset. This means that the training feature vectors in each one of the new subsets shows a higher preference for specific class(es). The node purity is the smallest when all classes

are equally mixed together in it, and the largest when the node contains only one class. A variety of node impurity measures can be defined. The most commonly used ones are Gini diversity index (Breiman et al., 1984) and Shannon's information theory (Shannon and Weaver, 1949).

Let y_1, y_2 denote the two classes in a binary classification task. The subset of training set at each node t is denoted as $\boldsymbol{X}_t$. Let $P(y_i|t)$ denote the probability that a sample in subset $\boldsymbol{X}_t$ belongs to class y_i, $i = 1,2$. Gini diversity index is defined as

$$I(t) = \sum_{i \neq j} P(y_i|t)P(y_i|t), \qquad (3-12)$$

Shannon's information theory defined node impurity as

$$I(t) = -\sum_{i=1}^{2} P(y_i|t) \log_2 P(y_i|t), \qquad (3-13)$$

Breiman et al. (1984) pointed out that the properties of the resulting final tree are insensitive to the choice of the splitting criterion. In this dissertation, Shannon's information theory (Shannon and Weaver, 1949) was adopted to measure the impurity of subset $\boldsymbol{X}_t$, also known as node impurity. $P(y_i|t)$ can be easily estimated by N_t^i/N_t, where N_t^i is the number of vectors in subset $\boldsymbol{X}_t$ that belongs to class y_i, and N_t is the total number of vectors in subset $\boldsymbol{X}_t$. After performing a binary split at node t, a subset $\boldsymbol{X}_{tY}$ with an answer "Yes" is assigned to node t_Y, and a subset $\boldsymbol{X}_{tN}$ with answer "No" is assigned to node t_N. The decrease in node impurity $\Delta I(t)$ is given by

$$\Delta I(t) = I(t) - \frac{N_{tY}}{N_t} I(t_Y) - \frac{N_{tN}}{N_t} I(t_N), \qquad (3-14)$$

where N_{tY} and N_{tN} are the numbers of vectors in subsets $\boldsymbol{X}_{tY}$ and $\boldsymbol{X}_{tN}$. By exhaustively searching for all candidate questions, the one that leads to the maximum impurity decrease is chosen.

3.2.1.2. Stop-Splitting Criteria and Class Assignment

A threshold probability value P_0 is necessary to stop the node splitting process at any node. Splitting stops when more than $P_0 \times 100\%$ of vectors in the subset belong to any one single class, i.e., $\max_{\mathrm{i}} P(y_i|t) > P_0$. In this model, 0.9 is selected to be the threshold value as this value will also ensure the tree grows large enough for pruning. Once a terminal node is determined, the class label is given by y_j where

$$j = arg \max_i P(y_i|t), \qquad (3-15)$$

3.2.1.3. Tree Pruning

The size of a classification tree is the key factor in developing the tree model. If a tree grows over more splits, it results in lower resubstitution estimate of the misclassification rate which is calculated based on training set. For instance, if the splitting is carried out to the point where each terminal node contains only one data case, the resubstitution estimate of the misclassification rate is zero. However, as the number of splits increase to past a certain point, the resubstitution estimate will be biased downward from the true misclassification rate. This problem is called overfitting. Too large a tree will overfit the training data and perform poorly on

testing data. On the other hand, if the size of a tree is too small, the tree results in high misclassification rates. Therefore, the suggested approach is to find the right sized tree by pruning after growing a tree with a large enough size.

Minimal cost-complexity pruning (Breiman et al., 1984) was employed as the pruning rule in this study. Due to its computational efficiency, the minimal cost-complexity pruning is one of the most common methods for pruning. The sequence of subtrees generated by this pruning process is nested meaning that the nodes that were previously cut off will not reappear in subsequent subtrees. The cost-complexity measure $R_\alpha(T)$ of tree T is defined as

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}|, \qquad (3-16)$$

where $R(T)$ is the resubstitution estimate for the overall misclassification rate of tree T, $\alpha \geq 0$ is the complexity parameter, and $|\tilde{T}|$ is the total number of terminal nodes in tree T. Each value of α is associated with a subtree $T(\alpha)$ that minimizes $R_\alpha(T)$. As α increase from 0 to a sufficiently large number, the size of $T(\alpha)$ decrease from its largest size to the smallest size (only for the root node). If a subtree that minimizes $R_\alpha(T)$ for a given value of α, it will remain minimizing $R_\alpha(T)$ until α increases to a jump point. Let $\{\alpha_k\}$ be the increasing sequence of the jump points. For any $\alpha_k \leq \alpha \leq \alpha_{k+1}$, $T(\alpha) = T(\alpha_k)$. Finally, a sequence of minimal cost-complexity trees $\{T_k\}$ are generated. The right sized tree T^*can be selected by validation sample estimates

$$R^v(T^*) = \min_k R^v(T_k), \qquad (3-17)$$

where $R^{ts}(.)$ denotes misclassification rate for validation sample.

3.2.2. Regression Trees

A regression tree (Breiman et al., 1984) is similar to a classification tree. It is also constructed by a sequence of binary splits of training set $\boldsymbol{X}$ into terminal nodes, as shown in **FIGURE 3-2**. Instead of return a class label, each terminal node output a response value $y(t)$. The entire tree construction process revolves around three components as classification tree:

- The selection of the splitting rules.
- The criterion to stop splitting and declare a terminal node.
- The assignment of a value $y(t)$ to each terminal node.

It turns out that the issue of the node assignment is easiest to solve. $y(t)$ can be easily obtained by averaging the response values of the terminal subset $\boldsymbol{X_t}$.

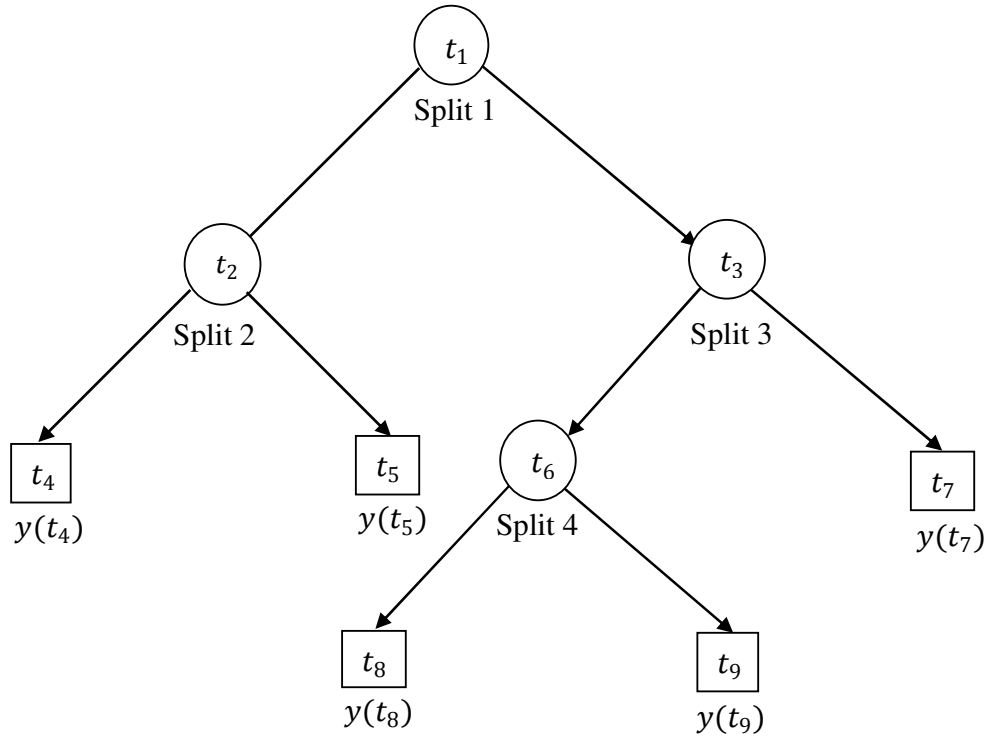


FIGURE 3-2 Regression tree structure.

The goal of the binary split at each node in regression tree is to reduce the overall resubstitution estimate of prediction error. The resubstitution estimate of prediction error of node t is often measured by the mean squared error of the node subset $\boldsymbol{X}_t$**.** It is formulated as

$$R(t) = \frac{1}{N_t}\sum_{i=1}^{N_t}(y_{ti} - \bar{y}_t)^2, \qquad (3-18)$$

where N_t is the size of node t, y_{ti} is the ith response value of subset falling into node t, and $\bar{y}_t$ is the average of response values of subset falling into node t. The best split of node t is the split that most decreases $R(t)$. For any split of node t into two descendant nodes t_Y and t_N, the decrease of $R(t)$ is

$$\Delta R(t) = R(t) - R(t_Y) - R(t_N), \qquad (3-19)$$

By adopting stop-splitting rule, a node was declared terminal node if node size $N_t \leq N_{min}$. Breiman (1984) suggests N_{min} for regression tree to be 5. The tree selection strategy is exactly the same as that used to select a classification tree. First, a large tree is grown by successively splitting. The starting tree is usually much larger than classification starting trees. Then minimal error-complexity pruning is performed exactly as minimal cost-complexity pruning in classification.

3.2.3. Advantages and Limitations of CART

CART is easy to understand and interpret. It is not sensitive to outliers. There is no worry about tuning a large number of parameters. Model results are easy to interpret since Boolean logic is used. The data training process is fast. The computational complexity of CART is logarithmic in the number of training data points. Suppose the training data contains n data points and m attributes, the computational cost of building one regression tree is $O(mn \log n)$. It is able to handle both categorical and numerical data. The major drawback of CART is its high variance of model results and it easily overfits training dataset. In practice, it is not uncommon that a small change in training data results in a very different tree, but ensemble learning methods such as bagging and random forest can address this problem.

3.3. Random Forest

Bagging or bootstrap aggregating is an approach to reduce variance of an estimated prediction function. Bagging seems to work especially well for high-variance and low-bias procedures. The essential idea in bagging is to average many noisy but approximately unbiased models. Trees are ideal candidates for bagging, since they are noisy and have relatively low bias. Random forest (Breiman, 2001) is an ensemble method based on the idea of bagging that build a large group of un-pruned trees. The main idea behind ensemble methods is to build a strong prediction model by combining a large group of weak models. The idea of random forest is to improve the variance reduction of bagging by reducing the correlation between the trees. Trees generated in bagging are identically distributed. Suppose K trees are generated in bagging with positive pairwise correlation ρ, and each with variance σ^2. The variance of the average is

$$\rho\sigma^2 + \frac{1-\rho}{K}\sigma^2. \qquad (3-20)$$

With the increase of K, the second term approach to zero, but the first term remains, and the pairwise correlation of bagged trees limits the benefit of averaging. A random forest reduces the variance of bagging by reducing the correlation between the trees. This is achieved through random selection of the input variables in the tree growing process.

The random forest algorithm for regression is described below.

1. For $k = 1$ to K:
    a. Draw a bootstrap sample from original training dataset.
    b. Grow a regression tree T_k with the bootstrapped dataset by recursively repeating the following steps for each node until the minimum node size is reached.
        i. Randomly select m predictor from p predictors.
        ii. Use the predictor variable among m predictors that makes the best split to split the node into two descendant nodes.
2. Output a collection of trees $\{T_k\}_1^K$.

To make prediction at a test data case, the new arrived data case is pushed down all the trees. Each tree will give an output value or a class label. The result is the average of the output values for regression, and majority voting for classification. In addition, Breinman (2001) makes the following recommendations:

- For regression, the default value for m is $p/3$ and the minimum node size is five.
- For classification, the default value for m is $\sqrt{p}$ and the minimum node size is one.

An importance feature of random forest is its use of out-of-bag (OOB) estimate, which is explained as follows. For each data case $(\boldsymbol{x}, y)$ in training set, aggregate the votes or take the average only over those trees built on bootstrap training sets which do not contain $(\boldsymbol{x}, y)$. Call this the OOB classifier. The OOB estimate for the generalization error is the error rate of the OOB classifier on the training set. It is similar to N-fold cross validation, but OOB estimate is unbiased. Thus, random forest can be fit in one sequence, with cross-validation being performed along the way. Once the OOB error stabilizes, the training can be terminated.

Random forest constructs variable importance to help user to understand the mechanism of the prediction process. One measure is obtained by accumulating improvement in split-criterion at each split in each tree separately for each variable over all trees in the forest. The other measure is constructed by out-of-bag samples. When the bth tree is grown, the out-of-bag samples are passed down to the tree, and the prediction accuracy is recorded. Then values for the jth variable are randomly permuted in the out-of-bag sample, the accuracy is again computed. The decrease in prediction accuracy caused by permuting is averaged over all trees, and is used as a measure of the importance of jth variable.

There are several advantages of random forest:

- There is no need to prune trees.
- It performs as well as Adaboost and sometimes better.
- It is robust to outliers and noise.
- It is faster than bagging and boosting.
- It provides useful internal estimates of error and variable importance.

- It cannot overfit training data. Increasing the number of trees does not cause the random forest to overfit.

The limitations of random forest are listed below:

- When the number of input variable is large, but the proportion of relevant variables is small, random forest is likely to perform poorly with small m.
- It can not predict beyond the range of training data for regression.

3.4. Genetic Fuzzy System

3.4.1. Development of Membership Function

A fuzzy set is a class of objects with a continuum of grades of membership, and a membership function assigns a grade of membership between zero and one to each object. Some common membership functions include triangular, trapezoidal, Gaussian, sigmoid and polynomial. The triangular membership function was selected as input membership function for this study for its simplicity and because it performs well in applications such as feedback control (Zhao and Bose, 2002). As shown in **FIGURE 3-3**, x is an input variable, c is the center of the membership function, w_1 and w_2 are the two widths of the membership function, and $\mu(x)$ denotes the degree of membership.

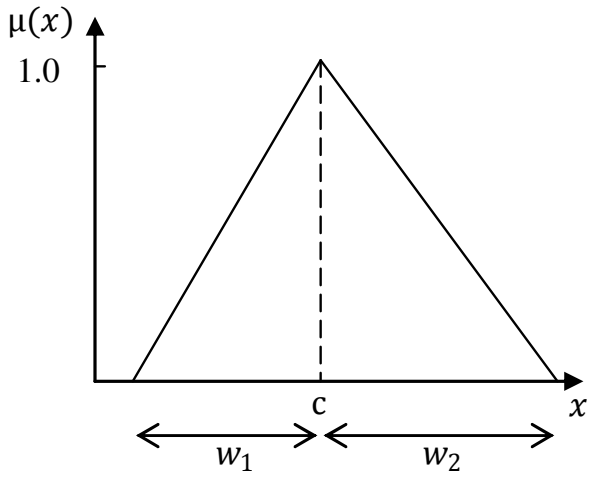


FIGURE 3-3 Triangular membership function.

The input membership function centers were determined by the entropy minimization principle (Christensen, 1980). For any input variable, assume a threshold value x in the variable range between x_1 and x_2 partitions the dataset into two regions. In a two-class problem, an entropy with value of x is expressed as

$$S(x) = p(x)S_p(x) + q(x)S_q(x), \qquad (3-21)$$

where

$$S_p(x) = -[p_1(x)\ln p_1(x) + p_2(x)\ln p_2(x)], \qquad (3-22)$$

$$S_q(x) = -[q_1(x)\ln q_1(x) + q_2(x)\ln q_2(x)], \qquad (3-23)$$

where

$p_k(x)$ and $q_k(x)$ are the conditional probabilities that the class k sample is in the region $[x_1,\ x_1 + x]$ and $[x_1 + x,\ x_2]$, respectively; $p(x)$ and $q(x)$ are the probabilities that all samples are in the region $[x_1,\ x_1 + x]$ and $[x_1 + x,\ x_2]$, respectively. $p(x) + q(x) = 1$

For any input variable, first a threshold value is found that holds the minimum entropy. The threshold value is then used to partition the dataset into two regions. The entropy minimization is applied again to each of these two regions to generate two new threshold values. The three threshold values are then used as centers of each input membership functions.

Next, the widths of input membership functions are determined. The widths of input membership functions affect the performance of the model. In order to optimize the model performance, a genetic algorithm (GA) (Forrest, 1993) was used to obtain the optimal widths of membership functions. GA is a population-based search technique that has been used in other transportation optimization studies (Edara and Teodorovic, 2008; Edara et al., 2011). Given the rule extracting and mapping method, discussed in the next section, GA identifies the widths of input membership functions that generate the most accurate output values. The root mean square error (RMSE) was used as the fitness function to evaluate the predicted output values in searching for the optimal widths.

In this dissertation, the output of the model is binary. Thus, the output membership function is a delta function, as shown in the **FIGURE 3-4**. The full membership is located at b and all other values have zero membership value.

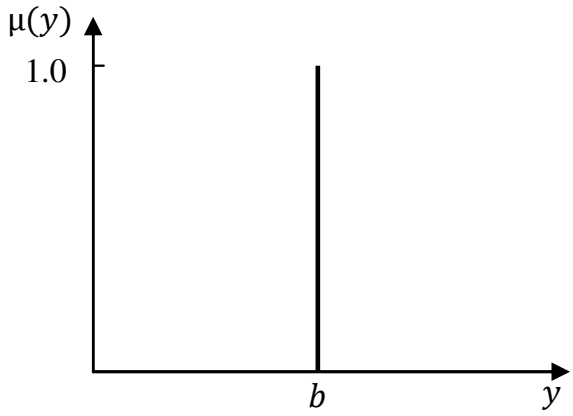


FIGURE 3-4 Delta membership function.

3.4.2. Rule Extraction and Mapping

After determining the membership function, fuzzy rules are extracted from input-output training data. The Learning From Examples (LFE) training procedure (Wang and Mendel, 1992) was used to construct fuzzy rule-base. The fuzzy sets of input and output membership functions is chosen for the rule representing data case $(\boldsymbol{x}, y)$ by choosing the ones with the highest degree of membership. For a two-input one-output fuzzy system, let X_i^j and Y^j denote the fuzzy sets with associated membership function $\mu_{X_i^j}(x_i)$ and $\mu_{Y^j}(y)$, respectively. The rule-base to be constructed contains rules of the form

$$R_i = \textbf{If } x_1 \text{ is } X_1^j \text{ and } x_2 \text{ is } X_2^k \text{ Then } y \text{ is } Y^l,$$

where associated with the ith rule is a degree defined by

$$degree(R_i) = \mu_{X_1^j}(x_1) \times \mu_{X_2^k}(x_2) \times \mu_{Y^l}(y), \qquad (3-24)$$

Suppose rules R_m are already in the rule-base, the following guideline for adding new rule R_i:

- If $degree(R_i) > degree(R_m)$ and the premises for R_i and R_m are the same, then the rule R_i would replace R_m in the existing rule-base.
- If $degree(R_i) \leq degree(R_m)$ and the premises for R_i and R_m are the same, then the rule R_i is not added to the rule-base, since previous data case is already adequately represented with rules in the fuzzy system.
- If rule R_i does not have the same premise as any other rule already in the rule-base, then it is added to the rule-base.

Rule-base generated from LFE procedure was then used for prediction of the output values. For input data case $(\boldsymbol{x}, y)$, let $\mu_{X_j}(x_j)$ denote membership value of the input variable x_j. Membership values that input data case $(\boldsymbol{x}, y)$ has in the ith rule are multiplied and results in

$$\mu_i(\boldsymbol{x}) = \prod_{j=1}^{n} \mu_{X_j}(x_j), \qquad (3-25)$$

where n is the number of input variables. The regression vector $\boldsymbol{\xi}$ is defined as

$$\xi_i(\boldsymbol{x}) = \frac{\mu_i(\boldsymbol{x})}{\sum_{i=1}^{R} \mu_i(\boldsymbol{x})}, \qquad (3-26)$$

where R is the number of rules in rule-base. The resulting mapping is

$$f(\boldsymbol{x}|\widehat{\boldsymbol{\theta}}) = \widehat{\boldsymbol{\theta}}^{\boldsymbol{T}} \boldsymbol{\xi}(\boldsymbol{x}), \qquad (3-27)$$

where $\widehat{\boldsymbol{\theta}}$ is the vector to be estimated, and $\widehat{\boldsymbol{\theta}}^{\boldsymbol{T}}$ is the transpose.

Recursive least squares (RLS) algorithm (Passino and Yurkovich, 1998) was used to estimate $\widehat{\boldsymbol{\theta}}$. As presented earlier, the output variable is binary and can either be 1 or 0. However, the de-fuzzification procedure may result in continuous output values. Therefore, in order to generate binary outputs, it is reasonable to choose a threshold value of 0.5 to classify the output values into two classes.

3.4.3. Advantages and Limitations of Fuzzy Logic Model

Fuzzy logic does not require detailed mathematical modeling and is especially suitable for modeling the remarkable ability of the human mind to learn and make rational decisions in an environment of uncertainty and imprecision. It relates input and output in linguistic rules, thus, it is easily understood. It simplifies knowledge acquisition and representation. One limitation of fuzzy logic is that it require more fine tuning and simulation before implementation.

3.5. Multilayer Feedforward Neural Network

Neural networks are powerful learning methods with widespread applications in many fields. The central idea of neural networks is to extract linear combinations of the input variables as derived features and then model the output variables as a function of linear combinations of the derived features. Multilayer feedforward neural network consist of two stages of regression. It is typically represented by the diagram in **FIGURE 3-5**. The neurons in the first layer perform computations of the first stage and constitute hidden-layer. The single neuron in the second layer performs the computation of the second stage and constitutes the output layer.

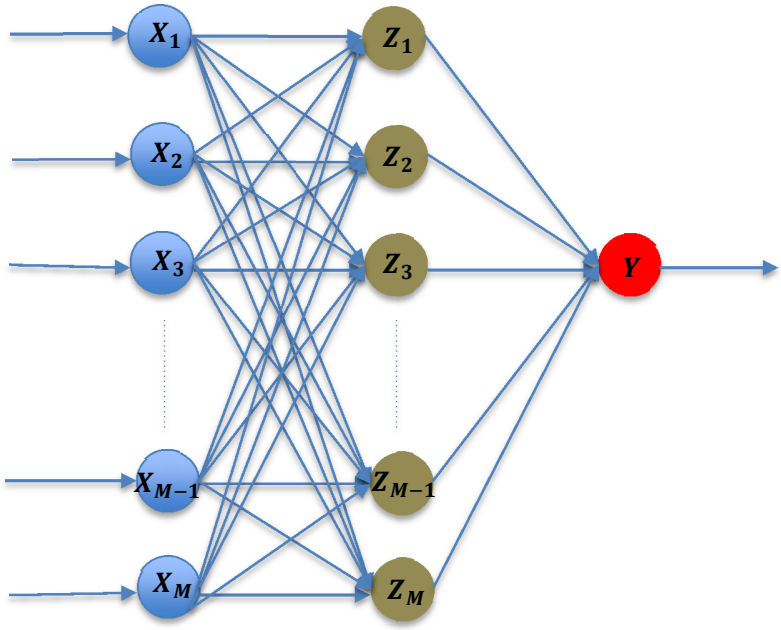


FIGURE 3-5 Schematic of a multilayer feedforward neural network with one hidden layer.

