DATA SCIENCE APPLICATION IN INTELLIGENT TRANSPORTATION SYSTEMS: AN INTEGRATIVE APPROACH FOR BORDER DELAY PREDICTION AND TRAFFIC ACCIDENT ANALYSIS

by

Lei Lin
January 5, 2015

A dissertation submitted to the
Faculty of the Graduate School of
the University at Buffalo, the State University of New York
in partial fulfillment of the requirements for the
degree of

Doctor of Philosophy

Department of Civil, Structural, and Environmental Engineering

ACKNOWLEDGEMENTS

First and foremost I would like to offer my greatest gratitude to my two supervisors, Dr. Adel W. Sadek and Dr. Qian Wang, who have supported me throughout my research studies with their patience and knowledge. Both of them have given me continuous encouragement and guidance on my research works in the past four and a half years. They always help me with useful suggestions to make research plans; they always respond very fast to my emails about any difficulties I met in the studies no matter how busy they are; and they always revise my paper sentence by sentence even a small typo in a figure. Without their help, this dissertation would not have been possible. I feel so fortunate to be one of their Ph.D. students. The moments that we have thought-provoking discussions, get promising experiment results, and work hard together on one paper will always be remembered by me.

Second, my special thanks to my family. My parents Jianzhou Lin and Jianrong Chu, who unconditionally support every decision I have made in my life, are my source of energy. My wife Yan Li, she always takes good care of me and also gives me huge help in getting my minor degree, the Master's Degree of Computer Science.

Finally, I would also like to thank my committee members, Dr. Panagiotis Ch. Anastasopoulos and Dr. Qing He for accepting the invitation to serve on my committee. I also feel very grateful for their precious advice and great ideas towards my dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ApEn	Approximate Entropy
AVI	Automatic Vehicle Identification
AVL	Automatic Vehicle Location
BMAP	Batch Markovian Arrival Process
CI	Conditional Independence
DTW	Dynamic Time Warping
ELC	Equally Likely Combination
ELV	Equally Likely Vehicles
EM	Expectation and Maximization
ETC	Electronic Toll Collection
FCM	Fuzzy C-means Clustering Method
FHWA	Federal Highway Administration
FP tree	Frequent Pattern Tree
k-NN	k Nearest Neighbor
HBDM	Hazard-based Duration Model
IAAFT	Iteratively Amplitude Adjusted Fourier Transform
ILD	Inductive Loop Detector
ITS	Intelligent Transportation System
LCC	Latent Class Clustering
LGP	Linear Gaussian Process

LRD	Long Range Dependence
MAPE	Mean Absolute Percentage Error
$\mathrm{M}/E_K/n$	Queueing Model with Exponential inter-arrival times and Erlang service times
NITTEC	Niagara International Transportation Technology Coalition
OPR	Object Purity Ratio
PH	Phase Types
ROPR	Relative Object Purity Ratio
SARIMA	Seasonal Autoregressive Integrated Moving Average Model
SPN	Spinning Network Method
SPSS	Statistical Package for Social Sciences
SVR	Support Vector Regression

ABSTRACT

With the great progress in information and communications technologies in the past few decades, intelligent transportation systems (ITS) have accumulated vast amounts of data regarding the movement iof people and goods from one location to another. Besides the traditional fixed sensors and GPS devices, new emerging data sources and approaches such as social media and crowdsourcing can be used to extract travel-related data, especially given the wide popularity of mobile devices such as smartphones and tablets, along with their associated apps. To take advantage of all these data and to address the associated challenges, big data techniques, and a new emerging field called data science, are currently receiving more and more attention. Data science employs techniques and theories from many fields such as statistics, machine learning, data mining, analytical models and computer programming to solve the data analysis task. It is therefore timely and important to explore how data science may be best employed for transportation data analysis. In this doctoral study, an integrative approach is proposed for data science applications in ITS. The proposed approach constitutes to an integration of multiple steps in the data analysis process, or integration of different models to build a more powerful one. The integrative approach is applied and tested on two case studies: border crossing delay prediction and traffic accident data analysis.

For the first case study, a two-step border crossing delay prediction model is proposed, consisting of a short-term traffic volume prediction model and a

multi-server queueing model. As such, this can be seen as an integration of data-driven models and analytical models. For the first step, the short-term traffic volume prediction model, an integration of data “width” decreasing (i.e., data grouping) step and model development step is applied. For model development, a model combination step of a Seasonal Autoregressive Integrated Moving Average Model (SARIMA) and Support Vector Regression (SVR) is applied to realize better performance than when using each single model. In addition, the spinning network (SPN) forecasting paradigm is enhanced for border crossing traffic prediction through the utilization of a dynamic time warping (DTW) similarity metric. The DTW-SPN is shown to yield several advantages such as computational efficiency and accuracy as demonstrated by a promising Mean Absolute Percent Error (MAPE) compared to SARIMA and SVR.

This dissertation also proposes the introduction of a data diagnosis step before short-term traffic prediction. In order to develop a methodology for model selection guidance, the author calculated the statistical measures of nonlinearity and complexity for multiple datasets and correlated those to the performances of multiple models SARIMA, SVR and k nearest neighbor (k-NN). Based on this, useful insights are revealed pertaining to parameter setting and model selection based on the data diagnosis results.

For the second step, namely the queueing model development, heuristic solutions are presented for two types of queueing models $M/E_K/n$ and *BMAP/PH/n*. These models take the predicted traffic volume as input, and use it to calculate future waiting time. The analytical results are compared to the results from a VISSIM model simulation results, and shown to be comparable. . Finally, an android smartphone app, which utilizes the two-step border prediction model methodology described above, is developed to collect, share and predict waiting time at the three Niagara Frontier border crossings.

For the second case study involving traffic accident data analysis, first an integration of a data “depth” decreasing step and a model development step is once again applied. To do this, the modularity-optimizing community detection algorithm is used to cluster the dataset, and for each cluster, the association rule algorithm is applied to yield insight into traffic accident hotspots and incident clearance time. The results show that more meaningful association rules can be derived when the data is clustered compared to when using the whole dataset directly. Secondly, an integration of a data “width” decreasing step (variable selection) and model development step is applied for real-time traffic accident risk prediction. For this, a novel variable selection method based on the Frequent Pattern tree (FP tree) algorithm is proposed and tested, before applying Bayesian networks and the k-NN algorithms. The experiment shows the models based on variables selected by FP tree always performed better than those using variables selected by the random forecast method. Lastly, an integration of the data mining model, M5P tree, and the hazard-based duration model (HBDM) statistical method is applied to traffic accident duration prediction. The M5P-HBDM method is shown to be capable of identifying more meaningful factors that impact the traffic accident duration, and to have a better prediction performance, than either M5P or HBDM.

The two case studies considered in this dissertation serve to illustrate the advantages of an integrative data science approach to analyzing transportation data. With this approach, invaluable insight is gained that can help solve transportation problems and guide public policy.

CHAPTER 1 INTRODUCTION

1.1 Data Science in Intelligent Transportation Systems

The recent years have witnessed a data explosion in many industries such as information technology, healthcare, retail, etc.. For example, according to Data Never Sleeps 2.0 Inforgraphic (DOMO, 2014), Google receives over 4 million search queries per minute. Wal-Mart, the retail giant, handles more than 1million customer transactions every hour, feeding databases estimated at more than 2.5 petabytes (The Economist, 2010).

The data explosion is also powerfully changing our intelligent transportation systems (ITS). ITS strive to improve transportation safety and mobility, and enhance American productivity, through the integration of advanced communications technologies into the transportation infrastructure and vehicles. ITS encompass a broad range of wireless and wire line communications-based information and electronics technologies (U.S. DOT, 2014). With the development of ITS, transportation professionals currently operate in an extremely data-rich environment, compared to the environment of a few decades ago. Every minute, the road sensors, Global Position System (GPS) devices, and smartphones record and accumulate huge amounts of movement information of people and goods from one location to another. Some researchers think that the large amount of data currently available can potentially lead to a revolution in intelligent transportation system (ITS), changing ITS from a conventional technology-driven system into a more powerful multifunctional data-driven intelligent transportation system (Zhang et al., 2011).

Given this, data science, which can be generally defined as the process of extracting useful knowledge from the data, is emerging as a new and important field of inquiry. More specifically, data science employs techniques and theories from many fields within the broad areas of mathematics, statistics, and information technology. This includes such theories and techniques as signal processing, probability models, machine learning, data mining, statistical learning, computer programming, data engineering, pattern recognition and learning, visualization, uncertainty modeling, data warehousing, and high performance computing (Wikipedia, 2014).

One important task of data science is to handle what is now referred to as “big data”. According to Douglas (2012), “big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization”. This definition clearly summarizes the three most important characteristics of big data, called the “3Vs” - volume (amount of data), velocity (speed of data in and out), and variety (range of data types and sources). ITS data can be definitely viewed as one type of “big data”. Companies such as IBM has recently brought in big data techniques to tackle challenging transportation problems such as real-time traffic data analysis and traffic conditions prediction and so on (Mukherjee, 2012).

Although most disciplines in data science, like statistics and machine learning, have been utilized in ITS data analysis previously, there are still a lot to be explored.

For example, it’s important to extract the most important aspects of the data given a variety of data sources. One also needs to choose the appropriate model or sometimes combine multiple models flexibly to answer one hard question. With new technologies still emerging, as reported by New York Times, “the new field of data science promises to revolutionize industries from business to government, health care to academia”. (New York Times, 2013)

1.2 ITS Data Introduction and Application

The wealth of transportation-related data presents the transportation community with an unprecedented opportunity in support of the planning, design, management, operations, and maintenance of the surface transportation system. The types and sources of ITS data are varied. This Chapter will briefly introduce several types of ITS data. Some of them have been widely used in transportation research in the past, while others have emerged recently.

1.2.1 Fixed Sensors

For several years, basic temporal traffic characteristics, such as traffic volume, occupancy and speed at a given point, have been recorded by fixed sensors on the side of roads. One of the most widely used types of fixed sensors is the inductive loop detector (ILD). Other fixed sensors such as optical detectors, ultrasonic detectors are also deployed. Although these fixed sensors are very useful, they fail in capturing the spatial aspect of traffic. In addition they suffer from their limited reliability, with their prohibitive cost in attaining significant coverage of the roadway network (Faouzi et al., 2011).

1.2.2 Probe Vehicles

The relevant probe vehicles techniques include Automatic Vehicle Location (AVL) systems onboard buses and other fleets, Automatic Vehicle Identification (AVI), ground-based radio navigation, cellular geo-location and the Global Positioning System (GPS) (Turner et al, 1998). Probe vehicles techniques can measure travel time directly. In addition, fundamental traffic characteristics like volume and speed can be inferred from probe vehicle data. Probe vehicle data can thus be used for real-time traffic operations monitoring, incident detection and route guidance applications.

1.2.3 Connected Vehicles

Connected vehicles technologies can create a safe interoperable wireless communications network that includes cars, buses, trucks, trains, traffic signals, cellphones, and other devices. Connected vehicles have the potential to provide transportation agencies with dramatically improved real-time traffic, transit, and parking data, making it easier to manage transportation systems for maximum efficiency and minimum congestion. Other environmentally relevant real-time transportation data can also be generated and captured by connected vehicles to support and facilitate green transportation choices. Another important application of connected vehicles data is to improve safety by increasing situational awareness and reducing or eliminating crashes through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) data transmission (U.S. DOT, 2014).

1.2.4 Inspection and Toll Stations

Toll stations are usually located along highways, at bridges and tunnels to collect usage fees which are then used to pay for the cost of the transportation infrastructure. Some stations also are intended to inspect travelers and/or vehicles (e.g., border crossing inspection). In order to decrease or eliminate delay on toll roads, electronic toll collection (ETC) systems like E-ZPASS in northeastern US, and NEXUS for US-Canada border crossing, have been used. As a natural by-product of the inspection or the toll collection process, traffic volume data are automatically available. These data can be used to predict travel demand, calculate and minimize delay time, evaluate the traffic safety (Abdelwahab and Abdel-Aty, 2002) and assess environmental impact near toll plazas (Chen and Li, 2010).

1.2.5 Traffic Accidents

Traffic accident data constitute also a rich data resource. . Countless research efforts have been conducted over the years to solve traffic accidents related problems. Currently, traffic accident data are usually derived from police crash reports, from which data like frequency of crashes at specific locations and the injury-severities of vehicle occupants can be extracted. A great deal of other information like time of day, age and gender of vehicle occupants, road-surface conditions, roadway lighting, and speed limits can also be derived to serve as explanatory variables in traffic accident modeling (Mannering and Bhat, 2014).

There are several types of traffic accident research studies including studies on hazardous locations/hot spot identification, and accident frequency/rate analysis designed to gain a better understanding of the factors that affect the probability of crashes; the idea behind such studies is that a better understanding of the likelihood of crashes can provide direction for policies and countermeasures aimed at reducing the number of crashes (Lord and Mannering, 2010). The second category of traffic safety studies are traffic accident injury-severities studies designed to understand the characteristics that may mitigate or exacerbate the degree of injury sustained by crash-involved road users, given that a crash has occurred (Savolainena et al., 2011). The third category is accident duration analysis, which can predict accidents' duration under various conditions (different local and regional traffic conditions, time of day, day of week, weather conditions, work zones, etc…). Based on this predicted duration information, the authorities can then allocate incident response personnel and resources more effectively, and inform travelers about incidents more accurately. The last category is real-time traffic accident risk prediction modeling that take advantage of high-resolution, complex and rapidly and continuously flowing data instead of employing aggregated measures of traffic flow variables (e.g., speed limits for speed, AADT for flow, etc.) for predicting traffic accidents (Hossain and Muromachi, 2012).

1.2.6 Social Media and Crowd Sourcing

With the popularity of mobile devices like smartphones and online social platforms, there now exists the opportunity to glean very useful travel-related data from mining publicly available *social media* data such Twitter, Facebook, Foursquare, and Google+. This could provide unique insight into traveler behavior, while offering a cost-effective alternative to the traditional methods of collecting travel behavioral data. Traffic information can also be extracted conveniently from some transportation related smartphone applications based on crowd sourcing theory, like WAZE, a community-based traffic and navigation application (app) acquired by Google in 2013, where drivers can share real-time traffic and road information, saving travel time, gas and money on their daily commute.

1.2.7 Web-based Mapping Services

Web-based mapping services like Google Maps, Bing Maps, MapQuest, NextBus, Nokia Maps (Here), TomTom, INRIX and many other service providers are providing traffic information (e.g., real-time travel time given an origin-destination pair). These traffic data can be downloaded through the Application Programming Interface (API) provided by the map servicers. Some researchers have shown that travel time data downloaded from “virtual sensors” through Bing Maps APIs have a strong correlation with data from infrastructure-based sensors (Morgul et al., 2014). For example, Tostes et al. (2013) applied image processing techniques to Bing Maps of Chicago to acquire and analyze the traffic flow intensity based on the change of the color-coded road links on the map (where green represents free-flow conditions and red congestion). They then developed a logistic regression model to predict future traffic flow intensity.

1.2.8 Computer Simulation Data

Simulation is an effective way to study the dynamic changes of a transportation system and the interactions among its different elements through the application of computer software. Simulation offers a few obvious advantages over real-world experiments (Huang, 2011). First, the depth of understanding that can be achieved by simulation and modeling can hardly be achieved in other ways. Second, the cost of simulation and modeling is much lower than that of other ways, since it does not require any building or construction in reality. Third, the speed of simulation and modeling is mainly constrained by computational resources, but not physical factors, which means it is much faster and offers greater efficiency. Fourth, in most transportation case studies, simulation and modeling is the only choice, because real-world experiments are too costly, impractical, or impossible.

Therefore simulation can also help generate data for ITS development and analysis. Examples include user behavior data recorded by a driving simulator for traffic safety studies, system performance information for traffic flow under conditions like inclement weather, waiting time data for a queueing system of a toll plaza, connected vehicle data and traffic emission data.

1.2.9 Weather Data

It is widely acknowledged that inclement weather could have a significant impact on transportation, such as traffic safety, traffic flow characteristics, and road infrastructure and agency productivity. For example, in traffic safety, a report by the Federal Highway Administration (FHWA) shows that every year on average 23% of all vehicle crashes (more than 1.3 million) are a result of inclement weather, and 6,250 people are killed and over 480,000 people are injured in weather-related crashes (FHWA, 2014).

Weather data include temperature, visibility, wind speed, and precipitation information. Weather data can be downloaded from commercial weather service APIs that provide real-time and historical weather information via the Internet.For example, the Weather Underground website combines data from personal weather stations with data from quality controlled and the automated airport weather stations (Weather Underground, 2014).

1.3 ITS Data Analysis

Given the different types of ITS data discussed above, the remainder of this Chapter will introduce several applications of data science in analyzing ITS data. The discussion will be organized under two headings: (1) the data analysis process; and (2) data analysis models used to drive value from the data.

1.3.1 The ITS Data Analysis Process

Usually, an ITS data analysis process may include one or more of the following important steps, data preprocessing, data fusion, data “width and depth” reduction, and model development, as shown in Figure 1-1.

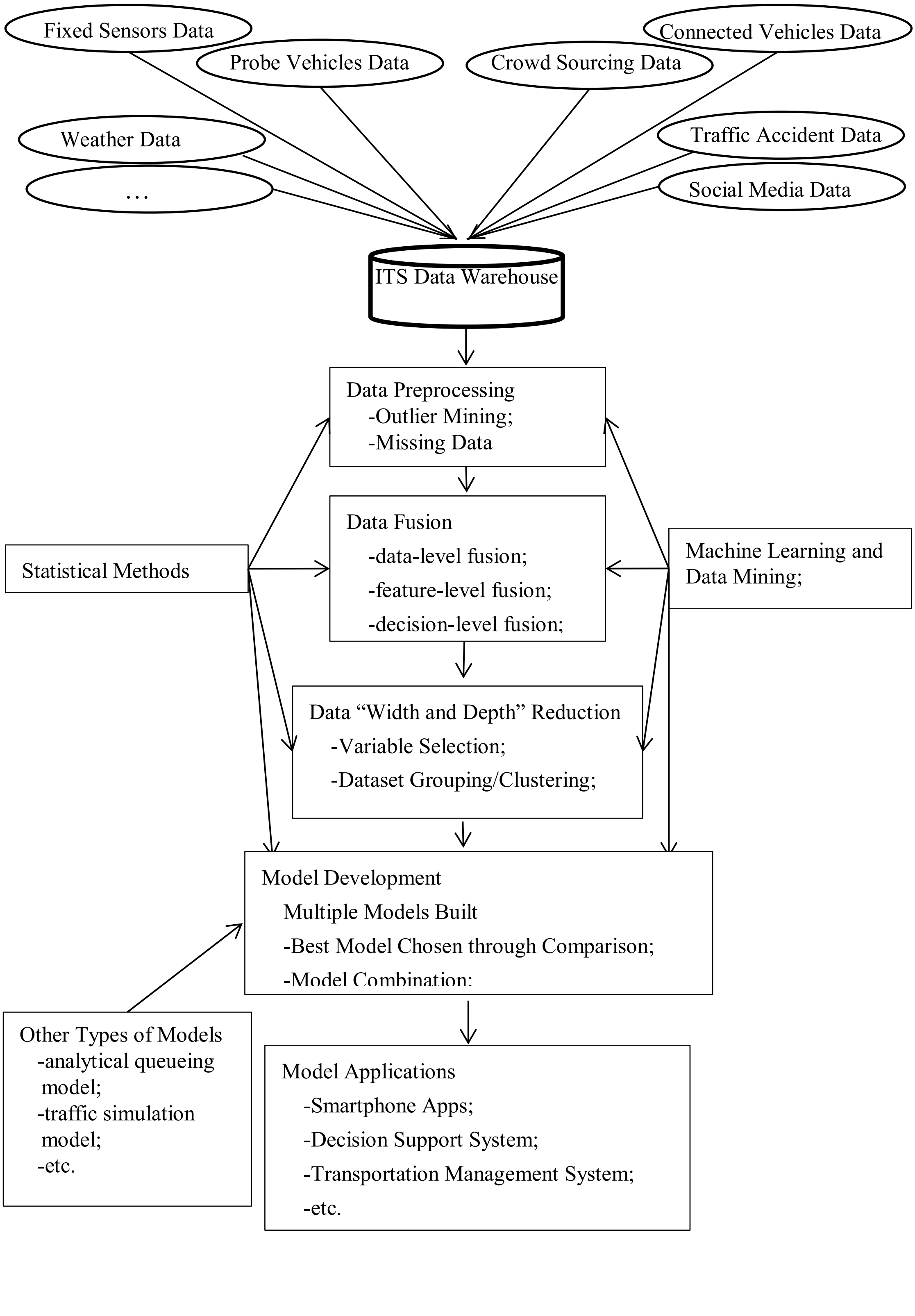


Figure 1-1 ITS Data Analysis Process

1.3.1.1 Data Preprocessing

As can be seen in Figure 1-1, the first necessary step in the ITS data analysis process is data preprocessing which may include outlier mining and missing data estimation. The objective of the data preprocessing is to produce a "clean and complete" traffic dataset that can be used for further analysis.

Outliers in ITS data may be the result of measurement error or equipment failure (faulty values); in that case it is essential and necessary to identify those kinds of outliers and remove them from the dataset. On the other hand, outliers may represent correct data points reflecting the ground truth. In that case, outliers may reveal important patterns in the data. Three typical outlier mining approaches can be identified: (1)the statistical-based approach; (2) the distance-based approach; and (3) the density-based approach. These approaches were applied on a travel time dataset and a traffic flow dataset by Chen et al.(2010).

The problem of missing data is another critical issue that needs to be addressed during the data analysis process. According to one research report, at any given time, approximately 25-30% of the detectors are off-line and contribute to missing data problems (Nguyen and Scherer, 2003). To deal with the missing data problem, techniques range from simply taking the average of the overall series to complex machine learning algorithms. A detailed review of data missing estimation techniques can be found in the paper published by García-Laencina et al. (2010). Most recently, Tan et al. (2013) proposed a tensor-based model to estimate missing traffic data and showed that their model can achieve a better imputation performance than the state-of-the-art approaches even when the missing ratio is up to 90%.

1.3.1.2 Data Fusion

A widely accepted definition of data fusion is “A multi-level process dealing with the association, correlation, combination of data and information from single and multiple sources to achieve refined position, identify estimates and complete and timely assessments of situations, threats and their significance” (Joint Directors of Laboratories, 1991). With such a variety of potentially useful ITS data, data fusion has become one indispensable part in solving ITS problems. For example, one application of ITS data fusion involves combining data from loop detectors and probe vehicles to better estimate travel time. Faouzi et al. (2011) gave a detailed review of data fusion applications in ITS systems such as the advanced traveler information systems, automatic incident detection, traffic forecasting and traffic monitoring.

There are three different levels of data fusion: data-level fusion, feature-level fusion and decision-level fusion (Hall and Llinas, 1997). Data-level fusion simply means the raw data can be directly combined if the sensor data are commensurate (i.e., if the sensors are measuring the same physical phenomena such as two visual image sensors or two acoustic sensors). Feature-level fusion means features are extracted from multiple sensor observations, and combined into a single concatenated feature vector which is used as an input for the models. Decision-level fusion involves fusion of sensor information, after each sensor has made a preliminary determination of an entity’s location, attributes, and identity. One example of decision level fusion to improve the classification accuracy for the severity of road traffic accidents based on the Dempster–Shafer algorithm, the Bayesian procedure and logistic model (Sohn and Lee, 2003).

1.3.1.3 Data “Width and Depth” Reduction

As can be seen in Figure 1-1, data “width” reduction refers to the variable selection (feature selection) step, whereas data “depth” reduction refers to data grouping and clustering. These two approaches have been shown to be useful in ITS data analysis, and are likely to become even more useful in the era of big data. The paragraphs below will introduce data width and depth reduction techniques. .

Different from feature-level fusion, in which different types of features can be concatenated and a new feature representation can be obtained for training and testing, variable selection (feature selection), or data width reduction methods, aims at identifying a subset of relevant features for improved or comparable recognition performance (Yeh et al., 2012). A large number of potential input variables can also result in the “garbage in-garbage out” effect (Papadokonstantakis et al., 2005), which refers to the times when computers unquestioningly process unintended, even nonsensical, input data ("garbage in") and produce undesired, often nonsensical, output ("garbage out") (Wikipedia, 2014). More is not always the better. Variable selection can thus help researchers identify and extract meaningful information (patterns, structure, underlying relationships, etc.) from the data. Only a small representative subset of the original feature space of the data may be needed to interpret the results (Fernández et al., 2014).

Besides that, the inclusion of a large number of explanatory variables may cause model overfitting (Sawalha and Sayed, 2006). It can also cause application-related problems such as long model running time and unreliable prediction results, particularly when a model is applied to new locations and larger data instances (Fernández et al., 2014).

Besides variable selection which can decrease the "width" of the dataset, data grouping and clustering techniques aim at reducing the "depth" problem of a dataset. Especially when the variation in observations is relatively large, clustering the data set first and fitting a model for each cluster accordingly would work better than developing a model based on the whole dataset (Sohn and Lee, 2003). For example, for short-term traffic volume prediction, studies have shown that the prediction results could be significantly improved if the input data were to be first grouped or clustered (Wild, 1997; Chrobok et al, 2004). This is mainly because daily traffic patterns vary significantly based on whether the prediction is for a weekday, weekend, or a special day (e.g., a holiday or a special event day). Another example where data clustering has proven very useful is in accident analysis where Abdel-Aty and Yu (2014), for example analyzed crashes by types, in the context of real-time crash risk assessment. Besides improving accuracy, dividing the dataset helps reduce computation time (Chung, 2004). Given this, preliminary data analysis is often conducted, prior to modeling, to determine if traffic data grouping and clustering is necessary.

1.3.1.4 Model Development

The last important step in ITS data analysis process is model development. Usually for the same ITS data analysis task, several models may be built and compared. One way is to simply pick up the model with the best performance, and the other way is to combine the results of a few models together, known as model combination (this is introduced later in this Chapter).

Note that model combination is different from decision-level data fusion, where each classifier is trained based on its corresponding sensor data and the results predicted by the different classifiers (and the associated features) are combined later via voting or other strategies to reach the final output (Yeh et al., 2012). In the case of model combination, the models are first generated by resampling using algorithms like bagging (Breiman, 1996), random forests (Breiman, 2001), and adaboosting (Schapire and Singer, 1999). As a result of the sampling, different subsets of the original dataset are obtained and utilized to train the different algorithms. Another approach is hybrid learning where different algorithms are trained on the original dataset (Geneva Artificial Intelligence Lab, 2009). Following model development, the prediction results of the different models are combined through fixed weights and/or Adaptive weights (Freitas and Rodrigues, 2006), or through majority voting (Lam and Suen, 1997).

The model combination method has been shown to improve the accuracy and robustness over a single model approach, for both regression (Clemen, 1989) and classification problems (Wang, et al., 2003). This is because in real-world problems, the true model is likely to be unknown, and hence choices and assumptions would have to be made to allow the problem under study to be acceptably modelled and solved (Freitas and Rodrigues, 2006). Thus a small perturbation of the training sample may change the prediction results of a single model (Breiman, 2001). Through model combination, this risk could be decreased.

Besides that, in time series prediction, the real-world time series are not purely linear or nonlinear, they often contain both linear and nonlinear patterns. As a result, a stand-alone ARIMA approach or a stand-alone neural networks (NNs) approach may not be adequate for modeling and forecasting time series. This is because the ARIMA model cannot deal with nonlinear relationships whereas NNs cannot handle both linear and nonlinear patterns equally well (Zhang, 2003). Given this, ARIMA models have been combined with some AI-based algorithms like NNs, support vector regression (SVR) in recent time series forecasting studies (e.g., Tseng, 2002; Zhang, 2003; Chen and Wang, 2007).

Examples of the model combination method for ITS data analysis include its application to short-term traffic volume prediction (Zheng, et al., 2006; Stathopoulos, et al., 2008; Tan, et al., 2009), travel time prediction (Van Hinsbergen and Van Lint, 2008), traffic accident severity classification (Sohn and Lee, 2003), computer vision related applications like vehicle license plate recognition (Dlagnekov, 2004), vehicle make recognition (Zhang, 2013), pedestrian detection (Nanni & Lumini， 2008), and road sign identification (Kouzani, 2007).

1.3.1.5 Model Application

After the models are developed, they can be integrated into all kinds of decision support systems, and transportation management systems for use by transportation authorities. They can also be developed into smartphone apps for travelers, a trend which is becoming more and more popular in routing, parking, and public transit applications.

1.3.2 ITS Data Analysis Models

From the previous introduction of the ITS data analysis process shown in Figure 1-1, we can see that there are mainly two groups of models which can be used within the different steps of the data analysis process. The first group is statistical models, and the second is machine learning and data mining. Historically, statistical models have played an important role in transportation data analysis, but nowadays methodologies from machine learning and data mining have also become popular in addressing the ITS data analysis challenges especially when it comes to the area of big data. The following sections discuss the advantages and limitations of the two groups of models in ITS data analysis. Some other models like queueing model and traffic simulation will also be briefly introduced.

1.3.2.1 Statistical Models

As defined by the American Heritage Dictionary, statistics is the mathematics of collecting, organizing and interpreting numerical data, particularly when these data concern the analysis of population characteristics by inference from samples.. Statistics have a solid and a widely accepted mathematical foundation, and can thus provide insight into the mechanisms creating the data (Karlaftis and Vlahogianni, 2011). A statistical model is a set of probability distribution functions on the sample space (McCullagh, 2002). The real-world phenomenon and behavior are then interpreted by checking the parameters of the statistical models (Schutt & O'neil, 2013).

Statistical models have been applied widely to ITS data analysis. Examples include:(1) outlier detection in traffic data using a discordancy test (Chen et al., 2010); (2) the use of the Dempster–Shafer algorithm in decision-level data fusion (Sohn and Lee, 2003); (3) the application of the classic Autoregressive Integrated Moving Average (ARIMA) model in traffic volume prediction (Ahmed and Cook, 1979); (4) the application of the multinomial logit model (MNL) in studying model choice behavior (Koppelman and Bhat, 2006); (5) the utilization of hazard based duration model (HBDM) in traffic accident clearance time analysis (Nam and Mannering, 2000), among others. However, the disadvantages of statistical models are that they cannot effectively deal with complex and highly nonlinear data (curse of dimensionality) (Karlaftis and Vlahogianni, 2011), a problem which is very common in ITS data analysis.

1.3.2.2 Machine Learning and Data Mining

On the other hand, machine learning is a subfield of computer science that deals with the construction and study of systems that can learn from data. Arthur Samuel in 1959 defined it as “a field of study that gives computers the ability to learn without being explicitly programmed”. Machine learning is employed in a range of computing tasks where designing and programming explicit, rule-based algorithms is infeasible (Wikipedia, 2014). Data mining is the process of discovering interesting patterns and knowledge from large amounts of data. The data sources can include databases, data warehouses, the Web, and other information repositories or data that are streamed into the system dynamically (Han et al., 2011). Machine learning and data mining are sometimes conflated; the former focuses on prediction, based on known properties learned from the training data, and the latter focuses on the discovery of unknown properties in the data (Wikipedia, 2014).

The promising performances of machine learning and data mining in dealing with the high nonlinearity in ITS data have been shown in comparisons with statistical models. Widely used machine learning and data mining models in ITS data analysis include support vector regress (SVR) for missing data estimation (Zhang and Liu, 2009), random forests for variable selection (Abdel-Aty et al., 2008), neural network (NN) in traffic volume prediction (Smith and Demetsky, 1994), Bayesian network models in real time traffic accident risk prediction (Hossain and Muromachi, 2012), among others. However, one primary disadvantage of the machine learning and data mining approach stems from the fact that they are essentially “black box” (the knowledge stored is not transparent to the user). These algorithms don’t generally focus on the interpretation of the parameters, and unlike statistical models for which the confidence intervals and posterior distributions for parameters and estimators are provided, machine learning algorithms like k-means or k-nearest neighbors, don’t have a notion of confidence intervals or uncertainty (Schutt & O’neil, 2013).

1.3.2.3 Other Types of Models

Besides statistical models, machine learning and data mining models, there are also some other types of models which play a role in ITS data analysis. First, there are queueing models where the system is described in terms of the distributions of inter-arrival time and service time, along with the number of servers. Van Woensel and Vandaele (2006) proved the validity of the queueing approach to uninterrupted traffic flows by comparing the queueing results with observed data on speed and flow; a detailed review can be found in Van Woensel and Vandaele (2007). Another example are traffic simulation models. Before applying traffic simulation models, however, they must be calibrated and validated using real-world traffic data to ensure the validity of the results. Brockfeld et al. (2004), for example, calibrated a microscopic traffic simulation model by feeding the GPS data of the leading vehicle into it, and validated the model by comparing the simulated headway with the measured headway of the following vehicle.

1.3.2.4 Summary

From the aforementioned sections, we can see that statistical models and machine learning and data mining have their own advantages and limitations. Although machine learning and data mining algorithms have shown better performances than statistical models in a few categories of ITS data analysis, their shortcomings, such as the "black box" property and the lack of correspondence between the real-world and the model parameters, are also worth noting. Besides that, as pointed out by Karlaftis and Vlahogianni (2011), it's sometimes unfair to compare a complex machine learning or data mining model with a simple linear regression or linear ARIMA model. They suggested that instead of solely considering model accuracy, the model simplicity and model suitability should also be compared.

The good news is that in ITS data analysis process, multiple types of models may be utilized together. For example, a statistical model could be applied to find the outliers in the data before a data mining algorithm is used for short-term traffic volume prediction. In the model combination step, the results of a statistical model and a machine learning model may be combined to increase the accuracy. A time series analysis used to predict traffic demands can be integrated with a traffic simulation model to realize the on-line prediction of freeway travel time (Juri et al., 2007).

1.4 Research Overview

After introducing the wide range of possible applications of data science in ITS data analysis and modeling, this section will give an overview of the research performed in this dissertation. As is well-known, the two primary objectives of the transportation engineering profession is to improve the efficiency and safety of the transportation system. Therefore, this dissertation explores the application of data science to two case studies from the real-world, the first one focused on improving transportation system efficiency and the second focused on its safety, as discussed below. Specifically, the focus of the first case study is on "border crossing delay prediction", so as to improve the efficiency of the transportation system, and the second case study involves "traffic accident data analysis" in order to improve safety.. The two case studies considered result in five research subtopics involving the application of data science methods and paradigms, as described below.

1.4.1 Two Case Studies

1.4.1.1 Niagara Frontier International Border Crossing Delay Prediction

The first case study involves predicting the border crossing delay at the Niagara Frontier International border crossing. The Niagara Frontier International Border include three main bridges connecting Western New York, U.S. to Southern Ontario, Canada, namely the Lewiston-Queenston Bridge, the Rainbow Bridge, and the Peace Bridge as shown in Figure 1-2.

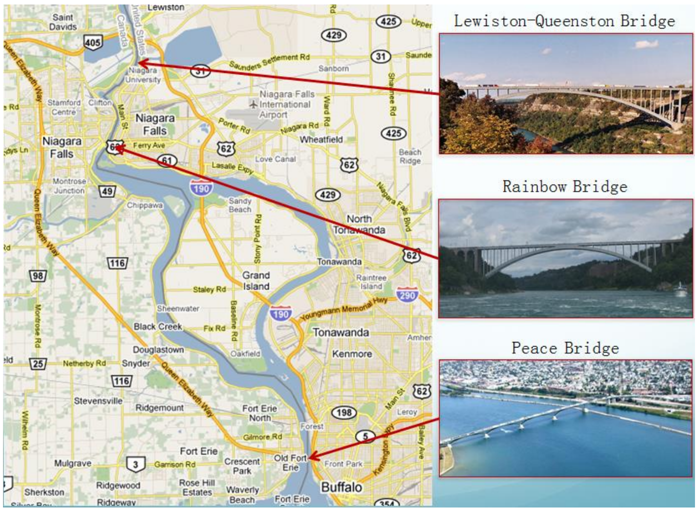


Figure 1-2 Three Bridges at Niagara Frontier International Border

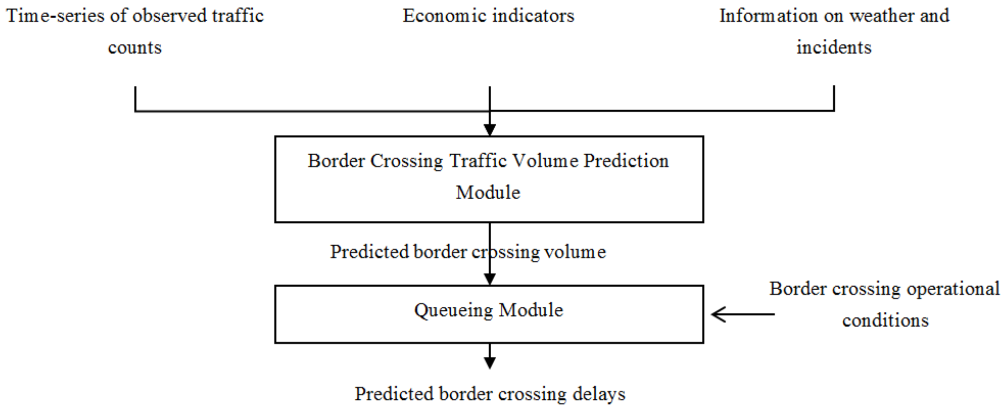
In recent years, and as a result of the continuous travel demand increase, coupled with a tighter security and inspection procedures after September 11, border crossing delay has become a critical problem. As reported by the Ontario Chamber of Commerce, the border crossing delay causes an annual loss of approximate $268.45 million for New York State. For the whole U.S., the cost is much higher and is estimated at about $4.13 billion every year (Ontario Chamber of Commerce, 2005). The report also warns that if the border delay issue is not adequately addressed, the U.S. stands to lose close to 17,500 jobs by 2020, and close to 92,000 jobs by 2030. Besides that, according to a press release in 2008 given by the U.S. Transportation Secretary, Mary E. Peters, the US-bound traffic from Canada encountered delays as high as three hours at several crossings, with delays costing businesses on both Canadian and the US sides as many as 14 billion dollars in 2007 (U.S. Department of Transportation, 2008).

Besides the economic impact of border crossing delays, border delays and the associated idling of traffic awaiting inspection also have a significant environmental cost. This includes the cost of the fuel wasted during idling, traffic-related pollutants and health hazards. A 10-year study by Lwebuga Mukasa et al. (2002) showed a positive relationship between increased commercial traffic volume at Peace Bridge border crossing between downtown Buffalo, New York and Fort Erie, Ontario, and the increased use of asthma health care.

To address these issues, the Niagara International Transportation Technology Coalition (NITTEC), a coalition of fourteen different agencies in Western New York and Southern Ontario, has been providing *current* or *instantaneous* border crossing delays to the public for years. The current border crossing times are communicated to the public via several information dissemination avenues, such as websites, on-road dynamic message signs and a traveler information phone system.

However, there is an inherent limitation associated with providing just the current border delay. This is because the current delay is likely to be quite different from the future delay that the travelers would experience by the time they arrive at the border, especially if there is a significant lag between the time when travelers need to act on the information provided and the time of their arrival at the border. If the future delay can be predicted, first, it would be more informative for travelers and businesses to select the time to depart and the route to pursue. Second, predicting border crossing delays would help customs and border protection authorities determine the needed staffing level to meet the expected travel demand. Third, with predicted border crossing delays, intelligent routing algorithms could be developed to optimally direct and route border-destined traffic in a fashion that would minimize the overall system travel time or the negative impacts on the environment.

Therefore this dissertation proposes a two-step border delay prediction model that is composed of two sequential modules as shown in Figure 1-3 below. The first module is designed to predict the traffic volume arriving at the border crossings for each time period. Given the predicted traffic volume as the input, the second model estimates the corresponding waiting time by solving a transient multi-server queueing problem.



Time-series of observed traffic counts
Economic indicators
Information on weather and incidents
Border Crossing Traffic Volume Prediction Module
Predicted border crossing volume
Queueing Module
Border crossing operational conditions
Predicted border crossing delays

Figure 1-3 Framework of the Two-step Border Crossing Delay Prediction

1.4.1.2 Traffic Accident Data Analysis

The second case study is traffic accident data analysis. Traffic accidents cause a great deal of loss of lives and property. According to the accidents report of the United States Census Bureau, there were 10.8 million accidents and 35,900 persons killed in 2009 (US Census Bureau, 2012). Besides that, As have been pointed out by a lot of researchers, traffic incidents account for more than 50% of motorist delays on freeways (Chin et al., 2004; Farradyne, 2000). The societal cost of such incidents comes in the form of lost productivity, wasted fuel, harmful emissions, and potential secondary incidents.

Countless research efforts have been conducted to solve these traffic accidents related problems. As previously mentioned, a few important types of traffic accident research studies include: studies on high accident frequency locations (hotspots) identification; accident frequency/rate analysis; traffic accident injury-severities analysis; accident duration prediction; and real-time traffic accident risk prediction models. This dissertation will use two traffic accident datasets collected from interstate I-190 in Buffalo, NY and I-64 in Norfolk, Virginia and study three subtopics that include: (1) traffic accident hotspots identification and clearance time analysis; (2) traffic accident duration prediction; and (3) real time traffic accident risk prediction.

1.4.2 Five Research Subtopics

In this dissertation, the research involving the border crossing delay prediction problem can be viewed as consisting of two research subtopics, whereas the traffic accident data analysis included three additional research subtopics, as described below.

1.4.2.1 Short-term Traffic Volume Prediction Model

In this task, , based on an analysis of the diurnal distribution of hourly traffic volumes at the Peace Bridge, six separate groups are defined and individual models including the Seasonal Autoregressive Integrated Moving Average (SARIMA) model and Support Vector Regression (SVR) are developed for weekdays (Monday -

Thursday), Fridays, Saturdays, Sundays, holidays and game days. For each group, a model was built by combining forecasts from the SARIMA model, with forecasts made by SVR. The two models' forecasts are combined using: (1) a simple fixed weight procedure; and (2) the fuzzy adaptive variable weight method, based on the Fresh Degree Function. The study's findings appear to confirm the hypothesis that, SARIMA model does a good job capturing the linear characteristics of the data (e.g., seasonality and trend), but SVR appears to outperform SARIMA in modeling the data's nonlinear aspects. The study also shows that combining forecasts from the two models, especially using the fuzzy adaptive variable weight method, yields excellent prediction performance.

Secondly, also under this research subtopic, the Spinning Network (SPN) method, a novel forecasting technique developed by Huang and Sadek (2009), is improved. The improvement is centered on the use of the Dynamic Time Warping (DTW) algorithm, to assess the similarity between two given time series, instead of using the Euclidean Distance as was the case with the original SPN. The performance of the DTW-SPN is then compared to that of three other forecasting methods, namely: (1) the original SPN (referred to as the Euclidean-SPN); (2) the SARIMA method; and (3) SVR. Both classified as well as non-classified datasets are utilized. The results indicate that, in terms of the Mean Absolute Percent Error, the DTW-SPN performed the best for all data groups with the exception of the "game day" group, where SVR performed slightly better. From a computational efficiency standpoint, the SPN-type algorithms require runtime significantly lower than that for either SARIMA or SVR. The performance of the DTW-SPN was also quite acceptable even when the data was not classified, indicating the robustness of the proposed forecasting method in dealing with heterogeneous data.

At last, considering that there is a lack of studies which focus on how to choose the appropriate prediction method based on the statistical characteristics of the dataset, the predictability of four different traffic volume datasets is diagnosed using various statistical measures including: (1) complexity analysis methods such as the delay time and embedding dimension method and the approximate entropy method; (2) nonlinearity analysis methods like the time reversibility of surrogate data; and (3) long range dependency analysis techniques like the Hurst Exponent. Following the diagnosis of the datasets, three short term traffic volume prediction models are applied: (1) SARIMA; (2) k Nearest Neighbor (k-NN); and (3) SVR. The results from the statistical data diagnosis methods are then correlated to the performance results of the three prediction methods on the four datasets in order to arrive at some conclusions regarding how to choose the appropriate prediction method.

1.4.2.2 Queueing Model

As the second step in estimate border crossing delay, the second research subtopic involved developing two classes of multi-server queuing models based on real-time traffic volume and inspection time data collected at the Peace Bridge. The two models are: (1) queueing models with exponential inter-arrival times and Erlang service times called $\mathrm{M}/E_K/n$; and (2) a more generic model with a Batch Markovian Arrival Process

(BMAP) and phase types (PH) services called *BMAP/PH/n*. The transient solution of the queueing models is obtained using heuristic methods. For validation, the queueing models' estimates are compared to the results from a detailed microscopic traffic simulation model of the Peace Bridge border crossing. The comparison shows that the transient queueing model, along its heuristic solution algorithm, is capable of predicting border crossing delay.

1.4.2.3 Traffic Accident Hotspots and Clearance Time Analysis

In this third research subtopic, the potential for using complex network analysis and data mining methods, namely a *modularity-optimizing community detection* algorithm and *association rules learning* algorithm, are explored to identify important accident characteristics based on an accident dataset compiled for interstate highway I-190 in the Buffalo-Niagara metropolitan area. The *community detection algorithm* is used first to cluster the data in order to reduce the inherent heterogeneity, and then the *association rule learning* algorithm is applied to each cluster to discern meaningful patterns within each, particularly related to high accident frequency locations (hotspots) and incident clearance time. The study results indicate that: (1) the *community detection algorithm* was quite effective in identifying clusters with discernible characteristics; (2) clustering helped in unveiling relationships and accident causative factors that remained hidden when the analysis was performed on the whole dataset.

1.4.2.4 Real-time Traffic Accident Risk Prediction

In the fourth subtopic, a novel variable selection method based on Frequent Pattern tree (FP tree) is proposed, as well as a new variable importance metric the Relative Object Purity Ratio (ROPR). The research develops two traffic accident risk prediction models a k-NN model and a Bayesian network based on accident data collected on interstate highway I-64 in Virginia. Prior to model development, two variable selection methods are utilized: (1) the FP tree method; and (2) the random forest method. The results show that the accident risk prediction models based on FP tree method perform better than the random forest based models, regardless of the type of prediction models (i.e. k-NN or Bayesian network), their parameter settings, and the types of datasets used for model training and testing.

1.4.2.5 Traffic Accident Duration Prediction

Both M5P tree based model and hazard-based duration model (HBDM) have been applied in traffic accident duration prediction. M5P can build tree-based models, like the traditional classification and regression tree (CART), but with multiple linear regression models as its leaves. However, in linear regression, the residuals are assumed to be distributed normally, which in turn means that the accidents duration is assumed to follow a normal distribution. Nevertheless, the distribution for the time to an event is almost certainly nonsymmetrical, and therefore hazard-based duration models (HBDM) may be a good choice for this kind of time-to-event modeling situation.

Therefore, in this last research subtopic, an M5P-HBDM algorithm is proposed to predict the traffic accident duration. According to the proposed algorithm, the leaves of the M5P tree model are replaced with HBDMs instead of linear regression models. The new M5P-HBDM model allows for classifying the dataset and for decreasing data heterogeneity. Moreover, the proposed model avoids the need to assume a normal distribution of traffic accident duration. The results show that for I-190 and I-64 traffic accident datasets, M5P-HBDMs can find more significant and meaningful variables, compared to eitherthe stand-alone M5P or the HBDM algorithm,. For the duration prediction performance, M5P-HBDMs also have the lowest overall mean absolute percentage error (MAPE).

1.5 Research Contributions

The most important contribution of this dissertation is in proposing an integrative data science approach and applying it to ITS data analysis. The integrative approach provides very promising insights into the data and results in the development of a few novel models for border crossing delay prediction and traffic accident data analysis. This section first discusses the dissertation contributions from the standpoint of the ITS data analysis process (e.g., the integration of data "width" reduction methods and model development for real-time traffic accident risk prediction). Secondly, it describes the dissertation's research contributions to ITS data analysis models (e.g., the integration of M5P and HBDM for traffic accident duration prediction). Lastly, this section summarizes the novel aspects of the research performed. Table 1-1 gives a summary of the research contributions in this dissertation.

Table 1-1 Research Contributions Summary

Two Aspects	Research Contributions	Relevant Research Topics	Novelty
ITS Data Analysis Process	Integration of Data "Width" reduction methods (variable selection based on frequent pattern tree) and Model Development	Real-time Traffic Accident Prediction	Yes
	Integration of Data "Depth" reduction methods (Dataset Grouping) and Model Development	Short-term Traffic Volume Prediction	Similar to previous research
	Integration of Data "Depth" reduction methods (modularity-optimizing community detection algorithm) and Model Development	Traffic Accident Hotspots and Clearance Time Analysis	Yes
	Integration of Data Diagnosis and Short-term Traffic Volume Model Development	Short-term Traffic Volume Prediction	Yes
	Model Combination of SARIMA and SVR using fuzzy adaptive variable weight method	Short-term Traffic Volume Prediction	Similar to previous research
	Model Application - an Android Smartphone Application for Niagara Frontier Border Crossing	Border Crossing Delay Prediction	Yes
ITS Data Analysis Models	Integration of Dynamic Time Warping (DTW) and Spinning Network (SPN)	Short-term Traffic Volume Prediction	Yes
	Integration of Data-driven Models (short-term traffic volume prediction models) and Analytical Methods ($M/E_K/n$ and BMAP/PH/n queueing models)	Border Crossing Delay Prediction	Yes
	Integration of M5P tree and Hazard-based Duration Model (HBDM)	Traffic Accident Duration Prediction	Yes

1.5.1 Dissertation Contributions to the ITS Data Analysis Process

In this dissertation, from the standpoint of the ITS data analysis process, the dissertation integrates data “width” and “data depth” reduction methods with model development. The dissertation also proposes the use of data diagnosis metrics to guide model selection and development, as well as the utilization of model combinations to improve prediction accuracy. Finally, as a demonstration, an android smartphone application that can collect, share and predict the Niagara Frontier border crossing waiting time is developed.

1.5.1.1 Integration of Data “Width” Reduction Methods and Model Development

For real-time traffic accident risk prediction, the research proposes a new FP tree based variable selection algorithm through the calculation of a new variable importance metric the Relative Object Purity Ratio (ROPR). The results show that the models based on variables selected by this new algorithm perform better than the models based on variables selected by random forests. *The relevant research has been documented and is to be submitted to Transportation Research Part C: Emerging Technologies in 2015.*

1.5.1.2 Integration of Data “Depth” Reduction Methods and Model Development

For border-crossing short-term traffic volume prediction, data grouping is applied to group the dataset into weekdays (Monday - Thursday), Fridays, Saturdays, Sundays, holidays and game days. The accuracy of border crossing short-term volume forecasts is shown to improve by classifying the volume data into groups and developing separate prediction model for each group. *The relevant work was published in the Proceedings of the Transportation Research Board 91st Annual Meeting (No. 12-3398).*

For traffic accident hotspots and duration analysis, data clustering is applied to decrease the heterogeneity of the accident dataset before building models for each cluster. An algorithm borrowed from social network analysis, namely a *modularity-optimizing community detection* algorithm is found to be quite appropriate to cluster the traffic accident data. This part of the dissertation representsthe first time that algorithm is applied to ITS data analysis. *The relevant research was published in the Journal of the Transportation Research Record, 2014.*

1.5.1.3 Integration of Data Diagnosis and Short-term Traffic Volume Model Development

Given a short-term traffic volume prediction dataset, it’s an extremely time consuming task to try all the possible prediction models and find the best one. Besides that, the best model may not be the best for other short-term traffic volume prediction datasets, and the whole procedure may have to be repeated again. This research analyzed the complexity and non-linearity characteristics of four different datasets: (1) an hourly traffic volume dataset for an international border crossing; (2) an hourly traffic volume dataset for I-90 in Buffalo, New York; (3) a 5-minute volume dataset from the Commonwealth of Virginia; and (4) a 2-minute volume dataset from Beijing, China through the use of appropriate pre-diagnosis procedures and statistical measures. Followig this, the data diagnosis results were correlated to the performances of three different model types, SARIMA, k-NN and SVR. This exploratory study provided insights into the process of selecting the best prediction models, as well as into how to select appropriate values for the different parameters of the short-term traffic prediction models. The integration of data diagnosis and short-term traffic volume model development is promising. *The relevant research was published in the Journal of the Transportation Research Record, 2013*.

1.5.1.4 Model Combination of SARIMA and SVR for Short-term Traffic Volume Prediction

The author applies the fuzzy adaptive variable weight method, based on the Fresh Degree Function, to combine the prediction results of SARIMA and SVR models for border crossing short-term traffic volume prediction. The results show that the model combination outperforms both SARIMA and SVR. *The relevant result was published in the Proceedings of the Transportation Research Board 91st Annual Meeting (No. 12-3398).*

1.5.1.5 Model Application - an Android Smartphone Application for Niagara Frontier Border Crossing

An Android smartphone application called the **T**oronto **B**uffalo **B**order **W**ait **T**ime (TBBW) is designed to collect, share and predict waiting time at the three Niagara Frontier border crossings. The innovative app offers the user three types of waiting times: (1) The app applies the *two-step border crossing delay prediction model* in practice to provide the future waiting time for the next 15 minutes and updates every five minutes; (2) the app can provide current waiting times based on *data-level fusion*, the data collected by border crossing authorities and the user-reported or "crowd-sourcing" data shared by the app's users; (3) the app can provide historical waiting times in the forms of statistical charts and tables to help users choose the crossing with the likely shortest wait time. *The relevant research has been accepted for presentation at the Transportation Research Board meeting in 2015.*

1.5.2 The Dissertation Contributions to ITS Data Models

In this dissertation, different kinds of statistical models, machine learning and data mining models, and probabilistic queueing models are integrated together to yield new models for accomplishing a specific task. The new integrative models developed in this dissertation are briefly introduced below.

1.5.2.1 Integration of Dynamic Time Warping (DTW) and Spinning Network (SPN) for Short-term Traffic Volume Prediction

This integration resulted in the development of an enhanced spinning network model called DTW-SPN, which uses Dynamic Time Warping (DTW) to evaluate the similarity between two time series instead of the Euclidean distance. The DTW-SPN showed superior performance compared to the original SPN, ARIMA and SVR models, and had a much lower computational running time. *The DTW-SPN research was published in Transportation Research Part C: Emerging Technologies in 2013.*

1.5.2.2 Integration of Data-driven Models and Analytical Methods for Border Delay Prediction

This integration resulted in a two-step border crossing delay prediction model, where a data-driven short-term traffic volume prediction model is utilized first, followed by an analytical multi-server queueing model in the second step. . For the multi-server queueing models utilized in the second step of the border delay prediction model, the original heuristic solution for $M/E_K/n$ queueing model proposed by Escobar et al. (2002) is improved to decrease the running speed for the on-line border crossing delay prediction requirement. Besides that, a new heuristic solution based on a new assumption called the Equally Likely Vehicles (ELV) is proposed to get the transient solution of *BMAP/PH/n* queueing model. The relevant research was published in *Transportation Research Part A: Policy and Practice in 2014*.

The structure of this two-step model allows one to consider multiple factors like the traffic volume, the weather, the accidents, the number of open lanes, and the service time in predicting border delay. The two-step model can also be incorporated within an optimization framework and used to derive optimal management strategies for a customs and immigration border control agency. While the two-step model was developed for the Niagara Frontier border crossing, it can be applied to other border crossings and toll stations.

1.5.2.3 Integration of M5P tree and HBDM for Traffic Accident Duration Prediction

This integration yields a new traffic accident duration prediction model called M5P-HBDM, which represents an integration of a machine learning algorithm (i.e., M5P tree) and a statistical model (i.e., HBDM). As mentioned in section 1.4.2.5, through M5P-HBDM, the leaves of the M5P tree model are replaced with HBDMs instead of linear regression models. This new M5P-HBDM showed promising performance. *The M5P-HBDM model research has been documented and is to be submitted to Accident Prevention and Analysis in 2015.*

1.5.3 Novelty Discussion

The integrative approach adopted in this dissertation exhibits several novel aspects, which to the best of the author's knowledge, have not been researched before. Those novel aspects can be summarized as follows:

(1) the introduction and design of a data diagnosis step that can be used to guide model selection for short-term traffic volume prediction models, as well as the setting of the parameters of those models;

(2) the integration of dynamic time warping and the spinning network (SPN) forecasting paradigm, resulting in an enhanced SPN called DTW-SPN;

(3) the integration of data-driven models and analytical methods to yield a two-step model for predicting future border crossing delay;

(4) the integration of the machine learning model, M5P tree, and the HDM statistical model for developing accident duration prediction models;

(5) proposing a FP tree based variable selection algorithm for data "width" reduction before developing models for real-time accident risk prediction;

(6) the application of a modularity-optimizing community detection algorithm for data "depth" reduction;

(7) the development of heuristic solutions for $M/E_K/n$ and *BMAP/PH/n* queueing models;

(8) the development of an Android smartphone application designed to collect, share and predict waiting time at the three Niagara Frontier border crossings.

1.6 Dissertation Organization

The current dissertation is organized in nine Chapters. Following this introductory chapter, Chapter 2 reviews the literature pertinent to the five research subtopics considered in this study. Chapters 3 to Chapter 5 then describe the author's effort to construct the two-step border crossing delay prediction model, which can be viewed as an integration of a data-driven model and an analytical model. Specifically, Chapter 3 describes the integration of data "depth" reduction methods (data grouping) and model development for border crossing traffic prediction, and the combining of statistical and AI models (namely SARIMA and SVR) using fuzzy adaptive variable weight method. Chapter 3 also describes the development of the enhanced SPN through the integration of DTW and the original SPN model, and proposes a data diagnosis step which can be used to guide model selection and parameter setting. Chapter 4 presents the heuristic solutions for two types of queueing models $M/E_K/n$ and *BMAP/PH/n*, which take the predicted traffic volume as the input to calculate the future waiting time. The verification of the queueing models using traffic simulation and sensitivity analysis can also be found in that Chapter. Chapter 5 introduces an Android smartphone application called the Toronto Buffalo Border Wait Time (TBBW) that was developed to collect, share and predict waiting time at the three Niagara Frontier border crossings. Next, Chapters 6 through 8 introduce the novel methods and models proposed for traffic accident data analysis. Specifically, chapter 6 focuses on hot spot identification and clearance time analysis using the modularity-optimizing community detection algorithm and the association rule learning algorithm; this work can be viewed as an integration of data "depth" reduction methods and model development. Chapter 7 proposes the integration of data "width" reduction methods (i.e., the novel FP tree based variable selection method) and model development for real-time traffic accident risk prediction. Chapter 8 introduces the integration of the data mining model M5P tree and the statistical model HBDM. The new M5P-HBDM models are then utilized to predict the traffic accident duration for two traffic accident datasets, and their performance are compared to those of stand-alone M5P trees and HBDMs. Finally, Chapter 9 summarizes the research contributions and gives some directions for future research.

CHAPTER 2 LITERATURE REVIEW

This Chapter will provide a brief review of the literature pertaining to the five subtopics mentioned in Chapter 1.4.2. Chapter 2.1 presents the historical studies related to each step in the two-step border crossing delay prediction model separately, namely the short-term traffic volume prediction model and the queueing model. After that, the other relevant studies about the border crossing delay problem are also introduced. Chapter 2.2 focuses on discussing the studies about the traffic accident data analysis such as hotspots analysis, real-time traffic accident risk prediction and traffic accident duration analysis. It's worth noting that the review will not follow the ITS data analysis process introduced in Chapter 1.3.1 strictly, only the researches relevant to the dissertation work are summarized.

2.1 Border Crossing Delay Prediction

This chapter reviews the previous studies that are relevant to the two steps in the border crossing delay prediction (i.e., the short-term traffic volume prediction and the queueing model) . It also reviews some other researches about the border crossing delay problem in general.

2.1.1 Short-term Traffic Volume Prediction

As opposed to long-term volume forecasting which provides traffic volume forecasts for relatively long prediction horizons (reaching up to 20 years), short-term traffic forecasting uses real-time traffic information from roadway sensors to predict traffic flow for much shorter prediction periods (ranging between 5 minutes to one hour). Short-term forecasting thus provides the functionality needed for on-line transportation system management and control. This Chapter summarizes the literature in short-term traffic volume prediction in terms of dataset grouping and clustering, data diagnosis, model development and model combination.

2.1.1.1 Data "Depth" Decreasing

There are different ways to group or cluster short-term traffic volume observations. The most common one is based on the characteristics such as time of day, day of the week, special events (e.g., holidays and sports games), inclement weather, and so on. Wild (1997) split a daily traffic volume time series data into several groups based on different combinations of day of week and the events like sports games, fairs and so on. Chrobok et al. (2004) defined four distinct data classes based on the combinations of days of the week and holidays. Stathopoulos and Karlaftis (2003) excluded the Saturday and Sunday traffic data, and divided the traffic flow data on weekdays into six separate groups by time period.

As for data clustering techniques, Van Der Voort et al. (1996) developed a method, called the Kohonen self-organizing map, to cluster traffic data, and then tuned a ARIMA model for each cluster. Park (2002) applied the fuzzy C-means (FCM) method to classify traffic flow patterns into several clusters before develoing a radial-basis-function (RBF) neural network model for each cluster. Yin et al. (2002)

developed a fuzzy-neural model that consists of two modules: the first module classified the input data into a number of clusters using a fuzzy approach, and the second module built a neural network model to capture the pattern within each cluster. Similarly, Srinivasan et al. (2009) also applied a fuzzy input fuzzy output filter to classify the input data into a number of clusters and built a multi-layer feed-forward neural network with evolution strategies for each cluster.

2.1.1.2 Data Diagnosis

This separate Chapter discusses one issue about the short term traffic volume prediction studies, which is lacking data diagnosis to connect the short-term traffic volume datasets with the selection of the prediction models. While there is an extensive literature on short-term traffic volume prediction, most of the previous studies considered only one modeling technique and a single data set. Even among the comparative studies in the literature, the focus has typically been on comparing the performance of multiple models on a single data set (William et al., 2006; Sun et al., 2007). The risk of using one data set to test multiple models is that the conclusions derived may be specific to the dataset considered. This has often led to inconsistent conclusions among the different studies regarding which modeling method is superior. In addition, single data based testing cannot address the essential questions that are of particular interest to practitioners, i.e., how to select prediction models based on the characteristics of a specific dataset.

A handful of researchers have begun to pay more attention to this issue. For example, Smith and Demetsky (1997) tested four prediction methods on two data sets. However, the two data sets were collected from the sites on the same highway, and there was no discussion in that study about the relationship between the attributes of a dataset and model performance. Other researchers pointed out the importance of data diagnosis before model selection, and proposed different measures to indicate data characteristics. For instance, Vlahogianni et al. (2006) discussed some statistical methods for detecting nonlinearity and non-stationarity of traffic volume time series, and Shang (2005) discussed the nonlinearity property of traffic volumes based on Chaos theory. However, no effort was made in those studies either to link data diagnosis results with model selection.

2.1.1.3 Model Development

For model development, this Chapter first introduces two groups of algorithms or approaches that have attracted great attentions: (1) statistical models such as time-series analysis; and (2) machine learning based methods such as Neural Networks (NN) and Support Vector Regression (SVR), among the numerous methods recently proposed for short-term traffic volume prediction. After that, this Chapter discusses the model combination application under this subtopic.

2.1.1.3.1 Statistical Models

With respect to the first group, the Box and Jenkins techniques (e.g., Autoregressive Integrated Moving Average (ARIMA) models) were firstly applied to the field of traffic forecasting by Ahmed and Cook (1979). Since then more and more advanced techniques from that family have been applied to traffic volume prediction, such as the seasonal ARIMA models (SARIMA) (Williams and Hoel, 2003; Smith et al., 2002), the ARIMA models with intervention x-variables (ARIMAX) (Williams, 2001; Cools et al., 2009), and the combination of Kohonen self-organizing map with ARIMA models (Van Der Voort et al., 1996).

In addition to the Box and Jenkins models, other multivariate time series techniques were exploited to increase prediction accuracy, including the state space model (Stathopoulos and Karlaftis, 2003) and the multivariate structural time series model (MST) (Ghosh et al., 2009).

Kalman filtering theory was also utilized for short-term traffic forecasting. Some examples are the initial examination by Okutani and Stephanedes (1984), the state space based method by Stathopoulos and Karlaftis (2003), and the work by Xie et al. (2007) who used a Kalman filter with discrete wavelet decomposition for short term traffic prediction.

Most recently, Min and Wynter (2011) adopted a multivariate spatial-temporal autoregressive model (MSTAR) to predict network-wide speeds and volumes in real time.

2.1.1.3.2 Machine Learning and Data Mining

On the AI side, among the most widely used methods are Neural Networks (NNs). Several NN topologies have been utilized in previous studies including the multilayer perceptron networks (MLP) (Smith and Demetsky, 1994; Chang and Su, 1995; Ledoux, 1997), radial basis function networks (Park et al., 1998), resource allocating networks

(Chen and Grant-Muller, 2001), and wavelet networks (Chen et al., 2006; Xie and Zhang, 2006).

NNs were also sometimes combined with other methods, such as fuzzy logic, genetic algorithms and empirical mode decomposition, to develop hybrid and more powerful predictions methods (e.g., Yin et al., 2002; Vlahogianni et al., 2005, and Wei and Chen, 2012).

Besides NNs, other AI methods were recently proposed for short-term traffic prediction. Dimitriou et al. (2008), for example, proposed an adaptive hybrid fuzzy rule-based system approach to predict traffic flow in urban arterial networks.

SVR has also recently been exploited for short-term traffic flow prediction. Specifically, Zhang and Xie (2008) compared a v-support vector machine (v-SVM) model with a NN model and concluded that the former performed better. Other examples of applying SVR to traffic prediction include Castro-Neto et al. (2009) and Hong et al. (2011).

2.1.1.3.3 Model Combination

Model combination has been applied in many short-term traffic volume prediction studies. Fuzzy logic has been used for model combination in short-term traffic volume prediction. Stathopoulos et al. (2008) employed a fuzzy rule based system to nonlinearly combine traffic flow forecasting results from a Kalman filter (KF) and a neural network model (ANN). Zhang and Ye (2008) also proposed a fuzzy logic system to combine multiple models like KF, exponential smoothing method (ESM), back propagation neural networks (BPNNs), and ARIMA model.

Besides the fuzzy logic based model combination, neural network (NN) is also employed to combine models. Zheng et al. (2006) applied a Bayesian combined neural network model to combine the traffic prediction results from two other NNs: BPNN and the radial basis function neural networks. Tan et al. (2009) used a BPNN model to aggregate the traffic flow prediction results based on moving average model (MA), exponential smoothing model (ES), and ARIMA model.

More recently, Dong et al. (2014) applied the support vector regression (SVR) to combine the statistical model ARIMA with the Elman neural network model to handle the linear and nonlinear patterns in traffic time series.

2.1.2 Queueing Model

This section discusses the literature relevant to the queueing model that is the second step of the border crossing delay prediction model. Queueing models have been used extensively to solve problems related to manufacturing processes, transportation systems, product distribution systems, call centers, among other applications (Gontijo et al, 2011). Queueing models can be categorized in a number of different ways. One categorization divides them into stationary queueing models and transient queueing models as explained below.

Let $N(t)$ denote the number of customers (i.e. vehicles) in the queueing system at time t measured from a fixed initial time moment $t = 0$, and let $p_n(t)$ denote the probability that $N(t) = n$ at time *t*. Because it is usually difficult to find the time-dependent solution $p_n(t)$ *analytically*, many applications in practice resort to consider only the steady state behavior of the queueing system after being in operation for a sufficiently long time. In that case, one is interested in the limiting behavior of $p_n = lim_{t \to \infty} p_n(t), n = 0,1,2, \dots$ (Medhi, 2003). These queueing models that study the limiting probability of $p_n(t)$ are called stationary queueing models.

Stationary queueing models are usually used to derive some useful performance measures such as the average waiting time, which in turn is often adopted as an objective function within an optimization framework to improve the efficiency of a system. For example, the study by Kim (2009) built a non-linear integer programming model to study the toll plaza optimization problem. In his model, the cost of the waiting time of the vehicles, as determined from the steady state solution of the queueing model, was minimized. Another example is the study by Ausín et al (2007), which minimizes the steady-state expected total waiting time by optimizing the number of servers based on the real data from a bank. Moreover, Zhang et al. (2011) proposed a two-stage queueing model to balance security and customer service goals for a border crossing system. Besides waiting time, some researchers tried to estimate additional measures of queueing systems, such as traffic intensity, based on stationary queueing models. Ke and Chu (2006), for example, proposed a consistent and asymptotically normal estimator of traffic intensity for a queueing system with distribution-free inter-arrival and service times. They also developed the confidence intervals for testing statistical hypothesis of intensity and derived the associated power function. Ke and

Chu (2009) constructed and compared new confidence intervals of traffic intensity for a queueing system based on different bootstrap methods, and introduced a new measure called relative coverage to assess the performances of the confidence intervals.

On the other hand, queueing models that consider time-dependent state probabilities are called transient queueing models. For systems that exhibit strong dynamic conditions, the transient solution of a queueing model is more meaningful. This is definitely the case for the border crossing problem studied in the dissertation, where the arrival and service rates exhibit strong dynamic patterns. Hence, the focus of this research is on the *transient* solution of a queueing model. An example of studies that considered transient queueing models is the study by Gupta (2011) which estimated air traffic delays using a transient $D(t)/M(t)/1$ queueing model. Moreover, the study by Escobar et al (2002) provided some preliminary results regarding the approximate transient solution for multi-server queueing systems with Erlangian service times, based on the equally likely combination (ELC) heuristic.

Besides the study by Escobar et al. (2002), Ausín et al (2008) used Bayesian inference to derive the transient behavior and the durations of the busy periods for a $GI/G/1$ queueing model, while Czachorski et al (2009) studied the transient behavior of multi-servers with general service time and inter-arrival time distributions in the context of a call center. In the aforementioned studies, approximate solutions were proposed since exact solutions for transient queueing models with general distributions for arrival and service processes are notoriously hard to derive. Similar to the stationary models, transient queueing models can also be used within an optimization framework to identify optimal system operating policies. For example, Parlar et al. (2008) derived the time-dependent operating characteristics of a queueing process that represents the operation of check-in counters at an airport. They formulated a stochastic dynamic programming model to determine the optimal numbers of the check-in counters.

2.1.3 Other Border Crossing Delay Studies

While an extensive literature on the short-term traffic forecasting problem and queueing models exist as surveyed above, only a handful of previous studies could be found for border crossings. Moreover, most of those previous studies focused on modeling border delays for off-line planning applications, and few for on-line prediction which is the focus of this research. Examples include a study by Paselk and Mannering (1994) that used duration models and a study by Lin and Lin (2001) that proposed a delay model for planning applications. Kam et al. (2005) described the development of a NN for predicting border crossing delays. However, the data used to develop the model came solely from a simulation model, and not from real-world observations as is the case in this dissertation. More recently, Haughton and Sapna Isotupa (2012) also applied the computer simulation to quantify the impacts of smoothing the commercial vehicle flows at a major Canada-US border crossing. Similarly, the inter-arrival and service time distributions are assumed but not based on the real-world observations.

2.2 Traffic Accident Data Analysis

This section first introduces the studies about data “width” and “depth” decreasing in traffic accident data analysis. Following this, three important subjects of traffic accident data analysis are reviewed, including hotspots analysis, real-time accident risk prediction and traffic accident duration prediction.

2.2.1 Data “Width” Decreasing

Accidents and crashes on road are highly complex phenomena that can be attributed to a wide range of variables, and in many occasions the quality as well as availability of detector data need to be compensated with surrogate variables. For example, the wealth of real-time traffic data offers more explanatory variables that may contribute to explaining traffic accident risk and patterns. This induces the classical situation involving large variable space and small sample size, which requires a suitable method to select the most important variables for traffic accident data analysis (Hossain and Muromachi, 2012).

The data “width” decreasing-variable selection problem has attracted attention from previous traffic accident data analysis research. As for the statistical model based research, Sawalha and Sayed (2006) found that using less but statistically significant explanatory variables can avoid over fitting and improve the reliability of a model. They suggested combining the t-statistics test and the likelihood ratio based scaled deviance test, for selecting significant explanatory variables. Different procedures were suggested for Poisson regression and negative binomial regression respectively due to the additional complexity introduced to the scaled deviance test for negative binomial regression models. As for the data mining models, classification and regression tree (CART) has been used to perform variable selection (Yu and Abdel-Aty, 2013; Pande and Abdel-Aty, 2006). Another ensemble learning method for classification and regression, called random forests, has also been widely used to rank explanatory variables (Abdel-Aty et al., 2008; Ahmed and Abdel-Aty, 2012). Recently, a hybrid model random multinomial logit (RMNL), formed by combining the random forests and logit models, was applied to calculate traffic accidents variable importance (Hossain and Muromachi, 2012).

2.2.2 Data "Depth" Decreasing

Several researchers have recently pointed out that heterogeneity inherent in traffic accident data often prevents the further exploration of these data (Savolainen et al., 2011). To deal with the issue, random effects and random parameter models have been proposed for traffic accident data analysis (Karlaftis and Tarko, 1998; Miaou et al., 2003). Such models capture the unobserved heterogeneity by using random error terms and allow each estimated parameter of a model to vary across each individual observation in the dataset (Lord and Mannering, 2010). Anastasopoulos and Mannering (2009), for example, demonstrated that random parameters model can account for the heterogeneity arising from a number of factors in accident records and other unobserved factors in their accident frequency study. However, random effects model and random parameters model may not be easily transferable, and are often difficult to estimate (Lord and Mannering, 2010).

Data "depth" decreasing based data clustering or grouping is another way to minimize the heterogeneity problem. For example, Valent et al. (2002) found that "Sundays" and "holidays" arise as significant risk factors when the analysis was performed for clustered data. Moreover, Mohamed et al. (2013) identified "bad visibility due to bad weather" as a factor that can increase the risk of fatal crashes in Montreal Canada, based on an analysis performed on a clustered dataset.

In traffic accidents studies, the two most widely used clustering techniques are the latent class clustering (LCC) and the K-means clustering method. On one hand, LCC has the advantages of being able to provide statistical criteria for deciding the number of clusters, and to calculate the probabilities for the new data points to belong to a given cluster (Depaire et al., 2008; de Oña et al., 2013). On the other hand, LCC heavily relies on the assumption of local independence among traffic accident variables to reduce parametric complexity and computing time, and was found to sometimes reach the local optimum rather than the global optimum. As for the K-means clustering, Sohn and Lee (2003) used it to cluster a traffic accident severity dataset before building a classifier for each cluster. Anderson (2009) applied the method to classify accident hotspots into relatively homogenous types based on their environmental characteristics. In addition, Mohamed et al. (2013) reported that for the Montreal accident dataset the K-means clustering method appeared to do a better job compared to LCC that tended to classify 90% of the accidents into the first two clusters.

Another widely applied way to do the data “depth” decreasing is based on traffic accident types (Golob et al., 1987; Ozbay and Kachroo, 1999; Giuliano, 1989). Recently some researchers classify the traffic crash data based on the visibility conditions like daylight, twilight and night conditions (Hong et al., 2014).

2.2.3 Model Development

The model development of traffic accident data, however, has for long been dominated by traditional statistical analysis methods that have yielded invaluable insight and helped guide policy over years. Recently, the techniques from machine learning and data mining have also been applied in traffic accident model development. This section summarizes the literature about hotspots analysis, real-time traffic accident risk prediction and traffic accident duration prediction. For each of them, the methodologies are divided into two groups: statistical models and machine learning and data mining.

2.2.3.1 Hotspots Analysis

As the first step of the highway safety management process, the identification of crash hotspots is very important. However, there is no universally accepted definition of a road accident ‘hotspot’ in the road accident literature (Anderson, 2009). Some studies defined hotspots (or black spots) as geographical locations with highly concentrated traffic accidents (Geurts et al., 2003; Xie and Yan, 2008), while some others detected hotspots based on quantitative measures such as the number of accidents divided by the traffic flow rate per period of time (Gregoriades and Mouskos, 2013). The identification and profile of hotspots can provide valuable insights for transportation authority actions.

2.2.3.1.1 Statistical Models

Among the statistical models identifying hotspots, Kernel Density Estimation has gained more and more popularity (KDE), especially in conjunction with Geographic Information Systems (GIS). KDE can calculate the density within an area with an user defined radius and an object location as the center by choosing a kernel function through which "distance decay effect" can be taken into account, the longer the distance between a point and the object location, the less that point is weighted for calculating the overall density (Xie and Yan, 2008). Xie and Yan (2008) proposed the network kernel density estimation to identify the hotspots by calculating the density over a linear unit instead of calculating the density over an area unit because of the linear nature of the road network spaces. Anderson (2009) applied a k-means clustering algorithm after creating the KDE map and found similar hotspots clusters based on collision and attribute data. Bíl et al. (2013) extended the standard KDE by combining it with the statistical testing of the cluster significance, and they found this modified KDE can both identify the clusters of traffic accidents and to determine which of them were significant.

Besides KDE, Montella (2010) summarized and compared seven commonly applied hotspot identification methods like crash frequency, equivalent property damage only crash frequency, crash rate, proportion method, empirical Bayes estimate of total-crash frequency, empirical Bayes estimate of severe-crash frequency, and potential for improvement. The quantitative evaluation tests showed that empirical Bayes estimate of total-crash frequency performs the best among the seven methods.

2.2.3.1.2 Machine Learning and Data Mining

Among data mining methods proposed for hotspots analysis, the association rules algorithm was used to identify accident circumstances that frequently occur together at high frequency accident locations and compared these patterns with those at low frequency accident locations for the road network in Belgium (Geurts et al., 2003; Geurts et al., 2005). Besides that, the Bayesian networks model was built based on an enriched accidents dataset and was used to predict the number of accidents under different scenarios that describe traffic condition at different time intervals and driver profiles to identify accident black spots on road networks (Gregoriades and Mouskos, 2013). Another machine learning and data mining algorithm used in hotspots analysis is that Nayak et al. (2011) used decision tree to find the best dataset portioning between crash prone and non-crash prone roads, they found the best road segment crash-proneness threshold was four to eight crashes in a four year period.

2.2.3.2 Real-time Traffic Accident Risk Prediction

As a key role in Active Traffic Management (ATM), real-time traffic accident risk prediction models have drawn more and more attentions. The real-time traffic accident risk prediction models can be integrated with the control strategies such as variable speed limits (VSL) to reduce the crash risk once a certain threshold of crash risk has been reached (Yu and Abdel-Aty, 2013).

2.2.3.2.1 Statistical Models

Lee et al. (2003) proposed an aggregate log linear model to predict the real time traffic accident for a freeway in Toronto, Canada, the results showed the speed difference between upstream and downstream detectors has a significant impact on the model performance. Abdel-Aty et al. (2004) applied the matched case-control logistic regression models to predict the freeway traffic accidents based upon loop detector data from Orlando, US. It was found that the coefficient of variation of speeds at the downstream station and the average occupancy of upstream station are significant in the final models. More recently, Xu et al. (2013) used a sequential logit model to predict the traffic accident risk based on the loop detector data from San Francisco Bay area, US. They found that traffic accident likelihood is high when the traffic density from the upstream, the speed variance from the upstream and/or downstream, volume difference between upstream and downstream station and the occupancy difference between upstream and downstream station are high.

2.2.3.2.2 Machine Learning and Data Mining

Typical examples of the data mining/ machine learning modeling approaches are summarized as below. Pande and Abdel-Aty et al. (2006) utilized neural networks to analyze and classify the crash and non-crash cases. The significant variables were chosen by a classification tree technique, and the results showed that the false alarm rates of this model were too high. Hossain and Muromachi (2012) applied the Bayesian Network model to predict the real-time traffic accident risk. They showed that Bayesian network has a very promising performance with an accuracy of 66% for the future crashes and a false alarm rate less than 20%. Yu and Abdel-Aty (2013) applied the support vector machine to a traffic accident dataset collected from Colorado, US. They showed the support vector machine performs better than the classic logistic regression models for real-time crash risk evaluation method.

2.2.3.3 Traffic Accident Duration Prediction

As mentioned previously, traffic incidents account for more than 50% of motorist delays on freeways (Chin et al., 2004; Farradyne, 2000). Different with the recurring congestion, for which travelers can plan their trips according to the expected occurrence and severity of recurring congestion, the nonrecurring traffic congestion introduced by incidents cannot be managed without real-time prediction (Garib et al., 1997). Therefore an efficient traffic incident management system (TIM) is needed to predict the durations of traffic incidents under various conditions (different local and regional traffic conditions, time of day, day of week, seasonal variations, weather conditions, work zones, etc…). Based on this predicted duration information, the authorities can then allocate incident response personnel and resources more effectively, and inform the travelers about the incidents more accurately.

The time sequential process of TIM can be divided into five phases (Zhan et al., 2011), including: (1) the incident detection phase representing the time interval from the occurrence of an incident to its detection; (2) incident verification phase as the time interval from the detection to the confirmation of the incident; (3) the incident response phase as the time interval between the confirmation time and the time when the first responder arrives on the scene; (4) the incident clearance phase representing the time interval from the arrival of the first responder to the time when the incident has been cleared from the freeway; and (5) incident recovery phase as the time it takes for the normal traffic condition to return after the incident clearance phase.

In the previous studies, there exist different understandings about the terms such as incidents versus accidents and duration versus clearance time. Usually incidents include the accidents. Besides that, the events like the vehicle breakdowns, spilled loads or other random events should also be called incidents but not accidents (He et al., 2013). For duration and clearance time, duration includes the whole process of five phases defined above. Clearance time is just incident clearance phase (Zhan et al., 2011; Alkaabi et al., 2011). Here we clarify that this dissertation only focuses on accidents but not incidents, and defines the duration as a time interval from the time of the accident detection to the time when the normal traffic condition returns.

2.2.3.3.1 Statistical Models

In the early stage of statistical models for traffic accident duration analysis, log-normal and normal distributions were applied a lot. Golob et al. (1987) analyzed the truck-involved incident duration in California. They found that the durations of the incidents categorized by type of collisions follow a log-normal distribution. Ozbay and Kachroo (1999) found a normal distribution of durations for homogeneous incidents grouped by incident type and severity.

Linear and polynomial regressions were used to identify the factors that may influence the traffic accidents duration. Giuliano (1989) assigned incidents into multiple categories and estimated incident durations using linear regression techniques for each category. Garib et al. (1997) developed a polynomial regression model to predict the incident durations. The results showed that in terms of the adjusted R-square values, 81% of the incident duration can be predicted as a function of six independent variables.

Another main group of statistical models for analyzing traffic accident durations is the hazard-based duration model. Nam and Mannering (2000) applied hazard-based duration models to evaluate incident durations. They mentioned that in comparison with regression approaches, hazard-based duration models have an advantage to allow the explicit study of duration effects. Recently, Alkaabi et al. (2011) and Chung (2010) also developed hazard-based duration models to analyze and predict traffic accident duration.

2.2.3.3.2 Machine Learning and Data Mining

Many machine learning and data mining algorithms have been applied in traffic accident duration analysis and prediction. There are quite a few studies that employed decision tree (DT) to predict the incident duration (He et al., 2013; Zhan et al., 2011; Smith and Smith, 2001; Ozbay and Kachroo, 1999). The main advantage of decision tree is that no probable distribution assumption is required for this model, which can find patterns in a given data set (Alkaa-bi et al., 2011).

However, Ozbay and Noyan (2006) pointed out that the decision trees can sometimes be unstable and insensitive to the stochastic nature of data. To remedy the situation, they applied another machine learning model, Bayesian networks (BNs), with the additional capability to estimate the duration in the presence of a real incidents for which data might only be partially available.

Wei and Lee (2007) proposed a sequential method of accident duration prediction by using two Artificial Neural Networks (ANN) methods. One of them can predict the duration once an incident is verified, and the other can provide multiple updates of the duration time after the incident verification. In the follow-up research, they applied genetic algorithm (GA) for feature selection to decrease the number of model inputs for the two ANNs (Lee and Wei, 2010).

Recently, Valenti et al. (2010) compared five incident duration models, namely multiple linear regression (MLR), DT, ANN, support vector machine (SVM), and k nearest neighbor (k-NN). The results showed that MLR performed the best for incidents with durations less than 30 minutes, while the SVM and ANN perform much better for the incidents with longer durations.

CHAPTER 3 SHORT-TERM TRAFFIC VOLUME PREDICTION

Chapter 3 to Chapter 5 involves with the two-step border crossing delay prediction model, which is also an integration of data-driven model and analytical model. This Chapter presents the researcher's three pieces of work on short-term traffic volume prediction. Chapter 3.1 introduces the integration of data "depth" decreasing step (dataset grouping) and model development step for border crossing traffic prediction and a model combination of SARIMA and SVR using fuzzy adaptive variable weight method. Chapter 3.2 introduces the integration of dynamic time warping and spinning network for border crossing traffic prediction. Chapter 3.3discusses the integration of the data diagnosis step with short-term traffic volume prediction model, statistical measures are calculated for multiple datasets and the connection with the performances of multiple prediction models. At last, Chapter 3.4 summarizes the conclusions from these three pieces of work.

3.1 Data "Depth" Decreasing and Model Combination for Border Crossing Traffic Prediction

3.1.1 Methodology

3.1.1.1 Dataset Grouping

The border crossing traffic volumes are impacted by the days of the week (weekday vs. weekend) as those at most of the other places. The different thing is the border crossing traffic is nationwide, which makes this system more complicate. The holidays and sport games of the both countries Canada and US may also cause the border crossing traffic have the significant variation. A logical grouping scheme that would identify the number of distinct traffic patterns or types of days (weekdays, weekends, holidays, sport games and so on) is necessary to be built. For each distinct pattern or day type, we would then develop a separate prediction model. Our approach was to try to tie the grouping scheme to easily identifiable properties of a given day in order to facilitate implementation of the prediction process.

To do this, we first considered the group of what may be called "ordinary days", defined as all weekdays excluding Fridays (i.e. Monday through Thursday). We also excluded the days which, we suspected, might need separate groups (i.e., holidays and game days). We then calculated the mean hourly border crossing traffic volume for each hour of an "ordinary day", and defined an interval of ± 15% of the average hourly volume (15% was chosen based on what may be regarded as acceptable prediction accuracy for the models). The traffic patterns of the "special" days (i.e. Fridays, Saturdays, Sundays, holidays and game days) were then compared to the "ordinary days" to determine qualitatively whether they differed enough to warrant having their own groups (i.e., whether they lied within the ± 15% band or not).

3.1.1.2 Combination of SARIMA and SVR

This study proposes a multi-model combined forecasting method, by combining forecasts from SARIMA and SVR. A brief description of SARIMA and SVR is provided below, followed by the methods used to combine the two methods' forecasts.

3.1.1.2.1 SARIMA Model

The SARIMA model is a tool to predict future values of a time series that exhibits seasonal trends. According to Box and Jenkins (2008), a time series $\{\mathrm{Z_t}|\mathrm{t} = 1,2,\dots,\mathrm{k}\}$ is generated by $\mathrm{SARIMA(p,d,q)} \times \mathrm{(P,D,Q)_s}$ if:

$$\Phi_{\mathrm{P}}(\mathrm{B^s})\varphi_p(\mathrm{B})\nabla_{\mathrm{s}}^{\mathrm{D}}\nabla^{\mathrm{d}}\mathrm{Z_t} = \mathit{\Theta}_Q(B^s)\theta_q(B)a_t \tag{3-1}$$

where B is the backshift operator defined by $\mathrm{B^aW_t} = \mathrm{W_{t-a}}$; *p, d, q, P, D, Q* are parameters with integer values and *s* represents the length of seasonal cycles; $\varphi_p(\mathrm{B}) = 1 - \varphi_1 B - \varphi_2\mathrm{B}^2 - \cdots - \varphi_p\mathrm{B}^p$ is the nonseasonal autoregressive operator of p order; $\Phi_{\mathrm{P}}(\mathrm{B^s}) = 1 - \Phi_{1s}B^{1s} - \Phi_{2s}\mathrm{B^{2s}} - \cdots - \Phi_{\mathrm{P}s}\mathrm{B^{Ps}}$ is the seasonal autoregressive operator of P order; $\theta_q(\mathrm{B}) = 1 - \theta_1 B - \theta_2\mathrm{B}^2 - \cdots - \theta_q\mathrm{B}^q$ is the nonseasonal moving average operator of q order; $\mathit{\Theta}_Q(\mathrm{B^s}) = 1 - \mathit{\Theta}_{1s}B^{1s} - \mathit{\Theta}_{2s}\mathrm{B^{2s}} - \cdots - \mathit{\Theta}_{Qs}\mathrm{B^{Qs}}$ is the seasonal moving average operator of Q order, $\nabla_{\mathrm{s}}^{\mathrm{D}} = (1 - \mathrm{B^s})^{\mathrm{D}}$ is the seasonal differencing operator of order D, $\nabla^{\mathrm{d}} = (1 - \mathrm{B})^{\mathrm{d}}$ is the nonseasonal differencing operator of order d; and a_t is the estimated residual at time *t*, which are - assumed to be identically and normally distributed with mean as zero and variance as σ^2, $\{a_t\} \sim WN(0,\sigma^2)$.

3.1.1.2.2 SVR Model

Support Vector Regression (SVR) shares several advantages of the Support Vector Machine (SVM) concept, a popular machine learning method based on statistical learning theory proposed by Vapnik (1995). SVM embodies the structured risk minimization principle and attempts to minimize an upper bound of the generalization error. Initially, SVMs were developed to solve classification problems, but with the introduction of Vapnik's insensitive loss function in 1997, SVM was extended to allow for solving nonlinear regression problems, resulting in the SVR method (Kim, 2003; Pai and Hong, 2005). SVR can be described as follows:

Given a set of data points $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\} \subset \mathcal{X} \times \mathbb{R}$, where $\mathcal{X}$ denotes the space of the input patterns and *m* is the total number of training samples, a linear regression function can be stated as $f(x) = < \omega, x > + b$ with $\omega \in \mathcal{X}, b \in \mathbb{R}$ where <,> denotes the dot product in $\mathcal{X}$, and b is a scalar threshold. Assuming an $\varepsilon -$ insensitive loss function, a function intended to allow for ignoring errors that fall within a certain band or distance from the true value, the ω and b can be obtained by solving Equation (3-2) below.

$$\text{minimize } \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{m}(\xi_i^+ + \xi_i^-) \tag{3-2}$$

$$\text{Subject to } \begin{cases} y_i - < \omega, x_i > - b \leq \varepsilon + \xi_i^+ \\ < \omega, x_i > + b - y_i \leq \varepsilon + \xi_i^-, \\ \xi_i^+, \xi_i^- \geq 0 \end{cases}$$

where,

$\varepsilon\ (\geq 0)$ is insensitive loss function, representing the maximum deviation allowed;

$C\ (> 0)$ is the penalty associated with excess deviation during the training; and

ξ_i^+, ξ_i^- are the slack variables corresponding to the size of the positive and negative excess deviation, respectively.

In the process of solving this optimization problem, SVR achieves nonlinear regression by mapping the training samples into a high dimensional kernel induced feature space, followed by linear regression in that space. Since the kernel mapping is implicit (depends only on the dot product of the input data vectors), it is possible to map the data to a very high dimension, and still keep the computational cost low. A Radial Basis Function (RBF) $K(\mathrm{x_i}, \mathrm{x_j}) = \exp\left(-\gamma|\mathrm{x_i} - \mathrm{x_j}|^2\right)$ is one common kernel function. The parameters of the SVR models, namely the penalty factor C and the gamma in the kernel function, are often optimized using the k-fold cross validation method (Stone, 1974; Chang and Lin, 2001).

3.1.1.2.3 Fuzzy Adaptive Variable Weight Method

This study used the fuzzy adaptive variable weight method based on the Fresh Degree Function to combine the SARIMA and SVR forecasts. The method is adaptive in the sense that the weights assigned to each model are a function of how well that particular model performed on recent forecasts. Furthermore, the use of the Fresh Degree Function (Ma and Liu, 2005), F(t), which is usually a function of the time series index, t (e.g., t, or t^2 or $\sqrt{\mathrm{t}}$), allows one to weigh the performance on the most recent forecast more heavily than prior forecasts. The method proceeds as follows (Tang, 1997).

The prediction error of model j at time index i is first computed according to Equation (3-3):

$$\mathrm{e_j(i) = y(i) - f_j(i)} \qquad (3\text{-}3)$$

where,

$\mathrm{e_j(i)}$ is the prediction error of model j at time i;

$\mathrm{y(i)}$ is the observed value at time i; and

$\mathrm{f_j(i)}$ is the prediction value of model j at time i.