The linear combinations of the input variable create new features Z_m, and then the output Y is modeled as a function of linear combinations of Z_m. Z_m are called hidden unit because the values Z_m are not directly observed. In general there can be more than one hidden layer.

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X), m = 1, \dots, M, \quad (3-28)$$

$$Y = g(\beta_0 + \beta^T Z), \quad (3-29)$$

where $X = (X_1, X_2, \dots, X_M)$ and $Z = (Z_1, Z_2, \dots, Z_M)$. The activation function $\sigma(x)$ is usually chosen as sigmoid $\sigma(x) = 1/(1 + \exp(-x))$. For regression, the output function $g(x)$ is usually chosen as identity function $g(x) = x$.

There are several advantages of using neural networks:

- It is very flexible. It is able to approximate any non-linear relationship between input and output variables. This applies to problems where the relationship may be quite dynamic or non-linear.
- It provide alternative to conventional techniques which are often limited by assumptions of normality, linearity, or variable independence.
- It can deal with both categorical and numerical data and it is simple to implement.

The major concerns about neural networks are:

- A number of parameters need to be chosen before training, such as number of hidden layers, number of nodes in hidden layers, selection of activation function, starting value for weights and learning rates.
- Neural network is often considered as "black box" model, because the results of the model are hard to interpret.

3.6. Nearest Neighbor Nonparametric Regression

Nonparametric regression describes the relationship between dependent and independent variables based on data. The nearest neighbor nonparametric regression is a pattern matching method that matches the current observations with those in a database of historical observations. The model is summarized as follows. Given a distance measure and forecast calculation, then:

1. Identify *k* nearest neighbors from training dataset for an unknown data case.
2. Forecast for the unknown data case using the forecast calculation based on the dependent values of the selected neighbors.

For this study, a common distance measure – Euclidean distance is used to identify k nearest neighbors. It is formulated as

$$D(\boldsymbol{x_j}, \boldsymbol{x_k}) = \sqrt{\sum_{i=1}^{l} (x_{ji} - x_{ki})^2}, \qquad (3-30)$$

where $\boldsymbol{x_j}$ and $\boldsymbol{x_k}$ are two data cases of l independent variables. The weighted by inverse of distance approach is selected for forecast calculation. It is calculated as follow:

$$\hat{y} = \sum_{i=1}^{k} \frac{y_i}{D_i} \Big/ \sum_{i=1}^{k} \frac{1}{D_i}, \qquad (3-31)$$

where $\hat{y}$ is the predicted dependent variable, y_i is the dependent value of the *i*th nearest neighbor, and D_i is the distance between the *i*th nearest neighbor and the unknown data case.

Nearest neighbor nonparametric regression is very simple to understand and easy to implement. It can be updated online at very little cost as new instance with known classes are presented. There are only a few parameters to tune: distance measure and k. On the other hand, there are some disadvantages. The major limitation of nearest neighbor nonparametric regression is high computational burden required if the training data is large. It is very sensitive to irrelevant or redundant features because all features contribute to the similarity.

3.7. AdaBoost

Boosting (Theodoridis and Koutroumbas, 2006) is one of the most powerful learning ideas introduced in the last two decades. Although boosting can be considered an ensemble

method, it is conceptually different from bagging. The motivation of boosting was to boost a "weak" learning algorithm into a "strong" algorithm with higher classification accuracy. At the core of a boosting method is the "weak" classifier, also called base classifier. A sequence of base classifiers is designed iteratively using different subsets of the training set with an iteratively-computed weighting distribution. At each iteration, the weighting distribution emphasizes the cases that are misclassified by the previous base classifier. The final boosting classifier is a weighted average of previously designed base classifiers.

AdaBoost or adaptive boosting is a popular algorithm from the family of boosting algorithms and one that has been extensively studied. The detailed training procedure of "AdaBoost.M1" (Freund and Schapire, 1996; Freund and Schapire, 1997) is described as follows. For a binary classification problem, let the training data be $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ with class labels $y_i \in Y = \{-1, 1\}$. Let w_i^t denote the weight of ith data point for the tth iteration and $h_t(x)$ denote the base classifier for the tth iteration. Define function $I(a) = 1$ if a is true, otherwise $I(a) = 0$.

1. Initialize the weights for training data: $w_i^1 = 1/N$, $i = 1,2, \dots, N$
2. For $t = 1$ to T:
    a. Set

$$p_i^t = \frac{w_i^t}{\sum_{i=1}^{N} w_i^t} \qquad (3-32)$$

    b. Fit a classifier $h_t(x)$ to the training data using weights w_i^t.
    c. Calculate the error of $h_t(x)$:

$$\epsilon_t = \sum_{i=1}^{N} p_i^t \cdot I(h_t(x_i) \neq y_i) \quad (3-33)$$

d. Compute $\beta_t = \epsilon_t/(1-\epsilon_t)$.

e. Update weights for training data:

$$w_i^{t+1} = w_i^t \beta_t^{\,1-I(h_t(x_i)\neq y_i)} \quad (3-34)$$

3. Output the final classifier:

$$H(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \left(\log \frac{1}{\beta_t}\right) h_t(x) \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (3-35)$$

In this study, the classification and regression tree (CART) (Breiman, 1984) was used as the base classifier $h_t(x)$. CART was used because it is easy to interpret and is relatively fast to build. CART can be applied to mixtures of numeric and categorical variables, and missing values. It is also immune to the effect of outliers and resistant to the inclusion of irrelevant input variables.

Some advantages of AdaBoost are as follow:

- It is immune to overfitting. With the increase in the number of base classifiers, the error rate on test data continue to decrease and finally levels off at a certain value.
- The base classifiers are usually simple models such as a decision stump (decision tree with one node), thus, it is simple and easy to program.
- It can achieve accurate classification with much less tweaking of parameters and settings as compared to other methods.

On the other hand, there is one major limitation about AdaBoost:

- AdaBoost is sensitive to noisy data and outliers.

CHAPTER 4: EXPERIMENTAL DESIGN AND RESULTS

In this section, the details of data collection and modeling process for both driver lane-changing behavior and traffic demand forecast are described. First, five AI classification methods, Bayes classifier, classification tree, genetic fuzzy system, random forest and AdaBoost were used to model driver lane-changing behavior. The modeling process for each model is presented, and the prediction results are compared and interpreted. Then, four methods, multilayer feedforward neural network, nonparametric regression, regression tree, and random forest, were developed for forecasting traffic flow for planned work zone events. Both daily and hourly traffic flow forecasting models were investigated. Daily prediction allows TMCs to provide a traffic operation plan before a work zone is deployed. The short-term prediction allows real-time traffic control in work zones. While daily forecasts were made 24 hours in advance using historical traffic data, short-term forecasts were made 1 hour, 45 minutes, 30 minutes, and 15 minutes in advance using real-time temporal and spatial traffic data. The short-term forecast models were further tested on special event traffic data.

4.1. Modeling Driver Lane-Changing Behavior at Natural Lane Reduction

4.1.1. Data Reduction

Traffic data provided by the Federal Highway Administration's (FHWA) Next Generation Simulation (NGSIM) project (Federal Highway Administration, 2011) was used to build the lane-changing models. NGSIM dataset is an open source dataset that has been used in

previous research for simulation model development and testing (Choudhury et al., 2009; Yeo et al., 2008). NGSIM data include vehicle trajectories on a segment of southbound US Highway 101 (Hollywood Freeway) in Los Angeles, California and a segment of Interstate 80 in San Francisco, California. US Highway 101 data was collected for 45 minutes from 7:50 a.m. to 8:35 a.m., on June 15, 2005. Interstate 80 data was also collected for 45 minutes from 4:00 p.m. to 4:15 p.m. and from 5:00 p.m. to 5:30 p.m., on April 13, 2005. Both datasets represents two traffic states – conditions when congestion was building up (period of the first 15 minutes) denoted as the transition period and congested conditions (period of the remaining 30 minutes). **TABLE 4-1** shows the aggregate speed and volume statistics of NGSIM dataset for every 15 minutes. For the congested period, the flows and speeds both decreased. As depicted in **FIGURE 4-1**, the study segment of US Highway 101 was located between an on-ramp and off-ramp and was 2100 feet long with five freeway lanes and an auxiliary lane. The study segment of Interstate 80 was 1650 feet in length, and also had five freeway lanes and an auxiliary lane, and one on-ramp. The driver's merging behavior from auxiliary lane to the adjacent lane is the lane-changing behavior occurred at bottleneck. Given the focus of this study on bottleneck lane changes, only trajectory data of vehicles in the auxiliary lane and the adjacent lane were used for model development. Hereafter, the auxiliary lane is referred to as the merge lane and the adjacent lane as the target lane.

Past research studies (Punzo et al., 2009; Kesting and Treiber 2008; Duret et al., 2008; Ossen and Hoogendoorn et al., 2008) have shown that NGSIM speed measurements exhibit noises (random errors). Data smoothing techniques such as moving average (Ossen and Hoogendoorn et al., 2008), Kalman filtering (Punzo et al., 2005) and Kalman smoothing (Ma and

Andreasson, 2007) have been used to improve speed data quality. In this study, the moving average method was adopted to smooth the speed measurements.

The longitudinal and lateral coordinates, speed, acceleration, and headway for each vehicle were obtained from trajectory data at a resolution of 10 frames per second. The speed and position of each vehicle were identified in 1-second intervals. The 1-second interval produced data with comparable sample sizes for both lane changing and non-lane-changing events. Other researchers (Meng and Weng, 2012) have also used a 1-second interval for analyzing lane-changing behavior of drivers. Since it is impossible to determine the intent of the driver using vehicle trajectory data alone, the observed behavior of drivers is modeled. During every 1-second interval, a driver's behavior is identified as either merge or no-merge. Merge events occurred when a vehicle's lateral coordinate began to shift toward the adjacent target lane direction without oscillations. Otherwise it was deemed as a non-merge event. A single driver could participate in several non-merge events but only one merge event.

A total of 686 observations were obtained from US Highway 101, 373 of them being non-merge and 313 of them being merge events. As discussed in Hastie et al. (2001), there is no general rule on how many observations should be assigned to training and validation. In order to obtain high accuracy, a large training data size is required. Other studies have used 80% of the dataset for training and 20% for validating the model (Martin et al., 2012, Edara et al., 2007). Based on these studies, the dataset was divided into two groups – 80% of observations were used for training and 20% were used for validation. The model was tested using the Interstate 80 dataset consisting of 667 observations, 459 of them being non-merge and 208 of them being merge events.

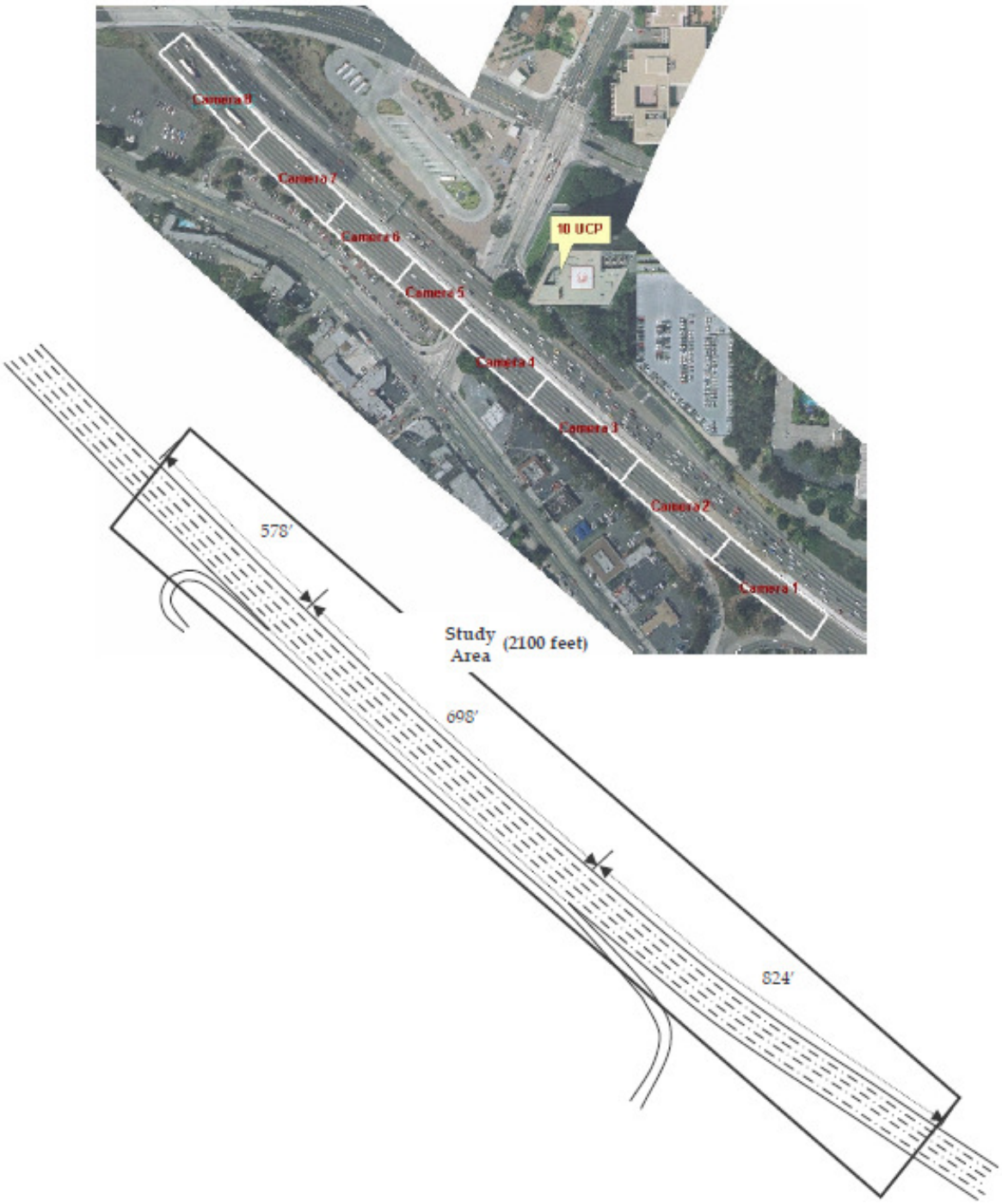
TABLE 4-1 Summary Statistics

A. Summary Statistics of US Highway 101 Dataset (Cambridge Systematics Inc., 2005)

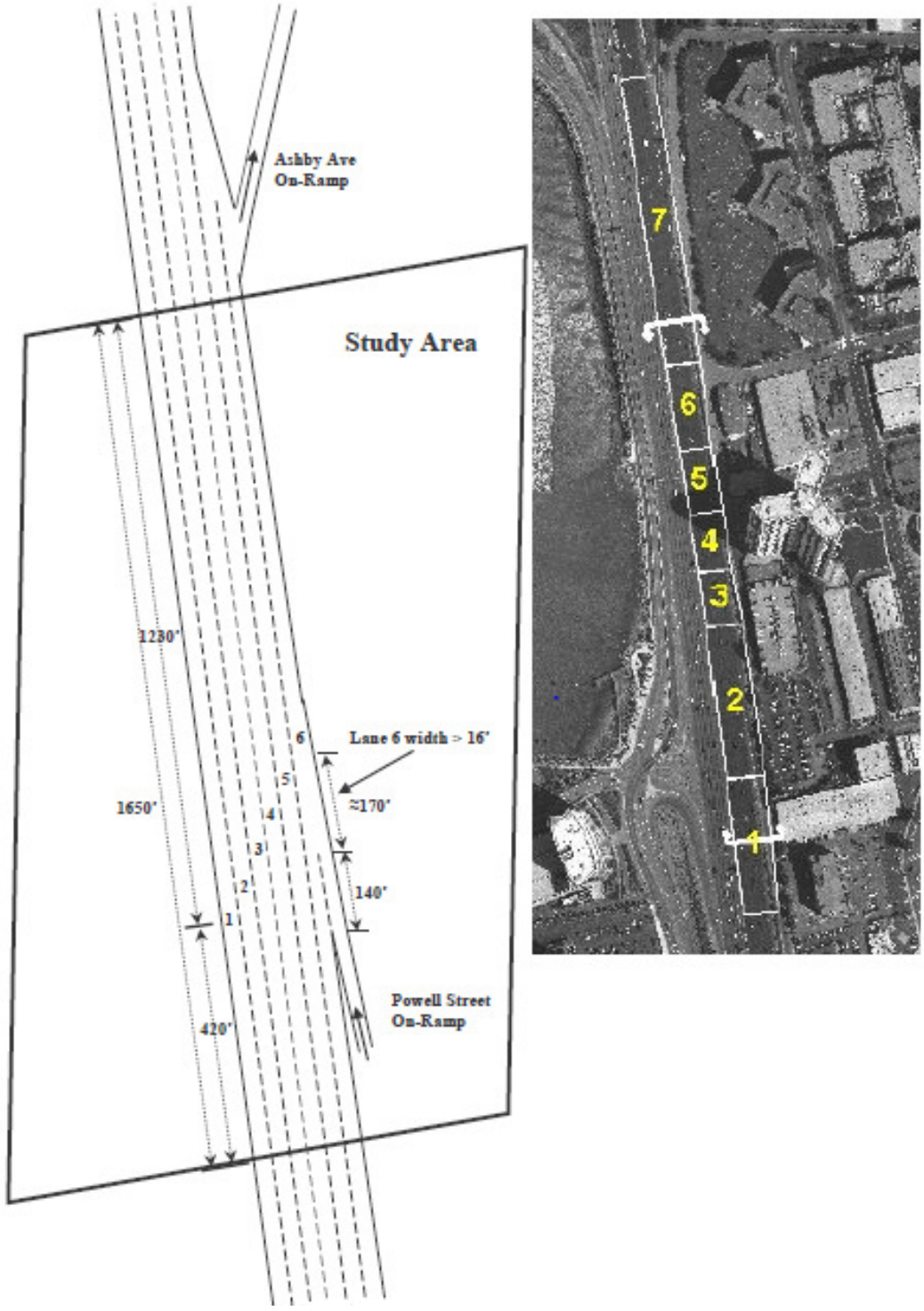
Traffic condition	Time period	Flow (vph)	Time mean speed	
			m/s	km/h
Transition	7:50 a.m. – 8:05a.m.	8612	12.55	45.16
Congested	8:05 a.m. – 8:20a.m.	8016	11.10	39.96
	8:20 a.m. – 8:35a.m.	7604	9.74	35.05

B. Summary Statistics of Interstate 80 Dataset (Cambridge Systematics Inc., 2005)

Traffic condition	Time period	Flow (vph)	Time mean speed	
			m/s	km/h
Transition	4:00 p.m. – 4:15p.m.	8144	9.92	35.71
Congested	5:00 p.m. – 5:15p.m.	7288	8.34	30.13
	5:15 a.m. – 5:30a.m.	7048	7.78	28.00


Camera 8
Camera 7
Camera 6
Camera 5
Camera 4
Camera 3
Camera 2
Camera 1
10 UCP
578'
Study Area (2100 feet)
698'
824'

(a)



(b)

FIGURE 4-1 US highway 101 (a) and Interstate 80 (b) study corridor from NGSIM (Federal Highway Administration, 2011).

4.1.2. Input Variables

At any given instant, a driver traveling in the merge lane assesses traffic conditions in both the target lane and the merge lane in order to decide whether to merge or not. Several factors may affect a driver's lane-changing decision. In this study, five factors or dimensions that were found to affect a driver's merging decision in previous studies (Ahmed et al., 1996; Hidas, 2005) were considered as input variables for the models. These factors are shown in **FIGURE 4-2** and defined below.

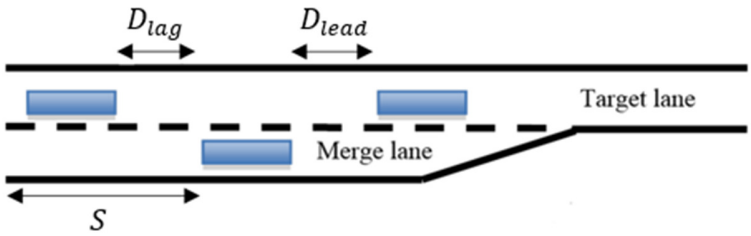


FIGURE 4-2 Schematic illustrating input variables.

- $\Delta V_{lead}(\mathrm{m/s})$: The speed difference between the lead vehicle in the target lane and the merging vehicle, in feet per second. ΔV_{lead} is

$$\Delta V_{lead} = V_{lead} - V_{merge}, \qquad (4-1)$$

where V_{lead} is the speed of the lead vehicle and V_{merge} is the speed of merge vehicle.

- $\Delta V_{lag}(\mathrm{m/s})$: The speed difference between the lag vehicle in the target lane and the merging vehicle, in feet per second. $\Delta \mathrm{V_{Lag}}$ is

$$\Delta V_{lag} = V_{lag} - V_{merge}, \qquad (4-2)$$

where $\mathrm{V_{Lag}}$ is the speed of the lag vehicle.

- D_{lead}(m): The gap distance between the lead vehicle in the target lane and the merging vehicle, in feet.
- $\mathrm{D_{Lag}}$(ft): The gap distance between the lag vehicle in the target lane and the merging vehicle, in feet.
- S(m): The distance from the merging vehicle to the end of the merge lane.

4.1.3. Model Results

4.1.3.1. Bayes Classifier

For the Bayes classifier the weights were first estimated using SVMs. The estimated weights shown in **TABLE 4-2** reveals that ΔV_{lead} has the largest weight, which indicates ΔV_{lead} is the most relevant feature in classifying the merge and non-merge events, indicating that a slight change in ΔV_{lead} may greatly change the distance. Speed differences ΔV_{lead} and ΔV_{lag} are more relevant than lead gap D_{lead} and lag gap D_{lag}. The distance from the beginning of the merge (auxiliary) lane, S, turned out to be the least relevant feature.