Following this, the method calculates the weighted average absolute prediction error for model j using the Fresh Degree function as shown below.

$$\mathrm{a_j(i)} = \sum_{\mathrm{p=0}}^{\mathrm{q}} \mathrm{F(i-p)}\,\left|\mathrm{e_j(i-p)}\right|\left[\sum_{\mathrm{p=0}}^{\mathrm{q}} \mathrm{F(i-p)}\right]^{-1} \qquad (3\text{-}4)$$

where,

$\mathrm{F(t)}$, the Fresh Degree Function $=$ t or $\mathrm{t^2}$, or $\sqrt{\mathrm{t}}$, $\mathrm{t} = 1,2,\ldots,\mathrm{s}$.

q = the length of the time series (i.e., the number of previous data points) used to calculate the weighted average absolute error, $\mathrm{a_j(i)}$ based on the Fresh Degree Function.

Next the method calculates the sum of the absolute prediction error for model j at time step i ($\mathrm{s_j(i)}$) using Equation (3-5). In doing so, the method typically uses a length from the time series, *l*, which is typically longer than the length q used in Equation (3-4) above.

$$\mathrm{s_j(i)} = \sum_{\mathrm{p=0}}^{\mathrm{l-1}} \left|\mathrm{e_j(i-p)}\right| \qquad (3\text{-}5)$$

The method then calculates the following: (1) $\mathrm{E_j(i)}$, which is the ratio of the weighted average absolute prediction error over the past q data points of prediction method j at time i, $\mathrm{a_j(i)}$, to the largest $\mathrm{a_j(i)}$ of all prediction methods, m (i.e., the max $\mathrm{a_j(i)}$); and (2) $\mathrm{E_{Aj}(i)}$, which is the ratio of the weighted average absolute error over the past q data points of prediction method j at time i,, $\mathrm{a_j(i)}$, to the largest sum of the l absolute error over the past *l* data out of all prediction methods, m (i.e. the max $\mathrm{s_j(i)}$), as follows.

$$\mathrm{E_j(i)} = \mathrm{a_j(i)} / \max_{1\le \mathrm{j} \le \mathrm{m}}[\mathrm{a_j(i)}] \qquad (3\text{-}6)$$

$$\mathrm{E_{Aj}(i)} = \mathrm{a_j(i)} / \max_{1\le \mathrm{j} \le \mathrm{m}}[\mathrm{s_j(i)}] \qquad (3\text{-}7)$$

Finally, $E_j(i)$ and $E_{jA}(i)$, are combined using Equation (3-8), which was derived based on a fuzzy set formulation, where α is another weighting parameter reflecting the importance of the more recent forecasts.

$$\tilde{k}_j(i) = 1 - [\alpha E_j(i) + (1-\alpha)E_{Aj}(i)] \tag{3-8}$$

The $\tilde{k}_j(i)$ are then normalized according to Equation (3-9), and used to derive the weight, f(i+1), for each prediction method, m, at time step (i+1) according to Equation (3-10).

$$k_j(i) = \tilde{k}_j(i) / \sum_{j=1}^{m} \tilde{k}_j(i) \tag{3-9}$$

$$f(i+1) = \sum_{j=1}^{m} k_j(i) f_j(i+1),\ j = 1,2,\dots,m, i = 1,2,\dots \tag{3-10}$$

3.1.2 Modeling Dataset

In this research, the focus is on predicting the next hour traffic volume on the Peace Bridge, particularly the traffic entering the United States from Canada. Hourly traffic volume and classification counts for the Peace Bridge, since 2003, are available for downloading on the Buffalo and Fort Erie Public Bridge Authority's website (Buffalo and Fort Erie Public Bridge Authority, 2014). This study mainly used the 2009 and 2010 passenger car traffic data. The data quality appeared to be excellent with very few hourly traffic counts missing (only 9 out of 8760 in 2009 and 7 out of 8760 points in 2010). For the missing counts, the time series was filled in with the average of the count for the hour before and the hour after the missing value. Given that there were only few data points that were missing, the study did not feel the need to utilize more elaborate imputation algorithms such as those described in the reference (Smith et al., 2003).

Figure 3-1a, Figure 3-1b and Figure 3-1c show how two holidays (one Canadian and one US) and one day with a US sports event have traffic patterns that differ from "ordinary days" (i.e. they fall outside the ± 15% band).

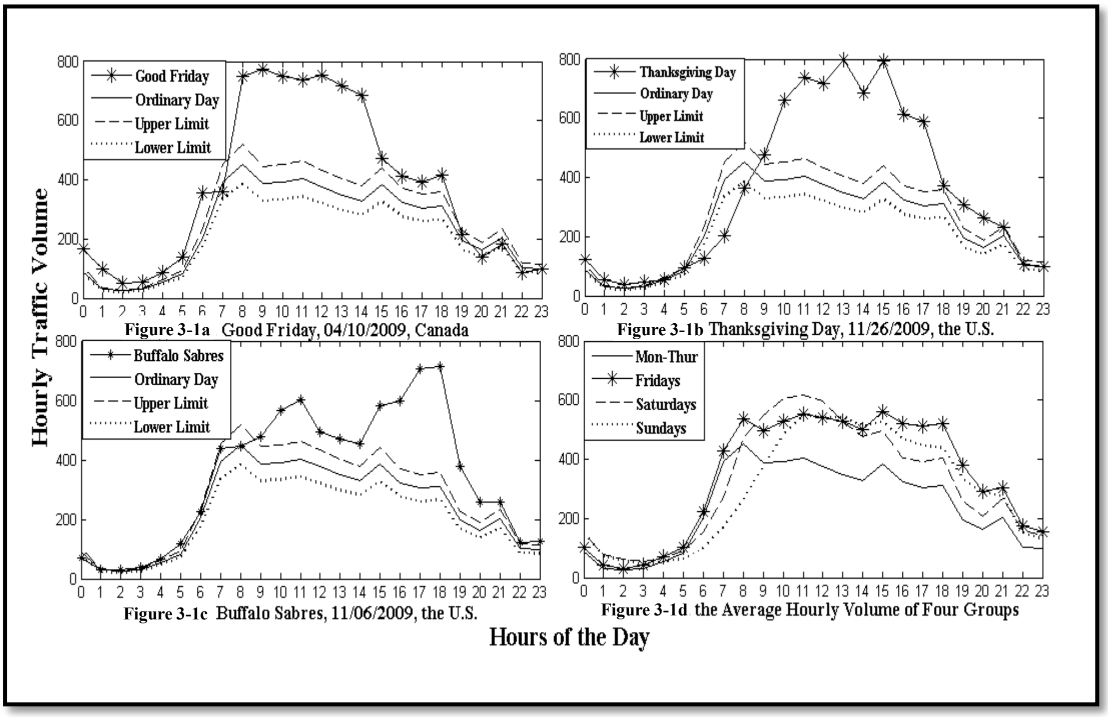


Figure 3-1 Hourly Traffic Volume at the Peace Bridge on Different Days

By following this procedure, the study identified the holidays with traffic patterns that differ significantly from "ordinary days", and a separate group or cluster was defined for those holidays. In total, there were 22 such holidays in 2009, 5 of which belonged to Canadian holidays, while the other 17 days (including three long weekends: Thanksgiving, Christmas and New Year's holidays) were U.S. or common holidays of the two countries.

For game days, the analysis showed that the days when the Buffalo Sabres' and Buffalo Bills' games were held, exhibited significantly different trends from ordinary days (Figure 3-1c). As a result, a separate group was defined for those days. In total, there were 50 game days in 2009, among which three were also holidays, and 48 game days in 2010. The study also looked at how traffic patterns generally varied by the day of the week. Figure 3-1d plots the average hourly traffic volumes for four additional groups (i.e. weekdays excluding Friday, Fridays, Saturdays, and Sundays). As can be seen, the diurnal distributions for these four groups differ significantly and warrant defining separate groups.

Based on this, six different groups, for which separate prediction models were developed, were defined: (1) weekdays excluding Fridays (a total of 181 days in 2009 - only 2009 data were used for this group since there were enough data for the analysis); (2) Fridays (35 days in 2009 and 37 days in 2010); (3) Saturdays (with 38 and 41 days in 2009 and 2010, respectively); Sundays (with a total of 83 days in 2009 and 2010); (5) game days (a total of 98 days in 2009 and 2010); and (6) holidays (a total of 22 days in 2009). The following Chapter will describe the process of model development and evaluation for the first five groups (for holidays, given the small data size, a different methodology is being devised based on a case-based reasoning approach). Finally, it needs to mention that the data used in the study included traffic volumes under non-recurrent events (e.g. accidents, emergencies, inclement weathers, etc.). No attempts were made to screen or exclude those data points.

3.1.3 Modeling Development and Results

3.1.3.1 Prediction Accuracy Measure

Two measures were utilized in this study to assess the accuracy of the models developed: (1) the Mean Absolute Percent Error (MAPE); and (2) the Root Mean Square Error (RMSE). It is noted that in calculating both MAPE and RMSE, only the hourly volume from 7:00 am to 21:00 (i.e. 9 pm) from each day is utilized. This is because the hourly volumes for the hours from 10 pm to 6:00 am are usually very small (less than 100 vehicles per hour), and hence suffer little delay.

3.1.3.2 SARIMA Model

The Statistical Package for Social Sciences (SPSS) was used to build the SARIMA models, with a separate model developed for each of the groups defined above. The seasonal cycle for each model was set to 24, corresponding to the 24 hours in a day. Because SARIMA models are typically used for off-line modeling, a simple procedure was developed to allow the SARIMA model to be used for on-line prediction. This procedure basically involved fitting the SARIMA model first using a training data set made of the first part of the time series (how the length of the training data was determined will be discussed next), and then recalibrating the SARIMA model to update the model parameters after each prediction. When recalibrating the model, the most recent observation is added to the training data set, and the first or oldest data point in the training time series is dropped to keep the computational burden of model recalibration manageable. This process, which was automated using a syntax file in SPSS, results in what may be viewed as a moving window that updates the part of the time series used for model calibration. Note that although there are more mathematically rigorous methods available to do so (e.g. a state-space representation of the SARIMA model coupled with a Kalman Filter (Shekhar and Williams, 2007)), the simple procedure described above was deemed adequate for the purposes of this study, especially since the computation time required for recalibrating SARIMA after each prediction was very short.

To determine the appropriate length of the training data set, different lengths were tried, and for each length, the prediction errors for a test dataset consisting of 888 data points (about 20% of the whole time series available for the weekdays group) were calculated. The results are shown in Figure 3-2a where it can be seen that, for the weekday group, a length of a training dataset of 960 hours (or 40 days) yielded the best performance in terms of the MAPE.

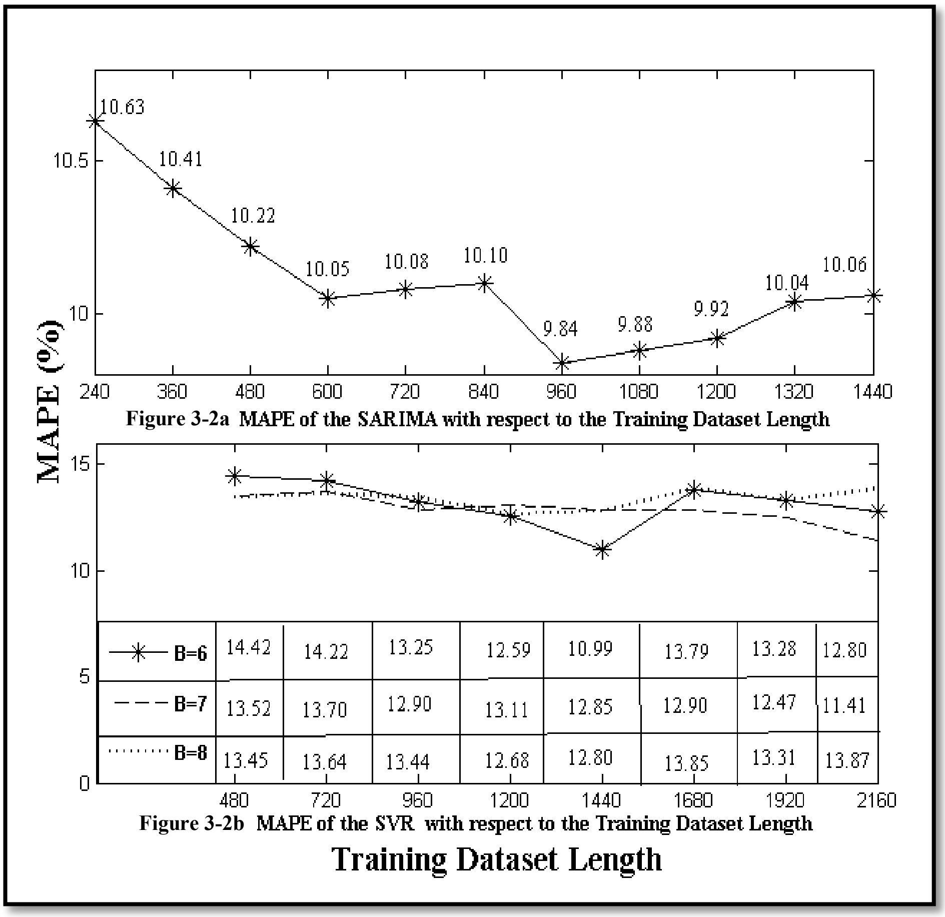


Figure 3-2a MAPE of the SARIMA with respect to the Training Dataset Length

	480	720	960	1200	1440	1680	1920	2160
—✳— B=6	14.42	14.22	13.25	12.59	10.99	13.79	13.28	12.80
– – – B=7	13.52	13.70	12.90	13.11	12.85	12.90	12.47	11.41
········ B=8	13.45	13.64	13.44	12.68	12.80	13.85	13.31	13.87

Figure 3-2b MAPE of the SVR with respect to the Training Dataset Length

Figure 3-2 Training Dataset Length and Model Performance

The same procedure was followed to develop the SARIMA models for the other four classes (i.e. Fridays, Saturdays, Sundays, and game days). Table 3-1 lists the size of the training and test data sets for each model, along with the model's MAPE and RMSE calculated for the test data set.

Table 3-1 Prediction Performance of the SARIMA Model

Data group	Size of full dataset	Size of training dataset	Size of test dataset	MAPE (%)	RMSE (veh/hr)
Weekdays (Mon - Thurs)	4,344 (181 days)	960 (40 days)	3,384 (141 days)	9.84%	47.95
Fridays	1,728 (35 days in 2009; 37 days in 2010)	960 (40 days)	768 (32 days)	10.28%	64.74
Saturdays	1,896 (38 days in 2009; 41 days in 2010)	720 (30 days)	1,176 (49 days)	10.80%	57.60

Sundays	1,992 (42 days in 2009; 41 days in 2010)	720 (30 days)	1,272 (53 days)	11.60%	57.38
Game Days	2,352 (50 days in 2009; 48 days in 2010)	1,440 (60 days)	912 (38 days)	15.17%	78.91

As can be seen from Table 3-1, with the exception of the game days, the MAPE for the models was between 9.84% and 11.60%. For game days, the MAPE was slightly higher (around 15%), probably because of the complex traffic patterns likely to occur on those days. Also note that for SARIMA, the best performance was achieved when predicting weekday traffic.

3.1.3.3 SVR Model

For the SVR model, the value of the insensitive loss function $(\varepsilon)$was set as 0.01, and the Radial Basis Function (RBF) was chosen as the kernel function. As opposed to the SARIMA model, SVR can be easily adapted for on-line prediction. Specifically, for this study, the input to the SVR model at a given time step is a vector, X(t), whose length is B, defined as $X(t) = [x(t), \dots, x(t - B + 1)]^{T}$. The SVR would then use X(t) to predict the next data point in the series, X(t+1). In other words, the model would always use the most recent B data points or hours to predict the next hour.

Before using the SVR model in this fashion, however, it needs to be calibrated by determining the optimal values for the cost factor C and the gamma parameter of the RBF. To do this, a training data set similar to the one used in conjunction with the SARIMA model, needed to be defined; we use the letter O to refer to the length of the training dataset. Moreover, to improve accuracy, we recalibrate the SVR model (i.e., determine new values for C and gamma) every P hours (P=120 hours in this study).

The reason the SVR model is only recalibrated every 120 time steps and not after each prediction as was the case with SARIMA, is that the calibration process is computationally intensive and hence cannot be practically performed after each prediction. Once again, to keep the size of the training data set, O, fixed, we adopted a moving window strategy which replaces the first P data points in O with the latest P points added.

To determine appropriate values for both B and O, multiple combinations of the values of the two parameters were tried, and the MAPE for a test period of 10 days was calculated for each combination. Figure 3-2b shows the results from these experiments performed on the weekday class. As can be seen, the combination of B=6 and O=1440 appears to yield the smallest prediction error for that class.

Table 3-2 shows the optimal values for the training dataset size, B, and the input vector length, O, for the five model classes. The table also lists the size of the test dataset used to calculate the MAPE and RMSE for each model. As can be seen, the MAPE for the SVR model ranged from 9.42% to 13.62%. An interesting observation regarding the SVR model performance is that the smallest prediction error for SVR (MAPE of 9.42%) was that corresponding to the game day model. This is in stark opposition to the SARIMA model where the game day SARIMA model had in fact the largest MAPE (15.17%). The RMSE of the game day SVR model is also lower than that in SARIMA model. This in turn tends to support the observation that traditional time series, given its assumption of linear correlation among the data points, may face difficulties when dealing with the non-linearity of complex patterns such as those observed on game days. For such patterns, the SVR paradigm appears to provide better performance.

Table 3-2 Prediction Performance of the SVR Model

Classes	Full Dataset Size	Size of Training Dataset, O	Input Vector Length, B	Size of Test Dataset	MAPE (%)	RMSE (veh/hr)
Weekdays (Mon Thru Thurs)	4344 (181 days)	1440	6	2904 (121 days)	10.37%	49.64
Fridays	1728 (35 days in 2009; 37 days in 2010)	1440	7	288 (12 days)	9.87%	58.28
Saturdays	1896 (38 days in 2009; 41 days in 2010)	1440	6	456 (19 days)	12.50%	64.23
Sundays	1992 (42 days in 2009; 41 days in 2010)	1440	6	552 (23 days)	13.62%	57.03
Game Days	2352 (50 days in 2009; 48 days in 2010)	1440	6	912 (38 days)	9.42%	52.17

3.1.3.4 Multi-model Combined Forecasting Method

The comparison of the SARIMA and SVR model results indicates that each model appears to have its own set of strengths and weaknesses. For example, the SARIMA model is good at handling the linear characteristics of the data, such as seasonality and trend, whereas SVR is capable of capturing the nonlinear characteristics. This Chapter develops methods to combine forecasts from the two models to improve the quality and accuracy of the predictions.

Specifically, two methods for combining the forecasts from the two models were investigated. The first method, called the simple or fixed weight method, simply compares the performance of each model (i.e. SARIMA and SVR) for predicting a specific hour of the day (e.g. 7:00 am or 8:00 am, etc.), over the whole training dataset. If for example, there were more instances in which SARIMA performed better than SVR for the 7:00 am volume prediction, SARIMA is selected for all future predictions for the 7:00 am hour (i.e., the weight assigned for SARIMA in this case would be 1, and for SVR 0). Our analysis of weekday predictions, for example, showed that SARIMA appeared to be the better model for 7 and 8 am, and for 2, 3, 6, 8 and 9 pm. For the remaining hours, SVR was the better model.

The second method is the Fuzzy Adaptive Variable Weight method previously described. When applying this method, the Fresh Degree Function, F(t), was assumed to be equal to t^2. It should be noted that in this method the performance of the model for predicting a given hour (e.g. 7 am) is evaluated based on the model's performance in predicting that same hour over the past few days and not on the model's performance over the past few hours of that day (i.e., the t index of F(t) is an index referring to the day number or sequence). This is because, as mentioned before, each model outperforms the other for predicting certain hours of the day. With respect to the other parameters of the weight method, q in Equation (3-4) was set to 3, the moving window length l in Equation (3-5) was set to 5, and the values of α in Equation (3-8) were set to 0.84, 0.7, 0.75, 0.84, 0.84 for the weekdays, Fridays, Saturdays, Sundays, and Game days, respectively. Those α values were chosen after trying many different values and picking the ones with the best performance.

Table 3-3 shows the magnitude of the improvement in the quality of the SARIMA and SVR predictions when their forecasts are combined using both the simple fixed weight method and the fuzzy adaptive weight method. For each class, the first five days were used to calculate the weights for the fuzzy adaptive weight method and performance was evaluated on the remainder of the dataset.

Two important observations can be made with regard to Table 3-3. First, it is clear that both methods for combining the results appear to improve the quality of the results. Specifically, for all five classes, both of the combined multi-model forecast methods are better than single model forecasts, with the exception of only the fixed weight method when used on the game day group, where the combined method performs better than SARIMA but worse than SVR. The second observation is that the fuzzy adaptive variable method clearly outperforms the simple weight method, and appears to yield a dramatic improvement in the quality of the results. Specifically, with the fuzzy adaptive variable method, the MAPE for all five classes is in the range of only 6% to 8%, and the RMSE for all 5 classes is lower than 45 vehicles/hour.

Table 3-3 Models' Prediction Performance Comparison

	MAPE (%)				RMSE (vph)			
	SARIMA	SVR	Fixed Weight	Variable Weight	SARIMA	SVR	Fixed Weight	Variable Weight
Weekdays (35 days)	10.10%	9.37%	7.80%	6.32%	99.60	48.95	74.88	37.49
Fridays (12 days)	10.09%	9.78%	8.36%	7.68%	57.42	51.55	47.49	43.49
Saturdays (19 days)	10.84%	11.80%	9.49%	7.59%	60.00	56.44	77.37	39.07
Sundays (23 days)	11.55%	12.23%	10.59%	7.65%	59.54	52.42	53.59	39.38

Game days (38 days)	15.31%	8.58%	11.93%	7.32%	79.94	48.52	64.60	43.73

For a more disaggregate view of the performance of the models, Figure 3-3 below compares the traffic volume predicted by SARIMA, SVR and the combined forecasting method, against the field observations for a period spanning a total of 60 hours. As can be seen, for some hours (e.g. those inside circle A), SARIMA outperforms SVR, whereas for other hours (e.g. inside circle B), SVR performs better than SARIMA. By combining the two methods, the final forecast is generally quite closer to the observed values (with naturally a few exceptions such as the hour enclosed by rectangle C).

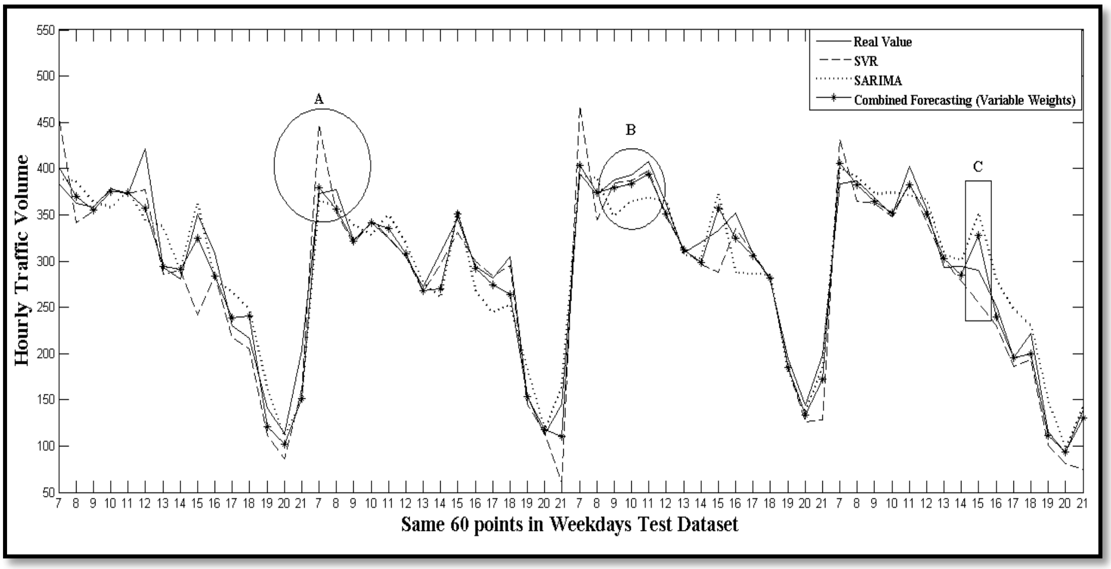


Figure 3-3 Traffic Volume Predictions of SARIMA, SVR, and the Combined Forecasting Method with Variable Weight

3.2 On-line Prediction of Border Crossing Traffic Using DTW-SPN

3.2.1 Methodology

In this research, the original Spinning Network method (SPN) is improved based on dynamic time warping (DTW), and it is compared with the SARIMA and SVR for the on-line prediction of border crossing traffic. The introduction of SARIMA and SVR can be found in Chapter 3.1.1.2. Here only SPN is introduced.

3.2.1.1 Spinning Network method (SPN)

The SPN method is a novel forecasting algorithm originally proposed by Huang and Sadek (2009). The method is inspired by the functionality of human memory in sensing, processing, and predicting the states of the surrounding environment, and attempts to mimic some aspects of human memory including: (1) the fuzzy nature of the information retrieved; (2) the instinctive association of ideas; and (3) the fact that the quality of the information retrieved is a function of the time and effort invested. While the method shares some features with the nearest neighbor approach, one of its key advantages is its dramatic computational efficiency compared to other forecasting algorithms including the nearest neighbor algorithm itself.

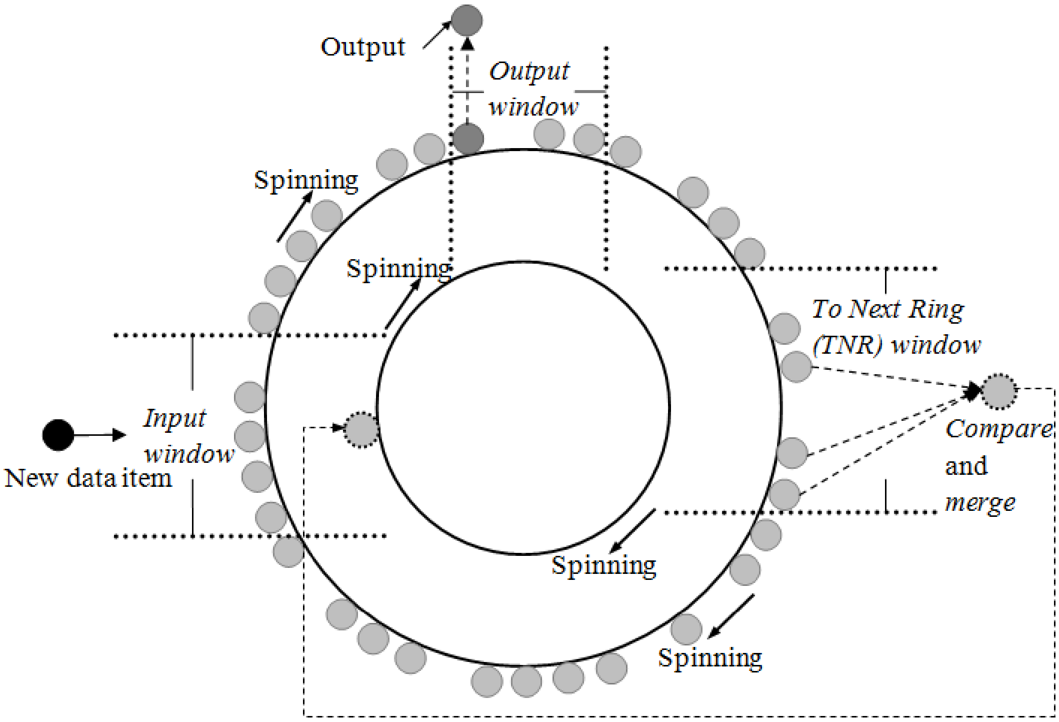


Figure 3-4 Spinning Network (SPN) (Revised based on Huang and Sadek, 2009)

As shown in Figure 3-4, the SPN consists of a set of consecutive rings on which the data items are stored and processed. Data items constitute the fundamental elements in the SPN. They denote the information that a "person" receives or recalls, and can be a vector, a matrix or an image. In the case of traffic volume prediction, a data item would take the form of a vector consisting of the current hourly traffic volume and the volumes collected in previous hours. Each ring has a fixed capacity that determines how many data items it can store. It also has the functionality of merging or consolidating similar data items and forwarding them to the next ring. This merge function has the benefit of saving space on the rings (i.e., reducing the number of data items stored and searched for) and of increasing the information content of data items, since the information content of a merged data item represents an integration of the information contained within the elements that got merged into it.

Each ring is exposed to three windows as also shown in Figure 3-4, representing the stages of receiving, consolidating, and outputting information, namely the input window, the To-Next-Ring (TNR) window, and the output window. Specifically, the input window accommodates an arriving data item by either placing it in one of its vacant cells if any, or merging it with the most similar data item in the window in case there is no vacant cell. The TNR window consolidates or merges similar data items and forwards the merged data item onto the next ring. Finally, the output window scans its cells and picks the data item most similar to the new data item entering the

SPN as the output (which in this case would represent the predicted hourly traffic volume for the next hour). When outputs are generated from all the rings, they are evaluated again, and the one that is most similar to the new data item is selected as the final prediction of the new data item. The rings spin continuously in a clockwise or counterclockwise direction so that the different parts of the ring are exposed to the input, TNR, and output windows, ensuring that a wide range of historical data items are examined.

As can be seen from the brief discussion above, the two key functions at the core of the SPN data processing are the “compare” function and the “merge” functions. The “compare” function in the original SPN algorithm used the Euclidean distance between two vectors to measure the degree of similarity between two data items. The “merge” function, on the other hand, combines similar data items by averaging their associated values. Given that the more a data item has been merged with others, the more stable and informative its value would be, the function also records the number of times a given item has been merged with others before the current merging, and uses that count as a weight when calculating the average.

3.2.2 Datasets

In this study, the same dataset described in Chapter 3.1.2 is used. The complexity of this dataset is calculated in Chapter 3.2.1.1.1, and its non-linearity is calculated in Chapter 3.2.1.1.2. Similarly, the dataset is also grouped into six different data groups as shown in Chapter 3.1.2. The DTW-SPN is built for the whole dataset and the separate grouped datasets, as well as other models.

3.2.3 Model Development

3.2.3.1 Enhanced SPN

Refinement of the original SPN algorithm was deemed necessary to allow it to handle the increased complexity and the non-linearity character of the Peace Bridge dataset, compared to the Virginia set used in the original study. As discussed before, both the “compare” and the “merge” functions of the SPN involve assessing the similarity between data items. The most straightforward way of evaluating the similarity of two data items is to calculate the Euclidean distance between them, and this was the approach implemented in the original SPN (referred to hereafter as the Euclidean-SPN). However, the Euclidean distance is a brittle distance measure that is incapable of dealing with elastic timing shifts in the time series data. For example, in the case of the two time-dependent sequences shown in Figure 3-5, sequence Y shares similar patterns with sequence X although its peak timings (or peak spans) are shifted (or distorted). If the Euclidean distance is used as the similarity metric, sequences X and Y would be regarded as quite different, and the similarity between the two patterns would go unnoticed.

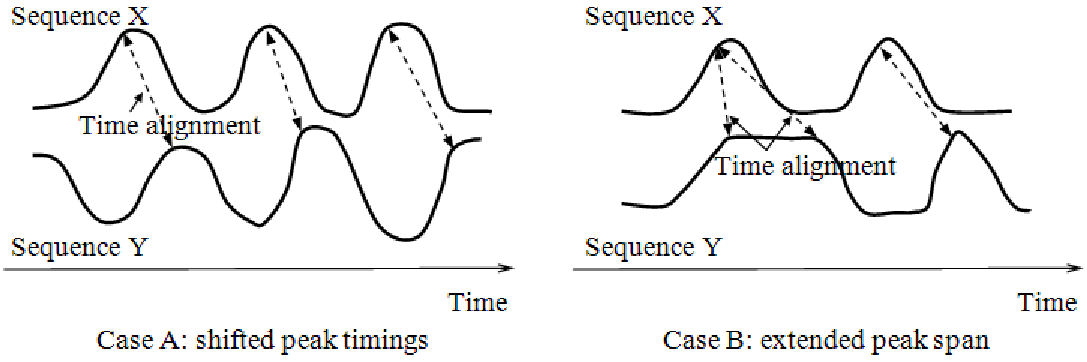


Figure 3-5 Similarity between Time-dependent Sequences

To solve this issue, an alternative algorithm, called Dynamic Time Warping (DTW), has recently been proposed as a similarity metric between time series sequences. DTW explores every possible time alignment to pair the elements of the two sequences, and then seeks the best pairing that returns the minimum distance. In comparison to the Euclidean distance based measure, it is much more robust and allows similar shapes to match even if they are out of phase in the time axis (Keogh and Ratanamahatana, 2005). The DTW distance is calculated in the following manner (Muller, 2007).

Given two time-dependent sequences $\mathrm{X} = [x_1, x_2, x_3, \dots, x_N]$ and $\mathrm{Y} = [y_1, y_2, y_3, \dots, y_M]$, an $N \times M$ accumulated distance matrix *V(N, M)* is constructed, with its entries calculated as in Equation (3-11). The last entry of the matrix, $v(N, M)$ records the minimum distance associated with the best time alignment between X and Y, and is defined as the DTW distance of the two sequence, i.e., $DTW(X, Y) = v(N, M)$. As can be seen, Euclidean distance is a special case of DTW when the fixed pairing $\{(x_1, y_1), (x_2, y_2), \dots, (x_{\min(N,M)}, y_{\min(N,M)})\}$ is chosen for distance calculation.

$$v_{(\mathrm{i},\mathrm{j})} = \begin{cases} \sum_{k=1}^{j} d(x_1, y_j), i = 1, j \in [1, M] \\ \sum_{k=1}^{i} d(x_i, y_1), i \in [1, N], j = 1 \\ d(x_i, y_j) + \min\{v_{(\mathrm{i-1},\mathrm{j})}, v_{(\mathrm{i-1},\mathrm{j-1})}, v_{(\mathrm{i},\mathrm{j-1})}\}, others \end{cases} \quad (3\text{-}11)$$

where,

$d(x_i, y_j)$ is the distance between x_i and y_j, it could be $d(x_i, y_j) = |x_i - y_j|$ or $d(x_i, y_j) = (x_i - y_j)^2$.

Different from Huang and Sadek (2009) where the focus was on the Euclidean distance, this research tested both the Euclidean distance and the DTW distance when developing the SPN model. The resulting models are labeled as the Euclidean-SPN (or Eu-SPN for short) and the DTW-SPN respectively, and are compared later to see which similarity measure performs better.

3.2.3.2 SPN parameters

As discussed in Chapter 3.3.1.1, the SPN has several parameters that may be tuned in order to improve performance. In this study, values of those parameters were determined primarily through experimentation. This involved changing the value of a given parameter until the value that yielded the best performance or the lowest Mean Absolute Percent Error (MAPE) was identified. The parameter tuning process and results are briefly discussed below.

3.2.3.2.1 Data item length

The length of the data items refers to the number of elements included in the input vector (referred to as a data item in SPN). In the case of the border crossing traffic prediction, the best prediction results were obtained when the input vector included the current hourly traffic volume along with the hourly volumes for the previous 18 hours. This meant that the total length of the SPN's data items had to be set to 20, with the first 19 elements constituting the input information for the prediction problem (called the historical span) and the 20th element representing the predicted traffic volume for the next hour (i.e., the prediction span). In other words, the hourly data elements in the historical span of a data item are used to evaluate whether two data items are similar or not. If the two data items are deemed similar, the known volume of the future hour in one data item can be used to predict the one-hour future volume of the other.

3.2.3.2.2 Number of rings and ring size

In the SPN, the rings hold the data items and serve, through the "merge" function, to consolidate similar data items, thereby making them more stable and informative. With respect to the number of rings, the SPN was found to perform the best on the border crossing dataset when the number of rings was set to four. Regarding the size of the rings, a general observation about the SPN is that the smaller the capacity of the rings the greater is the pressure to merge and consolidate data items, mainly because of the ring space constraint (i.e., each ring has a pre-defined capacity). In our study, after some experimentation, we set the size of the outer ring as 6,000 slots or data items, with the size of each inner ring being 10 slots less than the size of the ring immediately preceding it.

3.2.3.2.3 Input, TNR, and Output Windows' sizes

As mentioned before, there are three windows in the SPN model, namely the input window, the To-Next-Ring (TNR) window, and the output window. To increase precision in this study, the size of the input and output windows was set to be equal to the full size of the ring. The size of the TNR window was arbitrarily set to 10% of the size of the ring.

3.2.3.2.4 Spinning speed

Experimentation with different spinning speeds for the rings showed that the rings' spinning speed did not have a major impact on the SPN performance (this in fact should be expected because the size of the input and output windows was set to be equal to 100% of the ring size). Given this, the spinning interval was arbitrarily chosen as 4 ms (millisecond) for the outermost ring, with the interval increasing by 1 ms for each inner ring.

3.2.3.2.5 Threshold to Next Ring (TTNR)

The "Threshold to Next Ring (TTNR)" is a parameter associated with the TNR window. It denotes the minimum number of similar data items that need to be identified within the TNR window before a data "merge" can be conducted. This ensures that only common enough patterns are combined and forwarded to the next ring. A large value of TTNR would reduce the frequency of the "merge" operations (since merging would only happen in that case when a large number of similar data items are identified within the TNR window), and the outer ring would thus fill up quickly. In this study, the threshold was set to two data items.

3.2.3.2.6 Distance tolerance

The "distance tolerance" is a parameter associated with the merge function, which specifies how close two data items need to be in order to be merged. That parameter was found to have a significant impact on the SPN performance. Large values would encourage unnecessary data merging while small values would prevent merging. After some experimentation, a value of the distance tolerance parameter equal to 60 vehicles/hour (vph) was found to achieve a good balance. Here, the 60 vph is around 15% of the average hourly traffic volume in the dataset.

3.2.3.3 SARIMA model and SVR model

As two bench-marking models, the same parameter settings of SARIMA and SVR have been introduced in Chapter 3.1.3.2 and Chapter 3.1.3.3.

3.2.4 Evaluation Results

In this Chapter, we compare the prediction performance of the four models (i.e. DTW-SPN, Euclidean-SPN, SARIMA, and SVR) on both the classified dataset (i.e. the one divided into five groups) and the unclassified set. The models were tested on the hourly volumes from 7:00 to 21:00 of each day because the night hours (22:00-6:00) are of little interest due to their low traffic volumes (e.g., less than 100 vph). The test dataset included 127 valid days (1,905 hours). For the classified case, the test set was itself broken into the corresponding groups as follows: Weekdays (Mon-Thur) (525 hours), Fridays (180 hours), Saturdays (285 hours), Sundays (345 hours), and Game Days (570 hours). The discussion of the results from testing on the classified datasets is presented first, followed by the results from testing on the unclassified or the whole data set.

3.2.4.1 Comparisons of the four models based on the classified dataset

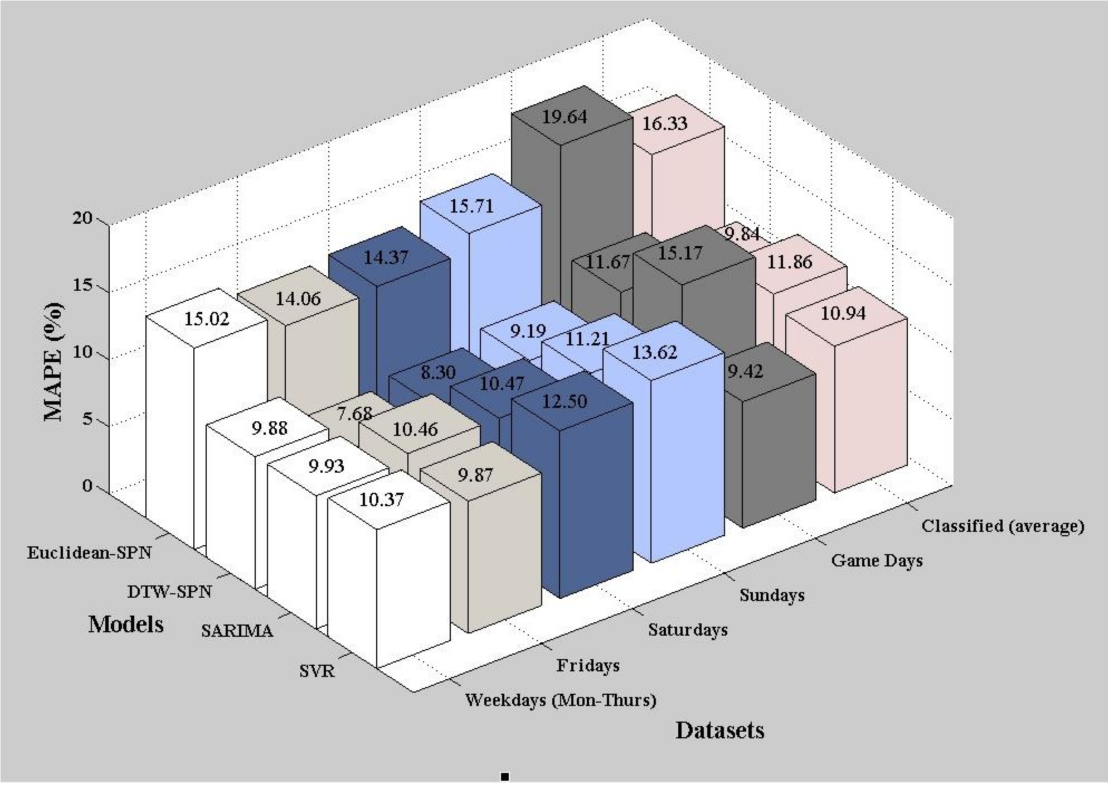


Figure 3-6 Prediction Performance of the Four Models for Different Data Classes

Figure 3-6 plots the MAPE of the four models when tested on the five groups. As can be seen, the DTW-SPN outperformed all other three methods for almost all the data groups. The only exception was the game day category where SVR performed the best closely followed by DTW-SPN. The superiority of DTW-SPN, compared to the other models, is also confirmed when one calculates the average MAPE for all the five groups. As can be seen also from Figure 3-6, DTW-SPN had the lowest average MAPE at 9.84%, followed by SVR at 10.94%.

The Euclidean-SPN, on the other hand, did not perform that well and its MAPE was invariably higher than the two benchmarking algorithms (i.e., SARIMA and SVR). One reason behind this could be that the parameters of the SARIMA and SVR were re-calibrated and tweaked for each data group, whereas the SPN's parameters were only calibrated once.

The results also appear to confirm the known strengths and weaknesses of traditional time series models such as SARIMA, compared to AI-based methods such as SVR. SARIMA is good at handling linear data sets such as weekday traffic that exhibits strong seasonality and trends, but is challenged when dealing with non-linear patterns (e.g., the game day group). SVR, on the other hand, appears to be capable of capturing the nonlinear patterns, particularly for the days that involve significant volume fluctuations such as game days.

The robustness of the four methods, or more specifically the capability to deal with the sudden changes in traffic volume levels, was also tested. To do this, we first identified what we refer to as, hours with abrupt traffic volume changes or hourly traffic volumes that are dramatically different from the volume in the previous hour. Specifically, the hours with an abrupt change were defined as those with volumes that are greater than 2.5 times (or lower than 0.4 times) of the preceding hourly volume. As found in Table 3-4, there are 36 abrupt points in total and 26 of them come from the Game Days group.

Table 3-4 Model Performance for the Abrupt Points in the Classified Datasets

MAPE (%)	Weekdays (6*)	Fridays (0)	Saturdays (2)	Sundays (2)	Game days (26)
DTW-SPN	10.63	NA	54.48	21.22	26.67
Euclidean-SPN	25.10	NA	61.91	49.02	55.78
SARIMA	17.56	NA	58.16	33.18	49.68
SVR	9.92	NA	13.06	23.89	9.78

Note: * the number in the parenthesis denotes the number of hours (or data points) with abrupt traffic volume changes.

As can be seen from Table 3-4, SVR outperforms the others in estimating abrupt hourly traffic volumes, except for the Sunday group. DTW-SPN ranks as the best for the Sunday group, and is the second best for the other groups. In contrast, Euclidean-SPN and SARIMA are not good at all at predicting these abrupt traffic volumes. These results demonstrate the comparable ability of DTW-SPN in capturing abrupt changes in traffic volumes although SVR is still the best in general.

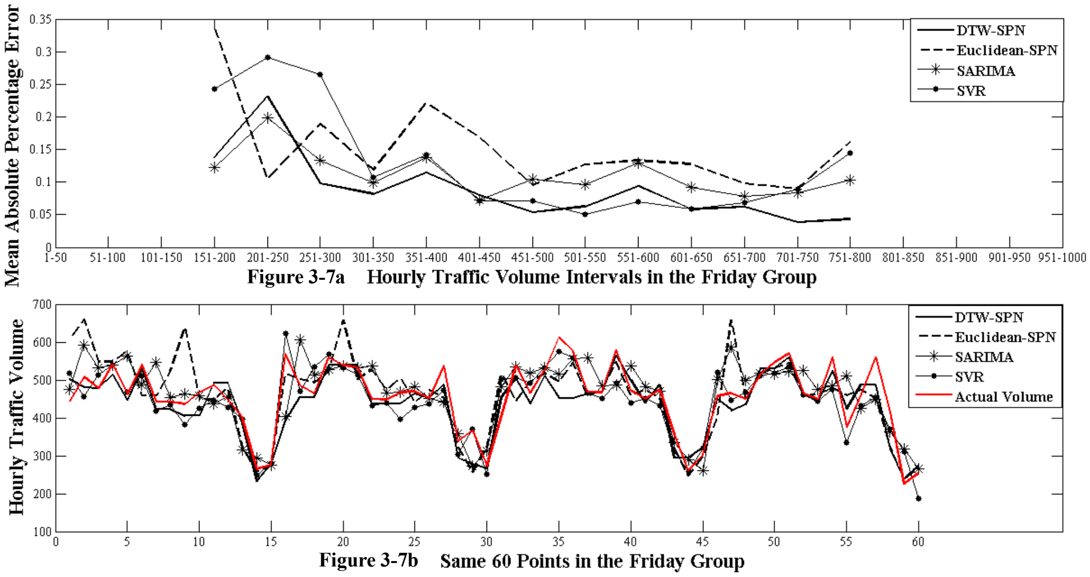


Figure 3-7a Hourly Traffic Volume Intervals in the Friday Group

Figure 3-7b Same 60 Points in the Friday Group

Figure 3-7 Estimation Performances of Four Models for the Friday Group

To assess the prediction performance of the models in more detail, the MAPEs of each model with respect to different levels of traffic volumes in the Friday group are shown in Figure 3-7a. As can be seen, DTW-SPN performs the best for almost all traffic volume levels when the volume is greater than 250 vph. For low traffic volumes such as 151-200 vph and 201-250 vph, SARIMA or Euclidean-SPN tends to be the best. To provide a different view of the models performance, the predicted versus actual hourly traffic volumes were plotted for a sample of 60 consecutive hours for the Friday group (Figure 3-7b). Consistent with the previous observations, DTW-SPN and SVR outperform Euclidean-SPN and SARIMA in estimating hourly traffic volumes.