TABLE 4-2 Weight Given by SVMs

Variables	Weights
ΔV_{lead}	0.7243
ΔV_{lag}	0.2449
D_{lead}	0.0292
D_{lag}	0.0012
S	0.0005

A Bayes classifier was developed from $k = 3$ and $\frac{\lambda_{21}}{\lambda_{12}} = 1$. The model's prediction accuracy on test data for merge and non-merge events is shown in **TABLE 4-3**. The accuracy of merge events was 92.3% and it was 79.3% for the non-merge events. The accuracy for overall data was 83.5%.

TABLE 4-3 Accuracy of Bayes Classifier for Test Data

		Predicted Class		Classification Accuracy	Overall Accuracy
		Merge	Non-merge		
Actual Class	Merge	165	43	79.5%	83.5%
	Non-merge	424	35	92.3%	

4.1.3.2. Classification Tree

A classification tree with 62 terminal nodes was constructed using training data before pruning. After applying the pruning rules, a sequence of 16 minimal cost-complexity trees were generated. The total numbers of terminal nodes $|\widetilde{T_k}|$ are shown in **TABLE 4-4**.

TABLE 4-4 Number of Terminal Nodes in Minimal Cost-Complexity Trees

Tree	$\|\widetilde{T_k}\|$
T_1	62
T_2	58
T_3	53
T_4	46
T_5	29
T_6	27
T_7	22
T_8	18
T_9	12
T_{10}	10
T_{11}	9
T_{12}	7
T_{13}	5
T_{14}	3
T_{15}	2
T_{16}	1

The relationship between total number of terminal nodes $|\widetilde{T_k}|$ and estimated misclassification rate for both training and testing data is presented in **FIGURE 4-3**. In **FIGURE 4-3** the estimated misclassification rate for training data $R(T_k)$ decreased sharply as

the tree initially increased in size and then decreased slowly. The estimated misclassification rate for testing data $R^{ts}(T_k)$ also decreased sharply initially, but after reaching its minimum value at 18 terminal nodes, the rate began to climb as tree size grows. Thus, the tree T_8 with 18 terminal nodes was selected as the right size classification tree model for predicting merge and non-merge events.

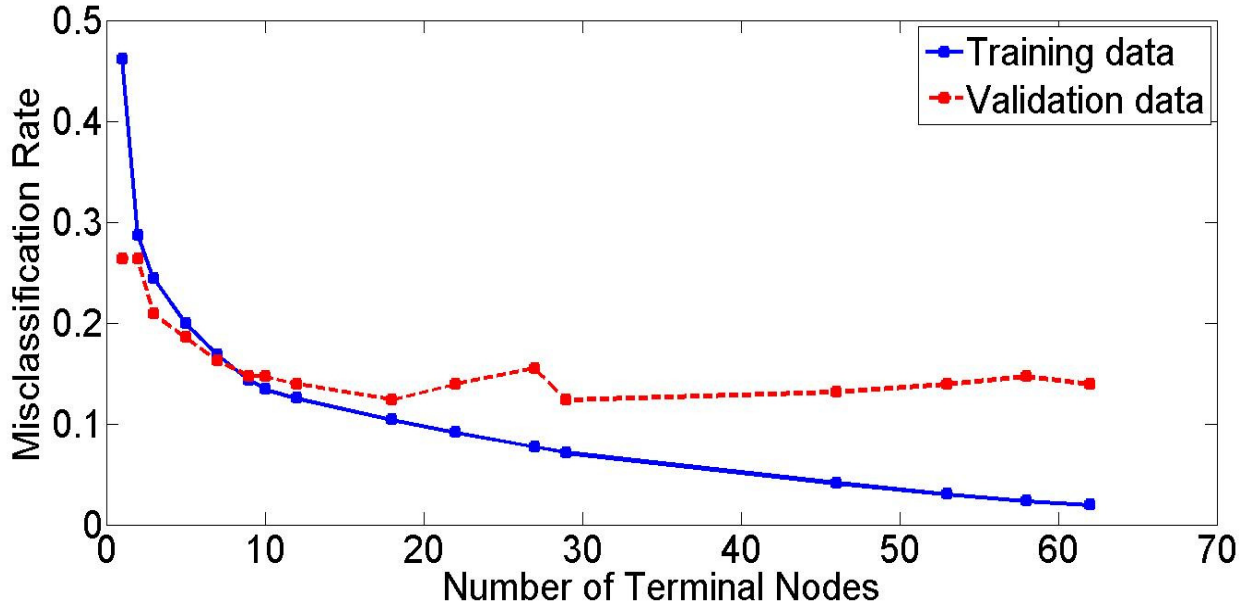


FIGURE 4-3 Relationship between number of terminal nodes and misclassification rate.

The tree structure is presented in **FIGURE 4-4**, where terminal nodes are represented by shaded squares and decision nodes are represented by circles. Number of observations, class labels, and prediction accuracies for terminal nodes are displayed beneath them. Node 1 was first split by using the relative speed between the lead and merging vehicles, ΔV_{lead}. This result further supports the finding from the Bayes classification model that ΔV_{lead} is the most relevant driver feature in making merging decisions. The decision making process of the classification tree model is intuitive. For example, as shown by terminal node t_8, a driver merges if the

merging vehicle is slower ($\Delta V_{lead} \geq 0\ m/s$) or slightly faster ($0 > \Delta V_{lead} \geq -2.7\ m/s$) than the lead vehicle and both the lead and lag gap is large ($D_{lag} \geq 2.4\ m$, $D_{lead} \geq 7.6\ m$). In contrast, terminal node t_7 is interpreted in natural language as: if the merging vehicle is much faster ($\Delta V_{lead} < -2.7\ m/s$) than the lead vehicle and the lead gap is small ($D_{lead} < 8.9\ m$), then the driver does not merge. For terminal node t_{14}, if the merging vehicle speed is much greater ($\Delta V_{lead} \geq -2.7\ m/s$) than the lead vehicle; lead gap is large ($D_{lead} \geq 8.9\ m$); distance from the end of the merge lane is far ($S \geq 138.7\ m$); and lag gap is not too large ($D_{lag} \geq 0.76\ m$); driver decides to merge, because as driver approaches the end of merge lane his or her merge behavior become more aggressive. These rules generated by classification tree are representative of everyday driving experiences. The prediction results of classification tree are presented in **TABLE 4-5**. The accuracy of both merge and non-merge events were 80.8% and 84.3%. The overall classification accuracy was 83.2%.

TABLE 4-5 Accuracy of Classification Tree for Test Data

		Predicted Class		**Classification Accuracy**	**Overall Accuracy**
		Merge	**Non-merge**		
Actual Class	**Merge**	168	40	80.8%	83.2%
	Non-merge	387	72	84.3%	

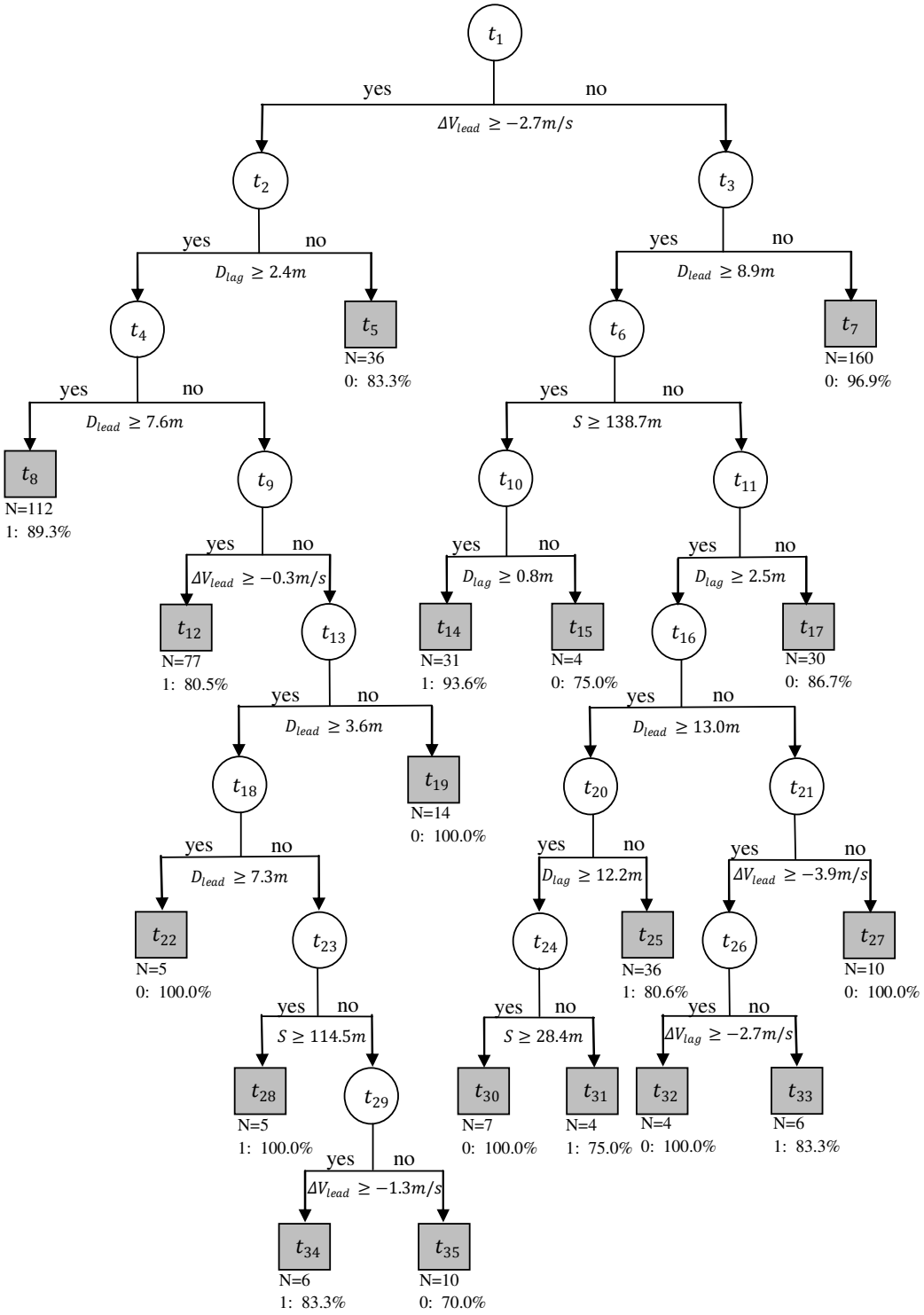
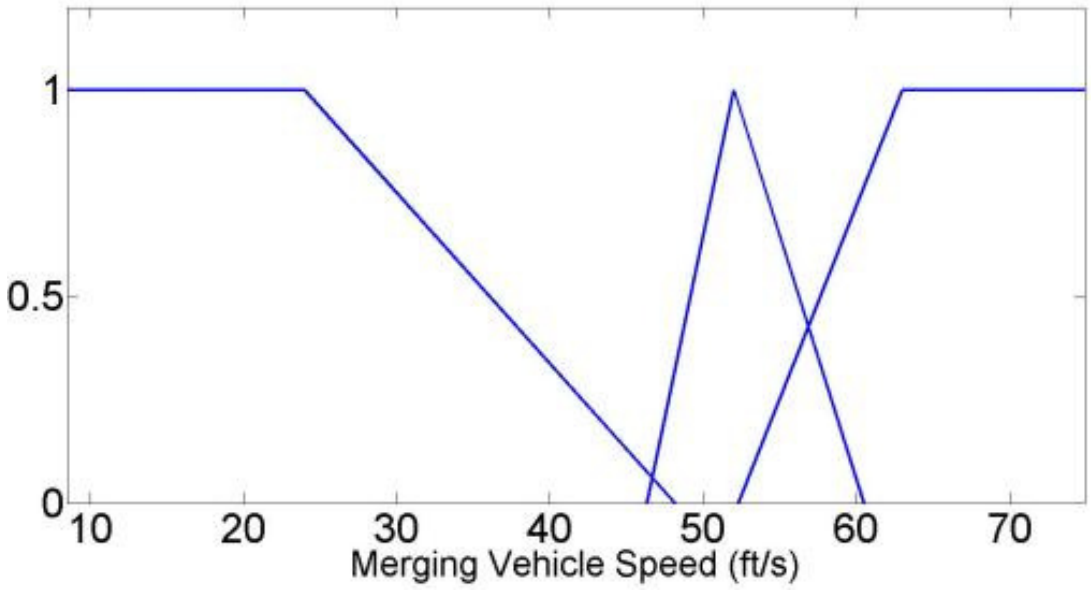


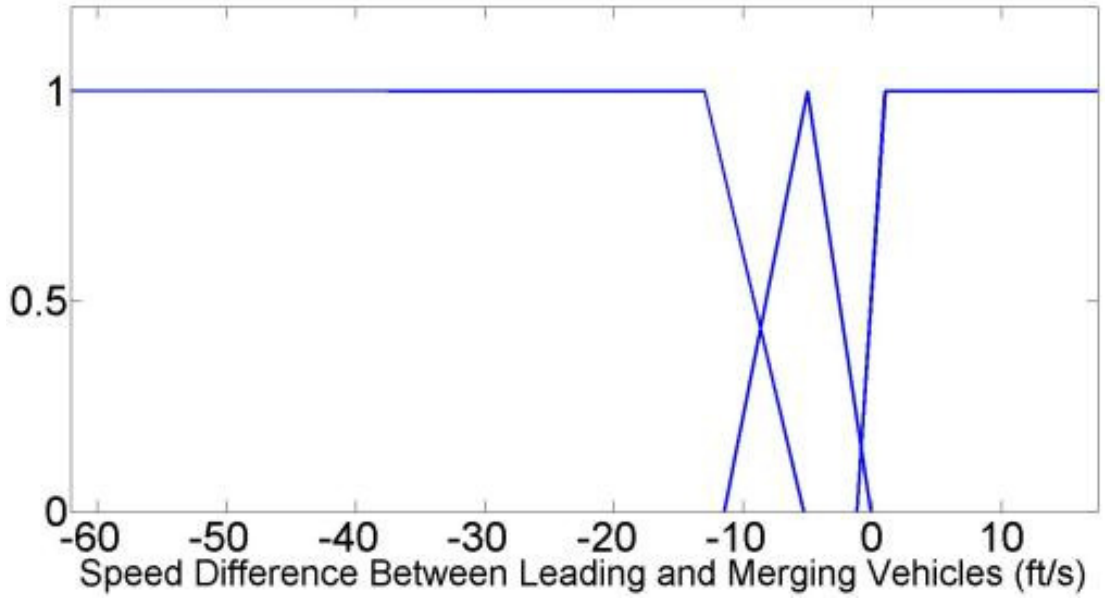
FIGURE 4-4 Classification tree model structure.

4.1.3.3. Genetic Fuzzy System

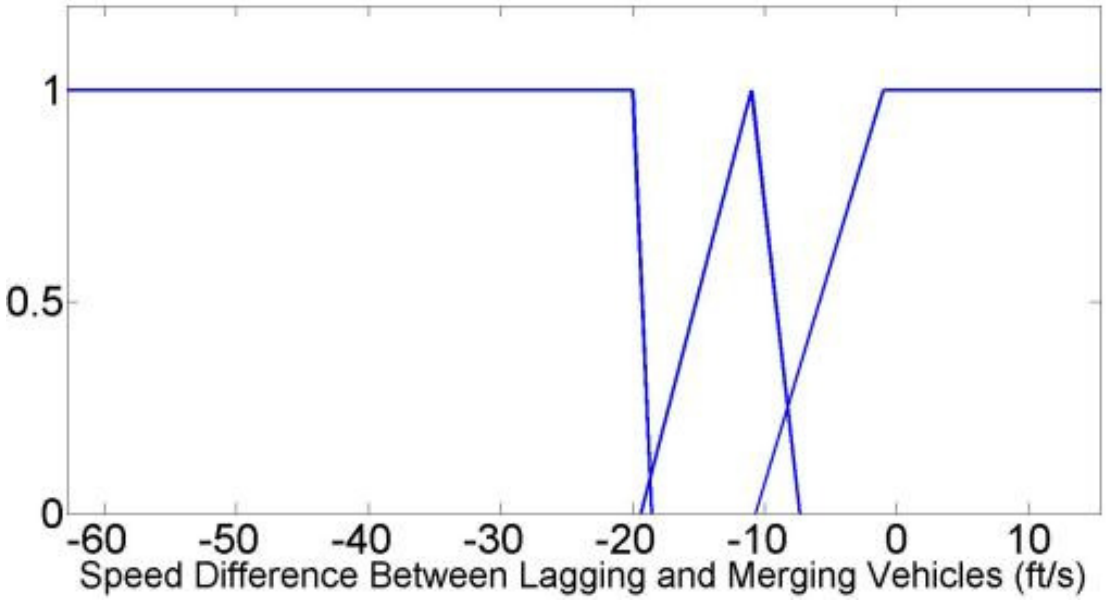
A population size of 20 individuals per generation and 50 generations was used in GA. Scattered crossover technique was used with crossover probability of 0.8. There are three membership functions for each input variable. The membership functions resulting from GA optimization are shown in **FIGURE 4-5** for the input variables. A total of 120 rules were generated from the training data. The performance of the fuzzy logic model was measured based on the number of instances in which the model prediction was identical to the observed output value. The performance was measured for both merge and non-merge events. **TABLE 4-6** shows the results for test data. The classification accuracy for both merge and non-merge events were 71.6% and 73.6%. The classification accuracy for overall data was 73.0%.



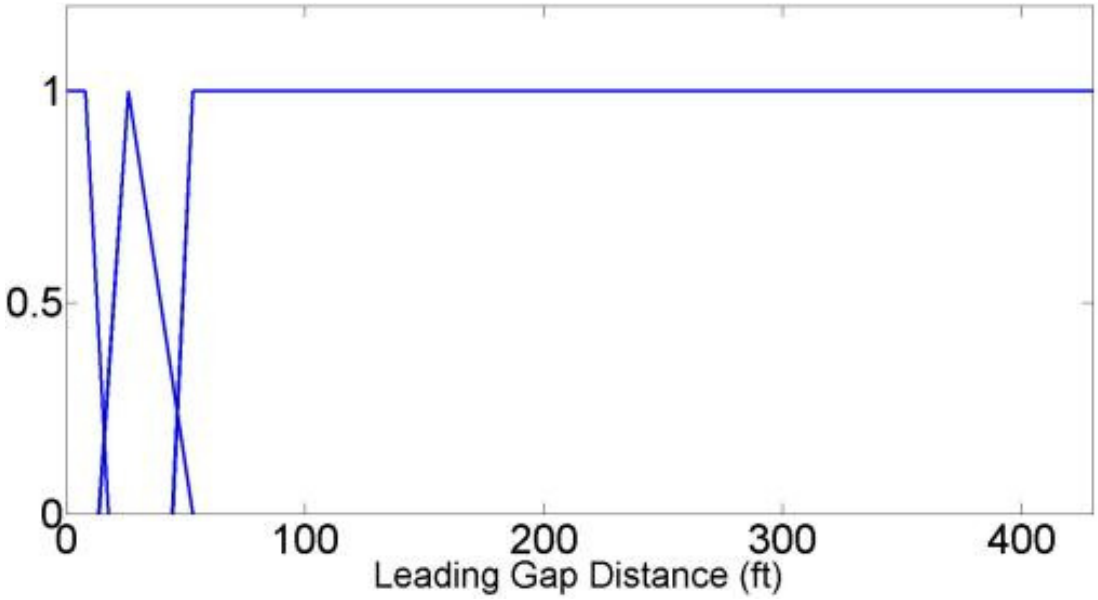
(a)


1
0.5
0
-60
-50
-40
-30
-20
-10
0
10
Speed Difference Between Leading and Merging Vehicles (ft/s)

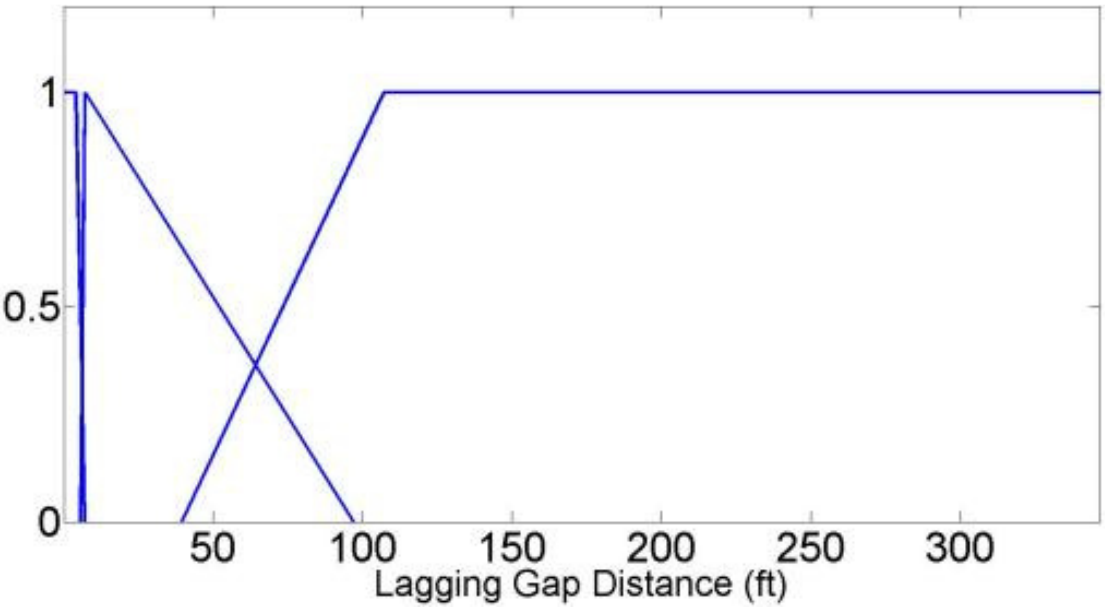
(b)


1
0.5
0
-60
-50
-40
-30
-20
-10
0
10
Speed Difference Between Lagging and Merging Vehicles (ft/s)

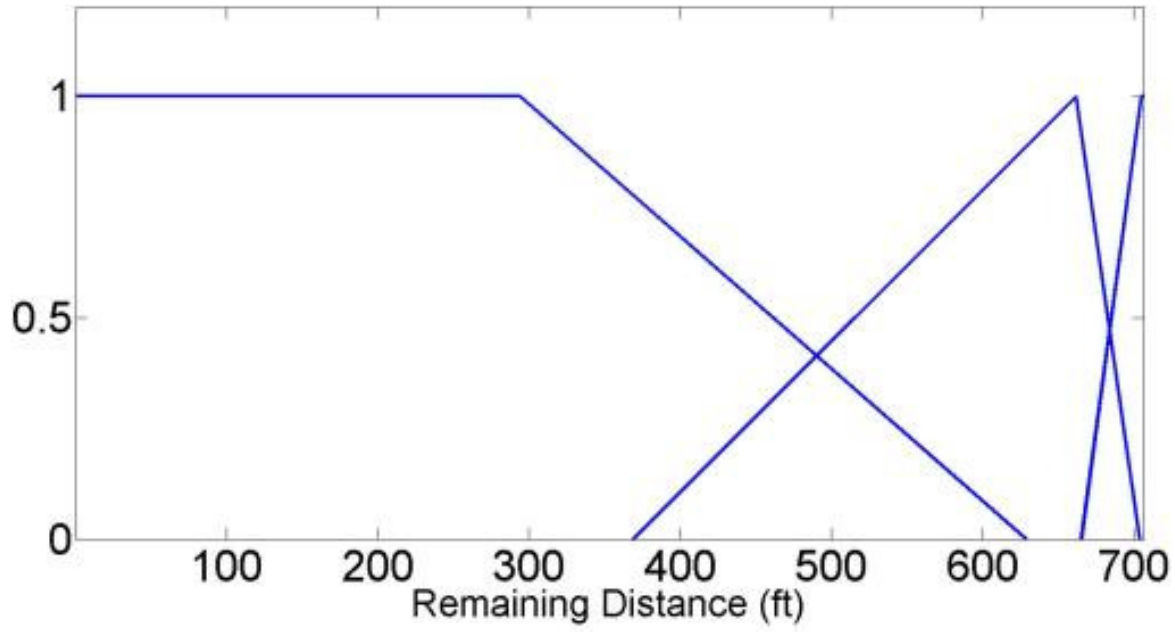
(c)


1
0.5
0
0
100
200
300
400
Leading Gap Distance (ft)

(d)


1
0.5
0
50
100
150
200
250
300
Lagging Gap Distance (ft)

(e)



(f)

FIGURE 4-5 Input membership functions, (a) merging vehicle speed, (b) speed difference between leading vehicle and merging vehicle, (c)speed difference between lagging vehicle and merging vehicle, (d) leading gap distance, (e) lagging gap distance, (f)the remaining distance to the end of the merge lane.

TABLE 4-6 Accuracy of Genetic Fuzzy System for Test Data

		Predicted Class		Classification Accuracy	Overall Accuracy
		Merge	Non-merge		
Actual Class	Merge	149	59	71.6%	73.0%
	Non-merge	338	121	73.6%	

4.1.3.4. Random Forest

The process of random forest development involves the fine-tuning of three parameters: minimum terminal node size, number of random selected variables at each node split, and number of trees. After an extensive trial-and-error process, the best performing random forest was produced using 150 trees with a minimum terminal node size of 1 and three randomly selected variables at each node split. The learning process of random forest is shown in **FIGURE 4-6**. **FIGURE 4-6** shows that the OOB error tends to be stable after approximately 100 trees. The classification results for test data are shown in **TABLE 4-7**. Random forest produced accuracy rates of 87.8% for non-merge events and 89.4% for merge events. The total classification accuracy was 88.3%.

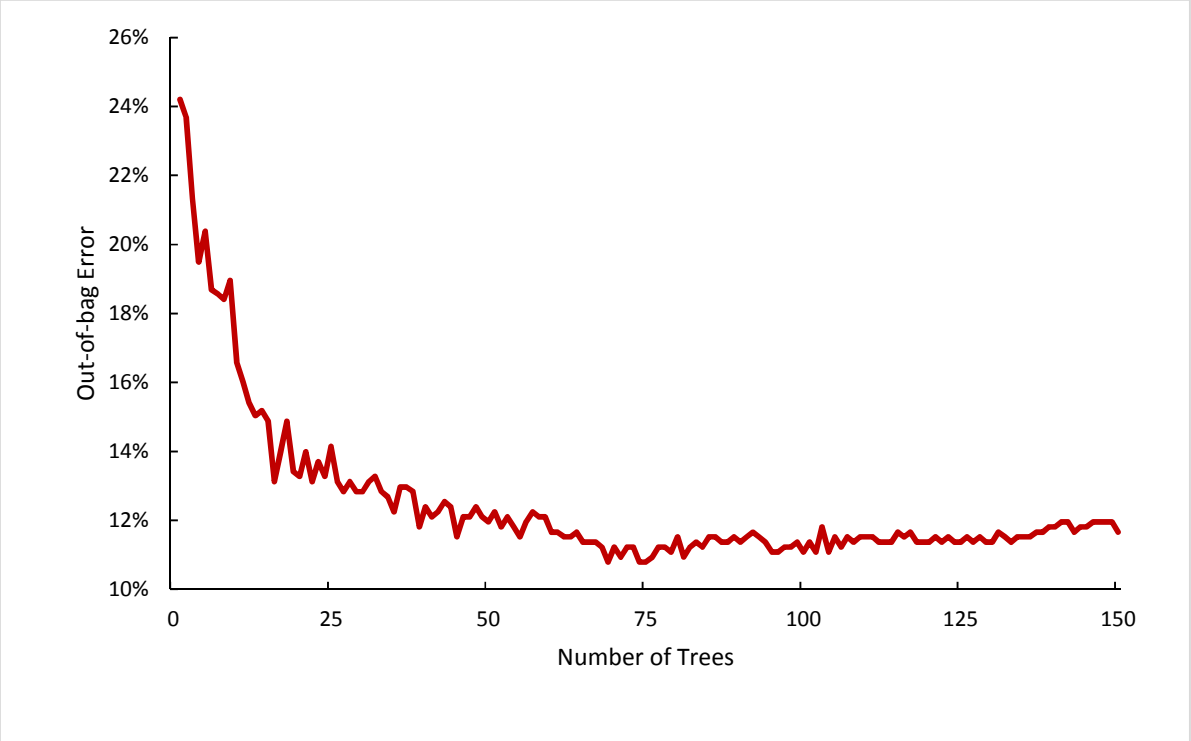


FIGURE 4-6 The learning process of random forest.

TABLE 4-7 Model Performance of Random Forest on Test Data

		Predicted Class		Classification Accuracy	Total Accuracy
		Merge	**Non-merge**		
Actual Class	**Merge**	186	22	89.4%	88.3%
	Non-merge	56	403	87.8%	

4.1.3.5. AdaBoost

A five-fold cross-validation was performed to estimate the classification accuracy. The data was split into five equal-sized parts. For the kth part, the model was fitted to other four parts of the data, and the classification accuracy of the fitted model for the kth part of the data was calculated. The process was repeated for $k = 1,2, \dots ,5$. Then, the five estimates of classification accuracy were combined to estimate the overall classification accuracy of the model. Two model parameters, the number of trees and minimum branch node size, were tuned to optimize model performance. After an extensive trial-and-error process, 150 trees with minimum branch node size being $1/7^{th}$ of the training set yielded the minimum cross-validation classification error. **FIGURE 4-7** displays the learning process of AdaBoost. **FIGURE 4-7** shows that the cross-validation error tends to be stable after approximately 140 trees. The classification results for test data are shown in **TABLE 4-8**. AdaBoost produced accuracy rates of 88.7% for non-merge events and 89.4% for merge events. The total classification accuracy was 88.9%.

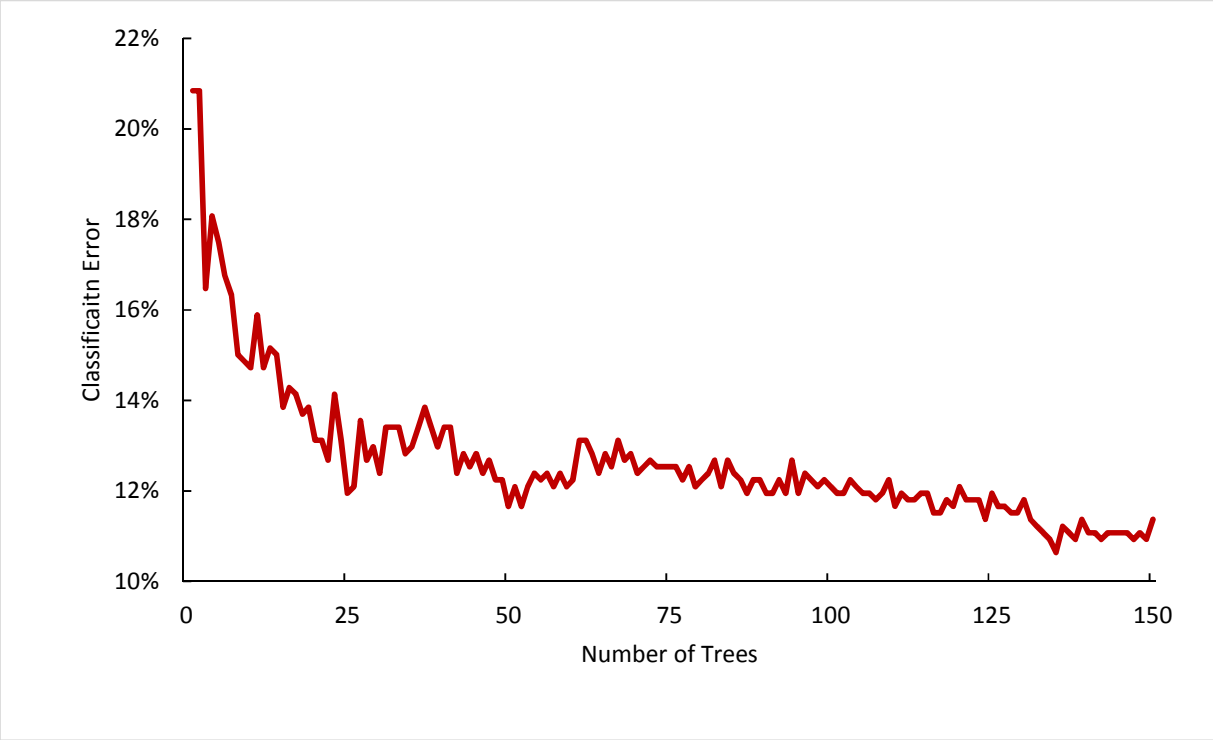


FIGURE 4-7 The learning process of AdaBoost.

TABLE 4-8 Model Performance of AdaBoost on Test Data

		Predicted Class		Classification Accuracy	Total Accuracy
		Merge	**Non-merge**		
Actual Class	**Merge**	186	22	89.4%	88.9%
	Non-merge	52	407	88.7%	

4.1.4. Model Comparison

One of the models from the literature, Binary Logit model, was also evaluated as a comparison. It was estimated using the same dataset and the same set of variables. The binary logit model is given by the following functions:

$$P = \frac{e^{y}}{1 + e^{y}}, \qquad (4-1)$$

where:

$y = \sum_k \beta_k x_k + c$;

P is the probability of choosing an alternative;

x_k is the kth explanatory variable;

β_k is the coefficient of the kth explanatory variable; and

c is the constant term.

For this model, regression coefficients were estimated by the method of maximum likelihood estimation. The coefficients of the Binary Logit model are presented in **TABLE 4-9**. The performance of Binary Logit model for the test data is displayed in **TABLE 4-10**.

TABLE 4-9 Coefficients of Binary Logit Model

Variable	Coefficient	p-value
$\Delta V_{lead}(\mathrm{m/s})$	0.163	<.0001
$\Delta V_{lag}(\mathrm{m/s})$	0.070	0.0043
$D_{lead}(\mathrm{m})$	0.061	<.0001
D_{lag} (m)	0.003	0.4721
$S(\mathrm{m})$	-0.004	<.0001
Intercept	1.967	<.0001

* Not significant at 0.05 significance level

TABLE 4-10 Accuracy of Binary Logit Model for Test Data

		Predicted Class		Classification Accuracy	Total Accuracy
		Merge	Non-merge		
Actual Class	Merge	199	9	95.7%	44.2%
	Non-merge	96	363	20.9%	

The performances of all proposed models and Binary Logit model for the test data is compared in **TABLE 4-11**. The Binary Logit model performed poorly compared to the models developed in the study. Although the prediction accuracy for merge events was 95.7%, the

prediction accuracy for critical non-merge events was only 20.9% and the accuracy for the whole test dataset was only 44.2% which was much less than the accuracies of proposed models. Random forest and AdaBoost had the highest prediction accuracy rate of 88.3% and 88.9% for the whole test dataset.

TABLE 4-11 Predicted Results Comparison

Models	Classification Accuracy		
	Non-merge	Merge	Overall
Bayes Classifier	79.5%	92.3%	83.5%
Classification Tree	84.3%	80.8%	83.2%
Genetic Fuzzy	73.6%	71.6%	73.0%
Random Forest	87.8%	89.4%	88.3%
Adaboost	88.7%	89.4%	88.9%
Binary Logit	20.9%	95.7%	44.2%

4.2. Traffic Demand Forecasting for Work Zone Bottleneck

4.2.1. Data Collection

In this dissertation, detailed urban work zone and traffic volume data were collected by MoDOT and detectors aligned with work zones on two different types of roadways in St. Louis,

Missouri. One was a segment on both westbound and eastbound of I-270 freeway between MO-370 and MO-367, and the other was a segment on both northbound and southbound on MO-141 between Clayton Rd. and I-55. I-270 is one of the busiest freeways in St. Louis. The length of I-270 segment of interest was about 12 miles and the free flow travel time along this segment was around 11 minutes. Eight detectors were permanently installed along this stretch of I-270. The segment of MO-141 stretches about 17 miles and is a signalized urban arterial. Free flow travel time for this stretch is about 23 minutes. There are 16 detectors along the segment. **FIGURE 4-8** and **FIGURE 4-9** depict the geometric characteristics and locations of detectors for both roadway segments. Traffic volume data were collected in half minute resolution by detectors 24 hours a day for 15 months from June 2012 to September 2013. A total of 69 and 92 work zones on I-270 and MO-141 were deployed during the traffic volume data collection period. Detailed work zone data such as work zone type, number of lanes closed, speed limit, begin log, duration, and work zone length were recorded by MoDOT. In order to predict work zone traffic demand, traffic volume data collected by the detector located between 1 mile and 3 miles upstream of work zone taper were used as dependent variable for model development.

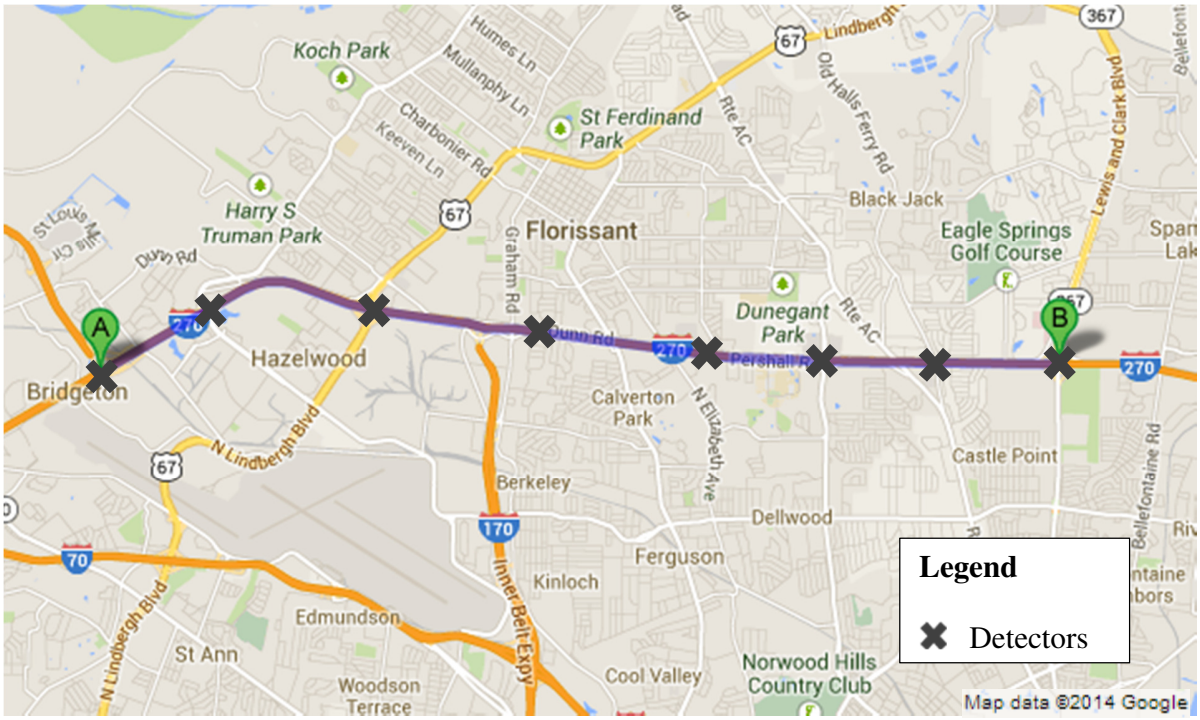


FIGURE 4-8 Study segment of I-270.

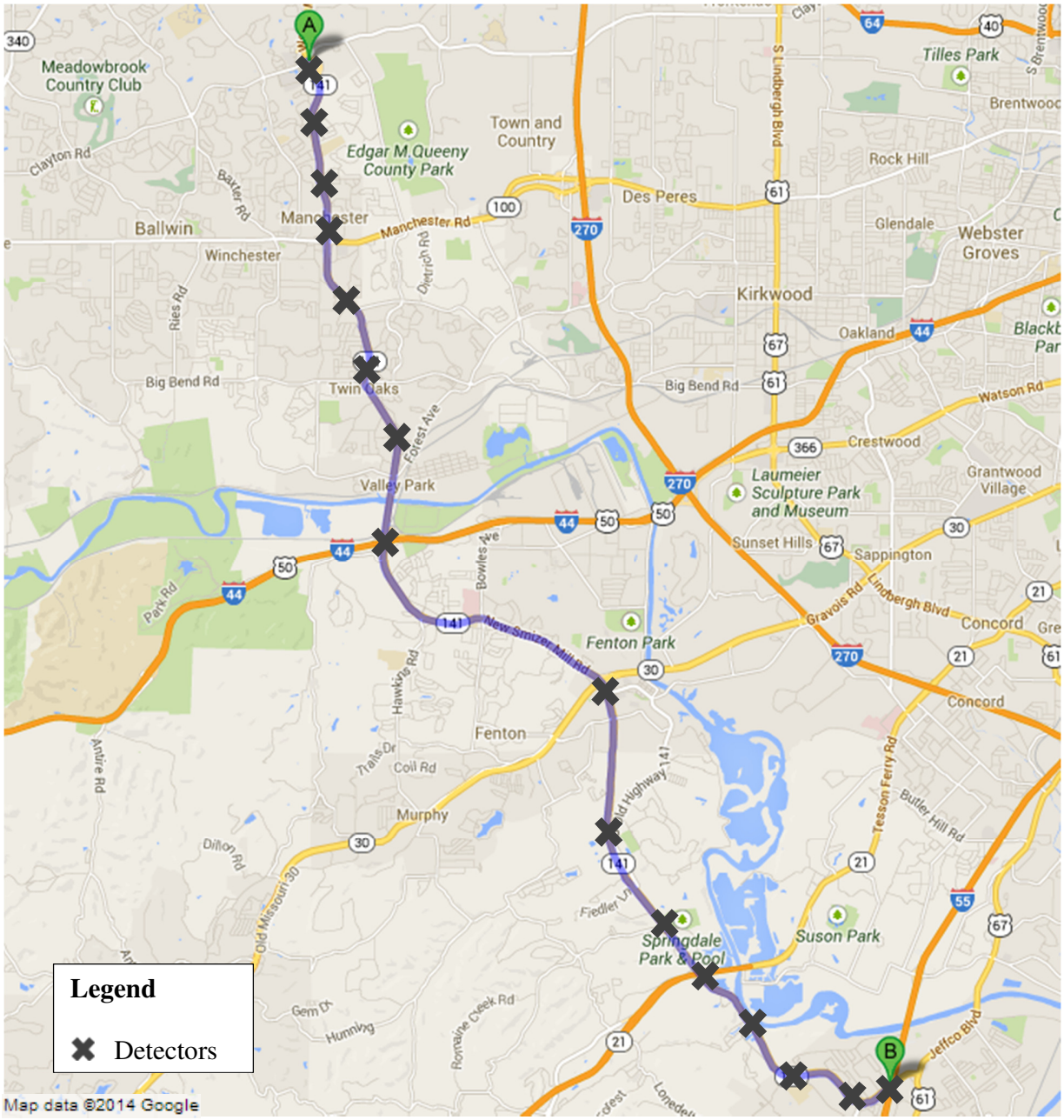


FIGURE 4-9 Study segment of MO-141.

4.2.2. Daily Traffic Demand Prediction

Multilayer feedforward neural network, regression tree and random forest were implemented for daily traffic prediction. The nearest neighbor nonparametric regression was not used for daily prediction since the data consisted of several categorical variables. Daily prediction provides prediction for a time horizon of 24 hours based on historical traffic data. Eleven variables that are relevant to work zone traffic demand were chosen as predictors for the daily traffic demand models. They are hour of the day, day of the week, month of the year, total number of lanes, number of closed lanes, work zone type, speed limit, direction, work zone begin log, work zone duration and work zone length. Some of these variables such as the work zone length, number of closed lanes, are commonly used in work zone capacity estimation models (Kim et al., 2001). Traffic volumes were aggregated into one hour intervals, since hourly volume is the most commonly used traffic flow parameter. For both I-270 and MO-141 datasets, 60% of the data were randomly selected as training data and the rest as testing data. Separate models were built for freeway (I-270) and signalized urban arterial (MO-141).