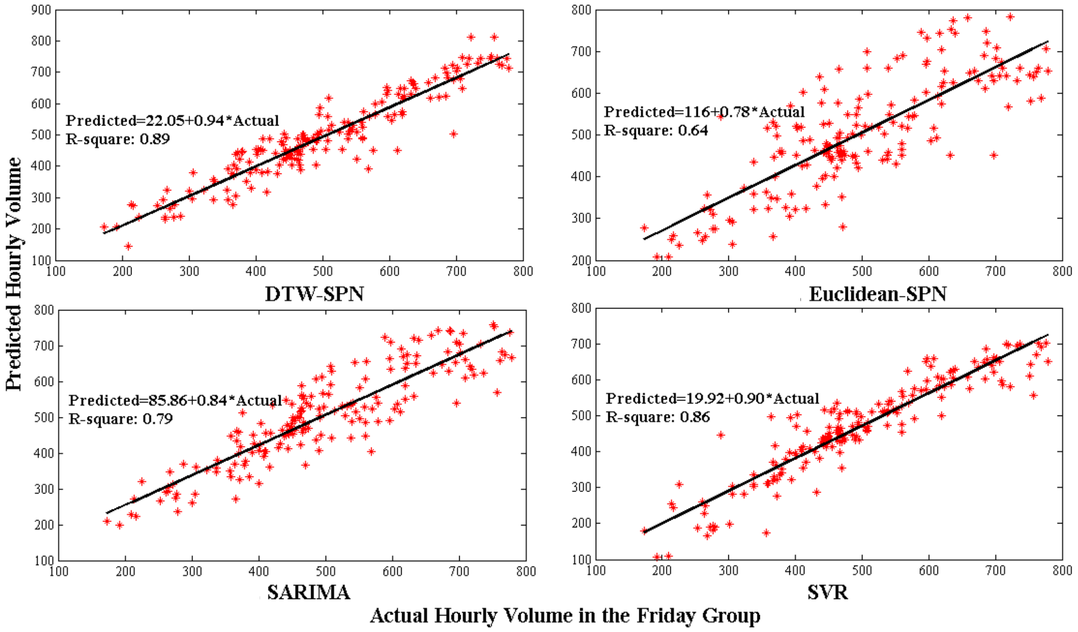


Figure 3-8 Predictions of Four Models versus Actual Volumes in the Friday Group

Finally, the plots of the models' predictions against the hourly traffic volume observations are shown in Figure 3-8. As can be seen, the linear fitting curve associated with DTW-SPN has the highest R-square (0.89), implying that DTW-SPN is the best model with the highest prediction accuracy among the four. The SVR model comes as a close second with an R-square value of 0.86.

3.2.4.2 Comparison of the four models based on the non-classified dataset

Besides comparing the models' performance on the classified dataset, their performance was also evaluated on the unclassified set that did not distinguish among the different day types. The results are shown in Table 3-5. As indicated by MAPE, the DTW-SPN model once again outperformed all other models, and had the lowest MAPE at 10.60%. The SVR came in second, followed by the SARIMA model and the

Euclidean SPN. In thus appears that the use of the DTW distance measure (as opposed to the Euclidean distance) has significantly improved the performance of the SPN, reducing the MAPE from 16.49% to only 10.60%.

Table 3-5 Prediction Performance of SPN, SARIMA, and SVR for the Unclassified Data

Method	MAPE (%) for the Entire Dataset (1,905 hours)	MAPE (%) for Hours Showing Abrupt Changes (36 hours)
DTW-SPN	10.60	27.55
Euclidean-SPN	16.49	69.44
SARIMA	16.38	52.95
SVR	14.57	12.59

Table 3-5 also shows the MAPE for the 36 hours with abrupt volume changes, as defined in Chapter 3.2.4.1 consistent with the previous findings, SVR yielded the best performance, followed by DTW-SPN. Both methods were much more accurate than either SARIMA or Euclidean-SPN. This confirms the ability of SVR and DRW-SPN to deal with the non-linearity of the traffic volume time series.

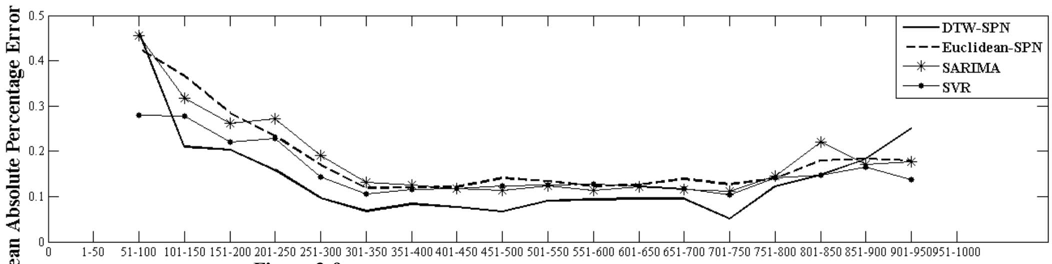


Figure 3-9a Hourly Traffic Volume Intervals in Whole Dataset

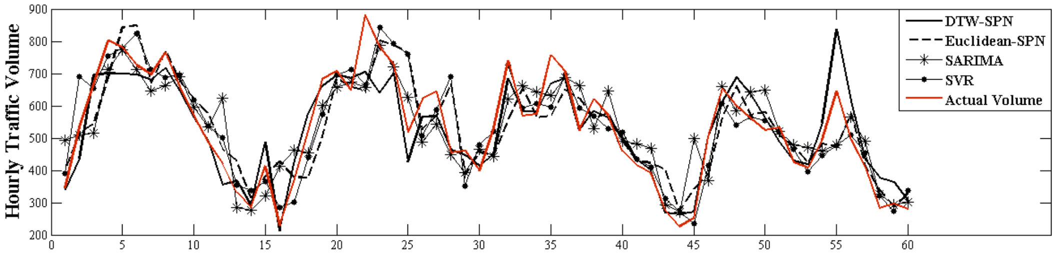


Figure 3-9b Same 60 Points in Whole Dataset

Figure 3-9 Estimation Performances of Four Models for the Whole Dataset

Figure 3-9 plots the four models’ predictions against the observed data. As can be seen from Figure 3-9a, DTW-SPN outperforms the others for the majority of traffic volumes ranging from 100 vph to 800 vph, while SVR performs slightly better for higher volumes. In contrast, the Euclidean-SPN and SARIMA perform worse than both the DTW-SPN and SVR. Figure 3-9b provides a zoom-in view of the models’ estimation performance in comparison to the actual observations for a sample data with 60 hourly traffic volumes. Consistent with the general case, DTW-SPN performs the best for most of the data entries, except for some hours such as the 55th hour.

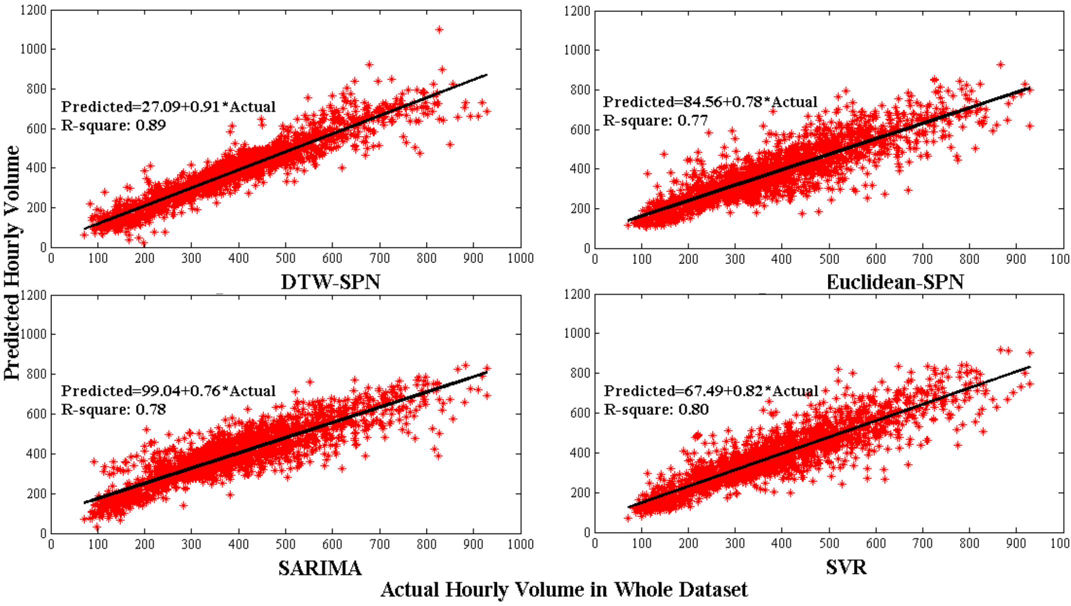


Figure 3-10 Hourly Volume Predictions versus Observations in the Whole Dataset

Figure 3-10 compares the four models’ predictions with the actual observations in the whole dataset (i.e., unclassified). As demonstrated by both the plots and the fitted regression models, DTW-SPN predictions appear to match the real-world observations the most: the fitting curve of the estimate-to-observation scatter plot has the slope which is closest to one; and the corresponding regression model has the highest R-square as 0.89. Among the remaining three models, Euclidean-SPN and SARIMA have similar performance, and both of them are worse than SVR.

3.2.4.3 Impact of data classification

The comparison between the classified case and the unclassified case reveals the role of data classification in improving model performance (the classified case in Figure 3-6 versus the unclassified case in Table 3-5). When the dataset is not classified, the MAPEs of DTW-SPN, Euclidean-SPN, SARIMA, and SVR were 10.6%, 16.49%, 16.38%, and 14.57% respectively. In contrast, after classifying the data by the day type, the average MAPEs were reduced to 9.84%, 16.33%, 11.86% and 10.94%, respectively. This indicates that classifying the data based on similarities of exhibited patterns (e.g., by day groups) is generally helpful in improving the prediction accuracy of all four models.

However, the more interesting observation in that context is with respect to the magnitude in the improvement of the performance of the SPN, as compared to the other model types, when the data is classified, and what that reveals regarding the robustness of the paradigm. Specifically, for both the DTW-SPN and the Euclidean SPN, the improvement in the performance resulting from classifying the data into groups is significantly less (0.76% for DTW-SPN and 0.16% for SPN) than that for SARIMA (4.52%) or SVR (3.63%). This seems to point out to the superior classification ability inherent in the SPN algorithm itself, even without the external help from classifying the data into different day types. The grouping ability of the SPN is naturally the result of the multiple comparisons and merging processed involved in the different windows and rings of network, which allowed the paradigm to exhibit high predictive accuracy even without the external classification of the data. This additional advantage makes the SPN method more practical for real-world traffic volume prediction since no additional effort is required for pre data classification.

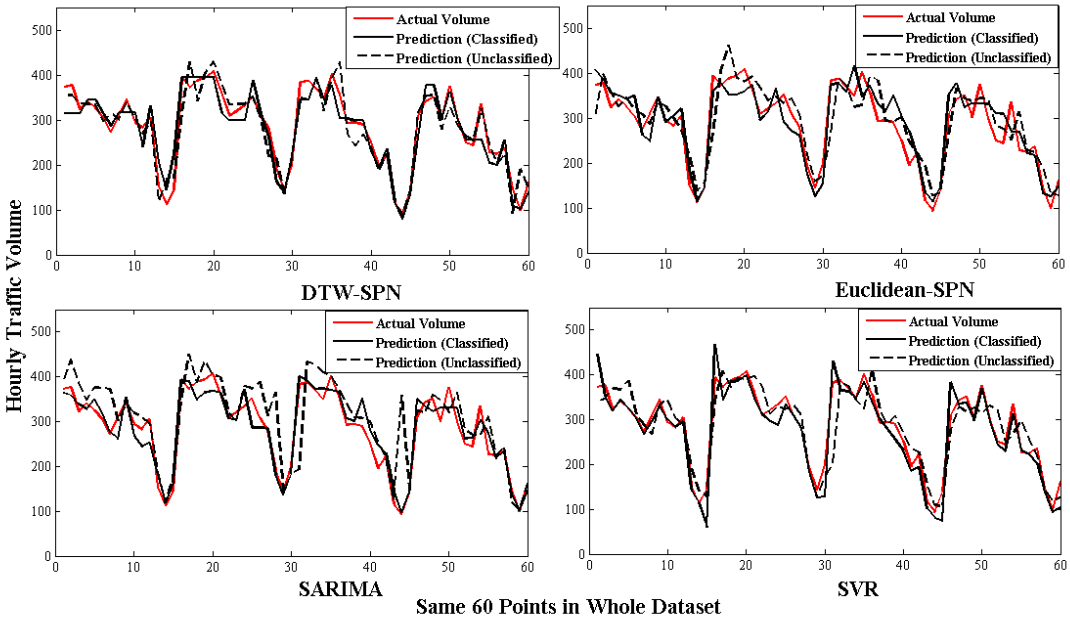


Figure 3-11 Comparison of Model Performance between the Classified Case and the Unclassified Case

As a further illustration of this last point, Figure 3-11 compares the four models' performance for the two cases of the classified (solid line) and the unclassified (dashed line) against actual volume observations for a sample of 60 data points. As can be seen in Figure 3-11, the improvement in the prediction performance of the SPN-based approaches (i.e., DTW-SPN or Euclidean-SPN) due to data classification was significantly less than the improvement for either SARIMA or SVR.

3.2.4.4 Running time comparison of the four models

In addition to prediction accuracy, running time is also considered as an important criterion for model comparison. For the SPN models, the computational burden mainly comes from the comparisons between the data items conducted on the three different windows of each ring, including: (1) the comparisons between existing data items in the input window of a ring and a new data item entering the ring; (2) the comparisons between the existing data items in the output window of a ring and a new data item; and (3) the periodical comparisons among the existing data items on the TNR window before each data merging and forwarding. Given that the testing data length is N_{te} and the sizes of the input window, output window, and TNR window are w_1, w_2, and w_3 respectively, the overall time complexity of the SPN, due to the comparison operations in the three windows of all the rings, can be represented as $O(N_{te} * (w_1 + w_2 + w_3^2/P))$, with P denoting the frequency of invoking the TNR process (i.e., the number of records in between two consecutive executions of the TNR process). If each comparison takes T_{com} , the total running time of SPN is $O(N_{te} * (w_1 + w_2 + w_3^2/P)) * T_{com}$. For the Euclidean-SPN model, the runtime of the compare function, T_{com}, is $O(B)$, where B refers to the length of a data item. For the DTW-SPN, T_{com} is larger and equal to $O(B^2)$ since the distance is calculated based on any random pairing of data elements between the two data items being evaluated. Therefore, the time complexity of the Euclidean-SPN model and the DTW-SPN model are $O(N_{te} * (w_1 + w_2 + w_3^2/P)) * B)$ and $O(N_{te} * (w_1 + w_2 + w_3^2/P)) * B^2)$ respectively.

For the SVR and SARIMA models, due to the moving window training strategy used for on-line prediction, their time complexity depends on both the training data length (N_{tr}) and the testing data length (N_{te}). For the SARIMA model, the time complexity associated with each model training is $O(m^3 N_{tr})$, with *m* being the order of the model (Lu et al., 2010). Therefore, the overall time complexity of SARIMA is $O(m^3 N_{tr}) * N_{te}/R = O(m^3 * N_{tr} * N_{te}/R)$, where *m* refers to the order of the SARIMA model (the *D* in Equation (3-1)) and *R* refers to the frequency of the re-calibration process (specifically the number of records or data points in between calibrations). In our study, given that the model is recalibrated at every prediction step, R is equal to 1. Similarly, for the SVR model, the time complexity for each training phase is $O(N_{tr}^3)$ (Zhao and Sun, 2009), and the total running time is $O(N_{tr}^3) * N_{te}/R = O(N_{tr}^3 * N_{te}/R)$ (*R* =100 for the SVR model).

Generally speaking, the SVR model is the most time consuming prediction method due to the involvement of N_{tr}^3 and the large value of N_{tr}. The time complexity of the SPN model can be controlled by adjusting the size of the input, output, and TNR windows. Table 3-6 summarizes the general time complexity of the four models, and shows the specific running times obtained for the unclassified Peace Bridge data set on a computer with 4.00 GB RAM and Intel® Core(TM) 2 Duo CPU.

As can be seen, given the parameter settings described in the model development Chapter, the Euclidean-SPN model is the fastest prediction method with just 1,836 seconds, followed by the DTW-SPN model at 10,656 seconds. SARIMA required a total run time equal to 39,636 seconds, whereas SVR needed the longest running time. Based on this, it can be concluded that not only the DTW-SPN yielded the highest overall prediction accuracy (as discussed in Chapters 3.3.4.1 and 3.3.4.2), but it is also significantly more computationally efficient compared to either SARIMA or SVR. Specifically, the runtime for DTW-SPN was about one quarter of the runtime for SARIMA and less than 1/15th of the runtime for SVR.

Table 3-6 Computational Time Complexity of SPN, SARIMA, and SVR for the Unclassified Data

Models	Time Complexity (general case)	Running Time (seconds)*
DTW-SPN	$O((w_1 + w_2 + w_3^2/P) * N_{te} * B^2)$	10,656
Euclidean-SPN	$O((w_1 + w_2 + w_3^2/P) * N_{te} * B)$	1,836
SARIMA	$O(m^3 * N_{tr} * N_{te}/R)$	39,636
SVR	$O(N_{tr}^3 * N_{te}/R)$	159,120

Note: * the running times were obtained for the unclassified Peace Bridge data set on a computer with 4.00 GB RAM and Intel® Core(TM) 2 Duo CPU.

3.3 Evaluating Short-term Traffic Prediction Models Based on Multiple Datasets and Data Diagnosis Measures

3.3.1 Methodology

3.3.1.1 Data Diagnosis

The goal of the data diagnosis is to assess the predictability of a time series and to identify which methods are more appropriate for prediction. Multiple measures were utilized in this study including: (1) measures of complexity, such as delay time and embedding dimension analysis and approximate entropy; (2) non-linearity indicators, such as the time reversibility of surrogate data; and (3) measures of long range dependence such as Hurst exponent. Given space limitations, only highlights of each measure are briefly introduced below. Interested readers are referred to the appropriate references for more details.

3.3.1.1.1 Complexity Measures

Delay Time and Embedding Dimension. The idea behind the delay time and embedding dimension method is to make a time-delay reconstruction of the phase or state space of the time series in which to view the dynamics of the system (Jayawardena et al., 2002; Liu et al., 2011). Suppose we have a time series$\{x_t\}$, a new time series, denoted by $\{y_t\}, y_t = \{x_t, x_{t-\tau}, \dots, x_{t-(m-1)\tau}\}$, is first constructed. The new space consisting of such vectors y_t is called the phase space or state space. The elements in y_t include (*m-1*) relevant past values ofx_t, and the relevant past values may lag τ time intervals from each other. Here, τ and *m*, are called the delay time (or lag value) and embedding dimension, respectively. Typically, the best value of delay is determined using the mutual information method which seeks to maximize the joint probability $p(X(t), X(t+\tau))$ given τ (Fraser and Swinney, 1986). For determining the embedding dimension, the false nearest neighbor algorithm can be used. The algorithm scans potential values of *m* in order to identify the optimal value that avoids the inclusion of false or irrelevant data the most (Kennel et al., 1992).

Approximate Entropy. Approximate entropy (ApEn) is a technique for measuring the magnitude of irregularity or unpredictability of fluctuations in a time series. Specifically, the measure denotes the likelihood that fluctuation patterns of a series do not repeat over time. Small values of ApEn usually indicate a predictable dataset with repetitive patterns, whereas larger values of APEn indicate more randomness. After phase space reconstruction discussed previously, the analyst needs to specify the threshold for the similarity criteria, r, which defines whether two patterns are similar. The details of how ApEn is calculated can be found in reference (Ho et al., 1997).

3.3.1.1.2 Non-linear Indicators

Time Reversibility of Surrogate Data. The method of surrogate data tests the nonlinearity of time series by verifying whether a series is consistent with the null hypothesis of a linear Gaussian process (LGP). A process known as the iteratively Amplitude Adjusted Fourier Transform (IAAFT) is first used to generate surrogate datasets from the original time series (Schreiber and Schmitz, 2000). This process is repeated several times, and after each time, a measure known as the time reversibility, r, which measures the asymmetry of a series under time reversal, is calculated (Schreiber and Schmitz, 2000; Disks et al., 1995). The time reversibility value for the original time series, r_0, is also calculated. Finally, the test checks to see if r_0 is within the distribution of r. If it is, the original time series is linear; otherwise, it is nonlinear (Merkwirth et al., 2009).

3.3.1.1.3 Long Range Dependence (LRD Indicators)

Hurst Exponent. Hurst exponent is a measure to characterize the long range dependence (LRD) of a time series (Barbulescu et al., 2010). In the time domain, LRD manifests as a high degree of correlation between distantly separated data points. The values of the Hurst exponent range from 0 to 1, and can be categorized into three groups with different implications: (1) Hurst = 0.5 implies a random time series; (2) 0< Hurst <0.5 indicates a trend-reverting tendency by which the increasing (or decreasing) trend observed at present is likely to flip at the next time instant; and (3) 0.5< Hurst <1 indicating a trend-reinforcing tendency by which the increasing (or decreasing) trend at present is likely to be maintained in the near future. Some researches show that for back propagation neural network models, time series with large Hurst exponent can be predicted more accurately than those series with H close to 0.50 (Qian and Rasheed, 2004). The details about how to calculate the Hurst exponent can be found in reference (Hurst, 1951).

3.3.1.2 Short-term Traffic Volume Prediction Models

In this study, SARIMA, SVR and k Nearest Neighbor (k-NN) are used to build the traffic prediction models for the multiple datasets and compared with each other, the introduction of SARIMA and SVR can be found in Chapter 3.1.1.2. Here only k-NN is introduced.

3.3.1.2.1 k Nearest Neighbor (k-NN)

k-NN is a prediction method which decides the output by finding the k-nearest neighbors (i.e. most similar) of the input in a historical dataset, and using their observed output (i.e. the predicted volume). The Euclidean distance is typically used to assess similarity. When k nearest neighbors are found, and assuming their corresponding output values are $v_i, \mathrm{i} = 1,2, \dots, \mathrm{k}$, the predicted value (v) can be determined by calculated the weighted average of the neighbors as follows:

$$\mathrm{v} = \frac{1}{k}\sum_{i=1}^{k} v_i \tag{3-12}$$

3.3.2 Modeling Datasets

The volume datasets chosen for testing purposes in this study represent, facilities with different characteristics (e.g. an international border vs. a commuter freeway), different locations (New York, Virginia and Beijing), as well different time resolutions (hourly vs. 5-minute vs. 2-minute). Specifically, dataset 1 came from interstate I-90 in Buffalo (detector M4183E), and is an hourly volume dataset. Dataset 2 came from the Peace Bridge international border crossing connecting Western New York and Southern Ontario, and is also an hourly volume dataset. Dataset 3 came from the westbound direction of Interstate 64 in the Hampton Roads area in Virginia, and has a resolution of 5 minutes. Finally, data set 4 is from the second ring road in Beijing (detector 02024) and has a 2-minute resolution. The lengths of the time series sets utilized in the study were 2000 observations for the hourly volume sets, 5,760 for the 5-minute data set, and 3,600 for the 2-minute Beijing data set.

3.3.3 Data Diagnosis Results

Before applying the different model development, the predictability of the four data is diagnosed by using the statistical measures previously discussed. The details are shown below.

3.3.3.1 Delay Time and Embedding Dimension

The mutual information method was first used to determine the best value for the time delay, τ, for the four datasets. The results are shown in Table 3-7, which lists the Mutual Information value for values of τ ranging from 1 to 5. As can be seen, for all four datasets, the best value of τ appears to 1 since it is the value corresponding to the maximum mutual information value (Wen and Wan, 2009).

Table 3-7 Mutual Information Values With Respect to Time Delay

Datasets	Mutual Information value				
	τ =1	τ =2	τ =3	τ =4	τ =5
I-90	9.08	9.07	9.08	9.07	9.07
Peace Bridge	6.59	6.58	6.57	6.57	6.58
Virginia	2.77	2.60	2.50	2.42	2.36
Beijing	4.37	4.23	4.11	4.01	3.93

To determine the value for the embedding dimension, *m*, Figure 3-12 plots the percentage of the false nearest neighbors as a function of the embedding dimension values, for the four datasets. As can be seen, the best values for the embedding dimension appear to be equal to 7, 6, 9 and 30 for the I-90, Peace Bridge, Virginia and Beijing datasets respectively, since those values lead to the lowest percent of false nearest neighbors. Determining the embedding dimension, *m*, serves the very important role of determining the length of the input vector for the k-NN and SVR prediction methods.

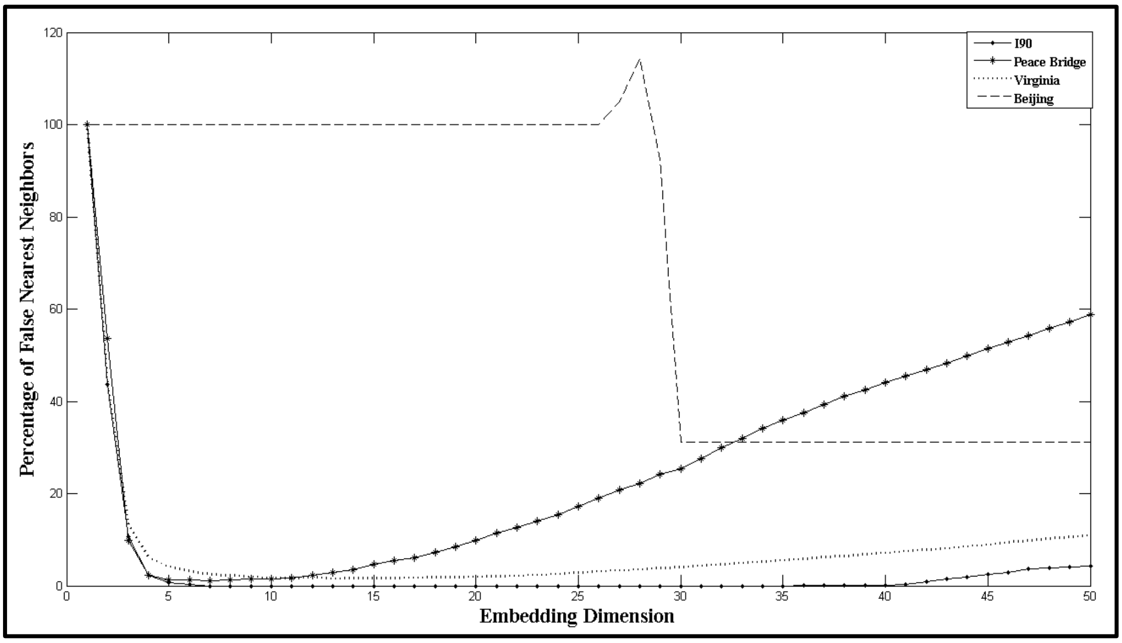


Figure 3-12 Percent of False Nearest Neighbors With Respect to Embedding Dimension

3.3.3.2 Approximate Entropy

Here, for each dataset, the time delay (τ) and embedding dimensions (m) values which had been calculated as described in the previous Chapter were used; that is to say, $\tau = 1$ for all the datasets and $m = 7$ for I-90, $m = 6$ for the Peace Bridge data, $m = 9$ for the Virginia data and $m = 30$ for the Beijing data. With these values, a new state space was built, and assuming that if the Euclidean distance between two vectors is lower than 20 percent of the standard deviation of the dataset the two patterns are similar, the approximate entropy for each dataset was calculated. The results were an approximate entropy value of 0.24 for I-90, 0.30 for the Peace Bridge, 0.21 for the

Virginia dataset, and 0.0049 for the Beijing dataset. Given that the approximate entropy is a measure of the predictability or irregularity of the dataset (with larger values indicating a harder to predict time series), it can be concluded that the Peace Bridge data is the most unpredictable with the least chance to have repetitive patterns, followed by I-90, Virginia, and finally Beijing which is easiest to predict. This is perfectly in agreement with what one should expect, since the Peace Bridge and the I-90 sets were hourly volumes, whereas the Virginia set was a 5-minute count, and the Beijing was a 2-minute count (naturally it is much harder to predict longer times in the future). Moreover, traffic at a border crossing is expected to be more irregular compared to a commuter freeway.

3.3.3.3 Time Reversibility of Surrogate Data

According to the discussion in the methodology Chapter, for each of the four datasets, 100 replicates of surrogate data were generated, and for each surrogate data set, the time reversibility value, *r*, was calculated and a distribution of *r* was generated, as can be seen in Figure 3-13. The time reversibility value of the original time series, $r_{o,}$, was also calculated (this is the location of the straight vertical line in each of the plots in Figure 3-13).

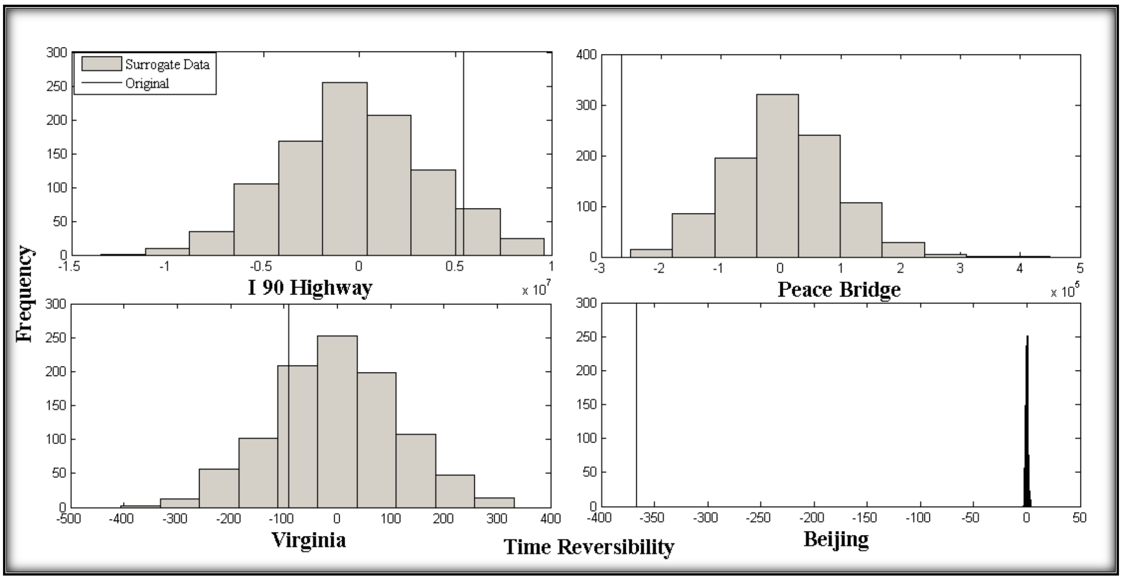


Figure 3-13 Test of Nonlinearity through Surrogate Data

As can be seen, the I-90 and the Virginia dataset appear to be exhibit linear patterns (since r_o lies within the distribution of r), whereas the Peace Bridge and the Beijing data exhibit nonlinearity.

3.3.3.4 Hurst Exponent

The calculated values for the Hurst exponent for the four datasets were as follows: Hurst$_{I\text{-}90}$ = 0.26, Hurst$_{PB}$ = 0.60, Hurst$_{Virginia}$ = 0.69 and Hurst$_{Beijing}$ = 0.91. Since, as was previously mentioned, the closer the value of the exponent to 0.5, the more random the data, the results seem to indicate that the Peace Bridge data is the most random while the Beijing data is the most stable one. Moreover, given that the Hurst value for the I-90 data is between 0 and 0.5, the set appears to exhibit a trend-reverting tendency. In contrast, the Virginia and the Beijing data exhibit a trend-reinforcing tendency (Hurst exponent within the range of 0.5 to 1).

3.3.4 Model Development

Following the characterization of each dataset, the three prediction methods (SARIMA, k-NN and SVR) were used to provide short-term forecasts for each of the four test datasets. Following the calibration of each prediction method, its performance on the different datasets is evaluated and correlated to the characteristics of the set as quantified using the data diagnosis measures described above. By doing so, the study hopes to glean useful insight into how to select the best prediction method for a dataset given its statistical characteristics. Furthermore, the insights gained from the data diagnosis measures are utilized to guide the design and calibration of the prediction methods.

3.3.4.1 SARIMA Models

The Statistical Package for Social Sciences (SPSS) was used to build the SARIMA models. An essential step in developing the SARIMA models was to determine the appropriate training data size for each dataset. For traffic volume SARIMA models, the appropriate seasonal period is one week, and therefore for the hourly data sets (i.e. the I-90 and the Peace Bridge), a seasonal period of 168 intervals was adopted (i.e. 24 x 7). For the five-minute volume and the two-minute volume data sets, and considering that the training dataset will be very large if the one-week seasonal period is assumed, we set the seasonal period to one day and attempt to predict traffic volumes for *weekdays only*. By excluding the weekends, and by assuming that weekdays are similar to one another, a seasonal period of one day would be adequate. With this assumption, the seasonal period was assumed to be equal to 288 intervals (24 x 12) for the Virginia data set, and equal to 720 intervals (24 x 30) for the Beijing data set.

When developing the models, various training dataset sizes were tested, and the values of the resulting mean absolute percent error (MAPE) was monitored (note that the size tested was always an integer multiple of the assumed seasonal period). To adapt SARIMA to on-line prediction, the training data set was updated at each step by adding the most recent observation and deleting the oldest. The results are shown in Table 3-8.

As can be seen, for the I-90 data, the prediction accuracy of SARIMA improved with the increase in the training data size, with the best performance reached with a size equal to 1,008 observations. This was also generally the trend for the other three datasets, although the best performance was achieved at slightly smaller sized training sets (i.e., 840 for the Peace Bridge, 864 for Virginia, and 720 points for Beijing). If one were to correlate these observations to the results of the statistical measures performed on the datasets, some interesting observations could be made. For example, because the Beijing data had a very low approximate entropy value (0.02), it was much easier to predict and hence required the least number of data points for training. Also, the Hurst exponent value for the Beijing dataset was 0.91 (close to 1.0) indicating a much stronger trend-reinforcing tendency compared to the Hurst value for Virginia and the Peace Bridge.

Table 3-8 Performance of SARIMA With Respect to Training Data Size

I90		PB		Virginia		Beijing	
Training Dataset	MAPE (%)	Training Dataset	MAPE (%)	Training Dataset	MAPE (%)	Training Dataset	MAPE (%)

Size		Size		Size		Size	
168	17.18	168	42.28	288	9.34	720	0.56
336	17.21	336	47.54	576	9.84		
504	7.79	504	27.31	864	6.54	1440	0.63
672	7.39	672	24.81	1152	8.80		
840	7.40	840	17.48	1440	7.58	2160	0.54
1008	6.95	1008	23.73	1728	10.31		

The prediction results also reveal important information about the performance of SARIMA. The SARIMA models perform well for the I-90 data, the Virginia data and the Beijing data, resulting in acceptable MAPEs lower than 10% in general. This was not the case for the Peace Bridge data for which the lowest MAPE was around 18 %. The inconsistent performances can be explained by the nonlinearity and less predictability of the Peace Bridge data as identified by the surrogate data and approximate entropy analyses. Theoretically speaking, SARIMA models are built on the linearity assumption, and thus may not perform well for nonlinear time series such as the Peace Bridge data.

3.3.4.2 K-NN Models

Two attributes needed to be specified for the K-NN model development, the length of the input vector (*B*), which refers to how many previous time steps are used to predict the next value, and the number of nearest neighbors (*k*). Various combinations of *B* and *k* were tested to see which one leads to the best performance for each dataset. As shown in Figure 3-14, the best (*B*, *k*) value combinations that returned the least prediction errors were (7, 3), (24, 3), (11, 1) and (30, 1) for the I-90, Peace Bridge, Virginia and Beijing data respectively. The question is now how do these values for *B* and *k* correlate with the values of the statistical data diagnosis measures calculated for the datasets?

With respect to *k*, the resulting values seem to correlate well with the values for the approximate entropy. The I-90 and Peace Bridge datasets had higher approximate entropies than the Virginia and Beijing datasets, which means that the probability that the former group of datasets will have more "different patterns" than "observed patterns" is higher than that for the latter group. As a result, more nearest neighbors may be needed to lower the risk of using a different pattern for prediction. In terms of the input vector length (*B*), the values were identical (or very close) to the embedding dimension, *m*, values, for the I-90, Virginia and Beijing datasets where the analysis indicated values for *m* equal to 7 for I-90, 9 for Virginia, and 30 for Beijing. However, this was not the case for the Peace Bridge. For the Peace Bridge, the optimal value of 24 is interesting since this can be easily explained by the strong seasonality of the data (i.e. the periodical variations of hourly volumes within the 24-hour cycle).

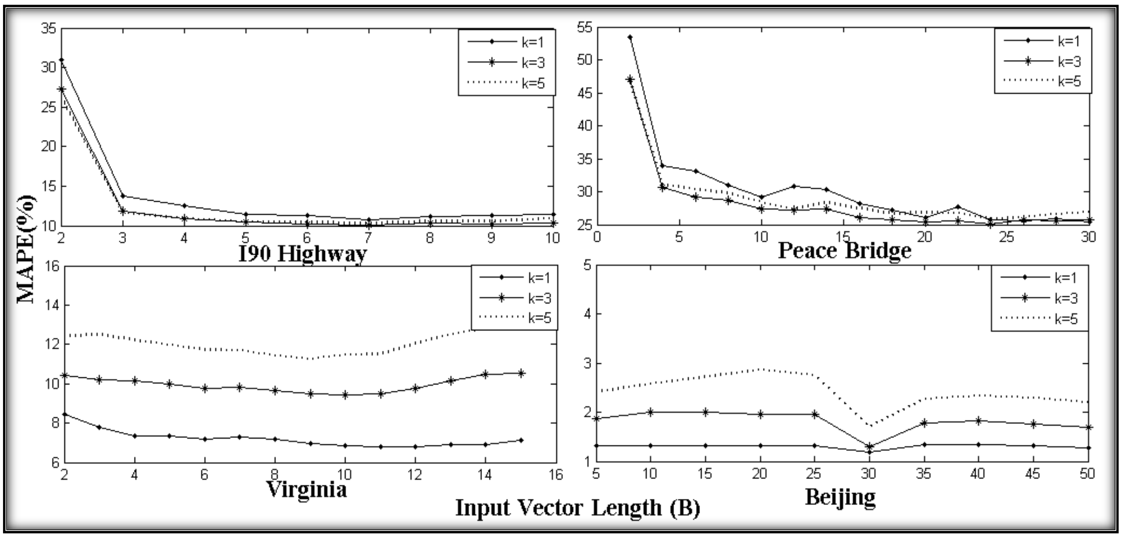


Figure 3-14 Performance of k-NN With Respect to Input Data Vector Length (*B*) and the Number of Nearest Neighbors (*k*)

3.3.4.3 SVR Models

The development of the SVR models required the specifications of the training data size (*O*) and input data vector length (*D*). The simplest way to jointly determine the values of the two parameters is enumeration and testing. For simplicity, we show here only partial results where the value of one parameter is fixed, and the other is varied and the impact of the variation on the MAPE is monitored. For the selection of the other parameters for SVR model, a detailed discussion can be found in Chapter 3.1.3.3.

With the value of the input vector initially fixed at 6 for all four datasets, the performance of SVR with respect to various training data sizes is shown in Table 3-9. The best training data sizes for the first two datasets appears to be around 600 data points. For Virginia data, the lowest MAPE is attained when the training data size is equal to 300. For Beijing data, there is a significant decrease in MAPE when the training data size is increased from 300 to 400, with the lowest MAPE achieved with data size equal to is 700. This shows that for SVR, the appropriate moving training dataset length is not really correlated to the Hurst Exponent values as was the case with the SARIMA model. This is because SVR does not consider the autocorrelation in the time dimension; instead, it considers the few “support vectors” to formulate the function.

Table 3-9 Performance of SVR With Respect to Training Data Size

MAPE	Training Data Size									
(%)	100	200	300	400	500	600	700	800	900	1000

I-90	19.77	19.01	18.75	18.47	18.32	18.22	19.32	19.68	21.74	22.73
PB	14.83	11.17	10.49	9.90	9.02	8.37	8.49	9.47	11.59	13.66
Virginia	37.58	29.7	13.58	13.80	17.22	17.11	17.33	18.20	19.03	20.41
Beijing	8.66	5.50	10.99	1.00	0.24	0.17	0.01	0.01	0.01	0.01

Next, with the training data size fixed at either 100 or 500 data-points, the length of the input vector was varied and the corresponding MAPEs were calculated. As shown in Figure 3-15, the vector length D affects the performance of SVR only for the Peace Bridge dataset. For the I-90, Virginia and Beijing datasets, the resulting MAPEs remain almost the same when the vector length varies from 2 to 10.

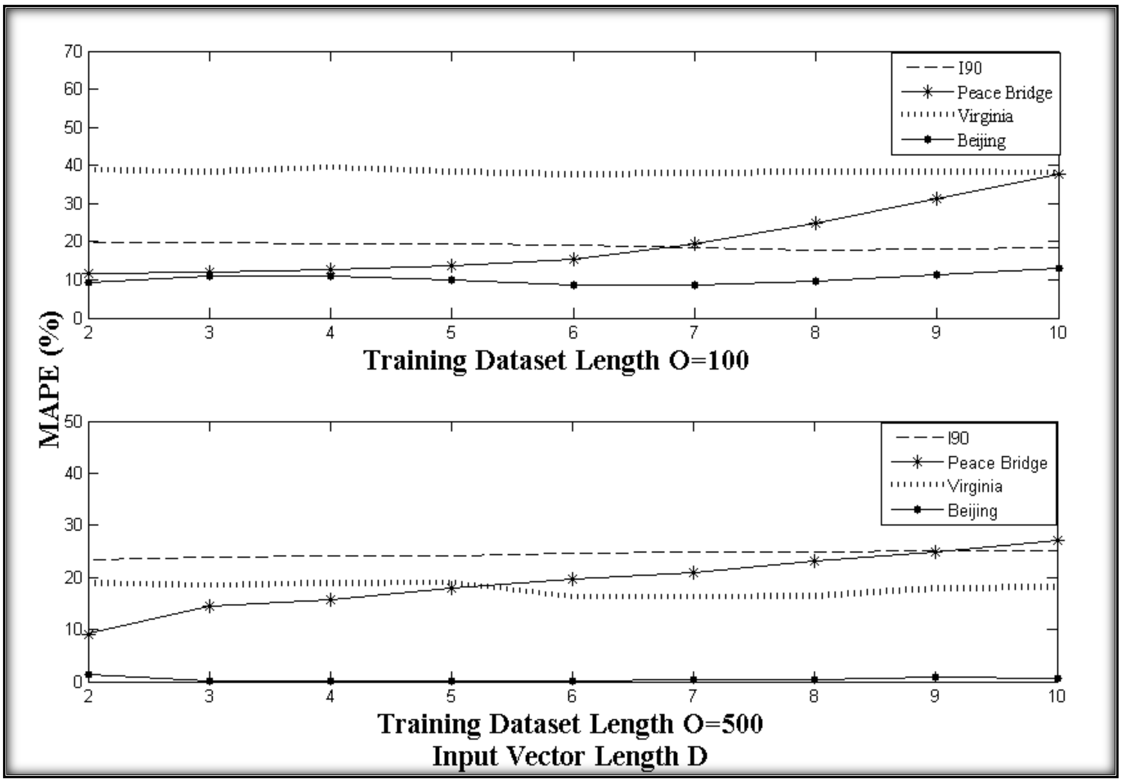


Figure 3-15 Performance of SVR With Respect to Input Data Vector Length D

3.3.4.4 Comparison of Model Performance

The best models calibrated in the previous Chapter were then applied to the four datasets. The results are shown in Table 3-10, along with the results from a naïve prediction model that basically uses the value of the observed volume at the current time step as the predicted value for the next step, and which is used for benchmarking. The table also lists the values of the data diagnosis measures for each dataset to give readers a clear view of the relationship between data characteristics and model selection.

Table 3-10 Comparisons of Three Prediction Models for Four Datasets

Datasets	Predictability			Performances (MAPE (%))			
	Complexity (AppEn)	Nonlinearity (SurroData)	LRD (HurstEn)	SARIMA	k-NN	SVR	Naïve Model
I-90	0.24	No	0.26	6.95	10.03	18.22	23.51
PB	0.30	Yes	0.60	17.48	25.13	8.37	42.34
Virginia	0.21	No	0.69	6.54	6.81	13.58	15.23
Beijing	0.0049	Yes	0.91	0.54	1.18	0.01	14.59

From Table 3-10, we can see firstly that, all the three models perform better than the Naïve model. Secondly, for all four datasets, SARIMA performs slightly better than k-NN; however, the advantage of k-NN is its low computational cost and ease of implementation. Thirdly, SARIMA and k-NN work much better for I-90 and Virginia traffic volume datasets than SVR, while SVR performs the best for Peace Bridge and Beijing traffic volume datasets. This shows that SARIMA and k-NN are more appropriate for linear datasets and SVR is definitely a good choice for nonlinear datasets. Another observation is that for both linear datasets and nonlinear datasets, the larger the approximate entropy is, the higher the MAPE is.

3.4 Conclusions

This chapter discusses the three pieces of work for the short-term traffic volume prediction model, which is the first step of the two-step border crossing delay prediction model.