Performance of machine learning methods is typically evaluated using error measures. The four models were evaluated using three common measures of effectiveness: root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). They are formulated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(v_i - \hat{v}_i)^2}, \qquad (4-7)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|v_i - \hat{v}_i|, \qquad (4-8)$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|v_i - \hat{v}_i|}{v_i}, \qquad (4-9)$$

where v_i is the observed traffic volume, $\hat{v}_i$ is the predicted or forecasted traffic volume, and N is the total number of observations.

Using the experience of other researchers along with trial-and-error processes, the best values for model parameters were determined for all proposed models. For the multilayer feedforward neural network, network structure of different hidden layers were tried. Back-propagation algorithm with Levenberg-Marquardt optimization (Hagan and Menhaj, 1994) was used to train the model. After varying the number of hidden nodes in the hidden layers, network structure of 2 hidden layer with 40 nodes in each produced the best results for the I-270 dataset. Network structure of one hidden layer with 60 nodes produced the best results for the MO-141 dataset. The regression tree yielded the best model performance for both datasets when the minimum terminal node size of 5 was used. In the process of developing random forest, the OOB error estimate stabilized after approximately 100 trees. The minimum terminal node size of 5 and the number of randomly selected input variables of 8 resulted in the best model performance for both datasets. Historical average traffic demand was used for comparison as a baseline. The historical averages were conditioned on hour of the day, day of the week, month of the year, total number of lanes, number of closed lanes, work zone type, and speed limit. There were instances when the work zone characteristics in the testing sample were not observed in historical training data. Thus, baseline predictions for such unobserved work zone characteristics were not made.

TABLE 4-12 summarizes the results from the test data for both I-270 and MO-141. Bold values indicate the smallest values for RMSE, MAE and MAPE. As shown in **TABLE 4-12**,

random forest outperformed regression tree, multilayer feedforward neural network and baseline predictor for all three measures. As expected, the ensemble method of bagging of random forest improved model prediction accuracy over the regression tree.

TABLE 4-12 Summary of Daily Prediction Results For (a) I-270 and (b) MO-141 Dataset

A. I-270 Dataset

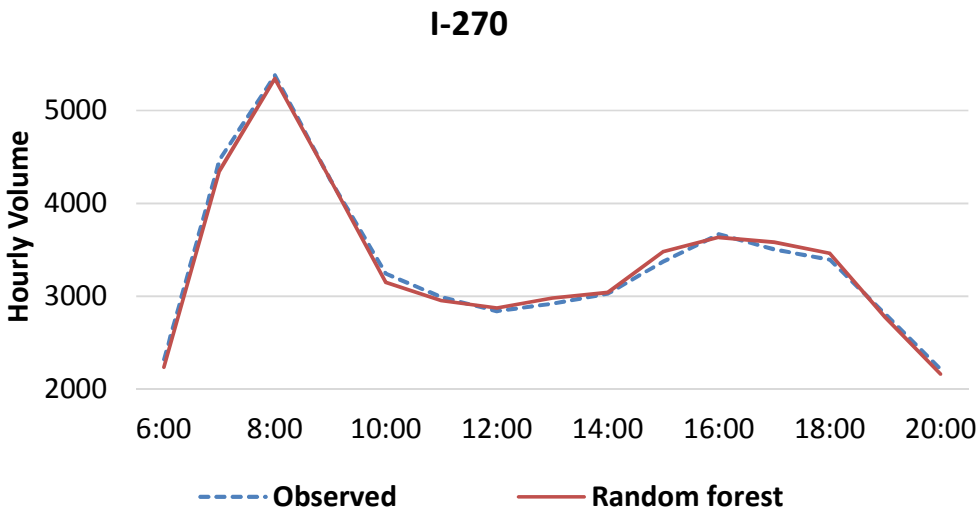
	RMSE	**MAE**	**MAPE**
Baseline Predictor	312.2	149.6	9.37%
Regression Tree	279.7	143.1	8.98%
Random Forest	**231.5**	**119.8**	**7.70%**
Neural Network	235.9	141.0	8.95%

B. MO-141 Dataset

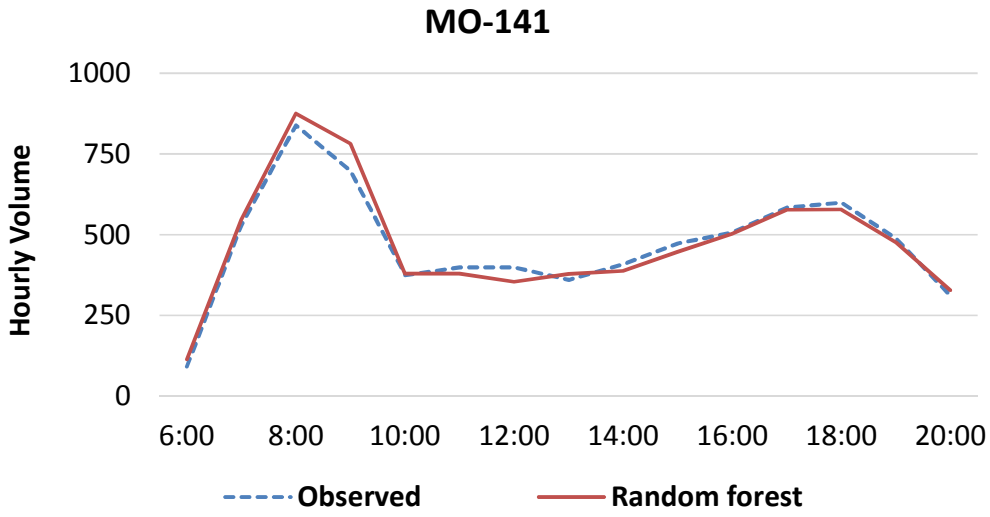
	RMSE	**MAE**	**MAPE**
Baseline Predictor	140.6	73.6	17.95%
Regression Tree	152.1	77.2	17.57%
Random Forest	**102.4**	**57.0**	**14.06%**
Neural Network	111.0	68.6	21.07%

FIGURE 4-10 illustrates an application of the random forest for modeling work zone flows on a randomly selected weekday. None of the observations on the randomly selected

weekday was used for model training. The temporal variation of predicted and observed traffic demand from 6:00 am to 8:00 pm on a weekday is plotted in **FIGURE 4-10**. **FIGURE 4-10** shows the predicted values of random forest fit the observed values very closely. The MAPE values for the randomly selected day were 1.81% for I-270 and 5.95% for MO-141.



(a) I-270

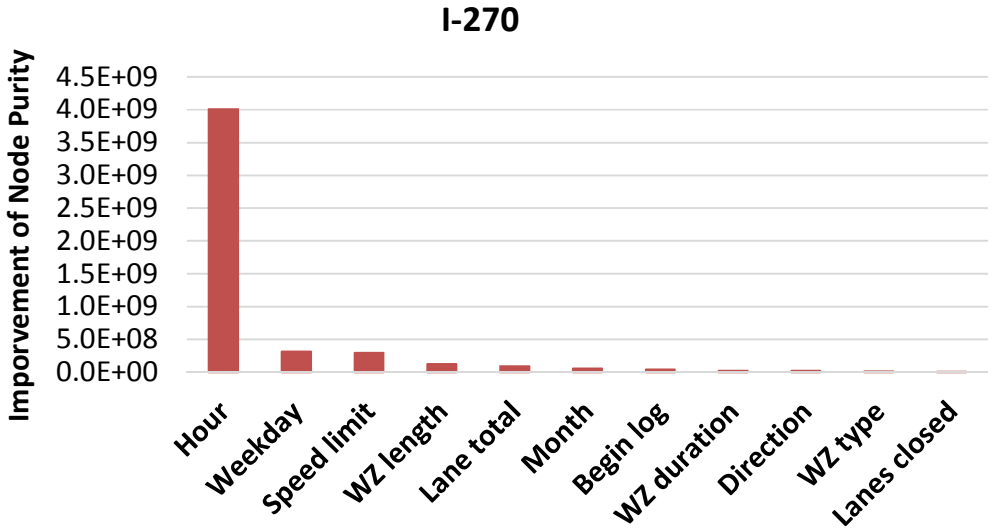


(b) MO-141

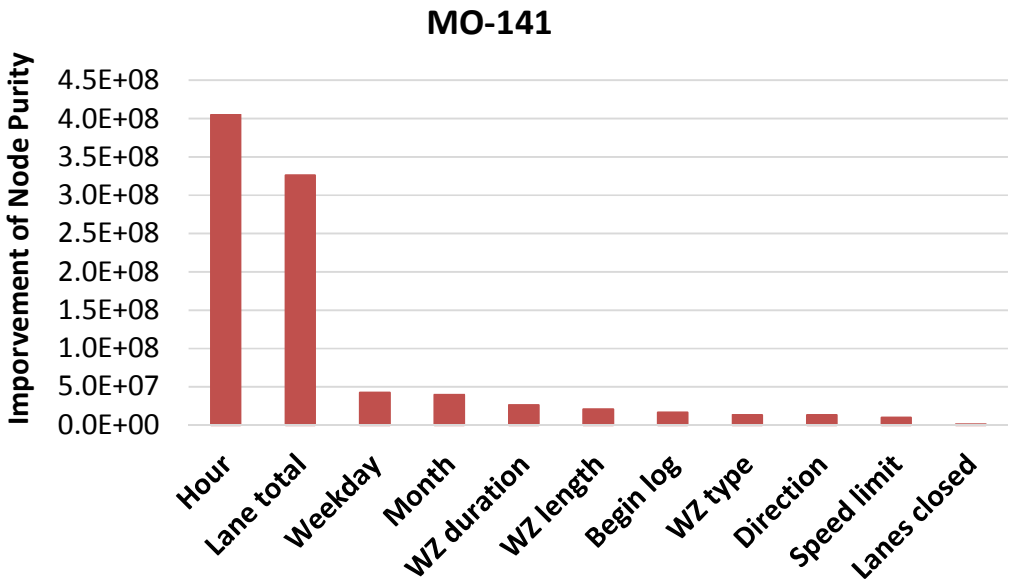
FIGURE 4-10 Temporal variation of predicted and observed traffic demand for daily traffic demand prediction.

The importance of random forest variables is presented in **FIGURE 4-11**. The Y axis in the figure is the accumulated improvement of node purity at each split in each tree of a variable. A high value of improvement indicates high variable importance. **FIGURE 4-11** reveals that hour of day dominates traffic prediction for freeway work zone. This is presumably due to the significant variation of traffic volume during the day as can be seen in **FIGURE 4-10**. Among the work zone characteristics, work zone speed limit and length of work zone exhibited the highest influence.

The time of day variable was once again the most influential for the MO-141 arterial corridor. Total number of lanes also had a very high influence on the daily forecasts. Work zone duration and length were the most influential among all work zone characteristics.



(a) I-270



(b) MO-141

FIGURE 4-11 Variable importance in daily traffic demand prediction.

4.2.3. Short-Term Traffic Demand Prediction

Multilayer feedforward neural network, nearest neighbor nonparametric regression, regression tree and random forest models were developed for short-term traffic prediction of work zones on the two routes. Let $V(t+i)$ denote the volume of the $i+1$th time interval in the future and $V(t)$ the volume of the nearest time interval in the future. Let $V(t-j)$ denote traffic volume of the jth time interval in the past and $V(t-1)$ the volume of the most recent time interval in the past. Let b denote the extent of the prediction interval and d the extent of the "look-back" interval. The traffic volumes at a location on a roadway in the near future ($V(t)$, $V(t+1)$, …, $V(t+b-1)$) are likely correlated to the traffic volumes in the recent past at the

location ($V(t-1), V(t-2), \ldots, V(t-d)$), adjacent upstream location ($V_{us}(t-1), V_{us}(t-2), \ldots, V_{us}(t-d)$), and adjacent downstream location ($V_{ds}(t-1), V_{ds}(t-2), \ldots, V_{ds}(t-d)$). Therefore, the short-term traffic demand prediction problem is essentially a problem that predicts the values of $V(t), V(t+1), \ldots, V(t+b-1)$, using the time series volumes from the detector of interest, and adjacent upstream and downstream detectors. In addition to temporal and spatial traffic volume information, other relevant factors such as hour of day, day of week, number of lanes, speed limit, direction, and work zone begin log were also considered in the models. Because traffic volumes at more distant locations have less influence, only observations taken from within 2 miles of the detector of interest were used in the models. Traffic volumes were aggregated in 15-min resolution, since 15-min interval is a reasonable time interval for TMCs to implement the operational strategy to the field in real-time traffic control and for drivers to react to the strategy change. The extent of prediction interval was set to 4 (b = 4). In other words, the proposed models provide prediction of traffic demand over 15-min intervals for up to 1 hour in advance. Prediction on multiple time periods into the future enables a wider range of applications. In addition, since forecasting1-minute traffic flow is a much more critical and complex problem, traffic flow forecast for 1-minute resolution for MO-141 is further analyzed. The extent of "look-back" was varied from 1 to 6 intervals; adding more "look-back" intervals after 3 did not improve model performance. Thus, the extent of "look-back" interval was set to 3. For both I-270 and MO-141 datasets, again, 60% of the data were randomly selected as training data and the rest were used for testing. Models were built separately for the I-270 freeway and the MO-141 signalized urban arterial.

Multilayer feedforward neural network of different hidden layers were tried. Back-propagation algorithm with Levenberg-Marquardt optimization (Hagan and Menhaj, 1994) was

used to train the model. The number of hidden nodes was varied, and network of two hidden layers with 40 nodes in the first hidden layer and 20 nodes in the second produced the best results for I-270 dataset. Network structure of two hidden layers with 40 nodes in the first hidden layer and 10 nodes in the second produced the best results for MO-141 dataset. The nonparametric model yielded the best prediction results for the I-270 and MO-141 datasets when the number of nearest neighbors was selected to be 9 and 7, respectively. A minimum terminal node size of 5 yielded the best regression tree results. For random forest, after 100 trees were built, further increases in the number of trees did not improve model performance. The minimum terminal node size of 5 and the number of randomly selected input variables being 6 resulted in the best model performance for both datasets. Instantaneous traffic demand was used as a baseline predictor for comparison. The instantaneous traffic demand for short-term prediction is to use the last measured traffic demand as a proxy for the traffic demand of future intervals. Instantaneous traffic demand provides accurate predictions in cases where traffic conditions change slowly over long time periods.

TABLE 4-13 presents the prediction results from the test data for both I-270 and MO-141. In the table, each "t", "$t + 1$", "$t + 2$", and "$t + 3$", is a 15-min interval into the future i.e., 15, 30, 45, and 60-minute prediction intervals. The performance of all five predictors worsens when predicting further into the future. Bolded values indicate the smallest values for RMSE, MAE and MAPE. **TABLE 4-13** shows that all the error measures for random forest are smaller than other models for all four 15-min prediction intervals. The performance of the baseline predictor worsens by a factor of greater than two between t and $t + 3$, while the random forest predictor MAPE increased slightly over 1% (I-270) and 4% (MO-141) between t and $t + 3$. All five predictors performed worse on MO-141. The comparison between random forest and

regression tree error measures shows that the ensemble method of bagging can increase prediction accuracy for the weak model. **TABLE 4-14** presents the 1-minute traffic flow prediction results for MO-141 dataset. **TABLE 4-14** shows that all the error measures for random forest are again smaller than other models for all four 1-min prediction intervals.

TABLE 4-13 Summary of 15-Minute Traffic Flow Prediction Results for (a) I-270 and (b) MO-141 Dataset

A. I-270 Dataset

		t	$t+1$	$t+2$	$t+3$
Baseline Predictor	RMSE	54.7	79.9	102.9	126.0
	MAE	38.2	53.5	69.5	85.9
	MAPE	10.62%	14.69%	19.26%	24.07%
Regression Tree	RMSE	50.2	57.4	59.8	63.2
	MAE	34.1	36.3	37.0	38.9
	MAPE	9.62%	10.42%	10.88%	11.25%
Random Forest	RMSE	**35.5**	**39.8**	**40.1**	**43.5**
	MAE	**23.9**	**25**	**26.1**	**27.3**
	MAPE	**6.85%**	**7.25%**	**7.71%**	**7.96%**
Neural Network	RMSE	39.7	44.4	50.0	53.2
	MAE	26.8	29.2	31.4	32.8
	MAPE	7.79%	8.68%	9.35%	9.68%
Nearest Neighbor	RMSE	39.8	49.8	57.3	64.4
	MAE	25.7	30.8	35.3	40
	MAPE	7.47%	9.13%	10.89%	12.74%

B. MO-141 Dataset

		t	$t+1$	$t+2$	$t+3$
Baseline Predictor	RMSE	26.9	40.9	54.3	67.5
	MAE	14.5	23.1	31.0	39.1
	MAPE	9.17%	16.24%	22.24%	29.21%
Regression Tree	RMSE	27.5	32.4	37.5	38.8
	MAE	14.2	17.9	19.7	20.1
	MAPE	8.33%	11.98%	12.92%	13.48%
Random Forest	RMSE	**20.5**	**23.1**	**25.5**	**28.3**
	MAE	**10.6**	**13**	**14.1**	**15**
	MAPE	**6.64%**	**9.36%**	**10.18%**	**10.88%**
Neural Network	RMSE	21.0	26.2	29.2	34.5
	MAE	11.8	15.3	17.2	19.8
	MAPE	8.32%	11.72%	13.15%	15.11%
Nearest Neighbor	RMSE	23.5	30.5	38.5	45.8
	MAE	12.7	16.8	20.9	24.9
	MAPE	8.24%	11.96%	14.87%	17.70%

TABLE 4-14 Summary of 1-Minute Traffic Flow Prediction Results for (a) I-270 and (b) MO-141 Dataset

		t	$t+1$	$t+2$	$t+3$
Baseline Predictor	RMSE	8.9	9.6	14.8	15.5
	MAE	4.2	4.5	6.6	7.5
	MAPE	13.47%	15.20%	18.70%	22.22%
Regression Tree	RMSE	8.6	9.1	9.6	10.1
	MAE	3.9	4.2	4.4	4.7
	MAPE	12.49%	13.62%	14.36%	15.47%
Random Forest	RMSE	**6.4**	**6.7**	**6.9**	**7.4**
	MAE	**3.0**	**3.1**	**3.3**	**3.5**
	MAPE	**9.79%**	**10.51%**	**11.28%**	**11.94%**
Neural Network	RMSE	6.5	6.8	7.1	7.6
	MAE	3.2	3.4	3.6	3.9
	MAPE	12.28	12.87	14.90	15.51
Nearest Neighbor	RMSE	6.8	7.2	7.7	7.9
	MAE	3.4	3.5	3.7	3.9
	MAPE	13.05%	13.25%	14.11%	14.58%

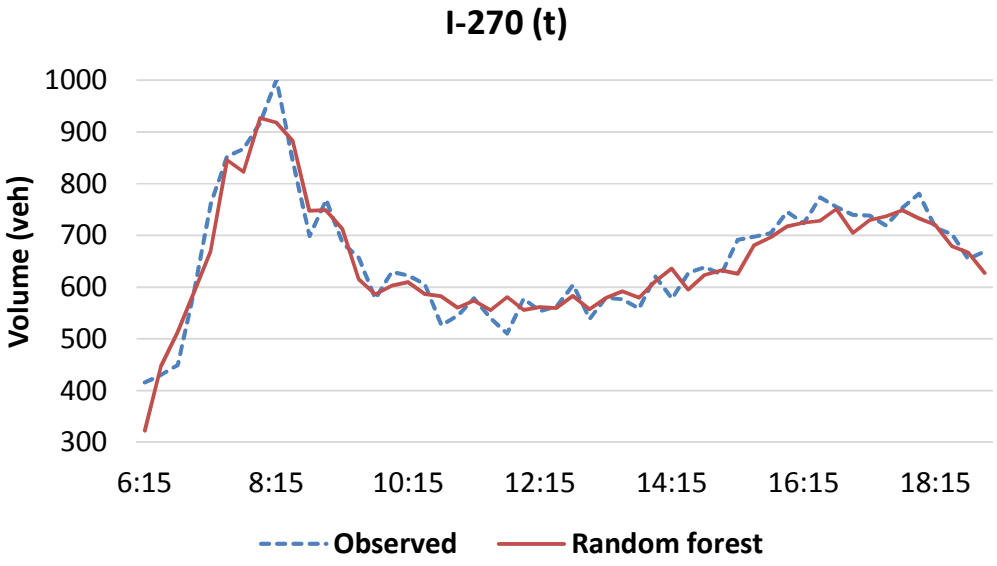
Because the main application of the short-term traffic flow prediction is for real-time traffic operation and control, the average computation time for model construction and prediction for test data are critical for real-time applications. They computation time for these models were recorded and presented in **TABLE 4-15**. These models were constructed and ran on 4GB RAM machine with Intel core i3 CPU. As shown in **TABLE 4-15**, all other models provide predictions within one second. For 1-min prediction interval, they are all suitable for real-time traffic predictions except nearest neighbor nonparametric regression. The total computation time for

random forest is faster than multilayer feedforward neural network and nearest neighbor nonparametric regression, but much slower than regression tree.