In Chapter 3.1, a multi-model combined forecasting method, combining forecasts from SARIMA and SVR, was proposed and used for the on-line prediction of hourly traffic volumes at the Peace Bridge in Western New York. By combining traditional time series analysis with SVR, the study managed to take advantage of the known strengths of time series analysis, while compensating for its weaknesses in capturing the non-linear aspects of the data, a phenomenon which AI-based methods such as SVR are known to be capable of capturing. Among the main conclusions and lessons learnt from the study are:

(1) The accuracy of border crossing short-term volume forecasts can be improved by classifying the volume data into groups and developing separate prediction model for each group. In this study, a convenient classification scheme involved dividing days into the following six groups: weekdays excluding Fridays, Fridays, Saturdays, Sundays, game days, and holidays.

(2) The SVR method appears to outperform SARIMA when predicting traffic volumes on special days (e.g. game-days), whereas SARIMA performs better than SVR during normal days (e.g. the weekday group). This is consistent with the known strengths and weakness of the two methods, namely the ability

of SARIMA, as a linear modeling approach to capture seasonality and trend, versus SVR superior ability to capture nonlinear effects.

(3) While the performance of both SARIMA and SVR methods on the border crossing traffic volume prediction problem appears acceptable, the accuracy is significantly improved by combining the forecasts from the two methods as demonstrated by the results shown in Table 3-3 and Figure 3-3.

(4) For combining the SARIMA and SVR forecasts, the Fuzzy Adaptive Variable Weight method appears to outperform the simple fixed weight method as indicated by Table 3-3.

Chapter 3.2 has developed an enhanced SPN which applies the DTW method to assess similarity among traffic volume data. The enhanced SPN was then used to predict hourly traffic volumes at the Peace Bridge international border crossing. The performance of the enhanced SPN (i.e. DTW-SPN) was then compared to three other forecasting methods, namely the original SPN algorithm (Euclidean-SPN) described in Huang and Sadek (2009), SARIMA, and SVR. When developing and comparing the models' performance, the study considered two cases, classifying the dataset into groups by day type and using the original dataset without classification. Among the main conclusions and lessons learnt from the study are:

(1) When the dataset was divided into day groups, DTW-SPN yielded the lowest MAPE for all data groups with the exception of the game day group where

SVR performed slightly best. The DTW-SPN also had the overall best performance when the MAPE was averaged over all the groups.

(2) DTW-SPN also performed the best when the whole dataset was used (i.e. the data was not broken into groups), and once again had the lowest MAPE. This demonstrates the robustness of the method and its ability to handle non-homogeneous time series to some extent.

(3) Euclidean-SPN did not perform very well on the Peace Bridge dataset. One reason could be that its parameters were not re-calibrated for the different data groups. On the other hand, Euclidean-SPN was the most computationally efficient and required a fraction of the processing time needed by SARIMA or SVR.

(4) DTW-SPN also appears to be significantly more computationally efficient compared to SARIMA or SVR. Specifically, for the case study considered, the total running time for DTW-SPN was about one quarter of the time for SARIMA and 1/15th the time for SVR.

(5) DTW-SPN and SVR are capable of capturing abrupt changes of hourly traffic while Euclidean-SPN and SARIMA performed poorly for these abrupt points. This implies the robustness of DTW-SPN along with SVR.

In terms of being "application ready", when comparing with SARIMA and SVR, our study shows that the SPN models appear to have several advantages such as computational efficiency, the ability to handle non-classified data sets, and not requiring training procedures. This means the SPN models can provide an efficient, straightforward and transferable method for short-term traffic volume prediction.

In Chapter 3.3, the predictability of four traffic volume datasets was first assessed using complexity, nonlinearity and long range dependency tests. Three prediction models, SARIMA, k-NN and SVR were then calibrated and tested for each of the datasets. The performances of the different prediction methods were then correlated to the results from the data diagnosis measures, hence providing some guidelines on how to choose the appropriate prediction method, and set its parameters, given the statistical characteristics of a given dataset. Among the main conclusions learned from the study are:

With respect to prediction model choice,

(1) For all the four datasets, SARIMA performs slightly better than k-NN, but k-NN has a faster running speed;

(2) For linear datasets, SARIMA and k-NN are more appropriate. For nonlinear datasets, SVR works better than SARIMA and k-NN; and

(3) The larger the approximate entropy is, the higher the MAPE is.

With respect to model's parameter setting,

(1) For SARIMA model, according to Hurst Exponent, Peace Bridge dataset has the weakest LRD, followed by Virginia dataset and then I-90 dataset, and we find the best training dataset length for Peace Bridge is 840, and for Virginia, it is 864, for I90, it becomes 1008. This shows that for SARIMA, a weaker LRD

indicates a smaller training dataset length. For datasets like Beijing with very small approximate entropy, the training dataset length can also be set small;

(2) For k-NN model, the vector lengths corresponding to the lowest MAPEs are consistent with the results of the delay time and embedding dimension analysis. This was true for the I-90, Virginia, and Beijing datasets. For the Peace Bridge, the best vector length is 24, because of its strong seasonality;

(3) For K-NN, the number of nearest neighbors, k, should be set higher for datasets with higher approximate entropies;

For SVR model, the setting of training dataset length does not appear to be sensitive to the Hurst Exponent value. There also does not appear to be a strong relationship between the delay time and embedding dimension and the vector length, with the best value of the input vector length varying with changes in the training dataset length.

CHAPTER 4 SOLUTIONS OF TRANSIENT MULTI-SERVER QUEUEING MODELS

This Chapter mainly focuses on the second step of the border crossing delay prediction model. With the predicted traffic as an input, the queueing models can be used to calculate the future border crossing waiting time. Chapter 4.1 introduces the real-world case study of Peace Bridge, how the data are collected and the queueing model $M/E_k/n$ and BMAP/PH/n are decided; Chapter 4.2 talks about the heuristic solutions for the two types of queueing models. Chapter 4.3 compares the results from the queueing model with the VISSIM traffic simulation model, some sensitivity analysis and optimization work are also introduced.

4.1 The Peace Bridge Case Study

As a case study for the development and solution of the border crossing delay prediction queueing models, this research considers the Peace Bridge, one of the four main border crossings connecting Southern Ontario, Canada and Western New York, US. The Peace Bridge carries an estimated 4.76 million cars on the annual basis, and is thus one of the busiest border crossings between the U.S. and Canada. When traveling through the bridge, traveling vehicles need to wait in line and go through security inspection before they can head to their destinations. The whole process can thus be considered as a queueing process with vehicle arrivals as the input flows, the inspection checkpoints as the service stations, and the queue length and waiting times as the performance indicators.

4.1.1 Estimation of Arrival and Service Process Distributions - Model 1

Information was collected regarding vehicle arrival patterns and security inspection processes in Peace Bridge in order to determine the appropriate distributions and the correct queueing model to use. Specifically, 700 observations of the vehicle inter-arrival times and 571 observations of the service times (i.e. inspection time) were collected from December 19, 2011 to January 10, 2012. The data were then used to define the appropriate probability distributions that best describe the arrival and service processes. As shown in Figure 4-1, the distribution of headways (i.e. the inter-arrival times) was matched best by the exponential distribution $f(x) = \exp(-x/9.63)/9.63, x \geq 0$ with a mean value of 9.63 seconds (for the first model, batch arrivals were ignored). The R-square for fitting that curve to the collected inter-arrival time data was 0.8721, and the Root Mean Square Error (RMSE) was 0.0064.

For the service times, the best fitting distribution was found to be the Erlang distribution $f(x) = x * \exp(-x/22.29)/22.29^2$with order 2 and mean of 44.58 seconds, as shown in Figure 4-2. Fitting that curve to the collected service time data points resulted in an R-square value of 0.8903, and RMSE of 0.009906.

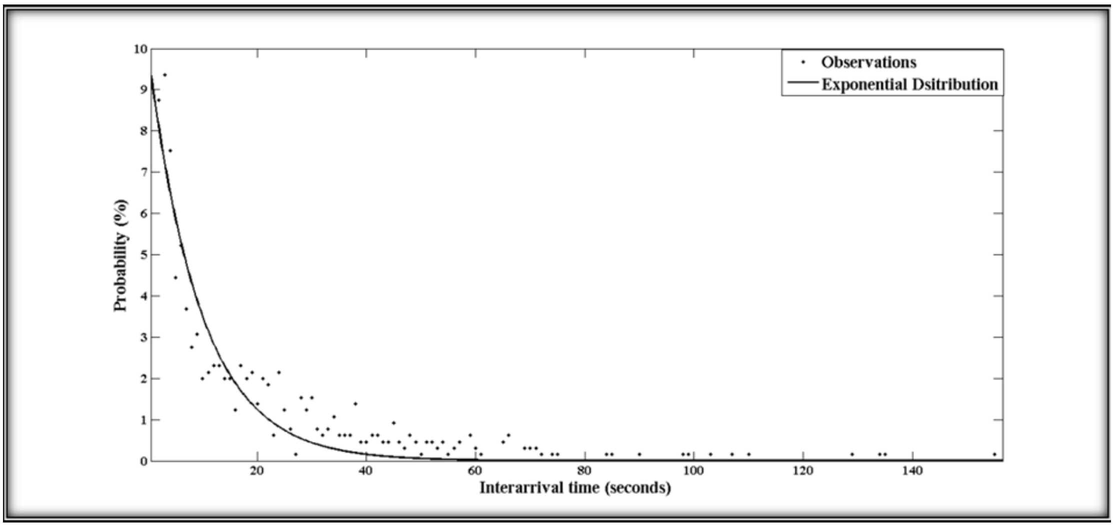


Figure 4-1 Inter-arrival Time Distribution

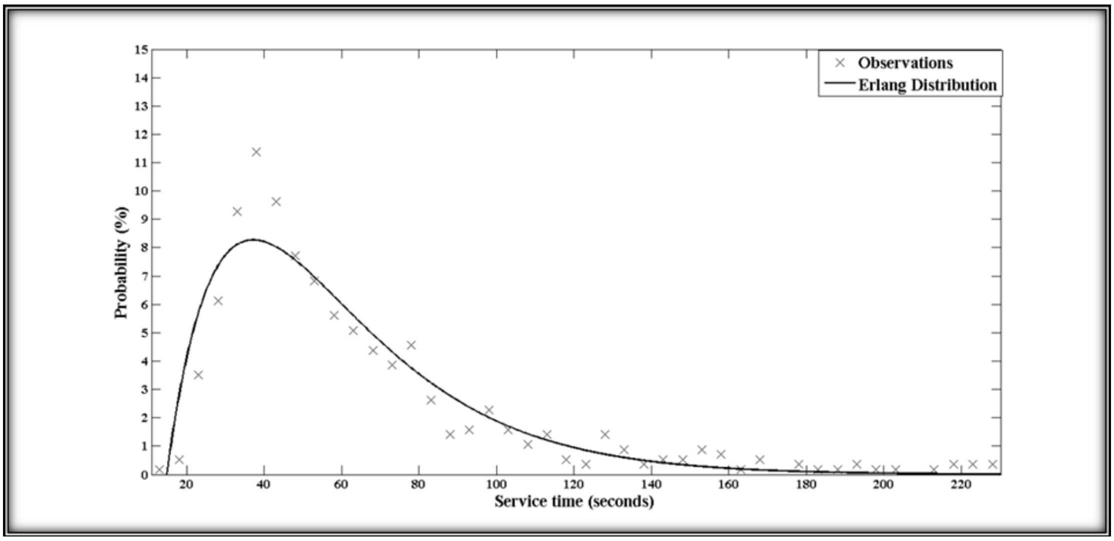


Figure 4-2 Service Time Distribution

4.1.2 Estimation of Arrival and Service Process Distributions - Model 2

In reality, the arrival process for transportation systems may not always be captured by an exponential distribution. In many cases, the arrival process may be the result of combining multiple streams with different exponential distributions. Moreover, it is quite common in the real-world for several vehicles to arrive simultaneously at the queueing system. Moreover, the service process could be more complicated than a simple Erlang distribution. To represent these complex arrival and service patterns, a more general modeling framework is to represent the system by a Batch Markov Arrival Process (BMAP) and a Phase Type (PH) distributed service process, as described below.

4.1.2.1 Batch Markov Arrival Process

Batch Markov Arrival Processes (BMAP) were introduced by Neuts (1979) in order to extend the standard Poisson process to account for more complex customer arrival processes in queueing systems. Let *J* be an irreducible, continuous-time Markov chain with finite state space $E = \{1, 2, \ldots, e\}$, where e is a finite, positive integer. Suppose J has just entered state i, $1 \leq i \leq e$, the process spends an exponential distributed amount of time in state i with mean λ_i^{-1} (Cordeiro & Kharoufeh, 2010). Let $\pi = \{\pi_1, \ldots \pi_i, \ldots \pi_e\}$ denote the probability distribution of entering state i for the process J, p_{ij} be the probability for the system to switch from state *i* to state j, $1 \leq \mathrm{j} \leq \mathrm{m}$ (j may be equal to i), and $\varepsilon_{b,i,j}$ represent the probability for *b* vehicles to arrive in batch during the system's transition from state *i* to state *j* (Daikoku et al. 2007). Obviously, the following conditions should exist:

$\sum_{1 \leq i \leq e} \pi_i = 1,$ (4-1)

$\sum_{1 \leq j \leq e} p_{i,j} = 1,$ (4-2)

$\sum_{b=0}^{B} \varepsilon_{b,i,j} = 1$, where B is the maximum batch size. (4-3)

Without loss of generality, we assume $\varepsilon_{0,i,i} = 0,\ 1 \leq i \leq e$. Let C and D_b $(1 \leq \mathrm{b} \leq \mathrm{B})$ be $e \times e$ matrices. C contains the transition rates of *J* for which no arrivals occur, and D_b $(1 \leq \mathrm{b} \leq \mathrm{B})$ contains the transition rates for which a batch size b occurs (Cordeiro & Kharoufeh, 2010). The $(i,j)^{th}$ elements $C_{i,j}$ and $D_{b,i,j}$ in C and D_b are given as below:

$$C_{i,j} = \begin{cases} -\lambda_i, & if\ i = j \\ \lambda_i p_{i,j} \varepsilon_{0,i,j}, & otherwise \end{cases}, \tag{4-4}$$

$$D_{b,i,j} = \lambda_i p_{i,j} \varepsilon_{b,i,j}, \tag{4-5}$$

So at last, the BMAP can be characterized by a set of $e \times e$ matrices $(\mathrm{C}, D_1, D_2 \ldots, D_B)$. And, $D = \sum_{1 \leq b \leq B} D_b \neq \mathbf{0}$, which means there must be some vehicles arriving.

As mentioned before, the Peace Bridge data included a total 700 observations of inter-arrival times. Out of those, 48 vehicles were observed to arrive in batch; unfortunately the study team did not record the exact size of batch for those 48 vehicles and the frequency at which this batch process occurred. For demonstrating our procedure therefore, we assume there are twelve batches with three vehicles and six batches with two vehicles and assign a random interval to each batch. In order to estimate BMAP (i.e. estimate parameters such as π, C and D_b (1≤b≤B)), an expectation and maximization (EM) algorithm proposed by Breuer (2002) is used. The algorithm converged after four iterations, yielding the following estimates for BMAP's parameters:

$$\pi = [0.7741, 0.2259],$$

$$\mathrm{C} = \begin{bmatrix} -0.1063 & 0.0776 \\ 0.2694 & -0.3558 \end{bmatrix},$$

$$D_1 = \begin{bmatrix} 0.0046 & 0.0222 \\ 0.0727 & 0.0078 \end{bmatrix},$$

$D_2 = \begin{bmatrix} 0.0001 & 0.0005 \\ 0.0016 & 0.0001 \end{bmatrix}$,

$D_3 = \begin{bmatrix} 0.0002 & 0.0011 \\ 0.0037 & 0.0005 \end{bmatrix}$,

Based on this, $p_{i,j}$ and $\varepsilon_{b,i,j}$, $1 \le i,j \le e$ can be calculated and used to determine the arrival intervals in our approximation/simulation approach for deriving the transient solution of the queueing described later. Specifically, we can first determine the initial state i by sampling from the inverse cumulative function of π. The next state of BMAP j and the batch size of arrival b in the state transition can also be sampled in the same way based on $p_{i,j}$ and $\varepsilon_{b,i,j}$. Moreover, the inter-arrival time interval in this transition can be determined by sampling from the exponential distribution with rate λ_i or using its mean value. When the system reaches state j, we continue to sample the next state, the arrival size and time interval. This procedure is repeated until the end of the prediction time horizon is reached.

4.1.2.2 Phase Type Distribution for Service Process

Consider a Markov process J on a finite state space $(0,1,\dots,p)$ where 0 is absorbing and the other p states are transient (Asmussen et al. 1996), a phase type (PH) distribution with parameter (π, A) is the distribution of the time until absorption into state 0 in this Markov process. π is the initial probability distribution of state, and it can be defined as $[\alpha_0\ \boldsymbol{\alpha_p}]$, where α_0 is the probability of starting the process at absorbing state 0, and $\boldsymbol{\alpha_p}$ is a $1 \times p$ vector containing the probabilities of starting at the other p states. Obviously, $\alpha_0 = 1 - \boldsymbol{\alpha_p}\mathbf{1}$, where $\mathbf{1}$ is a $m \times 1$ vector with all elements as 1. The $p \times p$ dimensional matrix A is called the phase-type generator (Asmussen et al. 1996). The $(i,j)^{th}$ elements $A_{i,j}$ are given as:

$$A_{i,j} = \begin{cases} -\sum_{k=0,k\neq i}^{p} \lambda_{ik}, & if\ i=j \\ \lambda_{ij}, & otherwise \end{cases}, \tag{4-6}$$

where λ_{ij} is the rate parameter of the exponential distribution, capturing the time that the Markov process spends at state i before it goes to state j.

With this, the infinitesimal generator of this process can be written as:

$$\mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0} \\ a & \boldsymbol{A} \end{bmatrix}, \tag{4-7}$$

where $a = -\boldsymbol{A}\mathbf{1}.$ Here, $\mathbf{1}$ is a $p \times 1$ vector with all elements as 1.

In this study, the KPC-Toolbox introduced by Casale et al. (2008), which uses the method of moment matching (Bobbio et al. 2005), was used to estimate the PH distribution of the observed service time for the Peace Bridge case study. The estimation results of π, A and Q are shown below:

$\pi = [0\ 0.0163\ 0.9837\ 0]$,

$$\boldsymbol{A} = \begin{bmatrix} -0.0126 & 0.0126 & 0 \\ 0 & -0.0488 & 0.0488 \\ 0 & 0 & -0.0488 \end{bmatrix},$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -0.0126 & 0.0126 & 0 \\ 0 & 0 & -0.0488 & 0.0488 \\ 0.0488 & 0 & 0 & -0.0488 \end{bmatrix},$$

In matrix Q, since $\lambda_{(i-1)(j-1)} = Q_{i,j}$, a positive value of $Q_{i,j}$ implies the *existence* of a transition from state $(i-1)$ to state $(j-1)$. Based on the values of matrix Q therefore, the PH distribution of the border service process for this study can be represented as in Figure 4-3 and is explained as follows.

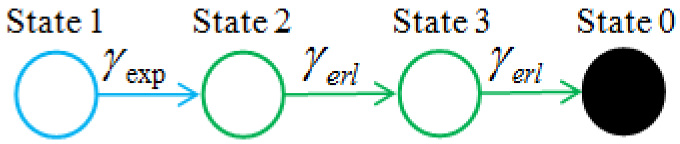


Figure 4-3 PH Distribution of the Border Service Process

As can be seen from Figure 4-3, the PH distribution for the border crossing case is a mixture of Exponential (with the subscript "exp") and Erlang (with the subscript "erl") distributions. There are also two types of service processes. The first service process starts at state 1 with a probability of 0.0163. After experiencing the service time that follows an Exponential distribution with a mean of 1/0.0126=79.36 seconds, it transitions to state 2, and then goes through state 3 before it arrives at state 0. When the transition from state 2 to state 0 occurs, the total service time is the summation of the two exponentially distributed service times, and thus follows an Erlang distribution with order $k = 2$ and mean of 2/0.0488=40.98 seconds. The second service process starts at state 2 with a probability of 0.9837(i.e., this process is much more likely), and the total service time from state 2 to state 0 follows the same Erlang distribution as in the first process.

4.2 Methodology

The methodology followed in this study can be viewed as consisting of two major steps: (1) queueing model development and solution; and (2) model validation. As previously mentioned, two groups of models were considered; first, the special case of an $M/E_k/n$ queueing model and then the more generic case of a *BMAP/PH/n* model. The transient solution for such models was derived using an approximation/heuristic approach. Specifically for the $M/E_k/n$ model, a slightly modified version of the Equal Likely Combination (ELC) heuristic of Escobar et al (2002), which reduces the computational burden, was utilized, whereas for the *BMAP/PH/n,* this study introduced a new heuristic (or assumption) which we call the Equally Likely Vehicle (ELV). In order to validate the model results empirically, the queue length and delay estimates derived from the queueing model solution were compared to those estimated from multiple runs of a detailed microscopic traffic simulation model. The details are shared below.

4.2.1 $M/E_k/n$ queueing models

As pointed out by Escobar et al. (2002), the exact solution of multi-server queueing models with exponential inter-arrival times and Erlangian service times ($M/E_k/n$) is quite challenging due to: (1) the very large number of possible system states, especially for systems with a large number of serves and/or Erlang distribution with high orders (i.e. high values of k); and (2) the complexity of the state transitions. To address these challenges, a handful of previous studies have proposed approximate solution methods to solve $M/E_k/n$ queueing model, utilizing ideas to simplify the size of the problem's state space. Among those approximation method is the ELC heuristic (Escobar et al., 2002) that is applied to this study with some modifications to better suit the problem at hand. A brief description of how the solution proceeds is given below. While the details of the formulation and solution procedure can be found in Escobar et al. (2002), enough details are included herein to allow the reader to follow the presentation.

4.2.1.1 System state description

The key step in deriving either the steady state or the transient solution of an $M/E_k/n$ queueing model is to transform the complex Erlang distributed service process to a simpler version. Since an Erlang distribution with an integer order *k* is equivalent to a sum of *k* independent exponential distributions, a service station with k^{th}-order Erlang distributed service times can be replaced by a chain of *k* service stations with exponentially distributed service times. With this, the process for a vehicle to go through an Erlang service station becomes equivalent to the process of passing through a sequence of *k* Exponential service stations.

Escobar et al. (2002) proposed a compact way to represent the system state. According to them, the system state can be represented by a three element descriptor *(l, m, r)*, where *l* refers to the remaining number of exponential service stations that need to be completed for the vehicles currently in the system (this will be referred as stages thereafter), *m* is the number of vehicles in the system, and *r* is what the researchers called the "pattern identifier". The pattern identifier is needed because there could be different instances or combinations where one would have *m* vehicles in the system, and *l* remaining stages or service stations. For example, in a queueing system with three Erlang service stations with order 3, the state (*6, 3*) can represent either of the two patterns shown in Figure 4-4. As can be seen, both patterns have three vehicles in the system. In pattern 1, one vehicle has just arrived at station 1, and the second vehicle has completed one stage of service in station 2, while the third vehicle has completed two stages in station 3. On the other hand, all three vehicles have completed one stage of service in Pattern 2. Finding the number of patterns associated with a given *(l, m)* combination amounts to solving the problem of finding *m* integers whose sum is *l*. This can be solved by writing a simple computer code or by enumeration.

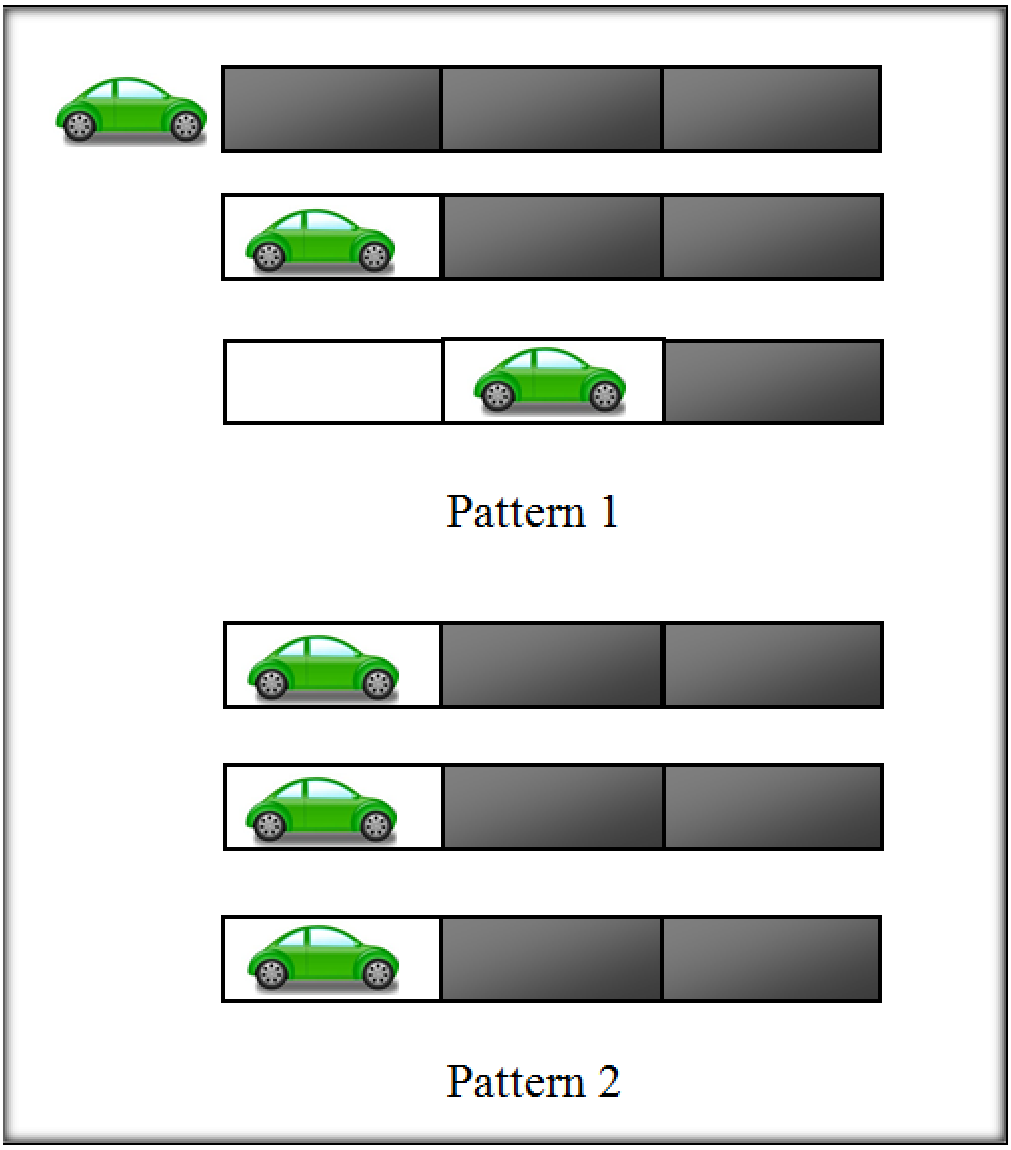


Figure 4-4 an Example for Stages, Patterns and States

4.2.1.2 State transition probabilities

Following the representation of the system states of a queueing model, the next step is to derive the state transition probabilities (herein we use $P_{l,m}$ to denote the probability of a given state *(l, m)* and we drop the pattern identifier index from the notation for simplicity). As previously, we calculate the state transition probabilities in this study on the basis of the equally likely combinations (ELC) heuristic method proposed by Marcos Escobar *et al.* (2002), which has been shown to be capable of simplifying the state transition calculation process, while maintaining the precision of the queueing model solution. While our solution approach is largely based on the framework proposed by Escobar et al. (2002), we introduce a slight modification to the solution algorithm which makes it more efficient. Specifically, the modification introduced involves updating the number of vehicles in the queueing system, when considering the vehicles' arrival process, only when a new vehicle joins the queue, and because we are considering the transient solution, we determine the time when a new vehicle arrives at the system by randomly sampling from the inter-arrival time distribution curve (or by simply using the mean value of the inter-arrival distribution curve). This is slightly different from the original ELC heuristic where the arrival process is considered at every time step. The modification introduced helps reduce the state space that needs to be considered, and thus increases the computational efficiency of the algorithm. Table 4-1 defines the variables and parameters that are used in the following Chapters.

Table 4-1 Notation for Queueing Models

Symbol	Definition
$\boldsymbol{l}$	the number of unfinished stages;
$\boldsymbol{M}$	the number of vehicles in the queueing system;
$\boldsymbol{N}$	the number of service stations in the queueing system;
$\boldsymbol{k}$	the order of Erlang distribution, or the number of stages added to the system once a vehicle joins the queue;
$\boldsymbol{t_s}$	the inter-arrival time sampled from the inverse cumulative function of an Exponential distribution;
$\boldsymbol{t_{av}}$	the average inter-arrival time of an Exponential distribution;
$\boldsymbol{N}$	the number of vehicles that have joined the queue;
$\boldsymbol{t_N}$	the time point when the N^{th} vehicle joins the queue;
$\boldsymbol{P_{l,m}}$	the probability of state (l, m);
$\boldsymbol{P_{(l_1,m_1)\to(l_2,m_2)}}$	the transition probability from state (l_1, m_1) to state (l_2, m_2);
$\boldsymbol{c_i}$	the different combinations for pattern i;
$\boldsymbol{C_{total\,(l,m)}}$	the total number of combinations producing a same state (l, m);
$\boldsymbol{s_{1,i}}$	the number of servers where only one stage is left for the vehicle in pattern i;
$\boldsymbol{D(l,m)}$	the set of patterns that satisfy state(l, m);
$\boldsymbol{\alpha_{l,m}}$	the probability that one more stage of service is conducted and one vehicle leaves from the system at state (l, m);
$\boldsymbol{\beta_{l,m}}$	the probability that one more stage of service is conducted but no

	vehicle in the system is completely served at state (l, m);
$\boldsymbol{\lambda}(\boldsymbol{t})$	the arrival rate at time *t*;
$\boldsymbol{\gamma}(\boldsymbol{t})$	the service rate at time *t*;
$\boldsymbol{\lambda}$	the average arrival rate of the Exponential distribution;
$\boldsymbol{\mu}$	the average service time of the Erlang distribution;
$\boldsymbol{\gamma}$	the average service rate of the Erlang distribution;

Now suppose that we are currently at time point, t, and that there are m vehicles and l unfinished stages in the system at that time step. For the next time step $(t+1)$, there may exist three possible types of state transition scenarios, as described below:

4.2.1.2.1 Scenario 1: one vehicle joins the queue

Let us assume first that the inter-arrival time period for vehicle $(N+1)$ has been determined, either by sampling from the inverse cumulative function of the inter-arrival exponential distribution or by just using the mean value for that distribution as previously mentioned, and it is denoted by t_s (for the sampled value) or by t_{av} for the average value. Also let t_N denote the time at which the N^{th} vehicle had joined the queue. Now if the next time step, $t+1$, which is equal to $t_N + t_s$ (or $t_N + t_{av}$ if we are using the average value), the state of the queueing system would transition from state (l, m) to state $(l+k, m+1)$ since the new arrival adds k unfinished stages and one more vehicle to the system . Here, k is the order of the Erlang distribution or the number of stages that need be finished for one vehicle to be fully served. In this case, the transition probability $P_{(l,m)\to(l+k,m+1)}$ should be 1. For other state transition scenarios mentioned later, $P_{(l,m)\to(l+k,m+1)}$ should be equal to 0, revealing that, without a new arrival, the transition from state (l,m) to state $(l+k,m+1)$ is impossible.

4.2.1.2.2 Scenario 2: one vehicle finishes its last stage of service and leaves the queue

Intuitively, the transition probability of this case can be represented as $P_{(l,m)\to(l-1,m-1)}$, and is calculated as follows. As was previously discussed, a given state may concern multiple patterns, *r*, and each pattern i may involve multiple combinations of servers' stages. For example, to produce pattern 1 shown on Figure 4-4, we could have one server with three stages left, a second with two stages left, and a third with one stage left. Moreover, the specific station server with one, two or three stages left may vary (e.g. server number one may have three stages left, or two or just one). the number of different combinations for resulting in a given pattern can be calculated as:

$$c_i = p!/(p_1!\,p_2!\,...\,p_x!) \tag{4-8}$$

where,

$p = min\{m,n\}$ is the number of active or busy service stations;

x is the largest number of unfinished stages for a server;

p_j is the number of servers with equal number of unfinished stages j, $j = 1, 2, ..., x$.

Meanwhile, when relating patterns to system states, the total number of combinations producing a same state (l, m) can be calculated as:

$$C_{total\ (l,m)} = \sum_{i \in D(l,m)} c_i \tag{4-9}$$

where,

D(l, m) is the set of the patterns that are associated with state(l, m).

To illustrate, consider once again the example shown in Figure 4-4. For pattern 1, there are three different numbers of unfinished stages for server (i.e. 1, 2 and 3 stages). Among them, the largest number of unfinished stages is 3, which means that $x = 3$. Since each number of unfinished stages corresponds to one server, $p_1 = 1$, $p_2 = 1$, and $p_3 = 1$. Based on this, by using Equation (4-8), the number of combinations for pattern 1, c_1, is equal to

$$c_1 = 3!/(1\,!\,1\,!\,1!) = 6$$

Similarly, for pattern 2, $x = 2$ and $p_2 = 3$ while $p_1 = 0$ since all the servers have two unfinished stages. Hence, the number of combinations for pattern 1 is equal to

$$c_2 = 3!/0\,!\,3\,! \quad = 1$$

Since there are only two patterns associated with state *(6, 3),* the total number of combinations for the state is $C_{total(6,3)} = c_1 + c_2 = 7$.

The basic assumption of the ELC method is that all the possible combinations in $C_{total\ (l,m)}$ are equally likely. Given this, the probability that one more stage of service is conducted and one vehicle leaves from the system can be calculated by:

$$\alpha_{l,m} = \sum_{i \in D} P_i * P_{1,i} = \sum_{i \in D} (C_i / C_{total\ (l,m)}) * (S_{1,i}/p) = \sum_{i \in D} S_{1,i} C_i / p C_{total\ (l,m)}, \tag{4-10}$$

where,

$P_i = C_i/C_{total\,(l,m)}$ is the probability of pattern i;

$P_{1,i} = S_{1,i}/p$ is the probability of having servers with only one stage unfinished in pattern i, and $S_{1,i}$ is the number of servers with only one stage unfinished in pattern i;

$p = min\{m, n\}$ is the number of active service stations;

$D(l, m)$ is the set of the patterns that satisfy state(l, m).

Finally, suppose $\gamma(t)$ is the service rate that is sampled for each time step, t, according to an Erlang distribution. Because the Erlang distribution with order k can be considered as a series of k consecutive, exponentially distributed tasks, the service rate for each stage is $k\gamma(t)$, with a corresponding service time of $1/k\gamma(t)$. Given the distribution, for each time step (i.e. from the current time step t to the next time step $t + 1$), the transition probability from state (l, m) to state $(l - 1, m - 1)$ can be calculated as follows:

$$P_{(l,m)\to(l-1,m-1)} = \alpha_{l,m} * \ k\gamma(t) * p, \tag{4-11}$$

where $p = \min(m, n)$ is the number of active service stations.

4.2.1.2.3 Scenario 3: one vehicle finishes one stage of service but still needs to stay in the queue for the other service stages

The probability of this can be represented as $P_{(l,m)\to(l-1,m)}$. Given the value of $\alpha_{l,m}$ (the probability that one more stage of service is conducted and one vehicle leaves from the system) was calculated as in Equation (4-10) above, the probability that one stage is finished while no vehicles leave the queue, $\beta_{l,m}$, can be calculated simply as:

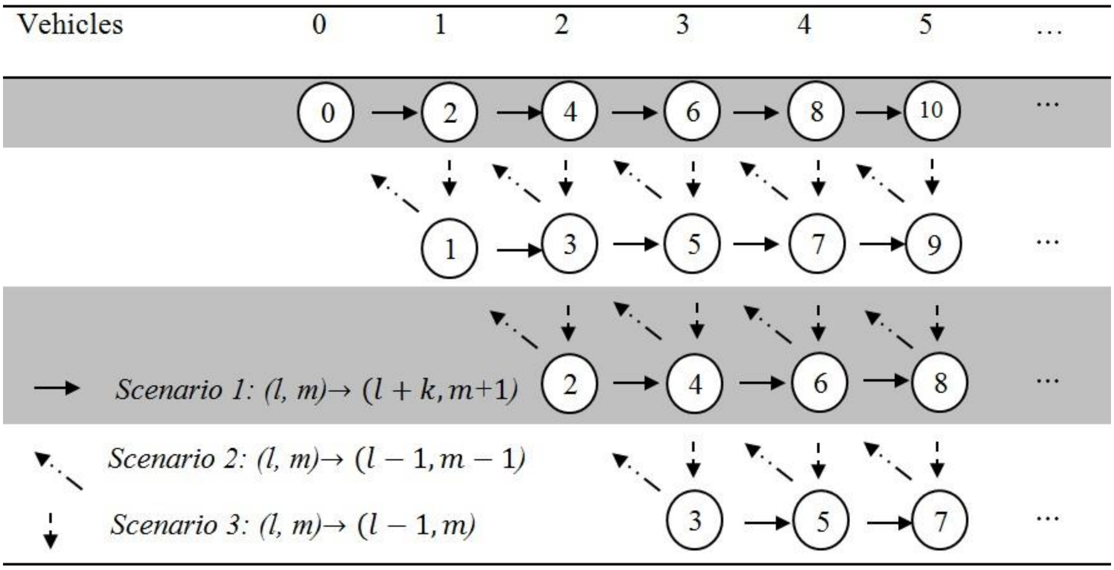
$$\beta_{l,m} = 1 - \alpha_{l,m}, \tag{4-12}$$

Similarly, the transition probability from (l, m) to state $(l-1, m)$ can be calculated as follows:

$$P_{(l,m)\rightarrow(l-1,m)} = \beta_{l,m} * \ k\gamma(t) * p, \tag{4-13}$$

where $p = \min(m, n)$ is the number of active service stations.

With these three scenarios discussed, the state-to-state transition diagram for the $M/E_{k=2}/3$ queueing model for example, can be depicted as shown in Figure 4-5.



Vehicles
0
1
2
3
4
5
...
0
2
4
6
8
10
...
1
3
5
7
9
...
2
4
6
8
...
3
5
7
...
Scenario 1: (l, m)→ $(l+k, m+1)$
Scenario 2: (l, m)→ $(l-1, m-1)$
Scenario 3: (l, m)→ $(l-1, m)$

Figure 4-5 State to State Transition Process for the $M/E_{k=2}/3$ Queueing Model

On Figure 4-5, the number in the circle denotes the number of uncompleted stages in the queueing system (i.e. *l*), whereas the number on the top of each column represents the number of vehicles in the system (i.e. *m*). In other words, each circle represents a given state (l, m). The different types of arrows show the different state to state transitions described in the left corner of Figure 4-5. The solid arrow going toward the right represents the first transition case discussed in Chapter 4.2.1.2.1 (a new vehicle arrives). The dotted diagonal arrow represents the transition case described in

Chapter 4.2.1.2.2 where one stage is completed and one vehicle leaves the system. Finally, the dotted vertical arrow moving downward represents the case described in Chapter 4.2.1.2.3 (one stage is completed but no vehicle departs).

4.2.1.3 State-to-State Transitions and Transient Solution Calculations

4.2.1.3.1 Calculation Approach Overview

In this Chapter, we briefly describe how the state-to-state transitions are calculated. Our discussion is once again largely based on the work of Escobar et al. (2002). However, we separate the description of the state transitions associated with serving vehicles from those associated with vehicle arrivals. In other words, in-between vehicle arrivals, we proceed in a time-step fashion to calculate the state transitions based on either state transition scenario 2 or 3 described above. On the other hand, when a new vehicle arrives, we calculate the state transitions based on scenario 1 above.

Because we are interested in the transient solution, our approach can be regarded as a hybrid between a simulation or numerical based approach on one hand, and an analytical approach on the other. The inter-arrival times are determined by sampling from the inter-arrival distribution curve, which determines when the next vehicle will arrive. In between the inter-arrival period, we sample the service rate from the service-time distribution at each time step which is one second in this study. The sampling mechanism was designed to facilitate the comparison to the VISSIM microscopic simulation results for validation as will be explained later in Chapter 4.2.3. Based on the sampled service rate, we calculate the state probabilities, as will be described in more detail in Chapter 4.2.1.3.2. Note that we also calculate the modeling results using the mean values of the inter-arrival and service times, and compare the results from both approaches (i.e. the randomly sampling based and the mean value based) to the VISSIM model's results in the validation Chapter.

4.2.1.3.2 State Probabilities Calculations

Let $P_{l,m}(t)$ represent the probability of state (l, m) at the current time step t. Naturally, at the initial state when no vehicle is in the system at time 0, $P_{0,0}(0) = 1$. Now consider the time interval in between when the N^{th} vehicle arrives and when the $(N + 1)^{th}$ vehicle arrives (as mentioned above, the inter-arrival time period between vehicle *N* and *N+1* can be either sampled from the cumulative arrival distribution curve (t_s) or assumed as the mean value of the inter-arrival time period , t_{av}). Right before the $(N + 1)^{th}$ vehicle arrives, the possible values for the number of the vehicles m in the system could range between 0 and N (depending upon how many vehicles have already been served and have left the queue). In other words, all the states (l, m), where $m \leq N$ may exist.

Now, at a given time step, the change in the probability for a state (l, m) can be calculated according to the transition probabilities from and to that state, *as indicated by the arrows going out of and toward the circles shown in* Figure 4-5 *above*. The following Equations will describe exactly how the calculations proceed to update the state probabilities for each time step t_i, within the time period which starts at the time when the N^{th} vehicle arrives and ends one time-step before the time when the

$(N+1)^{th}$ vehicle arrives (i.e. $t_i = t, \dots, (t + t_s - 1)\ or\ (t + t_{av} - 1)$. In these Equations, $\dot{P}_{l,m}(t_i)$ represents the change in the value for $P_{l,m}(t_i)$ at each time step, and therefore $P_{l,m}(t_i + 1) = P_{l,m}(t_i) + \dot{P}_{l,m}(t_i)$. This process is then repeated until the end of the analysis period of interest.

When calculating the change in the probabilities, one of the following two cases may occur, depending upon whether the number of vehicles in the system (*m*) is less than the number of servers in the system (i.e. no queue exists) or not. For each case, the calculations are as follows:

Case (1): The number of vehicles in the system (*m*) is less than the number of servers in the system (n), $m < n$:

In this case, we have the following three possibilities:

1a) For the special case of state $(0,0)$, the change in the state probability (see Figure 4-5 for possible state transitions), can be calculated by Equation (4-14) as shown below:

$$\dot{P}_{0,0}(t_i) = k\gamma(t_i)P_{1,1}(t_i), \tag{4-14}$$

1b) For the states where $l = mk$ (see Figure 4-5 for possible state transitions for states (mk, m)) with $m = 1,2, \dots, n-1$), , we have:

$$\dot{P}_{mk,m}(t_i) = -mk\gamma(t_i)P_{mk,m}(t_i) + \alpha_{mk+1,m+1}(m+1)k\gamma(t_i)P_{mk+1,m+1}(t_i), \tag{4-15}$$

1c) For the states where $l = mk - x$ with

$m = 1,2, \dots, n-1,\ and\ x = 1,2, \dots, m(k-1)$, we have:

$$\dot{P}_{mk-x,m}(t_i) = -mk\gamma(t_i)P_{mk-x,m}(t_i) + \beta_{mk-x+1,m}mk\gamma(t_i)P_{mk-x+1,m}(t_i) + \alpha_{mk-x+1,m+1}(m+1)k\gamma(t_i)P_{mk-x+1,m+1}(t_i), \tag{4-16}$$

Case (2): The number of vehicles in the system (*m*) is equal to or greater than the number of servers (*n*), $m >= n$:

The difference between case (2) and case (1) considered above is that for case (2) arriving vehicles may start forming a queue behind the servers. In this case, we would have the following three possibilities:

2a) For the states where $l = mk$ (i.e. states (mk, m)) with $m = n, (n+1), (n+2), \dots N)$,

$$\dot{P}_{mk,m}(t_i) = -nk\gamma(t_i)P_{mk,m}(t_i) + \alpha_{(n-1)k+1,n}nk\gamma(t_i)P_{mk+1,m+1}(t_i), \tag{4-17}$$

2b) For the states where $l = mk - x$ with$m = n, (n+1), (n+2), \dots, N$ and $x = 1,2, \dots, (n-1)(k-1)$,

$$\dot{P}_{mk-x,m}(t_i) = -nk\gamma(t_i)P_{mk-x,m}(t_i) + \beta_{nk-x+1,n}nk\gamma(t_i)P_{mk-x+1,m}(t_i) + \alpha_{(n-1)k-x+1,n}nk\gamma(t_i)P_{mk-x+1,m+1}(t_i), \tag{4-18}$$

2c) For the states where $l = mk - y$ with $m = n, (n+1), (n+2), \dots, N \; and \; y = (n-1)(k-1)+1, (n-1)(k-1)+2, \dots, n(k-1)$,

$$\dot{P}_{mk-y,m}(t_i) = -nk\gamma(t_i)P_{mk-y,m}(t_i) + \beta_{nk-y+1,n}nk\gamma(t_i)P_{mk-y+1,m}(t_i), \tag{4-19}$$

After the $(N+1)^{th}$ vehicle finally joins the queue at time point $t_i = (t + t_s)$ or $t_i = (t + t_{av})$, all the possible states in the queueing system will be updated with probability 1, and it would look as if the probability values of the possible states moved one step to the right in Figure 5 (i.e. the probability of a given state is now equal to the probability of the state to its left). The probabilities of the different states when a new vehicle arrives can thus be calculated as shown in Equation (4-20) below:

$$P_{mk-x,m}(t_i) = \begin{cases} P_{mk-x-k,m-1}(t_i), & state\ (mk-x-k, m-1) \text{ exists} \\ 0, & state\ (mk-x-k, m-1)\ not\ exist \end{cases} \tag{4-20}$$

for

$$m = (N+1), N, \dots, 1, 0\ and\ x = \begin{cases} 0,1,\dots,(n-1)(k-1),\dots,n(k-1), & if\ m > n \\ 0,1,\dots,(m-1)(k-1),\dots,m(k-1), & if\ m \le n' \end{cases}$$

An example of how the state probability calculations proceed and how the above Equations are used is given in Appendix A for readers interested in the details.