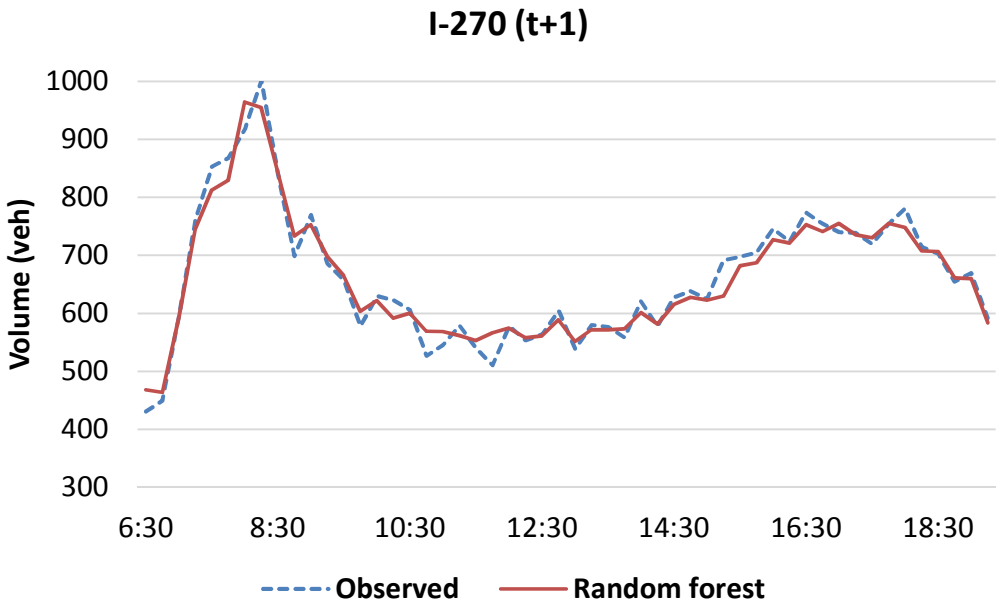
TABLE 4-15 Computation Times

	Model Construction	Prediction for Test Data	Total
Regression Tree	0.62s	0.03s	0.65s
Random Forest	41.48s	0.29s	42.17s
Neural Network	111.96s	0.70s	112.66s
Nearest Neighbor	N/A	158.08s	158.08s

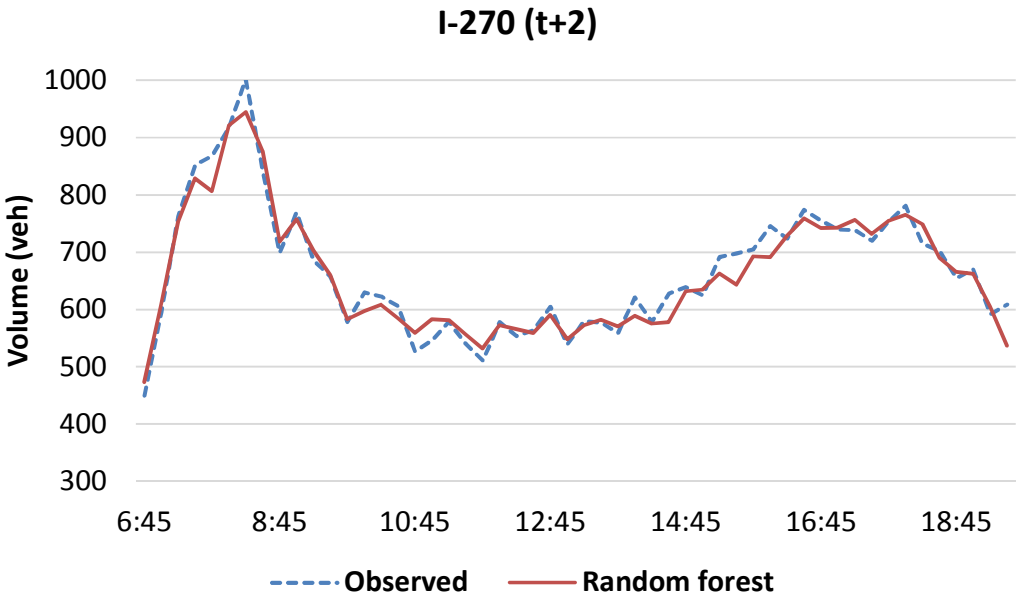
The work zone traffic demand of a randomly selected weekday on each study segment was forecasted by the best performed model. None of the observations on the randomly selected weekday was used in the model training process. **FIGURE 4-12** and **FIGURE 4-13** present the temporal variation of predicted and observed traffic demand from 6:00 am to 8:00 pm on a weekday on both I-270 and MO-141. Both figures show that the predicted values of random forest and the observed values are in close agreement for all four prediction time intervals. The random forest's MAPE values for this day for "t", "$t+1$", "$t+2$", and "$t+3$" prediction intervals were 4.34%, 2.63%, 2.92%, and 2.70% for I-270, and 4.20%, 5.39%, 4.89%, and 5.40% for MO-141.


I-270 (t)
Volume (veh)
1000
900
800
700
600
500
400
300
6:15
8:15
10:15
12:15
14:15
16:15
18:15
Observed
Random forest

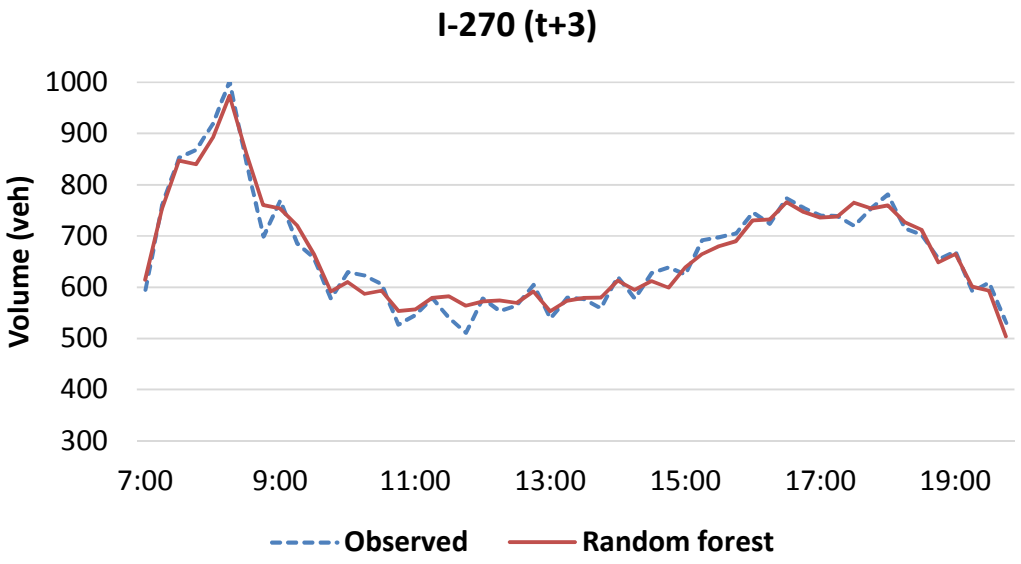
(a)


I-270 (t+1)
Volume (veh)
1000
900
800
700
600
500
400
300
6:30
8:30
10:30
12:30
14:30
16:30
18:30
Observed
Random forest

(b)

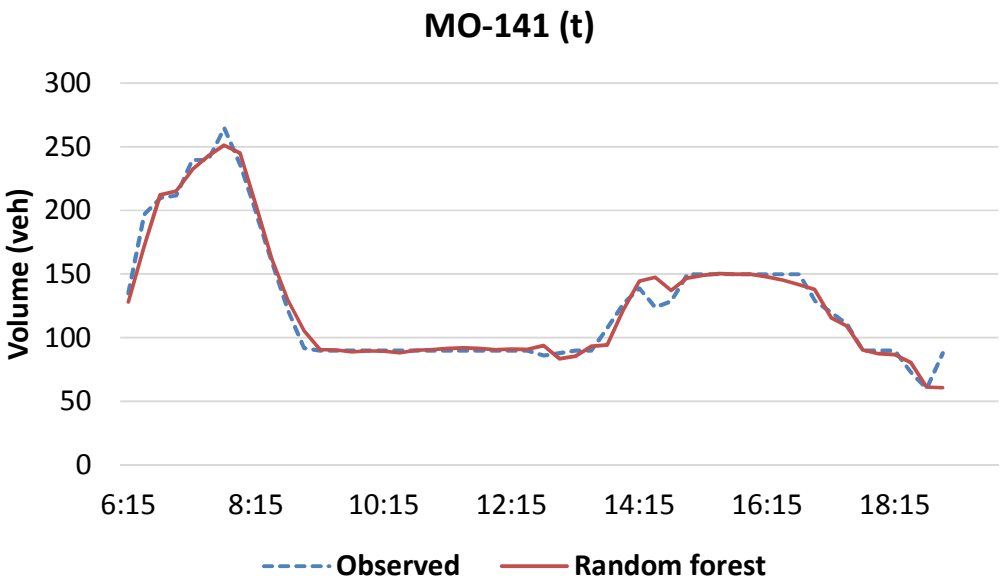


(c)

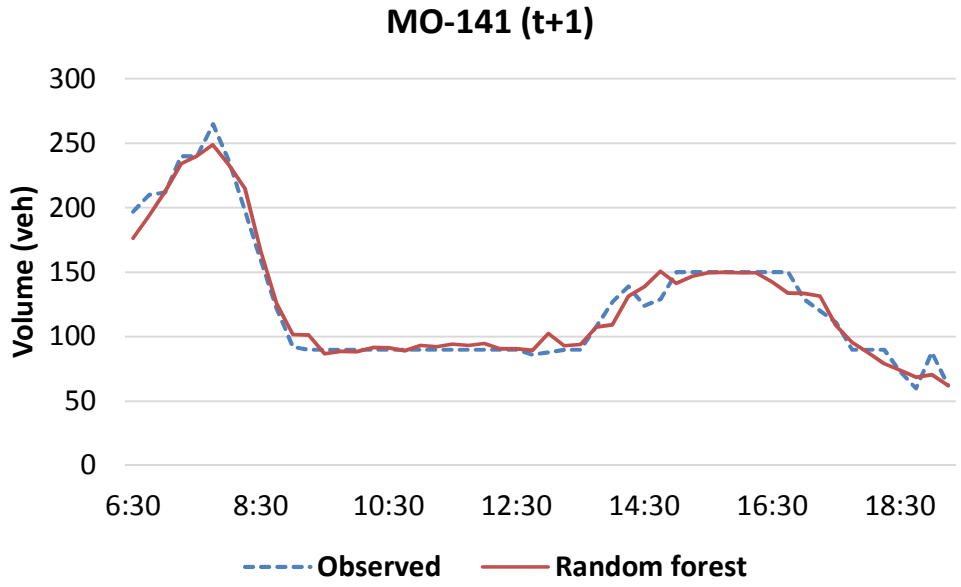


(d)

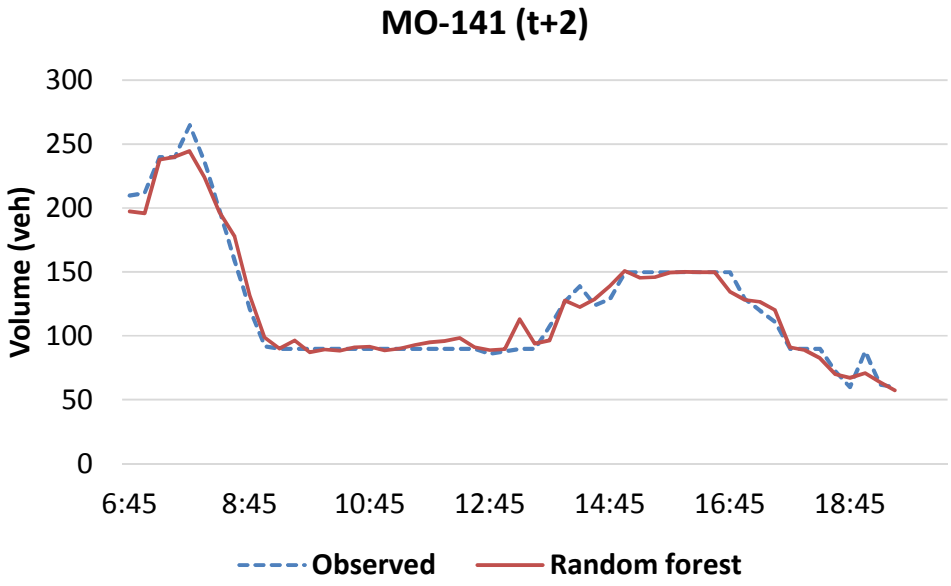
FIGURE 4-12 Temporal variation of predicted and observed traffic demand on I-270 for short-term prediction.


MO-141 (t)
300
250
200
150
100
50
0
Volume (veh)
6:15
8:15
10:15
12:15
14:15
16:15
18:15
Observed
Random forest

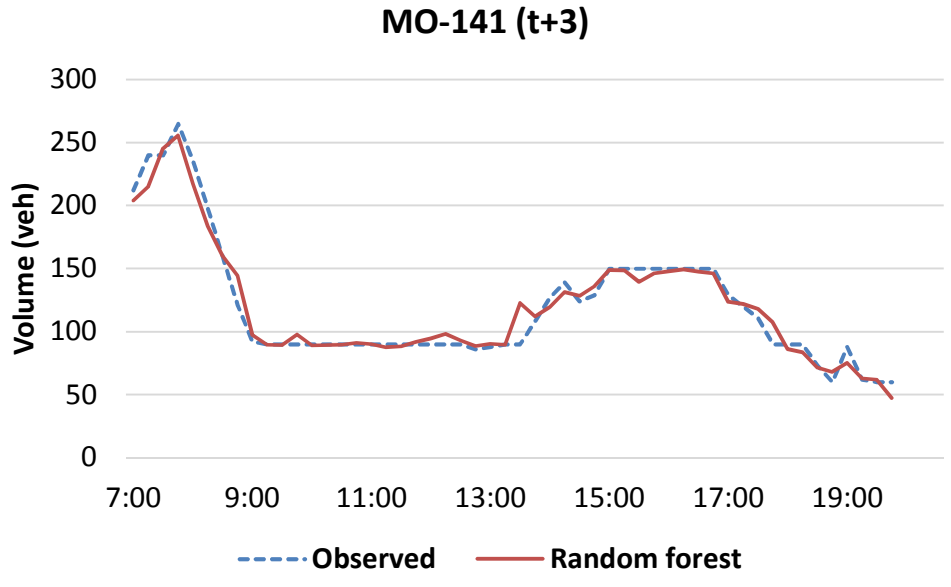
(a)


MO-141 (t+1)
300
250
200
150
100
50
0
Volume (veh)
6:30
8:30
10:30
12:30
14:30
16:30
18:30
Observed
Random forest

(b)



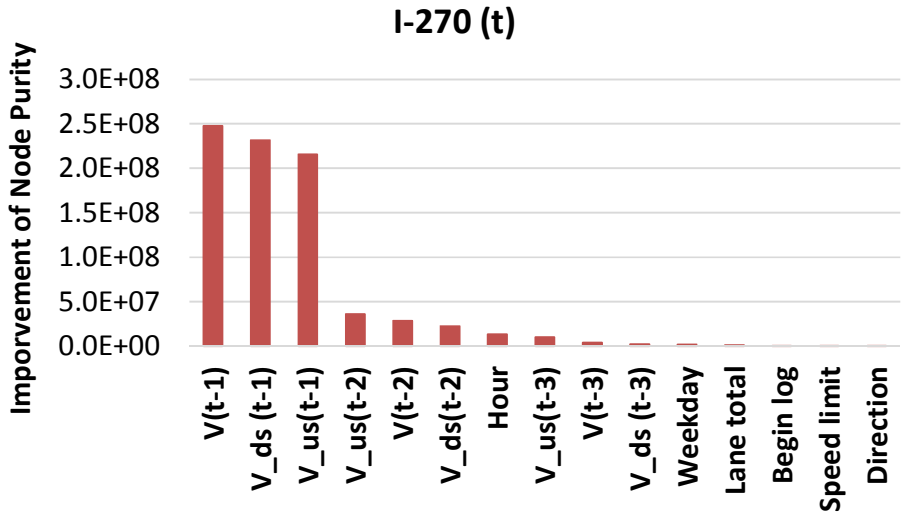
(c)



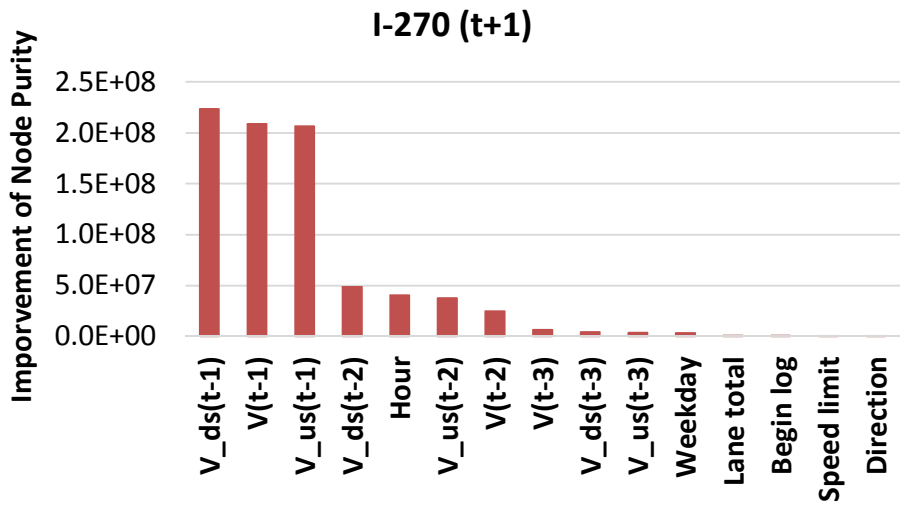
(d)

FIGURE 4-13 Temporal variation of predicted and observed traffic demand on MO-141 for short-term prediction.

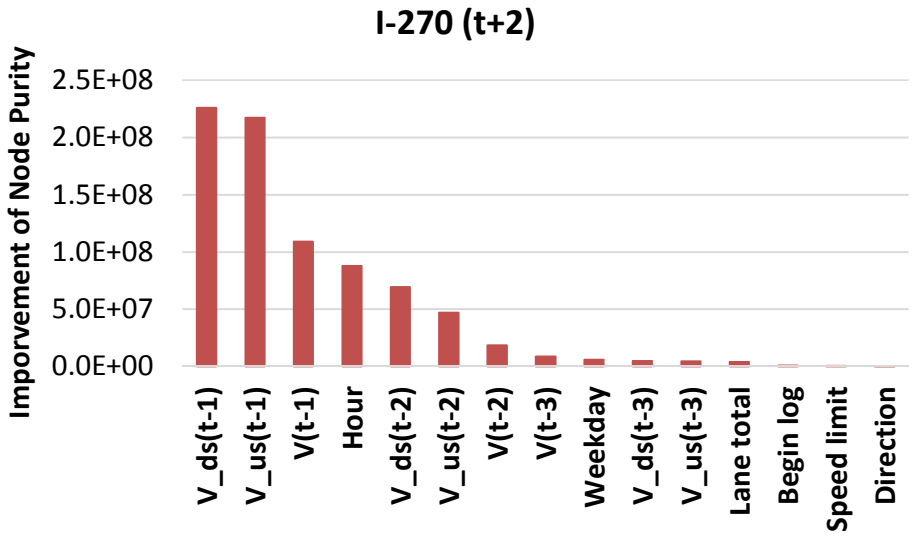
The importance of variables was explored for short-term prediction. **FIGURE 4-14** and **FIGURE 4-15** display the importance of variables in predictions of all four time intervals for I-270 and MO-141 datasets. **FIGURE 4-14** reveals that the most relevant variables in short-term traffic prediction for the I-270 dataset are the volumes at one "look-back" interval at the location of interest and the adjacent upstream and downstream locations, i.e. $V(t-1)$, $V_{us}(t-1)$ and $V_{ds}(t-1)$. As the forecast interval size increased from t to $t+3$, the importance of $V(t-1)$ decreased while the importance of hour of day variable (*Hour*) increased. When forecasting volume for immediately next time interval (i.e., t), the previous interval's $(t-1)$ volume served as a good proxy for the hour of day variation in flow. As the forecast interval size increased (to $(t+1$ and higher) this was not the case, since the previous interval became more removed from the forecast interval and thus the hour of day variable started to exhibit a higher influence as the forecast interval size grew. The downstream volume, $V_{ds}(t-1)$, became the primary variable from t onwards, which might be indicative of congestion spillback occurring due to downstream bottlenecks that were felt at the current location in the future time intervals.



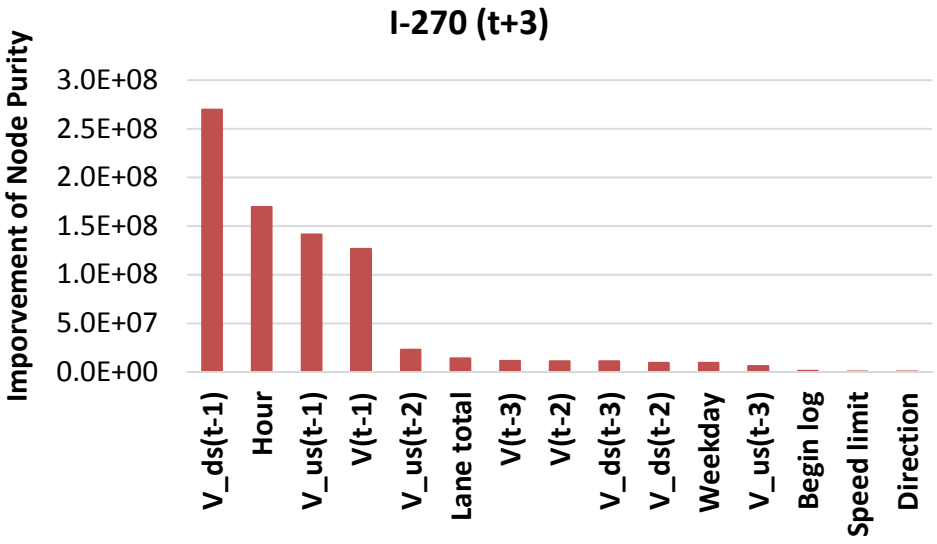
(a)


I-270 (t+1)
Imporvement of Node Purity
2.5E+08
2.0E+08
1.5E+08
1.0E+08
5.0E+07
0.0E+00
V_ds(t-1)
V(t-1)
V_us(t-1)
V_ds(t-2)
Hour
V_us(t-2)
V(t-2)
V(t-3)
V_ds(t-3)
V_us(t-3)
Weekday
Lane total
Begin log
Speed limit
Direction

(b)


I-270 (t+2)
Imporvement of Node Purity
2.5E+08
2.0E+08
1.5E+08
1.0E+08
5.0E+07
0.0E+00
V_ds(t-1)
V_us(t-1)
V(t-1)
Hour
V_ds(t-2)
V_us(t-2)
V(t-2)
V(t-3)
Weekday
V_ds(t-3)
V_us(t-3)
Lane total
Begin log
Speed limit
Direction

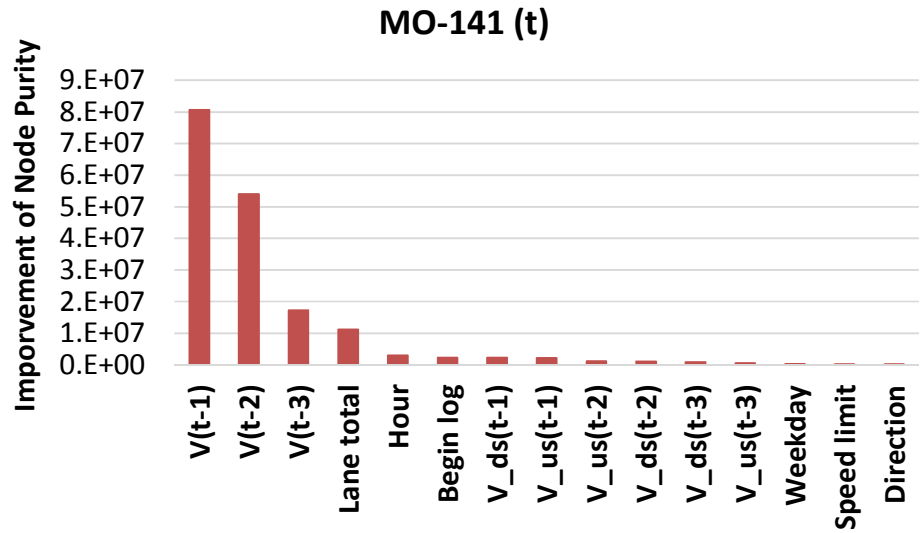
(c)



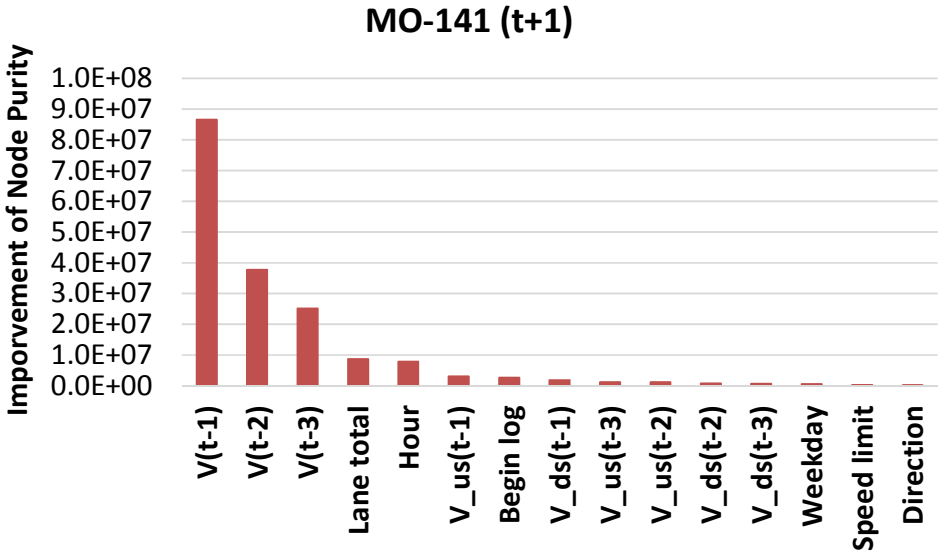
(d)

FIGURE 4-14 Variable importance in short-term prediction for I-270 dataset.

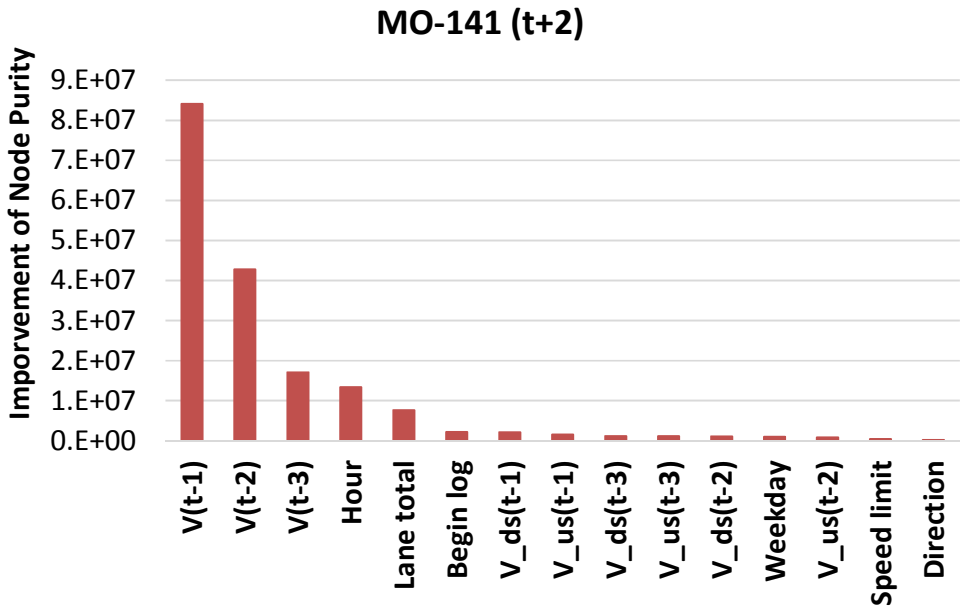
FIGURE 4-15 shows the most relevant variables in short-term traffic prediction for MO-141 dataset are the volumes of all three "look-back" intervals at the location of interest, i.e. $V(t-1)$, $V(t-2)$ and $V(t-3)$. Unlike the I-270 short-term prediction, the variables exhibiting the highest influence did not vary with the size of prediction interval. It appears that signalization reduced the influence of upstream and downstream traffic flow values.


MO-141 (t)
Improvement of Node Purity
9.E+07
8.E+07
7.E+07
6.E+07
5.E+07
4.E+07
3.E+07
2.E+07
1.E+07
0.E+00
V(t-1)
V(t-2)
V(t-3)
Lane total
Hour
Begin log
V_ds(t-1)
V_us(t-1)
V_us(t-2)
V_ds(t-2)
V_ds(t-3)
V_us(t-3)
Weekday
Speed limit
Direction

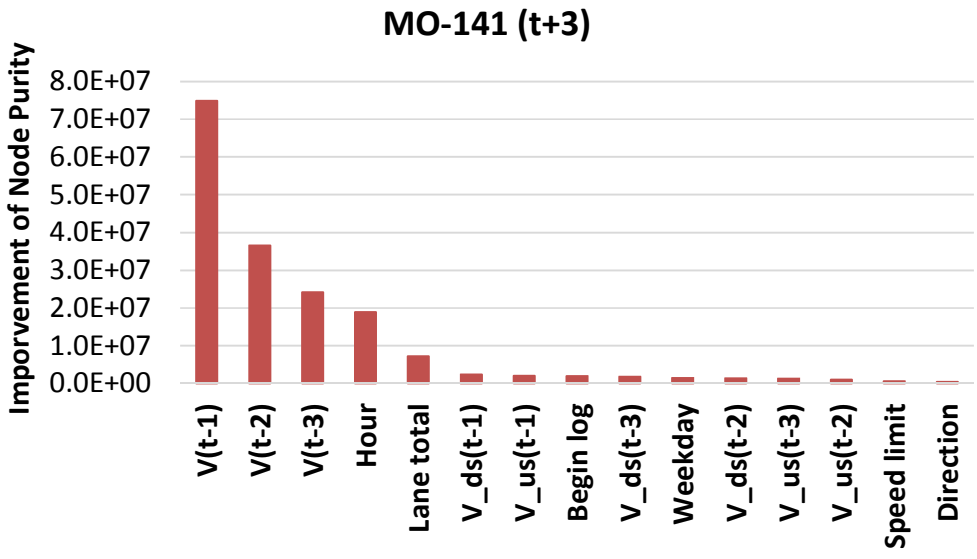
(a)


MO-141 (t+1)
Improvement of Node Purity
1.0E+08
9.0E+07
8.0E+07
7.0E+07
6.0E+07
5.0E+07
4.0E+07
3.0E+07
2.0E+07
1.0E+07
0.0E+00
V(t-1)
V(t-2)
V(t-3)
Lane total
Hour
V_us(t-1)
Begin log
V_ds(t-1)
V_us(t-3)
V_us(t-2)
V_ds(t-2)
V_ds(t-3)
Weekday
Speed limit
Direction

(b)



(c)



(d)

FIGURE 4-15 Variable importance in short-term prediction for MO-141 dataset.

4.2.3. Application to Special Event Traffic Demand Prediction

Special events are planned events that cause non-recurring delays just like work zones, thus they could be modeled in a way similar to work zones. Short-term traffic demand forecasting models were tested during special events. Special event traffic volumes were collected on westbound I-70 from downtown to I-170 in St. Louis, Missouri. I-70 is one of the busiest freeways in the metropolitan St. Louis area. The length of the study segment is approximately 11 miles and the free flow travel time along this segment is approximately 12 minutes. Fourteen detectors are located along this stretch of I-70. **FIGURE 4-16** shows a map of the locations of detectors for both roadway segments. A total of 129 St. Louis Cardinals baseball games were played from June 2012 to September 2013. Traffic volumes were gathered for half hour before the end of each game to one hour after the end of each game. Traffic volumes were aggregated in 5-min intervals.

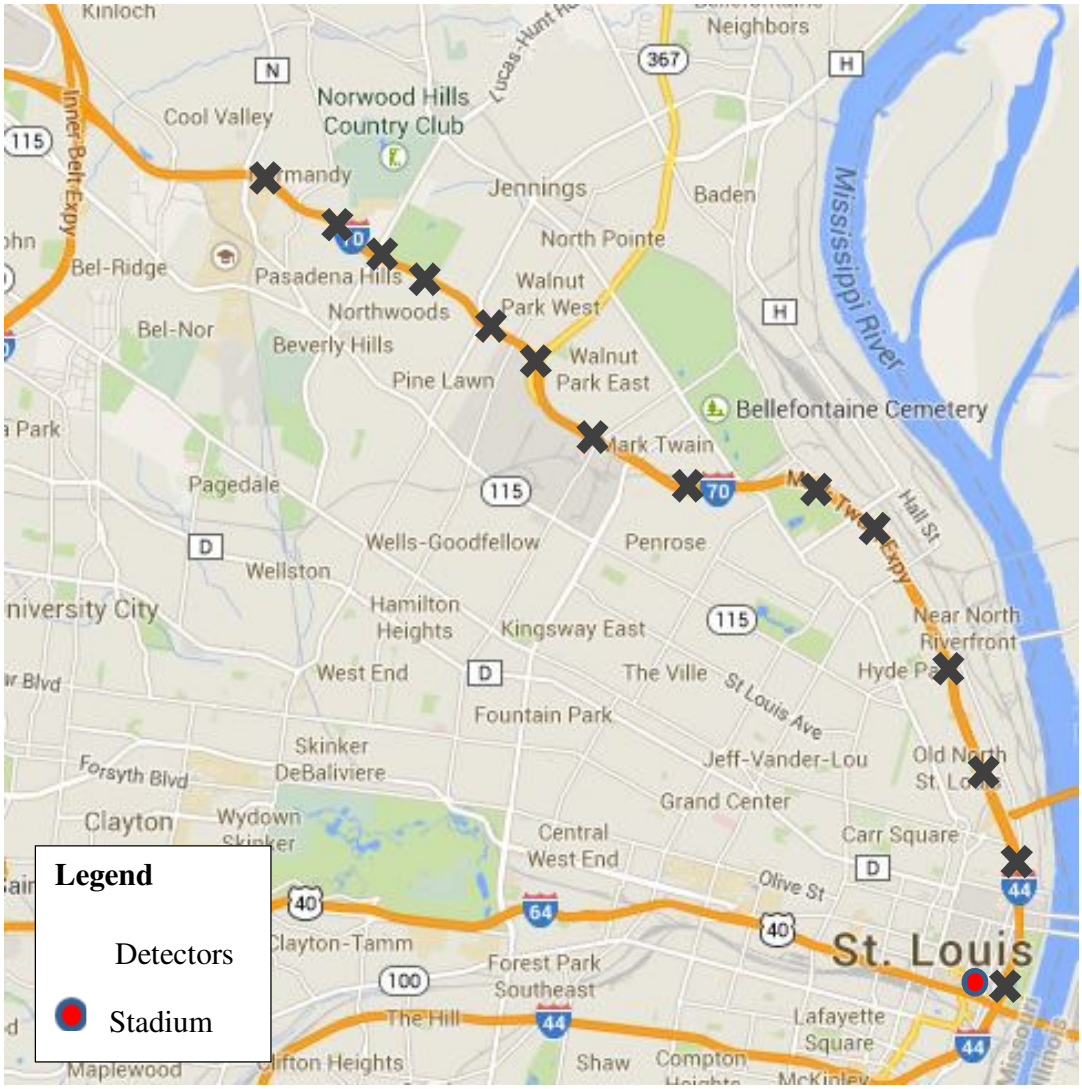


FIGURE 4-16 Study segment of I-70.

The same modeling techniques as applied to work zones were employed. Multilayer feedforward neural network, nearest neighbor nonparametric regression, regression tree, and random forest models were implemented for traffic prediction of special events. The models predict the traffic volumes at a location on a roadway in the near future using the time series volumes from the detector of interest, and adjacent upstream and downstream detectors. Baseball games occur at different times, such as daytime and nighttime, and weekdays and weekend, have

different levels of impact on traffic, thus hour of day and day of week were considered in the models. In addition, other relevant factors such as detector location and number of lanes were also considered as input variables. Traffic volumes were aggregated in 5-min resolution. The extent of prediction interval was set to 4. In other words, the proposed models provided predictions of traffic demand over 5-min intervals for up to 20 minutes in advance. Prediction on multiple time periods into the future enables a wider range of applications. The extent of "look-back" interval was set to 3. 60% of the data were randomly selected as training data and the rest were used for testing.

Multilayer feedforward neural network of different hidden layers were tried. Back-propagation algorithm with Levenberg-Marquardt optimization (Hagan and Menhaj, 1994) was used to train the models. The number of hidden nodes was varied, and a network of two hidden layers with 20 nodes in the first hidden layer and 30 nodes in the second produced the best results. The nonparametric nearest neighbor model yielded the best prediction results when the number of nearest neighbors was selected to be 10. A minimum terminal node size of 100 yielded the best regression tree results. For random forest, after 150 trees were built, further increases in the number of trees did not improve model performance. For both datasets, the best performing model used a minimum terminal node size of 5 and three randomly selected input variables. Instantaneous traffic volume was used as a baseline predictor for comparison. The so-called instantaneous traffic volume for prediction is to use the last measured traffic volume as a proxy for the traffic demand of future intervals. Instantaneous traffic demand provides accurate predictions in cases where traffic conditions change slowly over long time periods.

TABLE 4-16 presents the prediction results from the test data. In the table, each "t", "$t+1$", "$t+2$", and "$t+3$" is a 5-min interval into the future. As expected, the performance of

all five predictors worsens when predicting further into the future. Bolded values indicate the smallest values for RMSE, MAE and MAPE. **TABLE 4-16** shows that all the error measures for random forest are smaller than other models for all four 5-min prediction intervals, which means random forest outperformed regression tree, multilayer feedforward neural network, nearest neighbor nonparametric regression, and baseline predictor in terms of accuracy.

TABLE 4-16 Summary of Special Event Traffic Prediction Results

		t	$t+1$	$t+2$	$t+3$
Neural Network	RMSE	23.4	28.4	30.9	32.0
	MAE	18.5	22.1	24.3	25.2
	MAPE	8.35%	9.74%	10.89%	11.54%
Nearest Neighbor	RMSE	24.4	28.7	32.3	35.2
	MAE	19.2	22.6	25.4	27.4
	MAPE	8.66%	10.14%	11.44%	12.63%
Regression Tree	RMSE	25.2	29.3	30.8	33.4
	MAE	19.6	22.9	24.3	26.3
	MAPE	8.81%	10.19%	10.96%	12.11%
Random Forest	RMSE	**22.9**	**27.0**	**29.2**	**30.4**
	MAE	**18.0**	**21.1**	**23.0**	**24.0**
	MAPE	**8.12%**	**9.46%**	**10.30%**	**10.98%**
Baseline Predictor	RMSE	41.6	45.5	49.1	51.7
	MAE	28.8	32.3	35.5	38.3
	MAPE	12.52%	13.80%	15.00%	16.51%

CHAPTER 5: CONCLUSIONS

In this dissertation, two critical bottleneck related issues were investigated using AI techniques. First, more accurate lane-changing models were developed to enable more realistic traffic simulation of congested bottlenecks. Second, models for predicting demand at work zone bottlenecks were developed to improve work zone traffic management and operations. These models enable TMCs to proactively manage traffic operations at both natural and work zone bottlenecks.

The publicly available NGSIM vehicle trajectory dataset that consists of traffic conditions approaching congestion and congested conditions was used for lane-changing model development and testing. The model employed factors such as vehicle speeds relative to lead and lag vehicles in the target lane, lead and lag gap distances, and the distance from the end of merge lane. Bayes classifier, classification tree, genetic fuzzy system, random forest, and Adaboost were used to model driver lane-changing behavior at bottlenecks. They yielded prediction accuracies of 83.5%, 83.2%, 73.0%, 88.3% and 88.9%. They outperformed the conventional method of binary logit whose prediction accuracy was 44.2%. The results also demonstrate that ensemble learning methods, such as random forest and Adaboost, produced even higher prediction accuracy than single classifiers. In addition, ensemble learning methods are immune to overfitting, and the prediction results have less variance compared with single classifiers. For single classifiers, different training samples drawn from dataset may result in different model structures and results. Ensemble learning methods can solve the problem by building a large group of single classifiers with different training samples drawn from dataset.

The aforementioned models reflect driver merging behavior at bottlenecks more realistically, thus they can predict overall traffic mobility more accurately in microscopic traffic simulation tools. With more accurate microscopic traffic simulation of bottlenecks, transportation professionals are able to evaluate traffic control strategies with more accuracy. The methodology used for driver lane-changing behavior can be extended to the research of predicting driver lane-changing location at work zone bottlenecks by replacing the dependent variable, decision to merge or not, with the variable, location where drivers merge. Accurate prediction of driver's lane-changing location at work zone bottlenecks could be applied to determine the optimum merge sign location for work zone bottlenecks.

The lack of traffic flow forecast models for work zone bottlenecks motivated the second focus of this dissertation. To that end, four models were developed for short-term and daily traffic flow forecasting for work zone bottlenecks. Two of the models, regression tree and random forest, have not been investigated in previous traffic flow forecasting research. All three measures of effectiveness showed that random forest outperformed all other models for both short-term and daily forecasts. Random forest demonstrated higher prediction accurary than other three AI models. Also, random forest is not as computational complex as multilayer feedforward neural network and nearest neighbor nonparametric regression. Random Forest takes less compuational time for model building and prediction than multilayer feedforward neural network and nearest neighbor nonparametric regression, and it is suitable for real time application. Random forest is robust to outliers and noise, and cannot overfit training data. In addition, random forest yielded variable importance rankings to help the modeler to better understand work zone traffic flow behavior.

Of all variables, the 'hour of day' of the work zone was found to have the greatest impact in daily flow prediction for both freeway and signalized urban arterial. Short-term flow forecasts for freeway work zones depended the most on the flow values at the location of interest and at upstream and downstream locations during the latest 'look-back' interval, i.e., $V(t-1)$, $V_{us}(t-1)$ and $V_{ds}(t-1)$. In contrast, the short-term flow forecast for the arterial work zones relied the most on the flow in the previous three 'look-back' intervals only at the location of interest, $V(t-1)$, $V(t-2)$ and $V(t-3)$. The results showed that the most relevant predictors of short-term flow are traffic flow variables and not work zone characteristics. The results however do not mean that separate models are not needed for work zones. The relative importance of the traffic flow variables, time of day, day of the week, and other seasonality variables could be very different between normal operations and work zones. The contribution of this dissertation was in developing work zone traffic flow forecasting models. One possible reason for work zone characteristics not playing a significant role in the prediction is that traffic flow variables serve as a proxy for different work zone characteristics. For example, intensity of work activity and number of closed lanes affect traffic flow. Thus including traffic flow works as a surrogate (or proxy) for the combined influence of various work zone characteristics. The main purpose of the study was to build predictive models to facilitate work zone ITS operations.

A few related research topics can be explored in the future. First, lane-changing models will be incorporated into a microscopic traffic simulation program, and the accuracy of the resulting capacity values will be investigated. Second, forecasting models by the type of work zone may be developed. For example, maintenance and construction work activities involve different intensities and durations. Thus, separate models for these two types may be warranted. Third, the proposed models may be applied to other urban areas to further investigate forecasting

models. Last, other artificial intelligence and advanced time-series models, such as support vector machine, Bayesian network, and Markov decision processes may be explored in the future.

REFERENCES

1. "NGSIM U.S. 101 Data Analysis Summery Report," Federal Highway Administration, Washington, D.C., 2005.

2. Abbas, M., L. Chong, and A. Medina, "Neural-agent driver behavior modeling using naturalistic data," In *Proc. 13th International IEEE Conference on Intelligent Transportation Systems*, 2010.

3. Abdulhai, B., H. Porwal, and W. Recker, "Short-term freeway traffic flow prediction using genetically-optimized time-delay-based neural networks," Presented at 78th Annual Meeting Transportation Research Board, Washington, D.C., 1999.

4. Abdulhai, B., J. B. Sheu, and W. Recker, "Simulation of ITS on the Irvine FOT area using Paramics 1.5 scalable microscopic traffic simulator, Phase I: model calibration and validation," California PATH Research Report UCB-ITS-PRR-99-12, 1999.

5. Ahmed, K. I., M. E. Ben-Akiva, H. N. Koutsopoulos, and R. G. Mishalani, "Models of freeway lane changing and gap acceptance behavior," in *Proc. 13th International Symposium Transportation and Traffic Theory*, pp. 501-515, 1996.

6. Awad, W. H., and B. N. Janson. "Prediction Models for Truck Accidents at Freeway Ramps in Washington State Using Regression and Artificial Intelligence Techniques," In *Transportation Research Record, vol. 1635*, pp. 30–36, 1998.

7. Breiman, L., "Random forests," *Machine Learning,* vol. 45, pp. 5–32, 2001.

8. Breiman, L., J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Tree.* Belmont, CA, USA: Wadsworth, 1984.

9. Brijs, T., D. Karlis, F. Bossche, and G. Wets, "A bayesian model for ranking hazardous road sites," *Journal of the Royal Statistical Society Series A*, vol. 107, pp. 1–17, 2007.

10. Buturovic, L. J., "Improving k-nearest neighbor density and error estimates," *Pattern Recognition,* Vol. 26, no. 4, 1993, pp. 611-616.