4.2.1.4 Performance Measurement Calculations

With the probabilities calculated above, a number of useful performance measures can be calculated. In doing this, a unique advantage of our proposed hybrid numerical/analytical approach is that, as opposed to a purely simulation approach, the running time needed to derive these performance measures represents a fraction of the time needed to run a detailed microscopic traffic simulation model multiple times and to gather the required statistics. Specifically, after the state transition probabilities for all states (l, m) are calculated, the probability for *m* vehicles in the queueing system can be derived as follows:

$$P_m(t) = \begin{cases} \sum_{l=m}^{km} P_{l,m}(t), & 0 \le m \le n, \\ \sum_{l=n+(m-n)k}^{km} P_{l,m}(t), & m > n. \end{cases}, \quad 0 \le m \le N. \tag{4-21}$$

With $P_m(t)$ known, the average or most likely queue length at time *t*, $Q_m(t)$, can be calculated as:

$$Q_m(t) = L_{veh} * \sum_m P_m(t) * m, 0 \le m \le N \tag{4-22}$$

where, L_{veh} is the average length of the vehicle.

At the same time, the average delay for a vehicle that arrives at the border at time t, can be calculated by estimating the time it would take for the queue in front of the vehicles to be served as follows:

$$D_m(t) = \sum_m P_m(t) * m * \mu, \tag{4-23}$$

where, μ is the average service time of the Erlang distribution.

Besides the mean values, the variance or standard deviation of the expected queue length and/or delay can be calculated. First, when the sampling is deployed (i.e. the inter-arrival and service times are determined by sampling from their probability distributions), the variance of the delay or queue length can be calculated by running the model multiple times. Alternatively, when using the mean values of the inter-arrival and service times, the variance of the delay, for example, may be calculated from Equation (4-24), as follows.

$$V_m(t) = \sum_m P_m(t) * (m - \bar{m})^2 * \mu, \tag{4-24}$$

where, $\bar{m}$ is the average value of the number of vehicles.

4.2.2 BMAP/PH/n queueing model

Compared with $M/E_k/n$ queueing model, the *BMAP/PH/n* queueing model has more complicated system states and state transition scenarios. At the onset, it is important to note that “system state” is used herein to describe the state of the queueing system itself, and that this is different from the “state space” mentioned in relation to the BMAP or PH distributions. To calculate the transient measures of the queueing system, this research introduced a novel approach to describing the system states, along with a new assumption, which we call the Equally Likely Vehicles (ELV), to calculate the probabilities of the system states.

4.2.2.1 System state description

In order to make the description more understandable, we assume that "service type" i is equivalent to one of the service type distributions involved within that PH distribution (e.g., in our PH distribution, there are two service types: Exponential distribution and Erlang-2 distribution). Besides that, as was the case with the $\mathrm{M/E_k/n}$, we use "stage" $\mathrm{l_i}$ to represent the remaining number of exponential service stations that need to be completed for the vehicles $\mathrm{m_i}$ in service type i.

Now suppose there are s service types in the PH distribution, a natural way is to use $(l_i, m_i)_{1\le i\le s}$ to record the unfinished service stages l_i and number of vehicles m_i in service type i. Here $m_i \le l_i \le k_i * m_i,\ m_i \ge 0,$ and k_i denotes the number of the total service stages of service type i, like the order of the Erlang distribution. However, if there are *N* vehicles in the queueing system, the number of possible states will be s^N. This can be a very huge number. In order to save some space, we only calculate $(l_i, m_i)_{1\le i\le s}$ for the vehicles in the service stations, and use one more digit n_q to represent the number of vehicles waiting in the queue and not being served. The complete state representation is now $(l_1, m_1, l_2, m_2, \dots, l_s, m_s)|n_q$ in the BMAP/PH/n model. Using this representation, the total number of possible states will be $s^n * (N - n + 1)$ if N is greater than n and s^N if N is less than or equal to n.

4.2.2.2 State transition probabilities

Suppose that at time point t, N vehicles have joined the queue, and the queue is in state $(l_1, m_1, \dots, l_i, m_i, \dots, l_j, m_j, \dots, l_s, m_s)|n_q$, $1 \leq i, j \leq s, i \neq j$. For the next time step $(t + 1)$, there could be four possible state transition scenarios:

1) Scenario 1: one or more vehicles join the queue

Different from the $\mathrm{M}/E_k/n$ queueing model, b vehicles (and not just one) could arrive in the queue at t_{N+b}, and the initial service types when the b vehicles start being served can also be different. Now if the next time step, $t + 1$, is equal to t_{N+b}, there are three situations:

i) if there are empty servers, $\sum_{i=1}^{s} m_i < n$, and the number of empty servers is greater than or equal to b, all of the b vehicles will instantly be served, and the queueing system state will change from $(l_1, m_1, \dots, l_i, m_i, \dots, l_s, m_s)|n_q$ to $(l_1 + k_1 * m_{1b}, m_1 + m_{1b}, \dots, l_i + k_i * m_{ib}, m_i + m_{ib}, \dots, l_s + k_s * m_{sb}, m_s + m_{sb})|n_q$ with probability 1. Here, m_{ib} is the number of vehicles in b with initial service type i;

ii) if there are empty servers, $\sum_{i=1}^{s} m_i < n$, but the number of empty servers is less than b, $(e = n - \sum_{i=1}^{s} m_i)$ of the b vehicles will start their service process, and the rest will wait for their service in the queue, the state $(l_1, m_1, \dots, l_i, m_i, \dots, l_s, m_s)|n_q$ will become $(l_1 + k_1 * m_{1e}, m_1 + m_{1e}, \dots, l_i + k_i * m_{ie}, m_i + m_{ie}, \dots \dots, l_s + k_s * m_{se}, m_s + m_{se})|n_q + b - e$ with probability 1. Here, m_{ie} is the number of vehicles in e with initial service type i;

iii) if there are no empty servers, $\sum_{i=1}^{s} m_i = n$, all the b vehicles will wait in the queue without being served. The state will transfer from $(l_1, m_1, \dots, l_i, m_i, \dots, l_s, m_s)|n_q$ to $(l_1, m_1, \dots, l_i, m_i, \dots, l_s, m_s)|n_q + b$ with probability 1.

2) Scenario 2: one vehicle finishes its last stage of service and leaves the queue

Suppose after finishing the last stage of service type i, the vehicle leaves the queue (corresponds to the absorption state 0 in PH distribution), the queueing system would thus transfer from state $(l_1, m_1, \dots, l_i, m_i, \dots, l_s, m_s)|n_q$ to state $(l_1, m_1, \dots, l_i - 1, m_i - 1, \dots, l_s, m_s)|n_q$, $1 \leq i \leq s$. If $n_q > 0$, which means there are still vehicles waiting in the queue to be served, state $(l_1, m_1, \dots, l_i - 1, m_i - 1, \dots, l_s, m_s)|n_q$ must be transferred to state $(l_1, m_1, \dots, l_i - 1 + k_i, m_i, \dots, l_s, m_s)|n_q - 1$ or state $(l_1, m_1, \dots, l_i - 1, m_i - 1, \dots, l_j + k_j, m_j + 1, \dots, l_s, m_s)|n_q - 1$ according to the initial service type of the n_qth vehicle in the reverse order from N.

Suppose $\alpha_{(l_1,m_1,\dots,l_i,m_i,\dots,l_s,m_s)|n_q}$ is the probability that one more stage of service type i is finished and one out of m_i vehicles leaves the queue. To calculate this, we still assume the ELC heuristic. Therefore,

$\alpha_{(l_1,m_1,\dots,l_i,m_i,\dots,l_s,m_s)|n_q} = \alpha_{l_i,m_i}$, if the state transition is reasonable based on the Markov chain in PH distribution, (4-25)

Otherwise if the state transition is impossible based on the Markov chain in PH distribution,

$$\alpha_{(l_1,m_1,\dots,l_i,m_i,\dots,l_s,m_s)|n_q} = 0, \tag{4-26}$$

where, α_{l_i,m_i} can be calculated using the same way in $\mathrm{M}/E_k/n$ queueing model.

However, in order to calculate the queueing state transition probability that one stage of service type i is finished and one vehicle leaves the system in one time step (i.e. from the current time step t to the next time step $t+1$), one more assumption, which we call the Equally Likely Vehicles (ELV) heuristic, is needed. This heuristic assumes all vehicles have an equal probability to be served no matter what service type it is in, and therefore the probability that the vehicle to be served in the current time step is from service type i is given by:

$$v_i = \frac{m_i}{\sum_{I=1}^{s} m_I}. \tag{4-27}$$

Finally, the transition probability in this time step can be calculated as follows:

$$P_{(l_1,m_1,\dots,l_i,m_i,\dots,l_s,m_s)|n_q \to St} = v_i * \alpha_{(l_1,m_1,\dots,l_i,m_i,\dots,l_s,m_s)|n_q} * k_i \gamma_i(t) * m_i, \tag{4-28}$$

Where St is the corresponding queueing state depending on the initial service type of the n_q th vehicle in the reverse order from N;

$\gamma_i(t)$ is the service rate at time step t for service type i.

For example, let us consider how $\alpha_{(1,1,3,2)|0}$ is calculated. For the first pair $(l_1, m_1) = (1,1)$, which refers to the number of stages and the vehicles having the exponentially distributed service times, according to Equation (4-26), $\alpha_{(1,1,3,2)|0} = 0$. This is because in this study we know that after the vehicle finishes the Exponential distribution service type, it will continue the process with the Erlang-2 distribution service type. In other words , $(1,1,3,2)|0 \to (0,0,3,2)|0$ is impossible. For the second pair $(l_2, m_2) = (3,2)$, which records the number of stages and the vehicles having service type of Erlang-2 distribution, according to Equation (4-25),

$\alpha_{(1,1,3,2)|0} = \alpha_{l_2,m_2} = \alpha_{3,2}$, and we know that $\alpha_{3,2} = \frac{1}{2}$ from the previous Chapter. Given this, we can finally estimate, $P_{(1,1,3,2)|0\to(1,1,2,1)|0} = v_2 * \alpha_{(1,1,3,2)|0} * k_2 * \gamma_2(t) * m_2 = \frac{m_2}{m_1+m_2} * \alpha_{3,2} * k_2 * \gamma_2(t) * m_2$.

3) Scenario 3: one vehicle finishes its last stage of service type i and starts the first stage of another service type j

This means the transition from state $(l_1, m_1, \ldots, l_i, m_i, \ldots, l_j, m_j, \ldots l_s, m_s)|n_q$ to state $(l_1, m_1, \ldots, l_i - 1, m_i - 1, \ldots, l_j + k_j, m_j + 1, \ldots, l_s, m_s)|n_q$, $1 \le i, j \le s, i \ne j$, from the current time step t to the next time step $t + 1$ (corresponds to the transition from state 1 to state 2 in Figure 4-3).

Similarly, the probability can be calculated as:

$P_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_j,m_j,\ldots\, l_s,m_s)|n_q\to(l_1,m_1,\ldots,l_i-1,m_i-1,\ldots,l_j+k_j,m_j+1,\ldots,l_s,m_s)|n_q} =$

$v_i * \tau_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_s,m_s)|n_q} * k_i\gamma_i(t) * m_i$,

(4-29)

where,

$$v_i = \frac{m_i}{\sum_{l=1}^{s} m_l}, \quad (4\text{-}30)$$

$$\tau_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_s,m_s)|n_q} = \alpha_{l_i,m_i}, \quad (4\text{-}31)$$

if the state transition is reasonable based on the Markov chain in PH distribution. Otherwise,

$$\tau_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_s,m_s)|n_q} = 0. \quad (4\text{-}32)$$

4) *Scenario 4: one vehicle finishes one stage of service type i but still needs to finish other stages of service type i*

Now state $(l_1, m_1, \dots, l_i, m_i, \dots, l_j, m_j, \dots l_s, m_s)|n_q$ will become state $(l_1, m_1, \dots, l_i - 1, m_i, \dots, l_s, m_s)|n_q$, $1 \leq i \leq s$, from the current time step t to the next time step $t + 1$. Obviously, this probability is

$$P_{(l_1,m_1,\dots,l_i,m_i,\dots,l_j,m_j,\dots l_s,m_s)|n_q \to (l_1,m_1,\dots,l_i-1,m_i,\dots,l_s,m_s)|n_q} = v_i * \beta_{(l_1,m_1,\dots,l_i,m_i,\dots,l_s,m_s)|n_q} * k_i \gamma_i(t) * m_i, \tag{4-33}$$

where,

$$v_i = \frac{m_i}{\sum_{I=1}^{s} m_I}, \tag{4-34}$$

$$\beta_{(l_1,m_1,\dots,l_i,m_i,\dots,l_s,m_s)|n_q} = 1 - \alpha_{l_i,m_i}, \tag{4-35}$$

if the state transition is reasonable based on the Markov chain in the PH distribution. Otherwise,

$$\beta_{(l_1,m_1,\dots,l_i,m_i,\dots,l_s,m_s)|n_q} = 0. \tag{4-36}$$

4.2.2.3 State Probabilities and Transient Solution Calculations

In a simulation/approximation approach to determining the transient solution, the inter-arrival time and the arrival size can be sampled once the estimation of BMAP is realized. We can also estimate the initial service types of the newly arrived vehicles by sampling according to the PH distribution. Moreover, for each time step in the arrival intervals, the service rate can be determined either by sampling or using the mean value of the corresponding service time distribution, as done before. With these, the state probability can be calculated as following:

For a given state $(l_1, m_1, \ldots, l_i, m_i, \ldots, l_j, m_j, \ldots, l_s, m_s)|n_q$, $1 \le i, j \le s, i \ne j$, at time step t, assuming that N vehicles have joined the queue, for each service type pair (l_i, m_i), the probability $\dot{P}_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_j,m_j,\ldots,l_s,m_s)|n_q}$ is calculated as :

When $\sum_{I=1}^{s} m_I = 0$ (which refers to the case when no vehicle is in the queue),

$$\dot{P}_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_j,m_j,\ldots,l_s,m_s)|n_q}(t) = v_\alpha * \alpha * m_\alpha * ser_i * P_\alpha, \tag{4-37}$$

where,

$$ser_i = k_i \gamma_i(t), \tag{4-38}$$

$$v_\alpha = \frac{m_i+1}{\sum_{I=1}^{s} m_I + 1}, \tag{4-39}$$

$$\alpha = \alpha_{(l_1,m_1,\ldots,l_i+1,m_i+1,\ldots,l_s,m_s)|n_q}, \tag{4-40}$$

$$m_\alpha = m_i + 1, \tag{4-41}$$

$$P_\alpha = P_{(l_1,m_1,\ldots,l_i+1,m_i+1,\ldots,l_s,m_s)|n_q}, \tag{4-42}$$

When $\sum_{I=1}^{s} m_I > 0$ (some vehicles are in the queue), the transition probability can be generally expressed as,

$$\dot{P}_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_j,m_j,\ldots,l_s,m_s)|n_q}(t) = -m_i * ser_i * P_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_j,m_j,\ldots,l_s,m_s)|n_q} + v_\alpha * \alpha * m_\alpha * ser_i * P_\alpha + v_\tau * \tau * m_\tau * ser_i * P_\tau + v_\beta * \beta * m_\beta * ser_i * P_\beta, \tag{4-43}$$

where $ser_i = k_i \gamma_i(t)$, (4-44)

Now, if $\sum_{I=1}^{s} m_I + 1 \le n$,

$$v_\alpha = \frac{m_i+1}{\sum_{I=1}^{s} m_I + 1}, \tag{4-45}$$

$$\alpha = \alpha_{(l_1,m_1,\ldots,l_i+1,m_i+1,\ldots,l_s,m_s)|n_q}, \tag{4-46}$$

$$m_\alpha = m_i + 1, \tag{4-47}$$

$$P_\alpha = P_{(l_1,m_1,\ldots,l_i+1,m_i+1,\ldots,l_s,m_s)|n_q}, \tag{4-48}$$

Or if $\sum_{I=1}^{s} m_I + 1 > n$,

$v_\alpha = \frac{m_i}{\sum_{I=1}^{s} m_I}$, (4-49)

$\alpha = \alpha_{(l_1,m_1,\dots,l_i-k_i+1,m_i,\dots,l_s,m_s)|n_q+1}$, if the initial service type of the $(n_q+1)^{\text{th}}$ vehicle in the reverse order from N is i; or $\alpha = \alpha_{(l_1,m_1,\dots,l_i+1,m_i+1,\dots,l_j-k_j,m_j-1,\dots,l_s,m_s)|n_q+1}$, if the initial service type of the $(n_q+1)^{\text{th}}$ vehicle in the reverse order from N is j, $j \neq i$

(4-50)

$m_\alpha = m_i$, (4-51)

$P_\alpha = P_{(l_1,m_1,\dots,l_i-k_i+1,m_i,\dots,l_s,m_s)|n_q+1}$, if the initial service type of the (n_q+1)th vehicle in the reverse order from N is i; or, $P_\alpha = P_{(l_1,m_1,\dots,l_i+1,m_i+1,\dots,l_j-k_j,m_j-1,\dots,l_s,m_s)|n_q+1}$, if the initial service type of the (n_q+1)th vehicle in the reverse order from N is j,

$j \neq i$, (4-52)

$$v_\tau = \begin{cases} \frac{m_i+1}{\sum_{I=1}^{s} m_I}, & if\ (m_i+1) \le \sum_{I=1}^{s} m_I \\ 0, & if\ (m_i+1) > \sum_{I=1}^{s} m_I \end{cases}, \tag{4-53}$$

$\tau = \tau_{(l_1,m_1,\dots,l_i+1,m_i+1,\dots,l_j-k_j,m_j-1,\dots,l_s,m_s)|n_q}$, (4-54)

$m_\tau = m_i + 1$, (4-55)

$P_\tau = P_{(l_1,m_1,\dots,l_i+1,m_i+1,\dots,l_j-k_j,m_j-1,\dots,l_s,m_s)|n_q}$, (4-56)

$v_\beta = \frac{m_i}{\sum_{I=1}^{s} m_I}$, (4-57)

$\beta = \beta_{(l_1,m_1,\dots,l_i+1,m_i,\dots,l_s,m_s)|n_q}$, (4-58)

$m_\beta = m_i$, (4-59)

$P_\beta = P_{(l_1,m_1,\dots,l_i+1,m_i,\dots,l_s,m_s)|n_q}$, (4-60)

At last, after all the service type pair (l_i, m_i) is checked,

$P_{(l_1,m_1,\dots,l_i,m_i,\dots,l_j,m_j,\dots,l_s,m_s)|n_q}(\mathrm{t}+1)$ is updated as:

$$P_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_j,m_j,\ldots,l_s,m_s)|n_q}(t+1) =$$

$$P_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_j,m_j,\ldots,l_s,m_s)|n_q}(t) + \sum_{1\le i\le S} \dot{P}_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_j,m_j,\ldots,l_s,m_s)|n_q}(t)$$

(4-61)

If at time step $t+1$, b vehicles will join the queue, for state $(l_1, m_1, \ldots, l_i, m_i, \ldots l_s, m_s)|n_q$, according to the analysis of scenario 1 in Chapter 4.2.2, we can get the new state $(L_1, M_1, \ldots, L_i, M_i, \ldots L_s, M_s)|N_q$, and the probability of the new state should be:

$$P_{(L_1,M_1,\ldots,L_i,M_i,\ldots,L_s,M_s)|N_q}(t+1) = P_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_s,m_s)|n_q}(t+1), \quad (4\text{-}62)$$

$$P_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_s,m_s)|n_q}(t+1) = 0, \quad (4\text{-}63)$$

This calculation should be conducted according to the possible number of vehicles in the queue in a decreasing order. $\sum_{I=1}^{s} m_I + n_q = N, N-1, \ldots, 1, 0$.

With all the state probabilities at any time step t known, the probability that there are v vehicles in the system can be finally calculated:

$$P_v(t) = \sum P_{(l_1,m_1,\ldots,l_i,m_i,\ldots,l_s,m_s)|n_q}(t), \text{ if } \sum_{I=1}^{s} m_I + n_q = v, \text{ for } v = 0, 1, \ldots, N. \quad (4\text{-}64)$$

Similarly, this probability can be used to compute performance measures such as queue length and delay.

An example of how the state probabilities calculations proceed and how the above Equations are to be applied is given in Appendix B for readers interested in the details.

4.2.3 The Baseline micro-simulation VISSIM model

To validate the proposed numerical /analytical approach, the queueing model's results were compared to the results derived from a detailed microscopic traffic simulation model of the border crossing area developed in VISSIM (PTV, 2010). The VISSIM model is used as a “baseline” model to validate the performance of the proposed queueing models and their approximate transient solution procedure, due to the current unavailability of detailed field observations regarding actually experienced delays and the corresponding number of inspection stations that were open at the time. As mentioned in the Conclusions Chapter, we are planning to validate the models against field data in our future research. Figure 4-6 shows a screen shot of the VISSIM animation of traffic at the Peace Bridge. The orange part in the figure is the U.S. toll plaza for the private cars entering the U.S. from Canada, which is the focus of this case study. As can be seen, the number of the lanes for private cars changes from 1 to 10 as one gets close to the border (i.e., the maximum number of inspection stations or servers for our queueing model is thus 10). In the example shown in Figure 4-6, only 5 service stations are open.

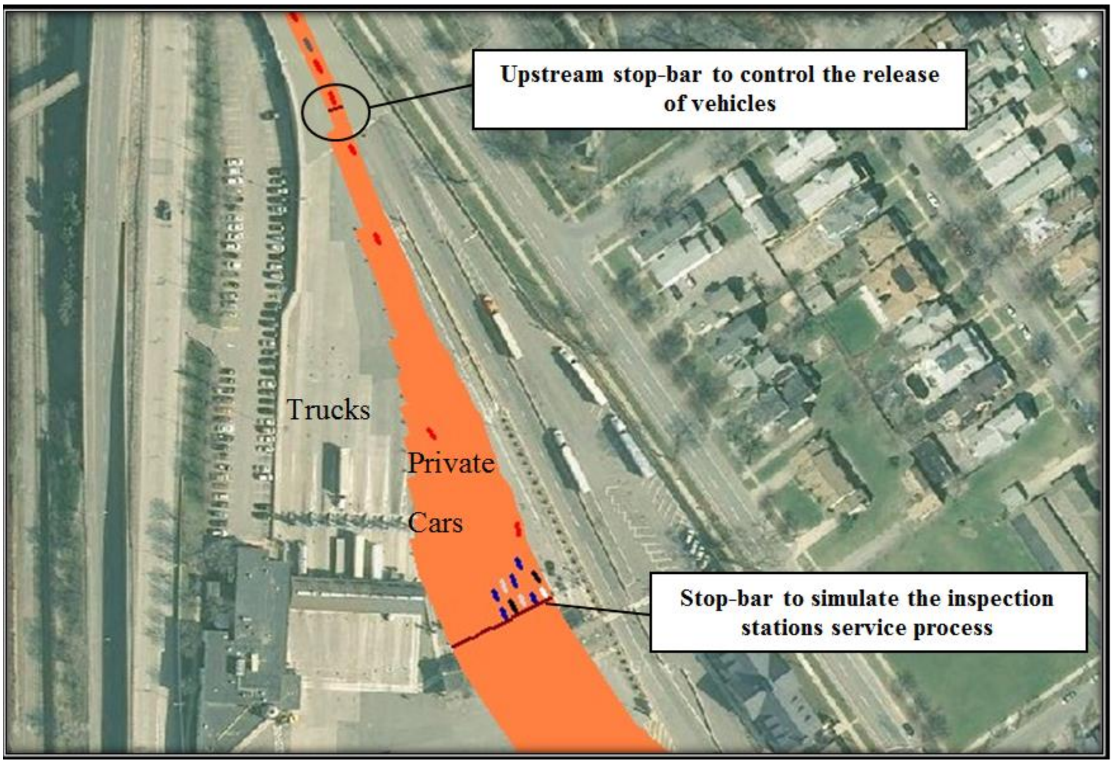

Upstream stop-bar to control the release of vehicles
Trucks
Private
Cars
Stop-bar to simulate the inspection stations service process

Figure 4-6 Queueing Model in VISSIM

In VISSIM, the dwell time distribution at the stop signs can be precisely controlled to follow a given probability distribution. Figure 4-7 shows an example on how the dwell time distribution at a stop sign may be adjusted to follow any desired probability distribution. Given this, a stop-bar was placed upstream the inspection plaza in order to control the release of vehicles and to make sure that vehicles' arrival at the toll plaza follow the desired probability distributions (i.e. the inter-arrival exponential distribution shown in Figure 4-1). Similarly, a second stop bar was placed, where the inspection stations are, to simulate the service time Erlang distribution shown in Figure 4-2. In other words, the VISSIM model was developed to mimic the operation of $\mathrm{M}/E_K/n$ queueing system (i.e., the first multi-server queueing model considered in this study).

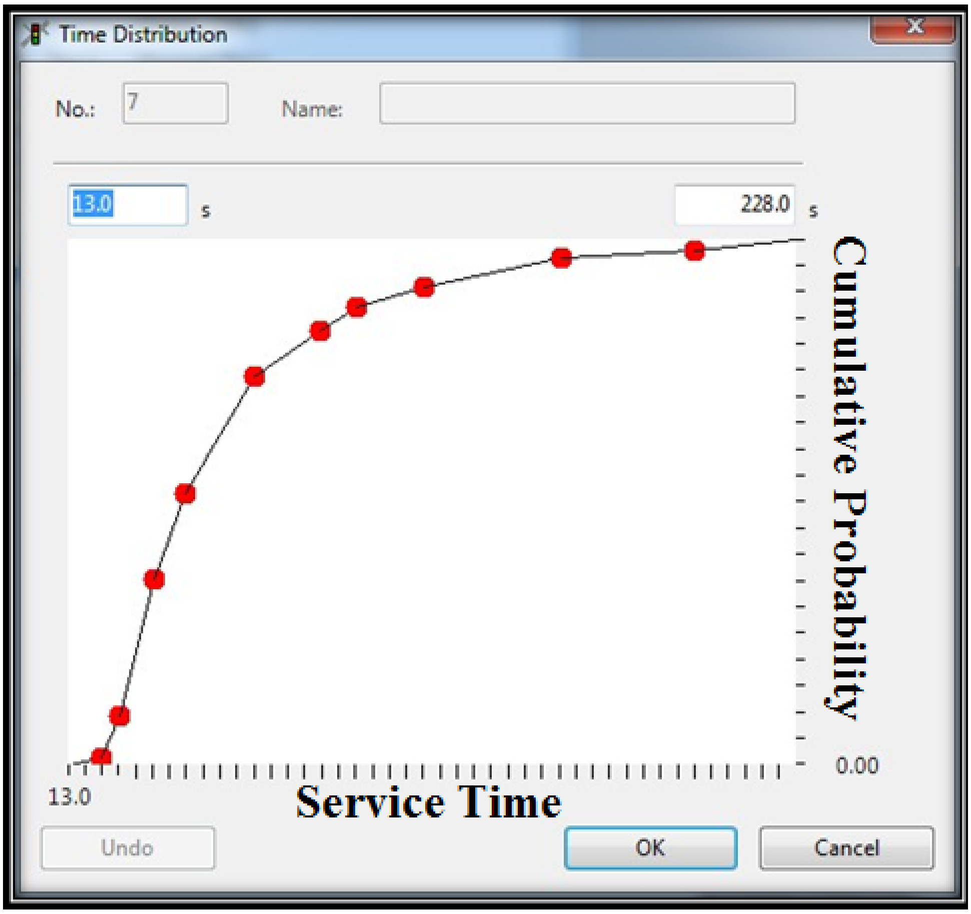


Figure 4-7 Setting of Dwell Time Distribution in Stop Sign to Simulate Service Time

With this, the traffic demand was defined in a ".fma" file, and VISSIM's optimal dynamic assignment module was used to simulate the drivers' choices of lanes or inspection stations so as to ensure that the number of vehicles waiting in line for each service station is almost the same. Other settings were realized through the COM interface, which can be used to customize the VISSIM model (PTV, 2010), using the C# programming language. For example, the COM interface was used to control the number of lanes/inspection stations that are open at a given time by dynamically controlling the the "LANECLOSED" parameter in VISSIM. The interface was also used to facilitate running the simulation model for multiple runs, using a different random seed number each time, and averaging the results to calculate the performance measures of interest (i.e., queue length and wait time) from the multiple runs.

4.3 Results

4.3.1 Validation Results

As mentioned above, due to the unavailability of accurate field data on border crossing delay at the time this study was conducted, validating the queueing model results involved comparing them to those derived from running a VISSIM model, keeping in mind . For these comparisons, a 20 minute prediction horizon was adopted (i.e., the models were used to estimate the queue length that a vehicle joining the queue 20 minutes later will encounter). The arrival volume was assumed to be equal to 400 vehicles per hour (vph), and an Erlang distribution with order 2 and mean 44.58 seconds was utilized to represent the service process for $\mathrm{M}/E_K/n$ queueing model, a mixture of an Exponential distribution with the mean of 79.36 seconds and an Erlang distribution with order 2 and the mean of 40.98 seconds was utilized to represent the service process for $BMAP/PH/n$ queueing model. The number of service stations was varied from 3 to 6 stations.

As previously mentioned before, two slightly different methods were utilized to derive the transient solution of the queueing models, sampling the arrival and service times from the corresponding distributions and using the mean values of the corresponding distributions. For the $\mathrm{M}/E_K/n$ queueing model, when sampling, the heuristic queueing model solution procedure was repeated 100 times so that the mean and standard deviation of the queue length were calculated. For the $BMAP/PH/n$ queueing model, because we always need to sample the arrival size and service types, the procedure based on sampling the arrival and service times method and the procedure based on using the mean values method were both repeated 100 times. The results from both methods were compared to the ones from the VISSIM model. For the VISSIM model, due to the fact that the model requires significantly more runtime compared to the runtime of the heuristic model solution, the model was run for only 10 times with different seed numbers. Finally for each model, the mean and the standard deviation of vehicles number in the queue can be seen in Table 4-2.

Table 4-2 Results of Analytical Approach and Simulation Approach

Predicted Traffic Volume (vph)	Number of Service Stations	Number of Vehicles in the Queue (Vehicles)									
		Random sampling method ($M/E_K/n$)		Mean value method ($M/E_K/n$)		Random sampling method ($BMAP/PH/n$)		Mean value method ($BMAP/PH/n$)		Simulation in VISSIM ($M/E_K/n$)	
		Mean	Standard Variance	Mean	Standard Variance	Mean	Standard Variance	Mean	Standard Variance	Mean	Standard Variance
400	3	43.38	9.13	40.73	5.16	50.78	4.72	47.93	6.15	41.9	7.24
	4	22.14	8.28	22	5.58	26.75	4.33	22.93	6.22	26.6	7.57
	5	12.7	6.14	8.28	3.40	8.40	2.18	8.65	3.09	15.4	5.29
	6	7	2.6	5.6	1.49	6.64	1.59	5.26	1.77	9.9	3.72

As can be seen, firstly, for the $M/E_K/n$ queueing models, the results from the heuristic solution method of the queueing model appear to be quite close to the VISSIM model results. The advantage of the analytical approach is naturally the very high computational efficiency compared to the simulation based approach, and the ability to incorporate the models within an optimization framework as will be described later. Moreover, it can be seen that the results of the $\mathrm{M}/E_K/n$ queueing models from the random sampling method are also generally close to the results from the mean value based method. Meanwhile, using the mean values proves to be even more computationally efficient in comparison to the random sampling method because it does not require multiple runs. Based on this small-scale validation study, it can be concluded that the formulated queueing model $M/E_k/n$, and the ELC heuristic solution procedure, offers a more efficient approach to estimate likely queue lengths and border crossing delays than microscopic traffic simulation.

Secondly, we can see that the results of the $BMAP/PH/n$ queueing models are quite close to those of the $\mathrm{M}/E_K/n$ queueing model and to the VISSIM simulation, except perhaps for when the number of the service stations was relatively small (i.e. 3) where the queueing lengths derived from the $BMAP/PH/n$ queueing models are slightly longer. These results appear to make perfect sense, since from the analysis of PH service process, we know that the probability of a vehicle going through the Exponential service process followed by the Erlang service process is relatively small (i.e., 1.63%). For the majority of cases, the service process follows an Erlang distribution similar to the first model. Even though the probability is low, the differences are expected to be more obvious effect when the number of service stations is small. The transient results of the $BMAP/PH/n$ queueing models thus appear to be realistic.

4.3.2 Sensitivity Analysis

This Chapter presents the results of a few sensitivity analysis tests conducted on the models. The purpose of this analysis is twofold, to demonstrate that the model results are reasonable and agree with intuition, and to provide some insight into how to effectively manage the border crossing in order to keep the delay within acceptable limits. Owing to the fact that there were minor differences between the $M/E_k/n$ and the $BMAP/PH/n,$ the sensitivity analysis and the subsequent optimization are based on the $M/E_k/n$ for simplicity.

4.3.2.1 Impact of an increasing travel demand, λ

Starting with a base mean travel demand level or volume of $\lambda = 500$ vph, the demand level was increased in increments of 50 vph (up to a demand level of 1000 vph), and the expected queue length and the delay a vehicle joining the queue 20 minutes later would encounter, was calculated. In this test, the number of service stations was assumed to be equal to 3 stations, and the mean service time μ of the Erlang distribution with order 2 was set as 30 seconds. The results are shown in Table 4-3, where it can be seen that there is a significant increase in the delay with increasing volumes. For example, doubling the traffic volume from 500 to 1000 vph would result in an almost fourfold increase in average vehicle delay (from 26.2 minutes/vehicle to 109.4 minutes/vehicle).

Table 4-3 Impact of Increasing the Traffic Demand Level

λ (veh/h)	500	550	600	650	700	750	800	850	900	950	1000
Queue length (no. of	52.4	70.5	84.1	102.7	119.9	135.4	152.2	168.8	186.2	201.9	218.9

vehicles)											
Delay (minutes)	26.2	35.2	42.0	51.3	59.9	67.7	76.1	84.4	93.1	100.9	109.4

Figure 4-8 shows the evolution of the magnitude of the likely delay a vehicle would encounter as the prediction time period changes from 1 to 20 minutes (i.e., the figure shows how the delay a vehicle encounter would change if it arrives 1 minute later versus 20 minutes later). The figure assumes a demand level of 500 vph, an average service time of 30 seconds, and 3 service stations open. As expected, the delay increases for vehicles joining the queue at later time periods. If an agency has a set policy to keep the delay under a certain threshold, for example, Figure 4-8 can be used to determine when additional service stations would need to be opened. For example, if the border crossing authority desires to keep delay below say 10 minutes, and assuming the case shown in Figure 4-8, the agency may need to open an additional inspection station around the 7th minute, when the delay is estimated to reach an average of 10 minutes/vehicle (see Figure 4-8).

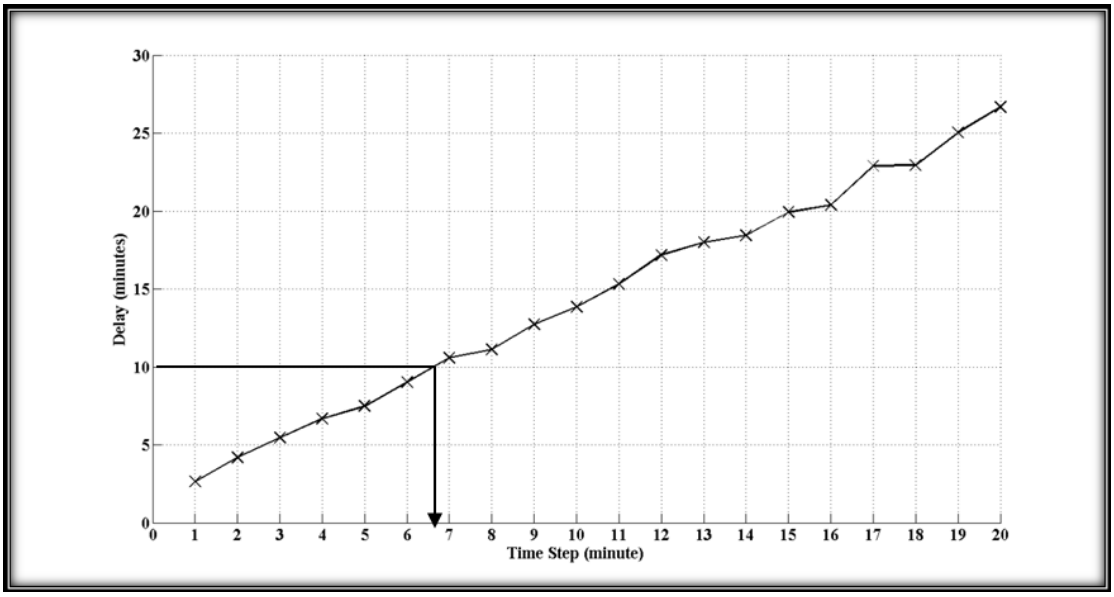


Figure 4-8 Delay Curve of 20 Minutes for $\lambda = 500\ \mathrm{veh/h}$ and $\mu = 30\mathrm{s}$ and $\mathrm{n} = 3$

4.3.2.2 Impact of opening additional service stations, n

Another sensitivity analysis test performed involves varying the number of inspection stations from 3 to 10 for the base scenario considered in Chapter 4.3.2.1 (i.e., arrival rate of 500 vph and an average service time of 30 seconds). The results are shown in Figure 4-9 which plots the delay for the vehicle joining the queue at the end of the 20-minute period. As can be seen, there is a dramatic reduction in the value of the delay as the number of open service stations increase from 3 to 4, and also from 4 to 5 stations. Having more than 5 lanes open, however, does not appear to be quite beneficial from a delay saving standpoint, since the drop in the delay beyond that point is somewhat marginal. Plots such as Figure 4-9 can also be used to determine the number of stations needed to keep delay below a certain threshold. In Figure 4-9 for example, if the agency would like to keep delay around 10 minutes/vehicles, 4 service stations would be needed.

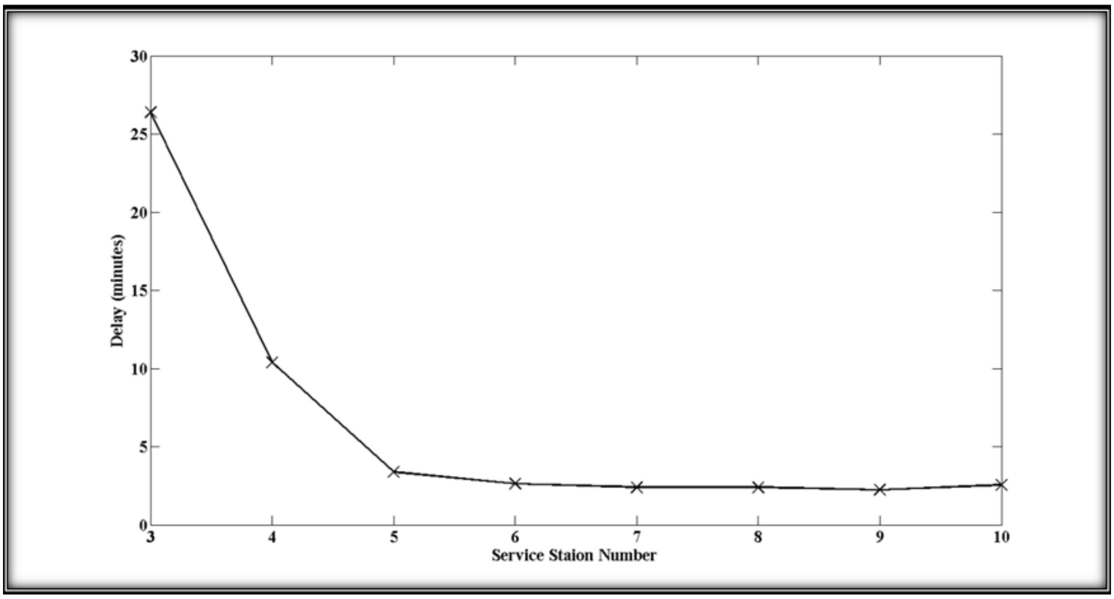


Figure 4-9 Delay at the End of 20 Minutes for Different Service Station Number n with $\lambda = 500\,\mathrm{veh/h}$ and $\mu = 30\mathrm{s}$

4.3.2.3 Impact of changes in the mean service time μ

The last test performed involves varying the value of the mean service time μ of the Erlang Distribution (in increments of 15 seconds), and determining the number of inspection stations needed to keep the delay at the end of the 20th minute below 10 minutes/vehicle. The results are shown in Figure 4-10, where it can be seen that if the mean service time is around 15 seconds, 3 service stations would be adequate. However, if the service time were to increase to 30 seconds, 4 stations would be needed, and so on.

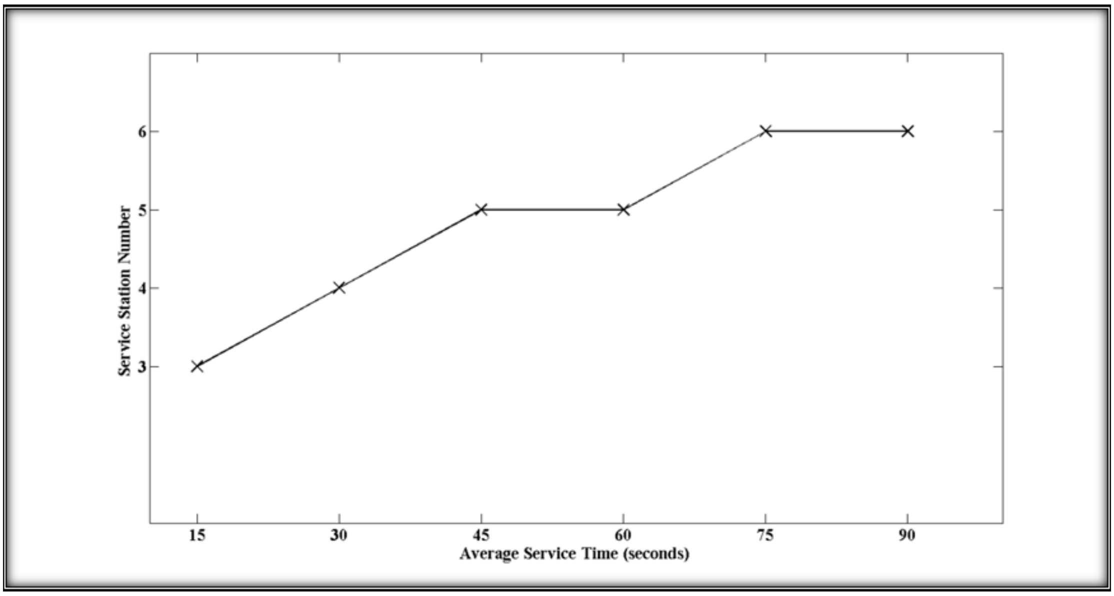


Figure 4-10 Service Station Number for Different Average Service Time of Erlang Distribution

4.3.3 Optimal Operating Policies

The sensitivity analysis described above pointed out the feasibility of using the queueing models to derive "optimal" operating policies for a customs and immigration border control agency. In this Chapter, we include a simple example that demonstrates how the optimization problem may be formulated. We also provide a brief discussion of the results obtained and the insight they provide into the operation of the border crossing system.

4.3.3.1 Optimization Problem Formulation

The goal of the optimization problem formulated herein is to minimize the total cost of the queueing system for a given time period of analysis, *T*, including the cost for both the travelers as well as the operating agency. While doing that, the problem strives to keep the expected delay below a certain threshold. Specifically, we view the total cost as consisting of the following three elements. The first element is the operating cost of opening the inspection stations, calculated by multiplying the assumed hourly cost of operating one booth by the number of booths or inspection stations open by the length of assumed analysis period, *T* (assumed to be 20 minutes in our study). The second element is the cost of the wait time travelers spent waiting in the queue at the border, calculated by multiplying the assumed monetary value for one hour of waiting time by the average number of vehicles in the queue during that time period by analysis period, *T*. The third element is a penalty term designed to capture the cost of switching between an open and a closed inspection lane (or vice versa). Two constraints are included: the first constraint is added to keep the average delay per vehicle below a certain threshold while a second constraint is included to make sure the number of inspection lanes open does not exceed the physical number of lanes available at the border crossing. The problem can thus be mathematically expressed as follows:

$$\min C_t = (C_{ope} * B_t + C_w * V_t) * T + C_{pun} \tag{4-65}$$

s.t.

$\frac{V_t * \mu}{B_t} \leq Th_w,$

$B_{min} \leq B_t \leq B_{max},$

where,

C_t is the total cost of the queueing system during time period *T*;

C_{ope} is the cost per hour to operate one booth;

C_w is the hourly cost of waiting time per vehicle;

B_t is the number of open booths at time period t;

V_t is the average number of waiting vehicles at time period *T*;

μ is the average service time (seconds);

T is the length of the analysis time period;

C_{pun} is the penalty cost for changing the number of open booths from one analysis time period to the next calculated as follows $C_{pun} = c * |B_t - B_{t-1}|$, where c is the penalty for switching for one booth.

$\frac{V_t * \mu}{B_t} \leq Th_w$ is the constraint that ensures that the average waiting time is less than a threshold value, Th_w;

$B_{min} \leq B_t \leq B_{max}$ is the constraint for the number of available booths.

4.3.3.2 Optimization Problem Results

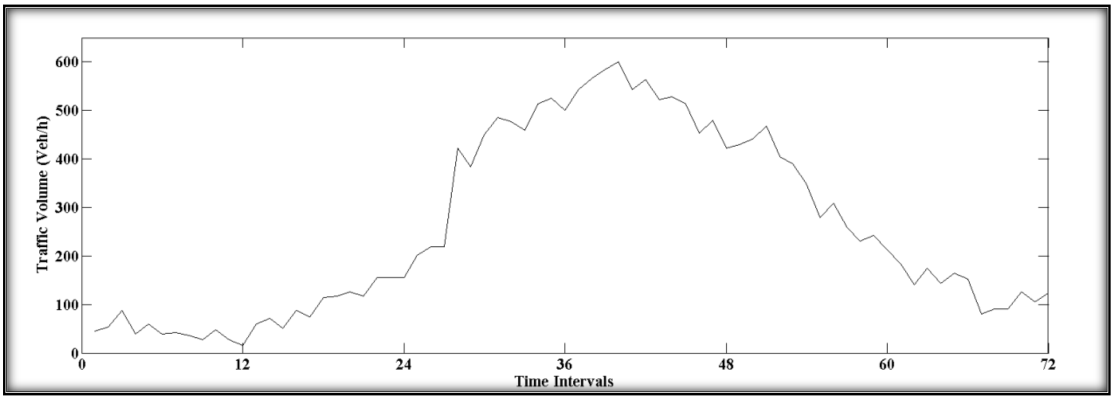


Figure 4-11 Traffic Volume of every 20 minutes' Interval for a Whole Day

To illustrate the application of the model for determining optimal operating policies for the border crossing, we considered the field-measured values for the hourly volumes at the Peace Bridge border crossing for a given day as shown in Figure 4-11, and applied the optimization model formulated above successively for all the successive 20-minute analysis periods of that day (i.e., we would calculate the predicted queue length and the average vehicle delay in 20-minute increments throughout the day and calculate the total cost of the queueing system for that time period). The values assumed for the other model's parameters were as follows. For the hourly operating cost of one booth (C_{ope}), four levels were assumed, a value of \$50 per hour, \$100 per hour, \$150 per hour and a value of \$200 per hour. For the monetary value of one hour of wait time (C_w), knowing the per capita annual income in New York in 2011 is \$31,796 (United States Census Bureau, 2012), and assuming one person works 250 days per year, 6 hours per day and there are 1.2 persons in one vehicle, the monetary value of one hour of wait time (C_w) was estimated to be around \$25. The penalty for switching one booth (c) from closed to open (or vice versa) was set as \$20. Given that the maximum number of inspection stations that can be opened at the Peace Bridge is 10, which meant that $B_{min} = 1$ and $B_{max} = 10$. Two levels of the accepted delay threshold (Th_w) were considered, 10 minutes and 30 minutes.

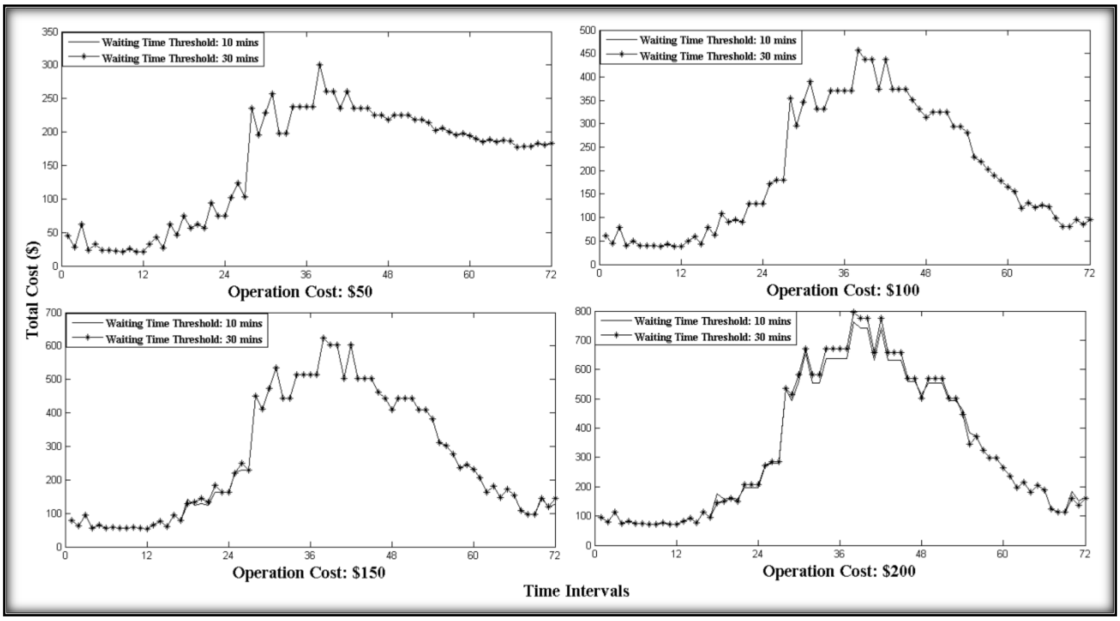


Figure 4-12 Total Cost of Optimizing the Queueing System for a Whole Day

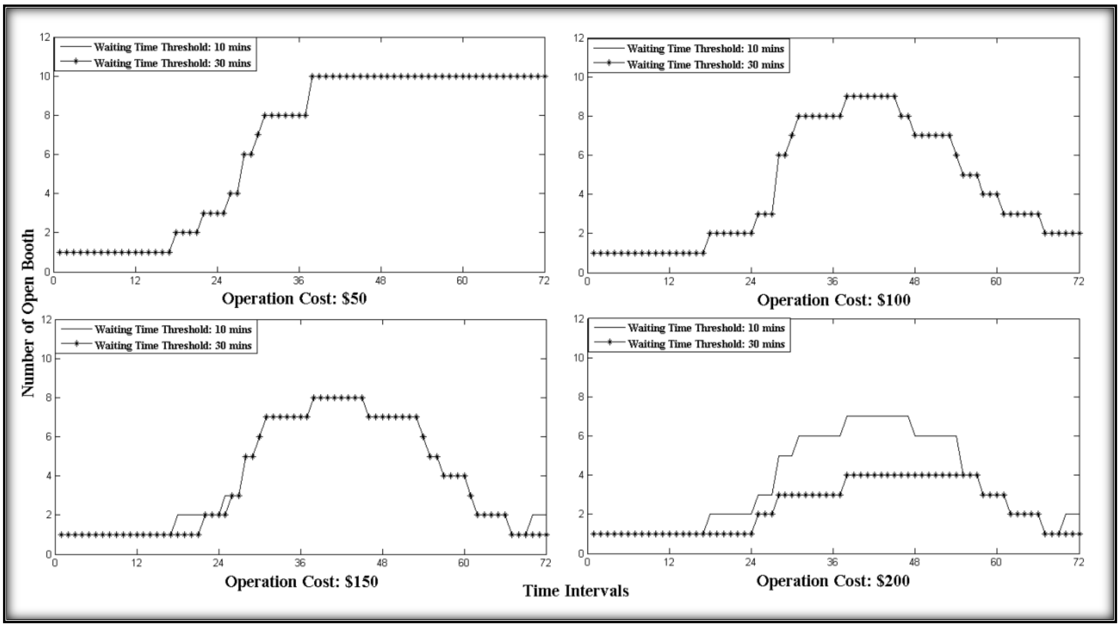


Figure 4-13 Number of Open Booth of Optimizing the Queueing System for a Whole Day

Figure 4-12 plots the values for the total cost obtained after solving the optimization model formulated above from the four assumed levels of hourly cost of opening a service booth (i.e., \$50, \$100, \$150, and \$200) and from two types of waiting time threshold (i.e., 10 minutes and 30 minutes). The model was simply solved by enumeration since the solution space of the problem is quite limited (i.e., it ranged from 1 to 10 for each time period considered), and since the main focus herein is to illustrate the possible applications of the queueing models formulated. Figure 4-13 shows the corresponding number of inspection lanes that are open for the four scenarios.

As can be seen and as to be expected, the total cost increases with the increase in the value assumed for the hourly operation cost (i.e., from \$50 up to \$200), and at the same time, the maximum number of open booths decreases. Specifically when the hourly operation cost (C_{ope}) is equal to \$50 per hour, the maximum number of booths are opened (i.e. 10) and that number remains open for the rest of the day. This is because of the penalty for switching one booth every 20 minutes is set as \$20, which is higher than the operation cost in 20 minutes (\$50/3 or \$16.67). Besides that, when $C_{ope} = \$50$ or $C_{ope} = \$100$, there were no differences in the total cost or in number of open booths when the waiting time threshold (Th_w) was set to 10 minutes versus when it was set to 30 minutes. This is because in these cases the cost of opening new booths is relatively lower than the delay cost for the travelers. As a result, the optimal solution was achieved at an average delay value less than the lower delay threshold (i.e. less than 10 minutes). In other words, the constraint $\frac{V_t * \mu}{B_t} \leq Th_w$ was not binding in this case. However, when C_{ope} is increased, for example, $C_{ope} = \$150$ or $C_{ope} = \$200$, minor differences in terms of the total cost between the two cases were observed (Figure 4-12), along with discernible differences between the numbers of inspection booths open (Figure 4-13). For the number of booths open, as to be expected, when a lower delay threshold is assumed (i.e. 10 minutes), more inspection stations will be needed to mitigate congestion.

Another interesting observation is that when $C_{ope} = \$150$ or $C_{ope} = \$200$, at some time intervals, the total cost for a 30-minute average delay threshold (Th_w) was slightly higher than that for the 10-minute average delay threshold. This is because the increase in the waiting cost of the vehicles, when a higher delay threshold was allowed, slightly outweighed the savings in the operating cost resulting from opening fewer lanes.

4.4 Conclusions

This study has formulated two groups of multi-server queueing models to predict border crossing delay, namely $M/E_k/n$ and *BMAP/PH/n* queueing models. The models were formulated based on real-world data collected from the Peace Bridge, and their transient solution was numerically derived based on heuristic approaches. The solution derived was then validated by comparing the results to those estimated from a well-calibrated microscopic traffic simulation model. Sensitivity analyses were performed to check the validity of the model's predictions and to gain insight into how to effectively manage the border crossing. To further demonstrate the potential applications of the models, they were incorporated within an optimization framework and used to derive optimal management strategies for a customs and immigration border control agency. Among the main conclusions of the study are:

(1) The transient solution of the $\mathrm{M}/E_K/n$ queueing models derived using the ELC heuristic appears to agree quite well with the values determined using the microscopic simulation models. The real advantage of the queueing modeling approach, however, is that its runtime is a fraction of the time needed to run the microscopic simulation model multiple times and gathering the required statistics;

(2) For the case study considered herein, the results of $BMAP/PH/n$ queueing models appear to be quite similar as to $\mathrm{M}/E_K/n$ queueing models except for the case when the number of server stations was small;

(3) The sensitivity analysis tests clearly demonstrate the reasonableness of the queueing model solution. Moreover, it shows that the models can be used to gain insight into how to best manage the border crossing; and

(4) The solution of the border management optimization problem described in Chapter 5 shows that when the hourly cost of opening and operating a new

inspection station is low, it becomes advantageous to open more lanes so as to keep the delay, and the associated wait time cost, on the low side. The case study considered also demonstrates that the solution of the optimal border crossing management strategy problem is quite sensitive to the assumptions regarding the cost of operating the inspection stations and the monetary value of waiting time.

CHAPTER 5 ANDROID SMARTPHONE APP FOR TORONTO BUFFALO BORDER WAITING

In recent years, one factor that deserves consideration is the emergence of social media applications using smartphones which allow people to easily create, share and exchange information. For example, Waze is a community-based traffic and navigation app, acquired by Google in 2013, where drivers can share real-time traffic and road information, saving travel time, gas and money on their daily commute (Waze, 2014). In this context, smartphone apps have become another important means for the public to get and share traffic information, in addition to the traditional methods such as variable message sign (VMS), radio, websites, and toll free phones. Given the above, the opportunity now exists to integrate the useful insights mined from traffic data with state-of-the-art techniques such as smartphone applications to provide accurate information about border crossing delay.

This Chapter introduces an Android smartphone application (app) called the Toronto Buffalo Border Wait Time (TBBW), which is designed to share waiting time among travelers of the three Niagara Frontier border crossings, namely the Lewiston-Queenston Bridge, the Rainbow Bridge, and the Peace Bridge. Three types of waiting times are offered based on users' preferences, including the current waiting time, the historical waiting time, and the future waiting time predicted by a real-time traffic delay prediction model. For the current waiting time, the app can provide both the data collected by the border crossing authorities and the user-reported or “crowd-sourcing” data shared by the community of users of the app. For the historical waiting time, the app provides statistical charts and tables to help users choose the crossing with the likely shortest waiting time. Moreover, the app can also provide future border waiting time for the next 15 minutes with an updating frequency of five minutes. The future waiting times are predicted by the two-step border crossing delay prediction model introduced in Chapter 3 and Chapter 4.

Figure 5-1 summarizes the characteristics of TBBW (shown in the green color), in comparison with the existing border crossing delay dissemination method (shown in the blue color). As can be seen, TBBW provides several options for border crossing delay estimates including, user-reported or “crowd-sourcing” wait time, historical, and future wait time, in addition to the current waiting time reported by the authorities. Travelers and border management authorities can then make better decisions based on this information.

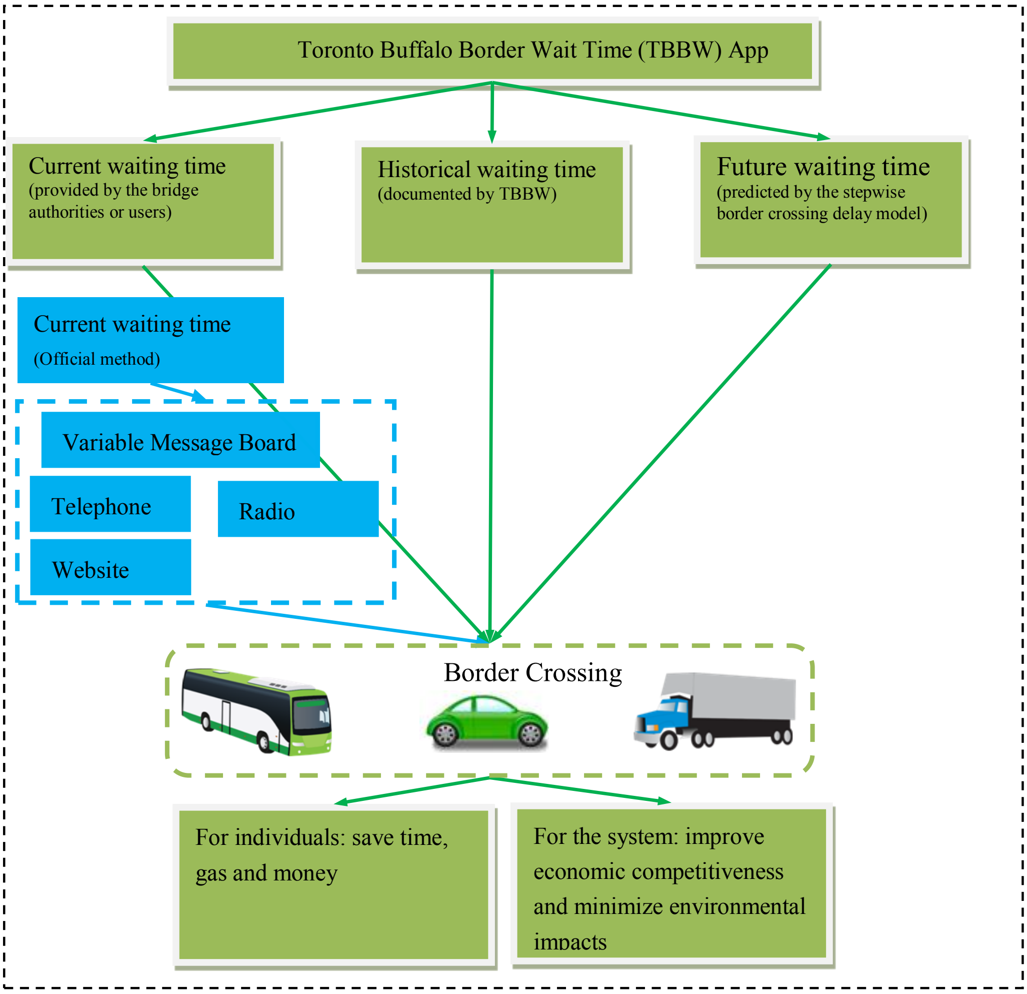

Toronto Buffalo Border Wait Time (TBBW) App
Current waiting time (provided by the bridge authorities or users)
Historical waiting time (documented by TBBW)
Future waiting time (predicted by the stepwise border crossing delay model)
Current waiting time (Official method)
Variable Message Board
Telephone
Radio
Website
Border Crossing
For individuals: save time, gas and money
For the system: improve economic competitiveness and minimize environmental impacts

Figure 5-1 Comparison of TBBW with the Other Ways to Share Border Waiting Time

5.1 Methodology behind the App

5.1.1 Integration of Data-driven Model and Analytical Model for Future Border Crossing Waiting Time

As mentioned earlier, this app has utilized the two-step border crossing delay prediction model to calculate the future border crossing waiting time. The two-step border delay prediction model is composed of two sequential modules. The first module is a data-driven short term traffic volume prediction model designed to predict the *traffic volume* arriving at the border crossings for each time period. Given the predicted traffic volume as input, the second model estimates the corresponding waiting time by solving an analytical transient multi-server queueing model.

5.1.1.1 Border Crossing Traffic Volume Prediction Module

In this app, however, SARIMA is chosen as the prediction method because of its easiness of implementation and its moderate computational cost. As previously reported in this dissertation, for a testing dataset with 1,905 hourly traffic volume points, the mean absolute percentage error (MAPE) was found to be equal to 16.38%. It needs to be noted here that the short-term traffic volume prediction module were built using data collected from the Peace Bridge, particularly the traffic volume entering the U.S. from Canada, due to the fine temporal resolution available (i.e., on the hourly basis) (Buffalo and Fort Erie Public Bridge Authority, 2014). The traffic volumes for the other bridges were only available to the study on a daily basis at the time (Niagara Falls Bridge Commission, 2014), and were thus deemed not sufficient for accurate waiting time prediction.

5.1.1.2 Transient Multi-server Queueing Module

Because the TBBW app requires that the predicted wait time be updated every five minutes, the predicted hourly traffic volume was split into a finer resolution (e.g., a five-minute resolution) before they were used for border wait time prediction by the queueing models. With the inter-arrival distribution known, this was done using the inverse cumulative function of the inter-arrival exponential distribution $F(x) = 1 - e^{-\lambda x}$, where λ is the predicted hourly volume or arrival rate.

Other input requirements of the queueing model included the number of inspection booths. However, the number of open inspection stations is typically not available ahead of time. To solve the problem, our approach at the moment involves running the queueing model for different numbers of open stations (1 to 10 in this study), and trying to estimate how many stations are actually open. Other venues to be explored in the near future are information offered by users or directly by the U.S. Customs and Border Protection.

Although the more general queueing model $BMAP/PH/n$ was also solved to get the transient results of the queueing system in Chapter 4.2.2, due to the computational time constraints in the real-world application, only the $M/E_k/n$ queueing model was implemented in the Android app.

5.1.2 Data-level Fusion for Current Border Crossing Waiting Time

For the current border crossing waiting times, there are two types of data sources. One is from the official border crossing authorities, and the other one is from the app users based on the idea of crowd sourcing. This fits the definition of data-level fusion mentioned in Chapter 1.3.1.2. The detailed introduction about the ways to share the current border crossing waiting time will be talked about in the following Chapter.

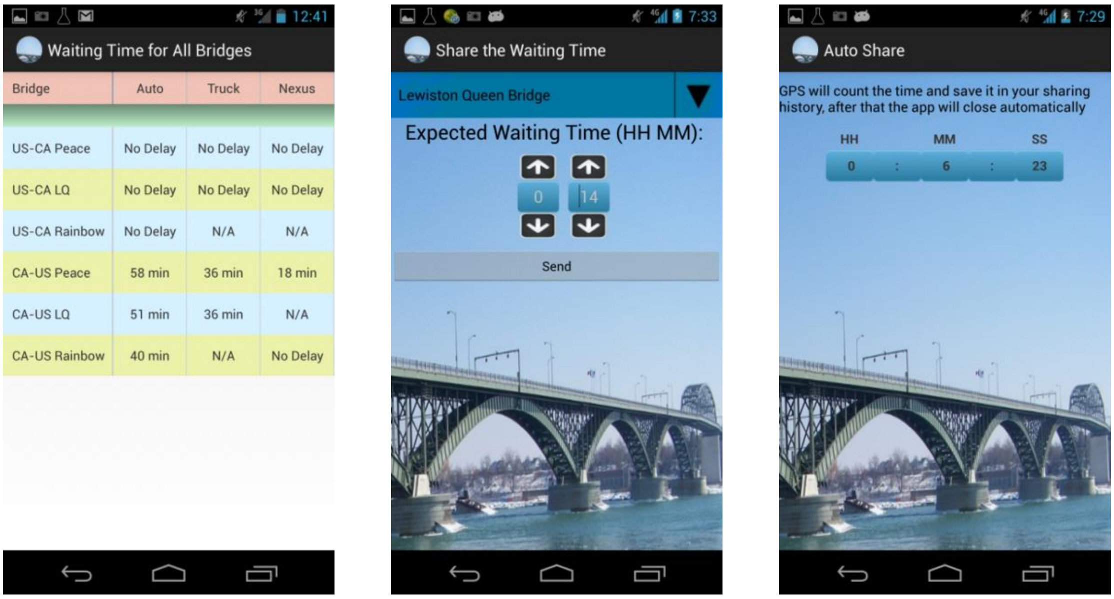
5.2 Datasets

Two types of data are used to develop the TBBW app. The first dataset contains the hourly traffic volume data collected at the Peace Bridge since 2003. This is used as the input to develop the stepwise border delay prediction model and to predict the future waiting times. The second dataset captures the current waiting times collected and maintained by the border crossing authorities. Such data are used as one source of the current waiting times provided by the app. In addition, they are stored for historical data analysis and also used as the ground truth to assess the performance of the border delay prediction model. All data are available for download from the websites maintained by the Peace Bridge authority and Niagara Falls Bridge Commission.

5.3 Innovative Features

The TBBW app was developed on the Android platform, the most popular mobile operating system used in the U.S. Specifically, in the third quarter of 2013, Android's share of the global smartphone shipment market was 81.3%. As of July 2013, there are more than one million apps available for Android in the Google Play Store where more than 48 billion apps have been downloaded by users as of May 2013 (Wikipedia, 2014). The developed TBBW app is innovative in terms of its ability: (1) to share current waiting time; (2) to store and analyze historical waiting time; and (3) to predict future waiting time, as described below.

5.3.1 Sharing Current Waiting Time Function



Waiting Time for All Bridges

Bridge	Auto	Truck	Nexus
US-CA Peace	No Delay	No Delay	No Delay
US-CA LQ	No Delay	No Delay	No Delay
US-CA Rainbow	No Delay	N/A	N/A
CA-US Peace	58 min	36 min	18 min
CA-US LQ	51 min	36 min	N/A
CA-US Rainbow	40 min	N/A	No Delay

Share the Waiting Time

Lewiston Queen Bridge

Expected Waiting Time (HH MM):

0 14

Send

Auto Share

GPS will count the time and save it in your sharing history, after that the app will close automatically

HH	MM	SS
0	6	23

Figure 5-2a Official Website

Figure 5-2b Manually Share

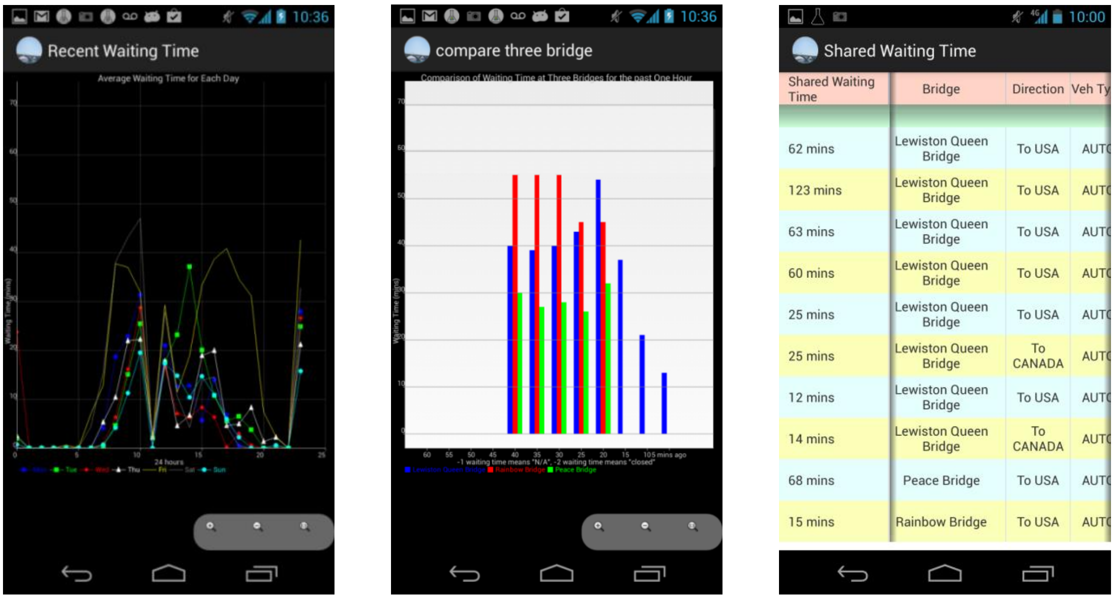
Figure 5-2c Automatic Share (GPS)

Figure 5-2 Three Ways to Share Current Waiting Time

The app employs two ways to collect current waiting time information. The first way involves downloading the waiting time data from the websites maintained by the Buffalo and Fort Erie Public Bridge Authority and the Niagara Falls Bridge Commission. The current waiting time for Peace Bridge and Lewiston Queen Bridge are provided and updated every five minutes, and for Rainbow Bridge, it is updated every one hour. The information is collected and uploaded in real time to the app as shown in Figure 5-2a.

Because the official current waiting time data is lagged (particularly for the Rainbow Bridge where it is only updated every hour), the app also provides a second way to collect the current waiting time data utilizing crowd sourcing ideas. Specifically, users are allowed to report their *experienced* border crossing delays that can be then processed and broadcasted to other users for their benefits (called crowd sourcing). The same concept has been widely applied in other traffic information sharing apps, such as the pre-mentioned Waze and the bus arrival time sharing app Tiramisu (Zimmerman et al., 2011). In TBBW, users can share their waiting times by manually inputting the data as shown in Figure 5-2b. They can also choose to automatically share their waiting times through their GPS-enabled smartphones as shown in Figure 5-2c. This option is necessary because if users are driving, it is unsafe and illegal to manually input waiting time.

5.3.2 Utilizing Historical Waiting Time Function



Shared Waiting Time	Bridge	Direction	Veh Ty
62 mins	Lewiston Queen Bridge	To USA	AUT
123 mins	Lewiston Queen Bridge	To USA	AUT
63 mins	Lewiston Queen Bridge	To USA	AUT
60 mins	Lewiston Queen Bridge	To USA	AUT
25 mins	Lewiston Queen Bridge	To USA	AUT
25 mins	Lewiston Queen Bridge	To CANADA	AUT
12 mins	Lewiston Queen Bridge	To USA	AUT
14 mins	Lewiston Queen Bridge	To CANADA	AUT
68 mins	Peace Bridge	To USA	AUT
15 mins	Rainbow Bridge	To USA	AUT

FIGURE 5-3a Average Waiting Times for Each Day of Week for Each Bridge

FIGURE 5-3b Comparison of Waiting Times at Three Bridges for the Past Hour

FIGURE 5-3c Waiting Times Sharing History by the registered user himself/herself

Figure 5-3 Three Ways to Utilize Historical Waiting Time

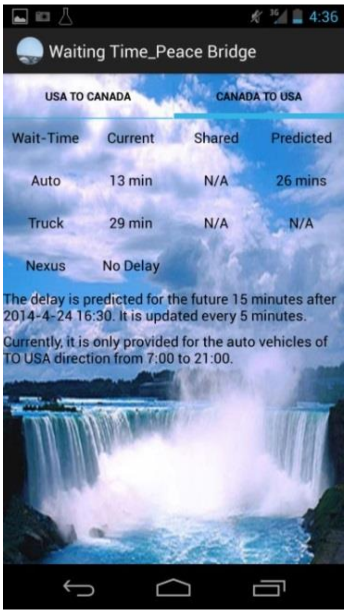
Mining and analyzing historical border crossing waiting time data in a proper manner can provide additional insight to travelers. In TBBW, three types of graphs and charts are created based on an underlying historical waiting time database.

As can be seen in Figure 5-3a, for each bridge, the average waiting times for each day of week are calculated and shown in one chart. This is the long term trend based on the historical data of the past month. The TBBW app also allows the users to compare the waiting times of the three bridges, based on the historical data of the past one hour, as shown in Figure 5-3b. Finally, because users may want to make decisions based on their own previous experiences, registered users can view their waiting times as another reference as shown in Figure 5-3c.

5.3.3 Predicting Future Waiting Time Function

Finally, in addition to current and historical analyses of wait times, the app is designed to *predict* the likely waiting time in the next 15 minutes (this estimate is also updated every 5 minutes). Predicting is based on utilizing the stepwise border crossing delay prediction model.

The TBBW interface of the predicted waiting time for passenger vehicles from Canada to U.S. through the Peace Bridge is shown in Figure 5-4.



Wait-Time	Current	Shared	Predicted
Auto	13 min	N/A	26 mins
Truck	29 min	N/A	N/A
Nexus	No Delay		

The delay is predicted for the future 15 minutes after 2014-4-24 16:30. It is updated every 5 minutes.

Currently, it is only provided for the auto vehicles of TO USA direction from 7:00 to 21:00.

Figure 5-4 Predicted Border Crossing Waiting Time

In order to test the prediction performance of the stepwise delay prediction model, the research compared the predicted waiting times with the historical waiting times recorded by the border authorities from 7:00 AM to 9:00 PM for each day of the whole month of May, 2014. Because the future waiting time is updated every five minutes, there should be a total of 5,580 predicted values for the month. However, because of several missing data points from the field observations (e.g., when the server was down and the official waiting time was recorded as “N/A”), a total of 3,103 observations were deemed valid for assessing the prediction model’s performance.

Table 5-1 Prediction Performance of the Two-step Delay Prediction Model

Data Group	Number of data points	Mean Absolute Difference (minutes)
Whole Dataset	3,103	9.22
Officially Recorded Waiting Time = 0 minutes (denoting less than 10 minute delays)	2,363	9.94

Officially Recorded Waiting Time >=10 minutes	740	6.95

The mean absolute difference (minutes) between the predicted waiting times and the officially recorded waiting times is shown in Table 5-1. As can be seen, the mean absolute difference for the whole dataset is 9.22 minutes. After checking the officially recorded waiting times, we find that there were a total of 2,363 data points where the wait times was recorded as being equal to 0 minutes, and the remaining 740 points had delays greater than or equal to 10 minutes. After discussions with the border crossing authorities, it was revealed that their practice was to report any wait time which was less than 10 minutes as 0 minutes delay. Given this, and in order to provide for a true evaluation of the predictive model accuracy, the testing dataset was split into two groups. The first group (2,363 data points) had an official reported delay of 0 minutes, which meant that the delay could be anywhere between 0 and 10 minutes. For that group, the mean absolute difference between the model's predictions and the officially reported delay times was as high as 9.94 minutes (it should be clear now that that absolute error is exaggerated, since the actual delay could have been anywhere between 0 and 10 minutes). The second group included points where the officially reported wait time was greater than or equal to 10 minutes. For that second group, the mean absolute difference was only 6.95 minutes.

For a more disaggregate view of the performance of the delay prediction model, the predicted waiting times and the historical waiting times for the peak hours 18:00-20:00 on April 22, 2014 are compared and shown on Figure 5-5.

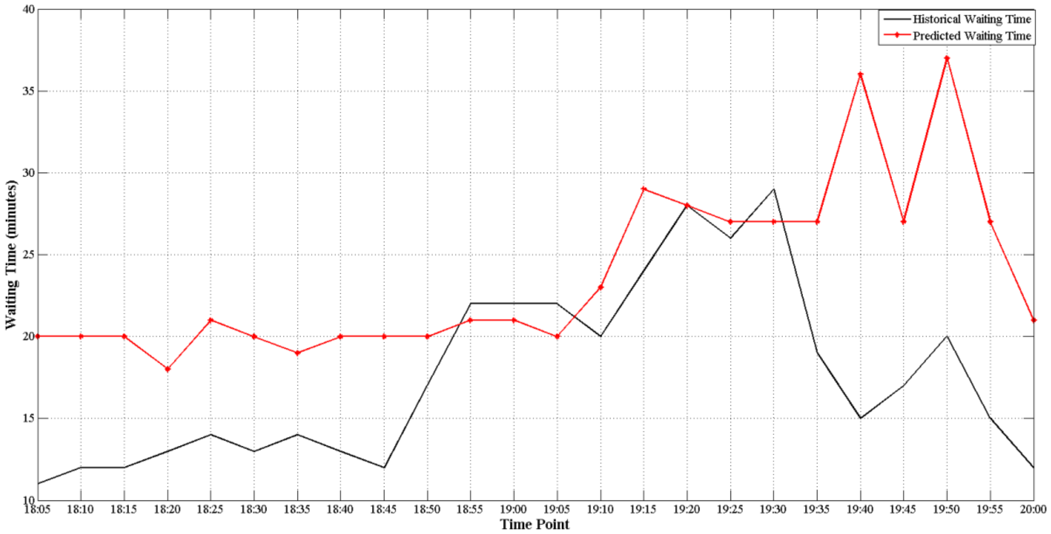


Figure 5-5 Prediction Performance for the Peak Hours of 18:00-20:00 on April 22, 2014

As can be seen in Figure 5-5, the mean absolute difference between the predicted waiting times and the observations is about 6.6 minutes. Most of the time, the difference is within 10 minutes, except for 19:40 for which the difference is around 20 minutes. This is most probably the result of the opening of additional inspection stations at that time without the model being aware of that (the reader may recall that there is currently no easy way for the app to discern the actual number of inspection stations open; it is hoped that in the future such information may be obtained from the Customs and Border Protection agencies). Another reason could be that the historical waiting time detected by the Bluetooth technology is lagging in time, since the Bluetooth technology provides an estimate of the delay at the time a vehicle had joined the queue some time prior to the reporting time (that time is actually equal to the time it took the vehicle to exit the system).

5.4 Comparison with Other Border Crossing Apps

A detailed comparison was conducted to demonstrate the innovative features of the TBBW app. A few observations can be made based on Table 5-2 below. First, although all other border crossing apps provide the waiting time for all border entries from Mexico to U.S. and from Canada to U.S., none of them provide the waiting time for the travelers leaving the U.S. through those borders. This is most probably because those apps all depend upon the data downloaded from the same U.S. Customs and Border Protection website that only provides the waiting time for travelers coming into the U.S. (U.S. Customs and Border Protection, 2014). In contrast, TBBW provides the waiting time for both directions. It also would be a simple extension to expand TBBW to include all the Canada-US and Mexico-US borders from that website.

Table 5-2 Comparison of TBBW with Other Border Crossing Apps

App name	Border Location		Bi-direction	Update Interval	Waiting Time Provided or Not			Login System	Crowd Sourcing	
	From	To			Current	Historical	Future		Manual	GPS
TBBW	Toronto	Buffalo	Yes	5 min	Yes	Yes	Yes	Yes	Yes	Yes
	Buffalo	Toronto								
Best Time to Cross Border	Mexico	US	No	5 min	Yes	Yes	No	No	Yes	No
	Canada	US								
Border Wait Times	Mexico	US	No	N/A	Yes	No	No	No	No	No
	Canada	US								
Border	Mexico	US	No	N/A	Yes	No	No	No	No	No

Times	Canada	US								
Border Info	Mexico	US	No	N/A	Yes	No	No	No	No	No
	Canada	US								
Border Check	Mexico	US	No	N/A	Yes	No	No	No	No	No
	Canada	US								

Second, all the apps provide the current waiting time downloaded from the official authorities, but only the “Best Time to Cross Border” app and our own app provide “crowd sourcing” based waiting time. The difference between TBBW and the “Best Time to Cross Border” app, however, is that TBBW also provides one more option, that is, the ability to share the waiting time automatically based on GPS location tracking, in addition to the ability to manually input waiting time data. The fact that TBBW can automatically calculate and report wait time is a huge advantage over the need for users to manually input the wait time themselves.

Third, only the “Best Time to Cross Border” and TBBW utilize historical data. The “Best Time to Cross Border” app can produce a seven day comparison graph that compares the average waiting time for each day of the week, and the individual day graph that shows the maximum, minimum and average waiting time at each hour for each day of the week. In addition to those functionalities provided by the “Best Time to Cross Border” app, TBBW offers an additional feature of comparing the waiting times at the three bridges for the past hour, and can also show the sharing history for registered users.

Fourth, TBBW is the only app that has registration and login functionalities. While unregistered users can still enjoy all the functions of this app, the ability to share waiting time with others is restricted to registered users. The registration and login function is deemed useful for a number of reasons. First, it can decrease the risk that some people share inaccurate waiting time intentionally. Second, through registration, users can receive the latest notification about border crossing traffic conditions. Lastly, as mentioned previously, the sharing history of registered users can be recorded in the database, which can help them make decisions based on their own experiences.

Finally, TBBW is the only one that can provide future waiting times. This is perhaps the most significant advantage of TBBW, in contrast to other border crossing apps that simply download and list the waiting times to the users. As shown in the previous Chapter, the stepwise border crossing delay prediction model, which is the engine behind this function, has been tested and has a promising performance.

5.5 Risks and Challenges

This Chapter will summarize the risks and challenges encountered while developing the app. Some of those challenges have been addressed, while others are left for future work.

5.5.1 The Need for More Data

A critical piece of information for wait time prediction which is missing at this point is the number of open lanes or inspection booths. Although the delay prediction model can estimate the number of open lanes, it would be better and more accurate if the real value were to be provided by the U.S. Customs and Border Protection agencies.

The authors plan to work with multiple agencies to explore methods for acquiring such data in the future.

5.5.2 Crowd Sourcing

As with any contribution-based crowd sourcing information system, a risk exists of low motivation to participate and of abuse (Steinfeld et al., 2011). To overcome this problem for TBBW, one can design a set of reward and penalty rules on the basis of the registration and login function. For example, when users share their border crossing waiting time with others, they can get some virtual points, and every period of time the user with the highest rank may be rewarded. Abuse can also be prevented through penalties. For example, users who intentionally share wrong border crossing waiting times can be identified and filtered by setting a threshold for the difference between the value provided by the user and a "best" estimate based on a combination of the officially reported waiting time and the average waiting time from other users. Users who abuse the system may also be restricted from sharing information.

5.5.3 GPS Location

Some privacy concerns may arise regarding the ability to share waiting time in an automatic fashion through the GPS location sharing function. To address this, the TBBW app was designed so that it does not store any of the users' GPS locations data; these data are only used to calculate the distances of the travelers from the borders and their speed, so an approximate waiting time can be estimated.

5.6 Conclusions

This Chapter introduced an android app TBBW which combines sophisticated transportation models with emerging mobile computing technologies to solve the wait time border crossing problem. The result is an app which can provide: (1) current border waiting time based on either officially reported data or user-shared waiting time based on crowd sourcing (manually input or automatically shared by GPS); (2) historical analysis of waiting time to further help users decide the “best” border crossing bridge; and (3) future or predicted border waiting time for the next 15 minutes. The performance of the prediction model was assessed by comparing its predictions to those reported by the authorities for month of May, 2014. The comparison demonstrated that the predictions are quite accurate, with a mean absolute difference of only 6. 95 minutes for delays greater than or equal to 10 minutes.

CHAPTER 6 DATA MINING AND COMPLEX NETWORKS ALGORITHMS FOR TRAFFIC ACCIDENT ANALYSIS

This Chapter starts to introduce the work on transportation accident data analysis, following by the Chapter 7 and Chapter 8. As mentioned in Chapter 1.5, this study is another integration of data "depth" decreasing (dataset clustering) and model development besides the dataset grouping before applying the traffic prediction models. Chapter 6.1 introduces the dataset clustering method called modularity-based community detection, and the data mining algorithm-the association rules learning for traffic accident hotspots and clearance time analysis. After that, the dataset in this study is introduced and processed in Chapter 6.2. The detailed clustering results and the analysis of the association rules for hotspots and clearance time are recorded in Chapter 6.3. Finally, Chapter 6.4 discusses the conclusions.

6.1 Methodology

6.1.1 Dataset Clustering

Recently, complex network analysis methods have been intensively used to understand the features of complex systems such as biological, social, technological and information networks. In the analysis, communities, also called clusters or modules, denote groups of system components that probably share common properties and/or play similar roles in graphs (Fortunato, 2010). For example, for a Facebook social network, communities represent people who share common interests, and therefore exploiting the affiliations of users to these communities provides an effective way to provide them with targeted recommendations and advertisements (Ferrara, 2012). For these methods to work, however, the problem needs to be formulated in the form of a network graph.

The modularity optimization method is one of the most popular methods used for community detection in graph and network analysis (Fortunato, 2010). Its premise is that the network is divided the best when the modularity (i.e., the degree to which a system's components may be divided) is maximized. Due to the generality of the method, the concept of modularity optimization can be applied to traffic accident clustering, by representing each accident record as one node in the network (analogous to a person in a social network).

Suppose the accident dataset contains N records, each of which contains information about a set of variables $A = \{c_1, c_2, \dots c_m, a_1, a_2, \dots a_n\}$. We divide those variables intro two groups: (1) the c_l variables, $1 \leq l \leq m$, which represents the causative factors behind the accident such as time of day, weather conditions, road geometric features (e.g. number of lanes), etc.; and (2) the accident attributes, a_k, $1 \leq k \leq n$, which represents the specific characteristics of a crash such as associated injuries, location, incident clearance time, etc..

6.1.1.1 Modularity-based Community Detection

This study used the *community detection* algorithm, for the first time, to cluster the data and reduce heterogeneity. The first step was to represent the data in the form of the network by treating each accident record as one vertex in the network (similar to a friend in a Facebook network). Then, the problem becomes to find out how these vertices are connected in the network. Because in this study the objective is to find out how causative factors contribute to the outcome (i.e. the accident characteristics), the grouping is based on the causative factors (i.e. the c_l variables).

According to the algorithm, two vertices (i.e. two accidents) i and j, $1 \leq i, j \leq N, i \neq j$ will be connected if the following condition is satisfied:

$$\sum_{1 \leq l \leq m} e_l \geq e_{th}, \quad (6\text{-}1)$$

where, $e_l = 1$, if the values of the factor c_l of i and j are the same, otherwise $e_l = 0$, and e_{th} is the similarity threshold defined by the user (i.e. this counts how many attributes are similar). If the two vertices i and j are connected, an undirected edge is drawn between them, and the weight of the edge can be calculated as:

$$W_{ij} = \frac{\sum_{1 \leq l \leq m} e_l}{m}, \quad (6\text{-}2)$$

Following the network formation, the *community detection* algorithm is applied to divide it into clusters so that each vertex belongs to only one cluster. The most popular quality function of a partition is the modularity of Newman and Girvan (2004), which can be calculated as following:

$$Q = \frac{1}{2T} \sum_{i,j} [W_{ij} - \frac{f_i f_j}{2T}] \delta(o_i, o_j), \quad (6\text{-}3)$$

where,

W_{ij} represents the weight of the edge between vertex i and j;

$f_i = \sum_j W_{ij}$ is the summation of the weights for the edges attached to vertex i;

o_i is the index of community or cluster vertex i is assigned to in a given iteration, and $\delta(o_i, o_j) = 1$, if $o_i = o_j$, otherwise $\delta(o_i, o_j) = 0$;

and $T = \frac{1}{2}\sum_{i,j} W_{ij}$.

As defined above, the *modularity* basically reflects the concentration of vertices within communities compared with random distribution of edges between all vertices regardless of communities. A positive modularity means that the weights of the edges within the communities exceed the weights expected on the basis of chance, and this is the main motivation behind maximizing modularity. However, because it is too difficult to enumerate and test all the ways to partition a graph, algorithms such as the one proposed by Blondel et al. (2008) for the fast unfolding of the communities are needed. Blondel et al.'s algorithm was the one utilized in this study (Blondel et al., 2008; Arenas et al., 2007).

As compared to traditional clustering techniques such as LCC and K-means clustering mentioned in Chapter 2.2.2, the *community identification* algorithm offers several advantages. First, the network transformation and the modularity optimization method are intuitive and easy to implement. Second, when building the network, because the causative factors are compared one by one and because there is no distance measure involved, as is the case with other techniques such as *K-means*, there is no need to normalize the data (which often introduces imprecision). Third, unlike the LCC method, the modularity optimization algorithm does not rely on the assumption of the independence among variables to decrease the complexity of computation; instead, it is extremely fast since the number of possible communities decreases drastically after a few iterations (Blondel et al., 2008). Fourth, the method provides a modularity based quality function, which can be used to measure the effect of clustering. Finally, the method, even for large dimensional datasets, requires the specification/calibration of only one parameter, the threshold e_{th}, as opposed to classical statistical analysis methods where the number of parameters may exponentially increase as the number of variables increases (Chen and Jovanis, 2000).

6.1.2 Hotspots and Clearance Time Analysis

6.1.2.1 Association Rule Learning

The concepts of association rules learning were firstly introduced by Agrawal et al. (1993). Given a traffic accident related variable set $A = \{c_1, c_2, \dots c_m, a_1, a_2, \dots a_n\}$, it can be transformed to a set of binary attributes called items $I = \{I_{1c}, I_{2c}, \dots I_{Lc}, I_{1a}, I_{2a}, \dots I_{Ka}\}$, where $I_{lc},\ 1 \le l \le L$ are the binary attributes associated with the causative factors, and $I_{ka},\ 1 \le k \le K$ are the binary variables related to accident attributes (i.e. the outcome). For example, the factor "Season" can be represented by four binary attributes, i.e., "spring", "summer", "autumn", and "winter". Each of the N accident records, referred to here as transactions T, has a unique transaction ID and is a subset of I. An association rule is an implication of the form, $X \Rightarrow Y$, where X and Y are sets of items in I, $X \subset I$, $Y \subset I$ and $X \cap Y = \varnothing$.The sets of items X and Y are called the body and head of the rules, respectively.

At a very high level, generating the association rules involves two basic steps. The first is to generate the frequent item sets in the data. X is called a frequent item set when its support, which refers to the frequency at which X appeared in the N transactions, is equal to or greater than the minimum support defined by user.

$$\frac{supp\{X\}}{N} \geq \sigma, \tag{6-4}$$

where,

$supp\{X\}$ is the number of transactions in N that contains item set X, and σ is the minimum support.

Now suppose item sets X and $X \cup Y$ are frequent item sets, the second step is to calculate the confidence of $X \Rightarrow Y$. This is based on the ratio of the number of transactions that contains $X \cup Y$ to transactions that only contains X. If the confidence is equal to or higher than the user-defined minimum confidence, $X \Rightarrow Y$ is an association rule.

$$conf(X \Rightarrow Y) = \frac{supp\{X \bigcup Y\}}{supp\{X\}} \geq \varepsilon, \tag{6-5}$$

where,

ε is the minimum confidence. Methods are then available to distinguish between the trivial and non-trivial rules (Geurts et al., 2003).