11. Ceylan, H., and M. G. H. Bell, "Traffic signal timing optimization based on genetic algorithm approach, including drivers' routing," *Transportation Research Part B*, vol. 38, no. 4, pp. 329–342, 2004.

12. Chang, L.-Y., and H.-W. Wang, "Analysis of Traffic Injury Severity: An Application of Non-Parametric Classification Tree Techniques," *Accident Analysis and Prevention*, vol. 38, no. 5, pp. 1019–1027, 2006.

13. Chen, H., S. Grant-Muller, L. Mussone, and F. Montgomery, "A study of hybrid neural network approaches and the effects of missing data on traffic forecasting," *Neural Computing and Applications*, vol. 10, pp. 277–286, 2001.

14. Chen, L., and C. L. P. Chen, "Ensemble learning approach for freeway short-term traffic flow prediction," in Proc. *International IEEE Conference System of Systems Engineering*, San Antonio, TX, 2007.

15. Chiu, S., and S. Chand, "Adaptive traffic signal control using fuzzy logic," In *Proc. 2nd IEEE International Conferences on Fuzzy Systems*, pp. 1371–1376, 1993.

16. Choudhury, C., V. Ramanujam, and M.Ben-Akiva, "Modeling acceleration decisions for freeway merges," *Transportation Research Record,* vol. 2124, pp. 45-57, 2009.

17. Chowdhury, M., and A. W. Sadek, "Advantages and Limitations of Artificial Intelligence," *Transportation Research Circular*, no. E-C168, pp. 6-8, November 2012.

18. Choy, M. C., R. L. Cheu, D. Srinivasan and F. Logi, "Real-time coordinated signal control through use of agents with online reinforcement learning," *Transportation Research Record*, *vol. 1836*, pp. 64–75, 2003.

19. Christensen, R., *Entropy Minimax Sourcebook.* vols. 1-4, Entropy Ltd., Lincoln, MA, 1980.

20. Clark, S., "Traffic prediction using multivariate nonparametric regression," *Journal of Transportation Engineering*, *ASCE,* vol. 129, no. 2, pp. 161–168, 2003.

21. Coufal, D., and E. Turunen, "Short term prediction of highway travel time using data mining and neuro-fuzzy methods," *Neural Network World*, vol. 3–4, pp. 221–231, 2004.

22. Daganzo, C. F., "Estimation of gap acceptance parameters within and across the population from direct roadside observation," *Transportation Research Part B,* vol. 15B, no. 1, pp. 1-15, 1981.

23. Dia, H., "An object-oriented neutral network approach to short-term traffic forecasting," *European Journal of Operational Research*, vol. 131, pp. 253–261, 2001.

24. Domeniconi, C., D. Gunopoulos, and J. Peng, "Large margin nearest neighbor classifiers," *IEEE Transactions on Neural Networks,* Vol. 16, no. 4, pp. 899-909, 2005.

25. Dougherty, M. S., and M. R. Cobbet, "Short-term inter-urban traffic forecasts using neural networks," *International Journal of Forecasting*, vol. 13, pp. 21–31, 1997.

26. Drew, D. R., L. R. LaMotte, J. H. Buhr, and J. A. Wattleworth, "Gap acceptance in the freeway merging process," Texas Transportation Institute, College Station, TX, 1967.

27. Duret, A., C. Buisson and N. Chiabaut, "Estimating individual speed-spacing relationship and assessing the Newell's car-following model ability to reproduce trajectories," Presented at 87th Annual Meeting Transportation Research Board, Washington, D.C., 2008.

28. DYMO, "Modelling of ITS applications: test of four dynamic models," DYMO final report. Centre of Traffic Simulation (CTR), Royal Institute of Technology (KTH) in co-operation with TRANSEK, ISSN 1104-683X, Stockholm, Sweden, 1999.

29. Edara, P., C. Sun, Y. Hou, "work zone deployment of variable advisory speed limits: mobility and safety evaluation," Presented at 93rd Annual meeting of the Transportation Research Board, Washington D.C., 2014.

30. Edara, P., and D. Teodorovic, "Model of an advance-booking system for highway trips." *Transportation Research C*, vol. 16, no. 1, pp. 36-53, 2008.

31. Edara, P., B. Smith, J. Guo, S. Babiceanu, C. McGhee, "A methodology to identify optimal placement of point detectors for travel time estimation," *Journal of Transportation Engineering, ASCE,* vol. 137, no. 3, pp. 155-173, 2011.

32. Edara, P., D. Teodorovic and H. Baik, "Using Neural Networks to Model Intercity Mode Choice" in *Intelligent Engineering Systems through Artificial Neural Networks*, New York, NY, USA: ASME, 2007, pp. 143-148.

33. Forrest, S., “Genetic algorithms: principles of natural selection applied to computation,” *Science*, vol. 261, pp. 872-878, 1993.

34. Foy, M. D., R. F. Benekohal and D. E. Goldberg, “Signal timing determination using genetic algorithms,” *Transportation Research Record,* vol. 1365, pp. 108–115, 1992.

35. Freund, Y., and R. E. Schapire. “Experiments with a new boosting algorithm,” *In Proceedings of 13th International Conference of Machine Learning*, 1996.

36. Freund, Y., and R. E. Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences,* vol. 55, pp. 119-139, 1997.

37. Gilmore, J. F., and N. Abe, “Neural network models for traffic control and congestion prediction,” *Journal of Intelligent Transportation Systems*, vol. 2, no. 3, pp. 231–252, 1995.

38. Gipps, P. G., “A model for the structure of lane changing decisions,” *Transportation Research,* vol. 20B, no. 5, pp. 403-414, 1986.

39. Hagan, M.T., and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, 1999, pp. 989–993, 1994.

40. Hamed, M. M., H. R. Al-Masaeid, and Z. M. Bani Said, “Short-term prediction of traffic volume in urban arterials,” *Journal of Transportation Engineering*, *ASCE,* vol. 121, no. 3, pp. 249–254, 1995.

41. Hastie, T., R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York, NY, USA: Springer-Verlag, 2001.

42. Henn, V., "Fuzzy route choice model for traffic assignment," *Fuzzy Sets and Systems*, vol. 116, no. 1, pp. 77–101, 2000.

43. Herman, R., and G. H. Weiss, "Comments on the highway crossing problem," *Operation Research,* vol. 9, no. 6, pp. 828-840, 1961.

44. Hidas, P., "Modeling vehicle interactions in microscopic simulation of merging and weaving," *Transportation Research Part C,* vol. 13, no. 1, pp. 37-62, 2005.

45. Huang, R., and S. Sun. "Kernel regression with sparse metric learning," *Journal of Intelligent & Fuzzy System: Application in Engineering and Technology*, vol. 24, no. 4, pp775-787, 2013.

46. Huisken, G., and E. C. van Berkum, "Comparative analysis of short-range travel time prediction methods," Presented at 82nd Annual Meeting of the Transportation Research Board, Washington, D.C., 2003.

47. Innamaa, S., "Short-term prediction of traffic situation using MLP-Neural Networks," Presented at 7th World Congress on Intelligent Transportation Systems, Turin, Italy, 2000.

48. Ishak, S., and C. Alecsandru, "Optimizing traffic prediction performance of Neural networks under various topological, input and traffic condition settings," Presented at 82nd Annual Meeting Transportation Research Board, Washington, D.C., 2003.

49. J. Hua and A. Faghri, "Development of neural signal control system—toward intelligent traffic signal control," *Transportation Research Record,* vol. 1497, pp. 53–61, 1995.

50. Kamarianakis, Y., and P. Prastacos, "Space-time modeling of traffic flow," *Computers & Geosciences*, vol. 31, no. 2, pp. 119–133, 2005.

51. Kesting, A., and M. Treiber, "Calibrating car-following models using trajectory data: a methodological study," Presented at 87th Annual Meeting Transportation Research Board, Washington, D.C., 2008.

52. Kim, D., "Neural networks for trip generation model," *Journal of the Eastern Asia Society for Transportation Studies*, vol. 4, 2001, no. 2, pp. 201–208, 2001.

53. Kim, T., D. Lovell, and J. Paracha, "A New Methodology to Estimate Capacity for Freeway Work Zones," Presented at 80th Annual Meeting Transportation Research Board, Washington, D.C., 2001.

54. Kita, H., "A merging-giveway interaction model of cars in a merging section: a game theoretic analysis," *Transportation Research Part A*, vol. 33, no. 3/4, pp. 305-312, 1999.

55. Kita, H., "Effect of merging lane length on the merging behavior at expressway on-ramps," in *Proc. 12th International Symposium Transportation and Traffic Theory*, pp. 37-51, 1993.

56. Ledoux, C., "An urban traffic flow model integrating neural networks," *Transportation Research Part C,* vol. 5, no. 5, pp. 287–300, 1997.

57. Lee, E. B., and C. Kim, "Automated work zone information system on urban freeway rehabilitation," *Transportation Research Record,* no. 1948, pp.77–85, 2006.

58. Lee, J., B. Abdulhai, A. Shalaby and E. H. Chung, "Real-time optimization for adaptive traffic signal control using genetic algorithms," *Journal of Intelligent Transportation Systems*, vol. 9, no. 3, pp. 111–122, 2005.

59. Levin, M., and Y. D. Tsao, "On forecasting freeway occupancies and volumes," *Transportation Research Record,* vol. 773, pp. 47–49, 1980.

60. Li, L., W.-H. Lin and H. Liu, "Type-2 fuzzy logic approach for short-term traffic forecasting," In *Proc. IEEE Conference on Intelligent Transport Systems*, 2006.

61. Lingras, P., and P. Mountford, "Time Delay Neural Networks Designed Using Genetic Algorithms for Short-term Inter-city Traffic Forecasting," IEA/AIE 2001, LNAI 2070, pp. 290–299, 2001.

62. List, G. F., and M. Cetin, "Modeling traffic signal control using petri nets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 3, pp. 177–187, 2004.

63. Lotan, T., and H. N. Koutsopoulos, "Models for route choice behavior in the presence of information using concepts from fuzzy set theory and approximate reasoning," *Transportation*, vol. 20, pp. 129–155, 1993.

64. Ma, X., and I. Andreasson, "Statistical analysis of driver behavior data in different regimes of the car-following stage," *Transportation Research Record,* vol. 2018, pp. 87-96, 2007.

65. Mahmassani, H., and Y. Sheffi, "Using gap sequences to estimate gap acceptance functions," *Transportation research Part B,* vol. 15B, pp. 143-148, 1981.

66. Martin, T., P. Harten, D. Young, E. Muratov, A. Golbraikh, H. Zhu, and A. Tropsha. "Does rational selection of training and test sets improve the outcome of QSAR modeling?" *Journal of Chemical Information and Modeling*. Vol. 52, No. 10, pp. 2570-2578, 2012.

67. Maunder, C. R. F., *Algebraic Topology.* New York, NY, USA: Dover, 1997.

68. Miaou, S., J. J. Song, and B. K. Mallick, "Roadway traffic crash mapping: a space-time modeling approach," *Journal of Transportation and Statistics*, vol. 6, no. 1, pp. 33–57, 2003.

69. Miller, A. J., "Nine estimators of gap acceptance parameters," In *proc. 5th International Symposium on the Theory of Traffic Flow and Transportation*, pp. 215-235, 1972.

70. Min, W., and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C,* vol. 19, no. 4, pp. 606–616, 2011.

71. Miranda–Moreno, L. F., A. Labbe, and L, Fu, "Bayesian Multiple Testing Procedures for Hotspot Identification," *Accident Analysis and Prevention*, vol. 39, no. 6, pp. 1192–1201, 2007.

72. Mozolin, M., J. C. Thill, and E. Lynn, "Trip distribution forecasting with multiplayer perceptron neural networks: a critical evaluation," *Transportation Research Part B,* vol. 34, pp. 53–73, 2000.

73. Murat, Y. S., and E. Gedizlioglu, "A fuzzy logic multi-phased signal control model for isolated junctions," *Transportation Research Part C*, vol. 13, no. 1, pp. 19–36, 2005.

74. Nakatsuji, T., and T. Kaku, "Development of a self-organizing traffic control system using neural network models," *Transportation Research Record,* vol. 1324, pp. 137–145, 1991.

75. Nakayama, S., and R. Kitamura, "Route Choice Model with Inductive Learning," *Transportation Research Record vol. 1725*, pp. 63–70, 2000.

76. Next Generation Simulation Fact Sheet, Washington, D.C., USA. (online). Available: ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.

77. Niittymaki, J., and M. Pursula, "Signal control using fuzzy logic," *Fuzzy Sets and Systems*, vol. 116, no. 1, pp. 11–22, 2000.

78. Nijkamp, P., A. Reggiani and T. Tritapepe, "Modelling inter-urban transport flows in italy: a comparison between neural network analysis and logit analysis," *Transportation Research C*, vol. 4, pp. 323–338, 1996.

79. Ossen, S., and S. P. Hoogendoorn, "Validity of trajectory-based calibration approach of car-following models in the presence of measurement errors," Presented at 87th Annual Meeting Transportation Research Board, Washington, D.C., 2008.

80. Pan, T., A. Sumalee, R. Zhong, N. and Indra-Payoong, "Short-term traffic state prediction based on temporal–spatial correlation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1242–1254, 2013.

81. Pappis, C. P., and E. H. Mamdani, "A Fuzzy Logic Controller for a Traffic Junction," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 10, pp. 707–717, 1977.

82. Park, B., C. J. Messer and T. Urbanik II, "Enhanced genetic algorithm for signal-timing optimization of oversaturated intersections," *Transportation Research Record*, vol. 1727, pp. 32–41, 2000.

83. Park, B., C. J. Messer and T. Urbanik II, "Traffic signal optimization program for oversaturated conditions: genetic algorithm approach," *Transportation Research Record,* vol. 1683, pp. 133–142, 1999.

84. Passino, K., and S. Yurkovich, *Fuzzy Control.* Menlo Park, CA, USA: Addison Wesley Longman, 1998, pp. 252–255.

85. Punzo, V., D. J. Formisano and V. Torrieri, "Nonstationary kalman filter for estimation of accurate and consistent car-following data," *Transportation Research Record,* vol. 1934, pp. 3-12, 2005.

86. Punzo, V., M. T. Borzacchiello and B. Ciuffo, "Estimation of vehicle trajectories from observed discrete positions and next-generation simulation program (NGSIM) data," Presented at 88th Annual Meeting Transportation Research Board, Washington, D.C., 2009.

87. Reggiani, A., and O. Tritapepe, "Neural networks and logit models applied to commuters' mobility in the metropolitan area of milan," *Neural Networks in Transport Systems*, 1998.

88. Russel, S., and P. Norvig, *Artificial Intelligence A Modern Approach*. 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003, pp. 653-663.

89. Sadek, A. W., "Artificial Intelligence Applications in Transportation," *Transportation Research Circular*, no. E-C113, pp. 1-6, January 2007.

90. Saito, M., and J. Z. Fan, "Artificial neural network–based heuristic optimal traffic signal timing," *Computer-Aided Civil and Infrastructure Engineering*, vol. 15, no. 4, pp. 293–307, 2000.

91. Sakai, A., S. Odagawa and Y. Masumoto, "Development of route calculation by genetic algorithm," Presented at 3rd Annual World Congress on Intelligent Transport Systems, Orlando, Florida, 1996.

92. Schintler, L. A., and O. Olurotimi, "Neural networks as adaptive logit models," *Neural Networks in Transport Systems,* 1998.

93. Shannon, C., and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL, USA: University of Illinois Press, 1949.

94. Smith, B. L., and M. J. Demetsky, "Short-term traffic flow prediction: neural network approach," *Transportation Research Record*, vol. 1453, pp. 98–104, 1994.

95. Smith, B. L., and M. J. Demetsky, "Traffic flow forecasting: comparison of modeling approaches," *Journal of Transportation Engineering*, *ASCE*, vol. 123, no. 4, pp. 261–266, 1997.

96. Smith, B. L., B. M. Williams, and K. R. Oswald, "Comparison of parametric and non-parametric models for traffic flow forecasting," *Transportation Research Part C,* vol. 10, no. 4, pp. 303–321, 2002.

97. Smith, S. A., "Freeway Data Collection for Studying Vehicle Interaction," Technical Report FHWA/RD-85/108, Federal Highway Administration, Office of Research, Washington D. C., 1985.

98. Srinivasan, D., and M. C. Choy, "Cooperative Multi-Agent System for Coordinated Traffic Signal Control," In *Proc. IEEE Intelligent Transport Systems*, vol. 153, no. 1, pp. 41–50, 2006.

99. Stathopoulos, A., and M. G. Karlaftis, "A multivariate state-space approach for urban traffic flow modeling and prediction," *Transportation Research Part C*, vol. 11 no. 2, pp. 121–135, 2003.

100. Sukthankar, R., S. Baluja, and J. Hancock, "Multiple adaptive agents for tactical driving," *Applied Intelligence*, vol. 9 (compendex), pp. 7–23, 1998.

101. Sun, D., R. F. Benekohal and S. T. Waller, "Multiobjective traffic signal timing optimization using nondominated sorting genetic algorithm," In *Proc. 2003 IEEE Intelligence Vehicles Symposium*, pp. 198-203, 2003.

102. Sun, H., H. X. Liu, H. Xiao, R. R. He, and B. Ran, "Short-term traffic forecasting using the local linear regression model," Presented at 82nd Annual Meeting Transportation Research Board, Washington, D.C., 2003.

103. Sun, S. and M. Xu, "Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 466–475, 2011.

104. Sun, S., C. Zhang, G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems,* vol. 7, no. 1, pp. 124–132, 2006.

105. Sun, S., R. Huang, and Y. Gao, "Network-scale traffic modeling and forecasting with graphic lasso and neural networks," *Journal of Transportation Engineering*, vol. 138, no. 11, pp. 1358-1367, 2012.

106. Teodorović, D., and S. Kikuchi, "Transportation route choice model using fuzzy inference technique," In *Proc. 1st International Symposium on Uncertainty Modeling and Analysis*, pp. 140–145, 1991.

107. Theodoridis, S., and K. Koutroumbas, *Pattern Recognition.* The 3rd ed., New York, NY, USA: Academic, 2006.

108. Toledo, T., H. N. Koutsopoulos, and M. Ben-Akiva. "Integrated driving behavior modeling". *Transportation Research Part C,* vol. 15, no. 2, pp. 96-112, 2007.

109. Trabia, M. B., M. S. Kaseko and M. Ande, "A two-stage fuzzy logic controller for traffic signals," *Transportation Research Part C*, vol. 7, no. 6, pp. 353–367, 1999.

110. USDOE, "Temporary Losses of Highway Capacity and Impacts on Performance," Oak Ridge National Laboratory (ORNL/TM-2002/3), 2002.

111. Van Lint, J. W. C., S. P. Hoogendoorn, and H. J. Van Zuylen, "Freeway travel time prediction with state-space neural networks: modeling state-space dynamics with recurrent neural networks," Presented at 81st Annual Meeting Transportation Research Board, Washington, D.C., 2002.

112. Vlahogianni, E. I., M. G. Karlaftis, and J. C. Golias, "Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach," *Transportation Research C,* vol. 13, pp. 211–34, 2005.

113. Wang, L. X., and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transaction on System, Man and Cybernetics*, vol. 22, pp. 1414–1427, 1992.

114. Williams, B. M., "Multivariate vehicular traffic flow prediction: an evaluation of ARIMAX modeling," *Transportation Research Record,* vol. 1776, pp. 194–200, 2001.

115. Williams, B. M., P. K. Durvasula, and D. E. Brown, "Urban traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transportation Research Record*, vol. 1644, pp. 132–144, 1998.

116. Xie, Y., D. Lord, and Y. Zhang, "Predicting motor vehicle collisions using Bayesian neural network models: an empirical analysis," *Accident Analysis and Prevention*, vol. 39, no. 5, pp. 922–933, 2007.

117. Yang, Q., and H. N. Koutsopoulos, “A microscopic traffic simulator for evaluation of dynamic traffic management systems,” *Transportation Research Part C,* vol. 4, no. 3, pp. 113-129, 1996.

118. Yasdi, R., “Prediction of road traffic using a neural network approach,” *Neural Computation and Application*, vol. 8, pp. 135–142, 1999.

119. Yeo, H., A. Skabardonis, J. Halkias, J. Colyar, and V. Alexiadis, “Oversaturated freeway flow algorithm for use in next generation simulation,” *Transportation Research Record,* vol. 2088, pp. 68-79, 2008.

120. Yun, S. Y., S. Namkoong, J. H. Rho, S. W. Shin, and J. U. Choi, “A performance evaluation of neural network models in traffic volume forecasting,” *Mathematical and Computer Modelling*, vol. 27, no.9–11, pp. 293–310, 1998.

121. Zhang Y., and Z. Ye, “Short-term traffic flow forecasting using fuzzy logic system methods,” *Journal of Intelligent Transportation Systems,* vol. 12, no. 3, pp. 102–112, 2008.

122. Zhang, H. M., “Recursive prediction of traffic conditions with neural networks,” *Journal of Transportation Engineering, ASCE,* November/December, pp. 472–481, 2000.

123. Zhang, L., H. Li and P. D. Prevedouros, “Signal control for oversaturated intersections using fuzzy logic,” Presented at 84th Annual Meeting of the Transportation Research Board, Washington, D.C., 2005.

124. Zhao, J., and B. K. Bose, “Evaluation of membership functions for fuzzy logic controlled induction motor drive.” In Proc. 28th Annual IECON Conference, vol. 1, IEEE Industrial Electronics Society, pp. 229-234, 2002.

125. Zuylen, H. V., “Difference between artificial intelligence and traditional methods,” *Transportation Research Circular*, no. E-C168, pp. 3-5, November 2012.

VITA

Yi Hou was a graduate research assistant in the Department of Civil Engineering at the University of Missouri-Columbia. He conducts research in areas of intelligent transportation systems, transportation modeling, transportation safety, artificial intelligence, and data mining. His research interests highlight a connection between machine learning and artificial intelligence approaches and transportation engineering research. To date he published 5 refereed journal articles in 4 different journals, 6 refereed conference proceedings, and 3 sponsored research reports. Before becoming a Ph.D. student at University of Missouri-Columbia, he received the B.S. degree in civil engineering from Southwest Jiaotong University, Chengdu, China, in 2008 and the M.S. degree in civil engineering from the University of Missouri, Columbia, MO, USA, in 2011.