6.2 Dataset Processing

The dataset used in this study included 999 traffic accidents observed at I-190 from 01/01/2008 to 10/31/2012. I-190 runs 28.34 miles (45.61 km) from its interChapter with I-90 near Buffalo, NY up north to Lewiston, NY via Niagara Falls. I-190 plays a critical role in the Buffalo-Niagara transportation network, especially in terms of connecting Western New York to Southern Ontario, Canada. Incidents and traffic flow are monitored by the Niagara International Transportation Technology Coalition (NITTEC), which serves as the region's Traffic Operations Center (TOC). Incident details are recorded every day through detailed incident log forms, which formed the basis for compiling the dataset used in this study. Table 6-1 lists both the causative factors and accident attributes variables that were available in NITTEC's incident logs, and were thought to be useful for analysis. After initial screening of the data, a total of 15 variables were selected (nine causative factors and six accident attributes) as shown in Table 6-1. The variables that were excluded did not exhibit enough variation over the dataset compiled (i.e., more than 95% of the records had the same value for the variable).

Table 6-1 Traffic Accident Variables in the I-190 Data

Variables	Values	Included
Causative Factors		
Season	Spring (March, April, May); Summer (June, July, August); Autumn (September, October, November); Winter (December, January, February)	Yes
Weekday	Yes (Monday 2 AM-Friday 9 PM, except holidays); no	Yes
Hour of the Day	morning (7 AM-9 AM); early afternoon (10 AM-12 Noon); afternoon (1 PM-3 PM); evening rush (4 PM-6 PM); evening (7 PM-9 PM); night (10 PM-6 AM)	Yes
Wind Speed	0 mph (miles per hour); 10 mph; 20 mph; 30 mph	Yes
Weather Conditions	clear; rain; snow	Yes
Direction	North; South	Yes
Lane Number on Main Road	1; 2; 3	Yes
Lane Number on Ramp	0 (away from exit); 1; 2;	Yes
Ramp Type	on ramp; off ramp; highway to highway on ramp; highway to highway off ramp;	Yes
Vehicle Type	Car; Truck/Tractor Trailer; Motorcycle	No
Accident Attributes:		
Location – Exit Number	Exit 1; …; Exit 25; Highway	Yes
Location relative to Road Configuration	Before the exit; at the exit; beyond the exit; highway; ramp; bridge; before the bridge; after the bridge	Yes
Number of Vehicles Involved	1; 2; more than 2	Yes
Clearance Time	0-15minutes; 16-30 minutes; 31-45 minutes; 46-60 minutes; 61-75 minutes; 76-90 minutes; more than 90 minutes	Yes
Blocked Lane Index	left lane at main road; middle lane at main road; right lane at main road; all lanes at main road; left lane at ramp; right lane at ramp; all lanes at ramp;	Yes
Blocked Lane Number	one lane at main road; two lanes at main road; three lanes at main road; one lane at ramp; two lanes at ramp	Yes
Injury	Yes; No	No
Roll Over	Yes; No	No
Congestion	Yes; No	No

6.3 Results

6.3.1 Community Detection

The only parameter that needed to be calibrated was the similarity threshold e_{th}, and given that the number of causative variables used for the comparison was 9 $(m = 9)$, the range for that parameter was from 1 to 9. Furthermore, because e_{th} determines the similarity criterion between two accident records, at least more than half of the variables should have the same values. This further narrowed the range to between 5 and 8 (it also does not make sense to require all 9 parameters to be similar). Given this, we experimented with four possible values for e_{th}: 5, 6, 7, and 8. This process was conducted with the help of the open visualization software Gephi (Bastian et al., 2014), and the resulting network characteristics are shown in Table 6-2.

Table 6-2 Network Clusters With Respect to the Similarity Threshold

Resulting Network Characteristics	$e_{th} = 5$	$e_{th} = 6$	$e_{th} = 7$	$e_{th} = 8$
Number of vertices	999	999	997	930
Number of edges	180,480	83,945	27,552	5,705
Number of clusters founded	3	5	8	33
Maximum modularity	0.213	0.296	0.47	0.647

As can be seen from Table 6-2, with the increase in the value of the similarity threshold e_{th}, the number of edges in the network decreases (since it becomes harder to find similar vertices to connect), and the number of clusters as well as the associated maximum modularity of the network increase. Since modularity represents the concentration of nodes within communities in comparison to the random distribution of edges among nodes regardless of communities, lower e_{th} makes the network more randomly connected. Therefore, it is better to choose larger e_{th}. However, when $e_{th} = 8$, although the maximum modularity is 0.647, the number of clusters is as high as 33. Besides, because connection requirements are more demanding, only 930 out of the 999 vertices are connected in that network (the remaining accidents were not found to be similar to any other accident which defies the purpose behind clustering). Given this, 7 was selected as the value for e_{th}, resulting in a total of 8 clusters. Figure 6-1 shows the resulting traffic accident network and the clustering results.

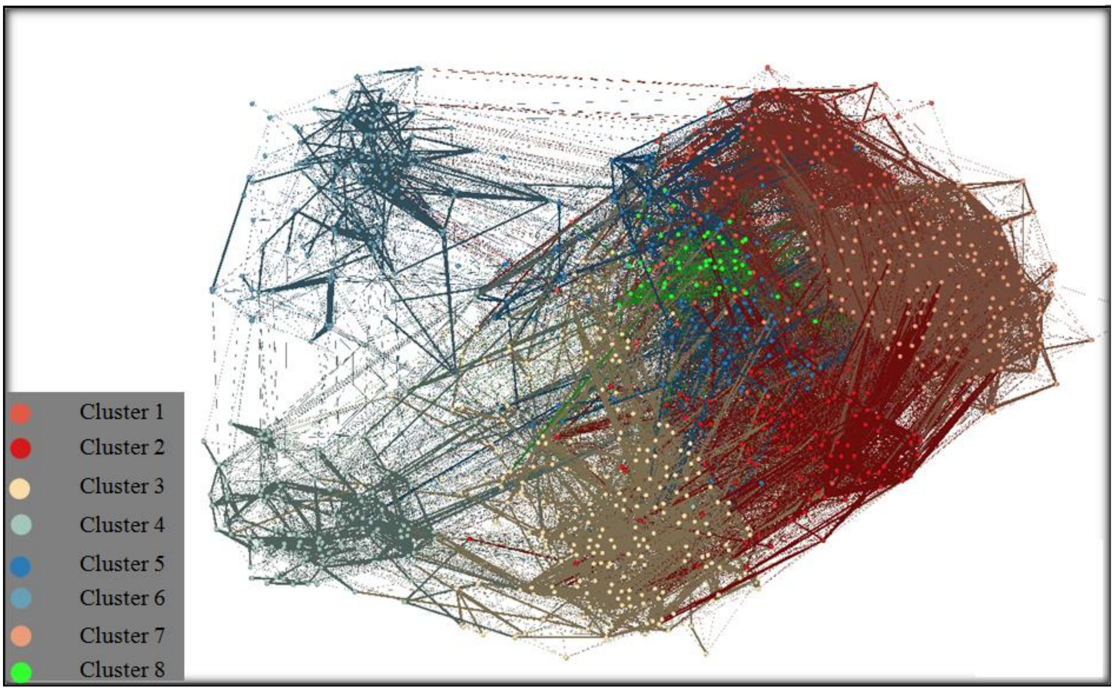


Figure 6-1 Resulting traffic accidents network and community detection ($e_{th} = 7$)

To identify the attributes of each cluster (in terms of describing a given accident type or condition), we followed the method used by Depaire et al. (2008) where the distributions of the variables in each cluster are analyzed to identify the dominant or skewed features (the cluster could then be named based on these features. For example, if 100% of traffic accidents in one cluster happen at non-weekdays, while the other clusters have low probabilities for that feature, we can refer to that cluster as the non-weekday accidents cluster). Table 6-3 shows the probabilities for each feature within the 8 clusters, where the dominant or skewed feature probabilities are underlined and highlighted.

Table 6-3 Causative Factors and Their Probabilities in Each Cluster (%)

Variable: Value (Environmental Feature)	**Cluster**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Season: Winter	14	50	21	34	16	45	29	94
Weekday: Yes	**99**	**95**	0	66	54	73	73	70
Weekday: No	1	5	**100**	34	46	27	27	30
Weather Conditions: Clear	80	70	84	65	85	44	73	0
Weather Conditions: Snow	1	16	5	24	0	31	14	**100**
Direction: South	**99**	0	60	55	**100**	**88**	0	**98**
Direction: North	0	**100**	40	45	0	12	**100**	0
Lane Number at Main Road: 3	**99**	**98**	**99**	61	0	2	0	74
Lane Number on Main Road: 2	0	0	0	37	**100**	**98**	**99**	26
Lane Number on Ramp: 1	99	81	90	0	**100**	15	72	**100**
Lane Number on Ramp: 2	1	19	10	0	0	**85**	28	0
Lane Number on Ramp: 0	0	0	0	**100**	0	0	0	0

The probabilities in Table 6-3 can clearly be used to characterize each cluster. For example, the first three clusters are all most likely to occur on the highway Chapters with three lanes at main road (with the occurring probabilities of 99%, 98% and 99%, respectively). Moreover, Cluster 1 and 2 can be claimed as weekday accidents in the southbound and northbound directions of I-190, respectively, while Cluster 3 includes non-weekday accidents only. All the Cluster 4 accidents (100%) occurred on highway Chapters away from exits, where the lane number on the ramp is 0. Clusters 5, 6 and 7 all involve accidents on roads with only two lanes. However, Cluster 5 seems to have involved accidents close to a ramp with one lane, whereas for Cluster 6, the ramp had two lanes. Moreover, Clusters 5 and 6 seem to involve accidents in the southbound direction, whereas accident s in Cluster 7 occurred in the northbound direction. Finally, Cluster 8 appears to involve accidents happening during snowy conditions (100%). Based on the results, the eight clusters can be described as shown in Table 6-4.

Table 6-4 Traffic Accident Types

Cluster	Traffic accident types	Size (%)
1	Traffic accidents on southbound Chapters with three lanes at main road on weekdays	17
2	Traffic accidents on northbound Chapters with three lanes at main road on weekdays	10
3	Traffic accidents on Chapters with three lanes at main road on non-weekdays	11
4	Traffic accidents on Chapters away from exits	13
5	Traffic accidents on southbound Chapters with two lanes at main road and 1 lane at ramp	9
6	Traffic accidents on southbound Chapters with two lanes at main road and two lanes at ramp	13
7	Traffic accidents on northbound Chapters with two lanes at main road	22
8	Traffic accidents on southbound Chapters with one lane at ramp in snowy days	5

6.3.2 Association Rule Analysis to Identify Hotspots

In this research, for the association rule analysis, a “hotspot” is defined as the place where the ratio of the number of accidents at that particular spot, to the number of accidents on the whole transportation system under consideration is greater than the minimum support σ, under the conditions defined by the body of an association rule. In order to identify accident hotspots and the characteristics of accidents that occur there, the association rule analysis algorithm was then run using the 9 causative factors as candidate variable for the body of each rule, and using the “Location-Exit Number” accident attribute as the head of each rule. The minimum support parameter was set to 0.05, and the minimum confidence to 0.50. The results are shown in Table 6-5 which lists the rules that had the highest confidence for a given location, along with a few other rules that provide some insight for the study. As can be seen, the analysis was performed twice: first, on the whole dataset without clustering, and then on each cluster. The dominant or skewed features for each cluster, as determined from the previous analysis, are shown in bold. Finally, the confidence level values shown in parentheses are those that result when the value of one causative factor is perturbed (e.g. for rule #5 in cluster 2, the confidence drops from 1.00 to 0.38, when the environmental condition changes from rain to clear).

Table 6-5 Rules on Hotspots from the Whole Dataset and the Clusters

Datasets	ID	Body	Head	Confidence
Whole Dataset	1	[direction: north]+[lane number at main road: 2]+[ramp type: off ramp]	[Exit 9: Peace Bridge]	0.67
	2	[lane number at main road: 2]+[lane number at ramp:1]+[ramp type: highway to highway off ramp]	[Exit 11: route 198]	1
	3	[lane number at main road: 2]+[lane number at ramp: 2]+[ramp type: highway to highway off ramp]	[Exit 16: I-290]	0.60
Cluster1	4	**[Weekdays: yes]**+[weather condition: clear]+**[direction: south]**+**[lane number at main road: 3]**+[lane number at ramp: 1]+[ramp type: highway to highway off ramp]	[Exit 7 Skyway]	1
Cluster2	5	**[weekdays: yes]**+[hour: 4 PM-6 PM]+[weather condition: rain (clear)]+**[direction: north]**+**[lane number at main road: 3]**+[lane number at ramp: 1]	[Exit 8: Niagara Street]	1 (0.38)
	6	([season: Winter]+)**[weekdays: yes]**+ **[direction: north]**+**[lane number at main road: 3]**+[lane number at ramp: 2]+[ramp type: off ramp]	[Exit 6: Elm/Oak Street]	0.90 (1)
Cluster3	7	**[weekdays: no]**+[direction: north]+**[lane number at main road: 3]**+[lane number at ramp: 2]+[ramp type: off ramp]	[Exit 6: Elm/Oak Street]	0.89
Cluster4	8	[season: winter]+[weekdays: yes]+[lane number at main road: 2]+**[lane number at**	Milepost 10-12	0.54

		ramp: 0]		
Cluster5	9	**[direction: south]**+**[lane number at main road: 2]**+**[lane number at ramp: 1]**+[ramp type: highway to highway off ramp]	[Exit 11: Route 198]	1
	10	([season: winter]+)[hour: 7 AM-9 AM]+**[direction: south]**+**[lane number at main road: 2]**+**[lane number at ramp: 1]**+[ramp type: off ramp]	[Exit 17: South Grand Island Bridge]	0.54(0.90)
Cluster6	11	[weekdays: yes]+[hour: 4 PM-6 PM]+**[direction: south]**+**[lane number at main road: 2]**+**[lane number at ramp: 2]**+[ramp type: highway to highway off ramp]	[Exit 16: I-290]	0.63
	12	[weekdays: yes]+[hour: 7 AM-9 AM]+[direction: north]+**[lane number at main road: 2]**+**[lane number at ramp: 2]**+[ramp type: highway to highway off ramp]	[Exit 16: I-290]	1
Cluster7	13	[weekdays: yes]+[hour: 4 PM-6 PM]+**[direction: north]**+**[lane number at main road: 2]**+[lane number at ramp: 2]+[ramp type: off ramp]	[Exit 9: Peace Bridge]	1
	14	[hour: 4 PM-6 PM]+[weather condition: clear]+**[direction: north]**+**[lane number at main road: 2]**+[lane number at ramp: 2]+[ramp type: off ramp]	[Exit 9: Peace Bridge]+[road structure: beyond the exit]	0.52
	15	**[direction: north]**+**[lane number at main road: 2]**+[lane number at ramp: 1]+[ramp type: highway to highway off ramp]	[Exit 11: Route 198]	1
Cluster8	16	[weekdays: yes]+**[weather condition: snow]**+**[direction: south]**+[lane number at main road: 3]+**[lane number at ramp: 1]**+[ramp type: highway to highway off ramp]	[Exit 7: Skyway]	1
	17	**[weather condition: snow]**+**[direction: south]**+[lane number at main road: 3]+**[lane number at ramp: 1]**+[ramp type: highway to highway off ramp]	[Exit 7: Skyway]+[road structure: before the exit]	0.6

	18	[weekdays: yes]+[hour: 10 PM-6 AM]+([wind speed: 10])+**[weather condition: snow]**+**[direction: south]**+[lane number at main road: 2]+**[lane number at ramp: 1]**+[ramp type: off ramp]	[Exit 9: Peace Bridge]	0.5 (0.75)

From the analysis on the whole dataset, three association rules with the highest confidence, for the corresponding three hotspots (Exits 9, 11 and 16), are selected. One common feature in body parts of the three rules is there are two lanes at main road, and two out of the three rules contain highway to highway off ramp feature, which appear to be problematic areas with a high accident frequency (this is quite intuitive because of the limitation of capacity and the excessive weaving that takes place there). As can be seen, the analysis on the non-clustered dataset yielded limited insight about the hotspots.

When the analysis was performed on the clusters, several more rules and causative factors are revealed. Specifically, 15 association rules are revealed, along with eight hotspots. For the hotspots, only one is located away from exits, and the rest are all close to exits. Furthermore, these seven exits identified are spatially correlated with one another, and fall very neatly in two definite *geographic* clusters; the first is [Exit 6, Exit 7, Exit 8, Exit 9, and Exit 11] – note that there is no Exit 10 on I -190; and the second is [Exit 16 and Exit 17]. Through comparing and analyzing the rules describing the same hotspot, a few additional insights can be gained as below:

Firstly, for Exit 6, when comparing Rules 6 and Rule 7, it becomes clear that the problem is consistently in the north direction no matter if it is a weekday or a non-weekday. Secondly, for Exit 7, when comparing Rules 4 and Rule 16, we can see that Exit 7 is always a hotspot with (confidence level = 1) regardless of the weather condition (both clear and snow). Rule 17 shows that the segment before Exit 7 is a hotspot in south direction when it snows. Thirdly, for Exit 9, Rule 13 provides more specific conditions than Rule 1. According to the rule, Exit 9 is a hotspot with confidence level 1 in the north direction for the peak hour 4 PM-6 PM on weekdays. Rule 14 shows that if it is the peak hour 4 PM-6 PM with clear weather, the segment beyond Exit 9 in north direction is also a hotspot. And Rule 18 shows that, in the south direction, Exit 9 may also be a hotspot when it is 10 PM-6 AM on weekdays with snow. Fourthly, for Exit 11, by checking Rule 9 and Rule 15, Exit 11 is always a hotspot with confidence 1 in both the north and southbound direction. This is consistent with the conclusion of Rule 2 on the whole dataset. Finally, for cluster 4 describing traffic accidents on highways away from exits, only one hotspot is found with a relatively low confidence 0.54, although it contains 13% of the total records. This seems to indicate that accidents along I-190 tend to happen close to exits more often.

Besides insight regarding hotspots, the associative rules shed additional light on the conditions under which accidents happen at those locations. This additional insight is gained by considering the role of the variables in the "body" parts of the rules. A few examples are described below.

Firstly, the variables "weekdays" and "hour of the day" appear to affect whether a location becomes a hotspot. Nine out of the 15 association rules generated from the clusters contain "[weekdays: yes]" in the body parts, and five of the nine rules contain "[hour: 7 AM-9 AM]" or [hour: 4 PM-6 PM]." This reveals the effect of weekday peak hours on traffic accidents. Another convincing example comes from Rule 11 and Rule 12. Exit 16-I-290 is a hotspot when it is 7 AM-9 AM in the morning towards north direction, and Exit 16 is also a hotspot when it is 4 PM-6 PM in the afternoon towards south direction.

Secondly, the feature "[season: winter]" can increase the confidence in claiming a location as a hotspot. For example, Rule 6 in Cluster 2 shows that if it is in winter, the confidence for Exit 6 to be a hotspot on weekdays will increase from 0.90 to 1. Similarly, Rule 10 in Cluster 5 shows that if it is 7 AM-9 AM on someday in winter, the confidence in claiming Exit 17 as a hotspot witness a large increase from 0.54 to 0.90. Besides that, the variable "wind speed" and "weather condition" are found to affect the confidence for some locations. Rule 18 shows that Exit 9-Peace Bridge has a higher risk 0.75 than the previous 0.50 if the wind speed is 10 miles per hour. Rule 5 shows that with the other features in the body part being the same, the "[weather condition: rain]," rather than "[weather condition: clear]," tend to make Exit 8 a hotspot with confidence level 1.

6.3.3 Association Rule Analysis to Identify Factors Affecting Incident Clearance Time

The *association rule* analysis was then repeated, this time using the accident attribute "incident clearance time" as the "head of the rules" to gain some insight into the factors affecting incident clearance time. For clearance time analysis, the minimum support is set as 0.05, and the minimum confidence is lowered to 0.30

(experiments showed this set of rules to have lower confidence levels compared to the hotspot analysis). The results are shown in Table 6-6.

Table 6-6 Rules on Clearance Time from the Whole Dataset and the Clusters

Datasets	**ID**	**Body**	**Head**	**Confidence**
Whole Dataset	1	[weekdays: yes]+[hour: 4 PM-6 PM]	[Clearance time: 31-45 minutes]	0.32
	2	[season: winter]+[lane number at main road: 3]	[Clearance time: 16-30 minutes]	0.34
Cluster1	3	**[weekdays: yes]**+[hour: 4 PM-6 PM]+[wind speed: 10]+**[direction: south]**+**[lane number at main road: 3]**	[Clearance time: 31-45 minutes]	0.35
Cluster2	4	**[weekdays: yes]**+[hour: 4 PM-6 PM]+[weather condition: clear]+**[direction: north]**+**[lane number at main road: 3]**+[lane number at ramp: 1] +[ramp type: off ramp]+[road structure: at the exit]	[Clearance time: 31-45 minutes]	0.58
	5	**[weekdays: yes]**+[weather condition: clear]+[Exit 8: Niagara Street]+**[direction: north]**+**[lane number at main road: 3]**	[Clearance time: 31-45 minutes]	0.55
	6	[season: winter]+**[weekdays: yes]**+[weather condition: clear]+**[direction: north]**+**[lane number at main road: 3]**+[lane number at ramp: 1]	[Clearance time: 16-30 minutes]	0.30
Cluster3	7	[season: autumn]+**[weekdays: no]**+[direction: north]+**[lane number at main road: 3]**+[lane number at ramp: 1]+[ramp type: off ramp]	[Clearance time: 46-60 minutes]	0.60
	8	**[weekdays: no]**+[Exit 8: Niagara Street]+ **[lane number at main road: 3]**+[lane number at ramp: 1]+[ramp type: off ramp]	[Clearance time: 46-60 minutes]	0.33
Cluster4	9	[season: autumn]+[weekdays: yes]+[lane number at main road: 3]+**[lane number at ramp: 0]**	[Clearance time: 46-60 minutes]	0.50
	10	[season: winter]+[direction: south]+[lane number at main road: 3]+**[lane number at ramp: 0]**	[Clearance time: 16-30 minutes]	0.47
	11	[weekdays: no]+[direction: south]+[lane number at main road: 3]+**[lane number at ramp: 0]**	[Clearance time: 31-45minutes]	0.37
	12	[weekdays: yes]+[direction: south]+[lane number at main road: 3]+**[lane number at**	[Clearance time: 16-30 minutes]	0.41

		ramp: 0]		
	13	[weekdays: yes]+[direction: north]+[lane number at main road: 3]+**[lane number at ramp: 0]**	[Clearance time: 31-45minutes]	0.31
Cluster5	14	[weekdays: no]+**[direction: south]**+**[lane number at main road: 2]**+**[lane number at ramp: 1]**	[Clearance time: 16-30 minutes]	0.31
	15	[weekdays: yes]+**[direction: south]**+**[lane number at main road: 2]**+**[lane number at ramp: 1]**	[Clearance time: 31-45minutes]	0.32
	16	[weekdays: yes]+ [Exit 9: Peace Bridge]+**[direction: south]**+**[lane number at main road: 2]**+**[lane number at ramp: 1]**+[ramp type: off ramp]	[Clearance time: 31-45minutes]	0.60
Cluster6	17	[Exit 16: I-290]+**[direction: south]**+**[lane number at main road: 2]**+**[lane number at ramp: 2]**+[ramp type: highway to highway off ramp]+[road structure: at the exit]	[Clearance time: 31-45minutes]	0.35
	18	[hour: 7 AM-9 AM]+**[lane number at main road: 2]**+**[lane number at ramp: 2]**+[ramp type: highway to highway off ramp]	[Clearance time: 46-60 minutes]	0.33
Cluster7	19	[weekdays: yes]+[hour: 1 PM-3 PM]+**[direction: north]**+**[lane number at main road: 2]**	[Clearance time: 0-15minutes]	0.52
	20	[weekdays: yes]+[Exit 9: Peace Bridge]+**[direction: north]**+**[lane number at main road: 2]**	[Clearance time: 16-30 minutes]	0.31
	21	[weekdays: yes]+[hour: 4 PM-6 PM]+**[direction: north]**+**[lane number at main road: 2]**	[Clearance time: 31-45minutes]	0.31
	22	[Exit 11: Route 198]+**[direction: north]**+**[lane number at main road: 2]**	[Clearance time: 31-45minutes]	0.34
	23	[season: winter]+([weather condition: snow])+**[direction: north]**+**[lane number at main road: 2]**	[Clearance time: 31-45 minutes]	0.34 (0.46)
Cluster8	24	[season: winter]+**[weather condition: snow]**+**[direction: south]**+[lane number at main road: 3]+**[lane number at ramp: 1]**	[Clearance time: 16-30 minutes]	0.52

Recalling in Table 6-1, clearance time is divided into seven intervals, each 15 minutes long. From Table 6-6, we can see when the analysis was performed for the whole dataset, two rules are shown: Rule 1 is associated with peak-hour 4 PM-6 PM on weekdays, and the clearance time of accidents is shown to be 31-45 minutes (with a confidence level of 0.32); Rule 2 is for winter, if accidents happen at Chapters with three lanes main road, the clearance time tend to be between 16-30 minutes (with confidence level of 0.32). As before, when the associate rule analysis is performed on the whole dataset, limited insight is gained.

For the clusters, 22 rules are selected; four have a clearance time of 46-60 minutes, 12 have 31-45 minute clearance times, 5 have 16-30 minutes, while the remainder has 0-16 minutes clearance times. Some of the main observations are summarized below.

Firstly, with respect to the “Weekday” variable, its impact on the incident clearance time appears to be mixed. For example, Rule 8 shows that on non-weekdays, accidents at Exit 8 have clearance time between 46 and 60 minutes with confidence 0.33. Also, according to Rule 11 and 12, on the southbound Chapters with 3 lanes on the main road, accidents on non-weekdays tend to have a longer clearance time than accidents on weekdays. On the other hand, when comparing Rule 14 and 15, we can see that with other factors being the same, accidents on non-weekdays are more likely to have clearance time of 16-30 minutes, while those on weekdays tend to have longer clearance time of 31-45 minutes. This indicates that there are other factors besides whether the accident is on a weekday or not that affects clearance time, but perhaps the dataset was not rich enough to reveal such factors.

Secondly, the variable “Hour of the Day” may have an impact on the clearance time of traffic accidents. Rules 3, 4 and 21, which correspond to a clearance time 31-45 minutes, all have the same feature “the peak hours 4 PM-6 PM” in their body parts; Rule 18 shows that at peak hours 7 AM-9 AM, accidents on Chapters with two lanes at main road and two lanes at highway to highway off ramp have a probability of 0.33 to experience 46-60 minutes. And Rule 19 which shows on weekdays at 1 PM-3 PM (i.e. off-peak) the clearance time of accidents on Chapters towards north with two lanes at main road tends to be short, 0-15 minutes with confidence equal to 0.52.

Thirdly, the feature “snow” appears to increase the likelihood of longer clearance time. According to Rule 23, in the winter for Chapters towards north with two lanes at main road, the confidence in the clearance time being 36-45 minutes (i.e. on the long side) is 0.34. During snowy condition, the confidence increases to 0.46.

Finally, the “direction” of the road may also affect the clearance time (because it could potentially impact the time needed to get to the incident scene). By comparing Rule 12 and Rule 13, we can see that for Chapters with 3 lanes on the main road on weekdays, accidents in the north direction has clearance time of 31-45 minutes with confidence 0.31, while accidents in the south direction has a probability of 0.41 to have clearance time of 16-30 minutes. Another similar example is for hotspot at Exit 9. Based on Rule 16 and Rule 20, on weekdays, the clearance time for accidents at Exit 9 in the southbound direction may be 31-45 minutes with a confidence level of 0.60, which is longer than 16-30 minutes at the same exit in the north direction (confidence of 0.31).

6.4 Conclusions

In this study, the *modularity-optimizing community detection* algorithm was used first to cluster accident data recorded for I-190 in the Buffalo-Niagara area. Following this, the *association rules learning* algorithm was used to gain some insight into accident hotspots and incident clearance times. To demonstrate the benefits of clustering, the *association rule* algorithm was applied to both the whole dataset (before clustering) and then to the clusters and the results were compared. The main findings are summarized as below:

(1) The community detection algorithm appears to do an excellent job in clustering the data into well-defined clusters;

(2) Clustering the data first before running the association rule learning algorithm appears to be a necessary step that can significantly improve the quality of the insight to be gained from the rules extracted. Specifically, when the association rule algorithm was run on the whole dataset in this study, the insight gained was very limited compared to that gained from running the analysis on the clusters.

(3) The association rule learning algorithm has the potential to reveal interesting insight about the characteristics of accidents, where they tend to occur, and the factors that affect incident clearance time.

CHAPTER 7 NOVEL VARIABLE SELECTION FOR REAL-TIME TRAFFIC ACCIDENT RISK PREDICTION

The Chapter is an application of the integration of data "width" decreasing and model development. It is organized as below. First, in the Chapter 7.1, we introduce the FP tree model and its variable importance score calculation algorithm. In Chapter 7.2, we describe the traffic accident datasets used for model training and testing. In Chapter 7.3, prior to the risk prediction model development, we describe and compare the FP tree and the random forest based variable selection methods in terms of their variable importance ranking results. After that, based on the variables selected by the FP tree and the random forest methods respectively, two traffic accident risk predictions models are discussed and compared in terms of their prediction performance, namely the k-NN model and the Bayesian network model. The Chapter ends with a summary of the main conclusions of the work.

7.1 Methodology

7.1.1 Variable Selection

This Chapter discusses the FP-tree algorithm used in this research for explanatory variable selection. The algorithm consists of two steps: variable discretization and variable importance score calculation. For the former step, the fuzzy c-means clustering method is used to convert a continuous variable to a series of discrete categorical variables; for the latter, we propose the "Relative Object Purity Ratio (ROPR)" as an importance score for each explanatory variable. This Chapter will also introduce the random forest method that is used as the bench-marking variable selection method. Finally, the two methods used for accident risk prediction, namely the k-NN model and Bayesian network, are briefly introduced.

7.1.1.1 Frequent pattern Tree (FP tree)

The Frequent pattern tree (FP-tree) algorithm was proposed by Han et al. (2004). It yields a compact representation of all relevant frequency information in a dataset. A brief introduction of the FP-tree algorithm follows.

Suppose $I = \{i_1, i_2, i_3, \ldots, i_m\}$ be a set of items. Let TN be a set of transactions or records in a database DB, and each transaction $Tran$ is a set of items, $Tran \subseteq I$. A pattern X also contains a set of items, $X \subseteq I$. X is called a frequent pattern when its support, referring to the frequency at which X appears in the TN transactions, is equal to or greater than the minimum support threshold, σ .

$$\frac{supp\{X\}}{TN} \geq \sigma \tag{7-1}$$

where,

σ is a threshold value defined by user.

A FP-tree includes a root labeled as “null”. It also includes a set of item-prefix sub-trees as the children of the root. There are two important fields for each node in the item-prefix sub-trees: *item name* and *count*. *Item name* tells which item this node represents, and *count* records the number of transactions represented by the portion of the path reaching this node.

The FP-tree can be constructed according to the following sequential steps:

1. Scan the DB once. Calculate the support of each item, if the item is a frequent item as determined by Equation (7-1), put it in the list of frequent items *F*. Sort *F* in the support-descending order to form the F-List;

2. Create the root of an FP tree, *T*, and label it as "null";

3. For each transaction *Tran* in DB, select the frequent items in *Tran* and sort them according to the order of the F-List;

4. Suppose the sorted list of *Tran* is $[p|P]$, where p is the most frequent item in this transaction and *P* is the remaining list. Next, run the function *insert_tree (*$[p|P]$*, T).*

The function, *insert_tree* ($[p|P]$, *T*), is defined as follows. If *T* already has a child node *N*, the item name of *N* is the same as the item name of p, and the *N*'s count is incremented by 1; otherwise, create a new node *N* with its *count* initialized as 1, and set its parent link to *T*. If *P* is nonempty, the function *insert_tree* (*P, N*) is run recursively.

Figure 7-1 shows an example of a FP tree. Suppose there are *TN* transactions $[x_1, x_2, \ldots, x_{TN}]$ in database DB, each transaction contains the values of *n* explanatory variables V_e, $1 \leq e \leq n$, and one response variable V_r which, in our case, denotes whether an accident occurs or not. The FP tree is then built on the *TN* transactions with n explanatory variables, among which the continuous variables are first transformed to discrete variables by using the Fuzzy C-means clustering method (FCM) as will be discussed in Chapter 2.1.2.

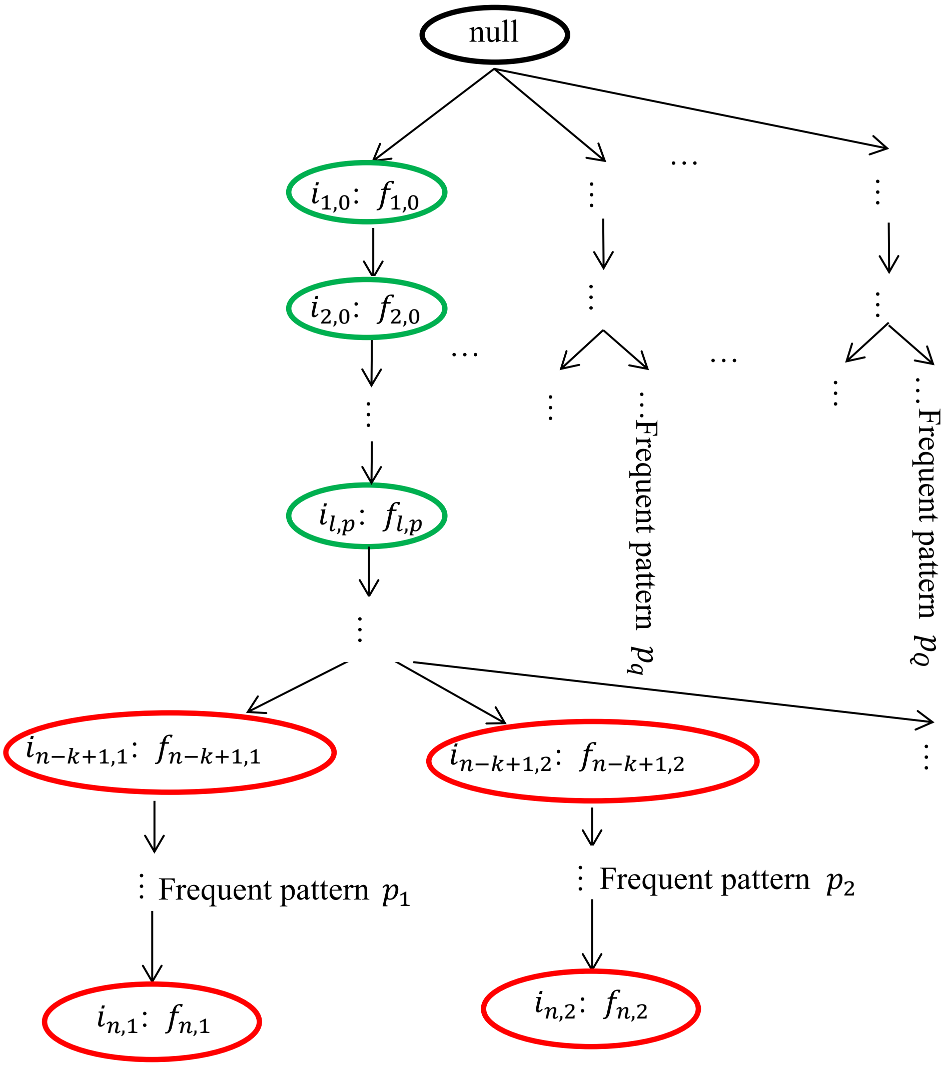


Figure 7-1 Frequent pattern (FP) tree.

As demonstrated in Figure 7-1, there are Q frequent patterns in the example, and each pattern p_q, $1 \leq q \leq Q$, is represented by one branch in the tree. Each branch consists of n nodes, and each node (node *l*) is labeled by *item name* $i_{l,p}$ and *count* $f_{l,p}$. As discussed before, *item name* $i_{l,p}$ denotes the variable name and its discrete state associated with the node and the branch, and *count* $f_{l,p}$ represents the number of records reaching the node through the preceding branch. l is the node order in a frequent pattern, $l = 1,2, \ldots, n$. The pattern status indicator p can take two values: if this node is a "*shared node*" by two or more frequent patterns, $p = 0$; otherwise,

$p = q$, and we call it an "*exclusive node*". There are k exclusive nodes that are marked as red for frequent pattern p_1 and p_2 in Figure 7-1.

7.1.1.1.1 Variable importance calculation

A novel FP tree based variable importance score calculation method is proposed to rank and select the significant explanatory variables for accident risk prediction. The method proceeds as follows.

1. For each frequent pattern p_q, calculate its *object purity ratio* r_q (OPR). OPR refers to the proportion of records falling into this frequent pattern, where their response variable V_r takes the object value o (in this research the object value o is set as 1 which indicates an accident occurrence). r_q can thus be calculated as follows:

$$r_q = \frac{num_q(V_r = o)}{f_{n,q}} \tag{7-2}$$

where,

$num_q(V_r = o)$ is the number of records in frequent pattern q which have the response variables V_r as o;

$f_{n,q}$ is the number of records allocated to frequent pattern q.

One issue associated with OPR is that its value is in reference to the proportion of records taking the object value in the whole dataset DB, which can thus lead to inconsistent variable ranking. In this context, it is the difference between the OPR value of a pattern and the average behavior of the entire data that actually distinguishes a pattern. Therefore, we propose the *relative object purity ratio* r_{rq} (ROPR) in this research, where, in its modified version, ROPR represents the absolute difference between the OPR and the proportion of records taking the object value in the whole dataset DB.

$$r_{rq} = abs(\frac{num_q(V_r = o)}{f_{n,p}} - \frac{num_{DB}(V_r = o)}{TN}) \tag{7-3}$$

where,

$num_{DB}(V_r = o)$ is the number of records with the response variables V_r as the object value o.

2. Given an observed record located in this frequent pattern, one intuitive thought is that the higher the ROPR is, the purer the frequent pattern is and the more likely the object response value will take place (i.e., in our case, that an accident will occur). Here, we assume that only the items that are in the exclusive nodes play a role in differentiating one frequent pattern from the others. Therefore, the importance score of an item is determined as follows: for each transaction $Tran$ in DB, find its corresponding frequent pattern p_q and exclusive nodes E_q; for each item in $Tran$, if it exits in E_q, add the ROPR to the item's importance score IS_i, otherwise, keep IS_i unchanged.

$$IS_i = \sum_{1 \le q \le Q} \sum_{1 \le Tran \le TN} r_{rq} * d_{Tran} * d_q * d_e \,, \quad 1 \le i \le m \tag{7-4}$$

where,

$d_{Tran} = 1$ if item i is in transaction $Tran$; otherwise $d_{Tran} = 0$;

$d_q = 1$ if p_q is the frequent pattern of the corresponding transaction $Tran$; otherwise $d_q = 0$;

$d_e = 1$ if item i is in the exclusive node set E_q; otherwise $d_e = 0$.

3. After the importance score of each item is calculated, the remaining step is to calculate the importance score of a variable (IS_v).

$$IS_v = \sum_{1 \le i \le m} IS_i * d_v,\ 1 \le v \le n \tag{7-5}$$

where,

$d_v = 1$ if item i is one discrete value of variable v; otherwise $d_v = 0$.

At last, the explanatory variables can be ranked based on the variable importance scores. The complete Matlab code of this FP tree based variable selection method can be found from the link provided in Appendix.

7.1.1.1.2 Variable discretization for FP tree

The FP-tree algorithm requires each transaction in the database to be a set of discrete items. However, in traffic accident risk prediction database, continuous variables such as traffic speed and traffic volume are quite common. In this research, the Fuzzy C-means clustering method (FCM) is used to transform the continuous variables to the discrete variables. FCM is an extension of the k-means methods in which each data point can be a member of multiple clusters with a membership value (soft assignment) (Jain, 2010). FCM is briefly described below (Hung and Yang, 2001):

Suppose there are *TN* transactions $[x_1, x_2, \ldots, x_{TN}]$ recorded for a continuous variable (e.g., travel speed values over time). FCM is to optimally categorize these transactions to different discrete clusters by solving a minimization programming problem as defined below. The decision variable (μ_{ij}) to be solved is the probability for transaction *i* belonging to cluster *j*. The objective ($J_\beta(U, W)$)) is to minimize the summation of the weighted squared distances from each transaction to the center of its associated cluster. The first constraint indicates that the sum of the clustering probabilities for a transaction always equals 1 while the second constraint ensures that C non-empty clusters will be formed at the end.

$$minimize \ \ J_{\beta}(U,W)=\sum_{j=1}^{C}\sum_{i=1}^{TN}(\mu_{ij})^{\beta}d_{ij}^{2} \tag{7-6}$$

Subject to

$\sum_{j=1}^{C}\mu_{ij}=1, \quad i=1,...,TN$;

$0<\sum_{i=1}^{TN}\mu_{ij}<TN, \quad j=1,...,C$.

$\mu_{ij}\in[0,1]$, $i=1,...,TN$ and $j=1,...,C$;

where,

U is the membership function matrix with element μ_{ij};

μ_{ij} is the probability for the i^{th} transaction belonging to the j^{th} cluster;

d_{ij} is the distance from a transaction x_i to $w_j^{(t)}$, $d_{ij}=\| x_i-w_j^{(t)} \|$;

$w_j^{(t)}$ denotes the center of the j^{th} pattern cluster in the t^{th} iteration;

W is the cluster center vector;

β is the exponent associated with μ_{ij} to control fuzziness or amount of overlapping clusters.

Derived from the method of Lagrange multipliers, the following optimal results can be reached:

$$\mu_{ij}^{(t)}=\frac{1}{\sum_{k=1}^{C}(\frac{d_{ij}}{d_{ik}})^{\frac{2}{(\beta-1)}}}, \ i=1,...,TN, \ j=1,...,C \tag{7-7}$$

If $\mathrm{d}_{ij}=0$ then $\mu_{ij}=1$ and $\mu_{ik}=0$ for $k\neq j$, (7-8)

$$W_j^{(t)} = \frac{\sum_{i=1}^{TN} (\mu_{ij}^{(t-1)})^{\beta} x_i}{\sum_{i=1}^{TN} (\mu_{ij}^{(t-1)})^{\beta}}, \quad j = 1, \ldots, C, \tag{7-9}$$

The FCM solution algorithm is composed of the following steps:

1. Set initial time step $t = 0$. Initialize the cluster center matrix $W^{(0)}$, given a predetermined number of clusters C, and initialize the membership matrix $U^{(0)}$ by using Equation (7-7) and (7-8);

2. Increase time step t by one, compute the new cluster center matrix $W^{(t)}$ by using Equation (7-9) and the new membership matrix $U^{(t)}$ by using Equation (7-7) and (7-8);

3. Continue on Step 2 until $\max_{ij}\{|\mu_{ij}^{(t)} - \mu_{ij}^{(t-1)}|\} < \varepsilon$, indicating that little improvement can be made on the clustering probabilities. Here, ε is a user defined positive threshold value.

7.1.1.2 Random forest

Random forest is an ensemble learning method for classification and regression. It is widely used to rank the importance of variables in a natural way. Again, suppose there are *TN* records or transactions $[x_1, x_2, \ldots, x_{TN}]$ in database DB, each record includes one response variable V_r and a set of explanatory variables $V = [V_1, \ldots, V_n]$, a classification and regression tree (CART) $\hat{f}$ for predicting V_r can be built (Breiman et al., 1984) . The prediction error of $\hat{f}$ based on a validation subset of DB is then defined as

$$R\left(\hat{f}, \overline{DB}\right) = \frac{1}{|\overline{DB}|} \sum_{i \in DB} I\left(\hat{f}(V_i) = V_{ir}\right), \tag{7-10}$$

where,

$I(e) = \begin{cases} 1, if\ e is true \\ 0, if\ e is\ false \end{cases};$

$\overline{DB}$ is the validation data subset;

V_{ir} is the observed value of the response variable of the i^{th} record.

However, CART is known to be unstable as a small perturbation of the training sample may change the prediction results. To overcome this, Breiman introduced the random forest algorithm (Breiman, 2001): the trees are built over n_{tree} bootstrap samples $\overline{DB}^1, \dots, \overline{DB}^{n_{tree}}$ of the training data DB; for each tree, different from the CART algorithm, a subset of variables n_{var} is randomly chosen for the splitting rule at each node; each tree is then fully grown until each node is pure. The trees are not pruned. The resulting learning rule is the aggregation of all the tree-based estimators denoted by $\hat{f}_1, \dots, \hat{f}_{n_{tree}}$ (Gregorutti et al., 2013). The class with the maximum number of votes among the n_{tree} trees in the forest is the predicted class of an observation. The Gini criterion is used to select the split with the lowest impurity at each node. As a useful byproduct of random forests, the Gini variable importance measure can be calculated once the forest is formed: at each split, the decrease in the Gini node impurity is recorded for variable V_i in $[V_1, \dots, V_n]$, and the average of all the decreases in the Gini impurity in the forest where V_i forms the split is its Gini variable importance. At last, the variables can be ranked according to the Gini variable importance measure (Archer and Kimes, 2008). Besides this, Breiman also proposed other measures like the permutation importance, the z-score and so on (Breiman, 2001).

7.1.2 Models

7.1.2.1 k nearest neighbor (k-NN)

k-NN is applied for short-term traffic volume prediction in Chapter 3.2, which is a way of regression. Here it is used for classification and introduced again. When k nearest neighbors are found, the following Equation (7-11) can be used to determine the class of the object (Murphy, 2012):

$$p(y = c \mid X, D, k) = \frac{1}{k} \sum_{i \in N_k(X,D)} I(y_i = c) \qquad (7\text{-}11)$$

where,

$N_k(X, D)$ are the *k* nearest neighboring points to object X in point set D;

$I(e) = \begin{cases} 1, \textit{if e is true} \\ 0, \textit{if e is false} \end{cases}$;

y_i is the response variable of neighboring point *i*;

y is the response variable of object X;

c is the one of the possible classes.

7.1.2.2 Bayesian Network

By the chain rule of probability, a joint distribution can be represented as follow

$$p(x_{1:V}) = p(x_1)\,p(x_2|x_1)\,p(x_3|x_2,x_1)\,p(x_4|x_1,x_2,x_3)\ldots p(x_V \mid x_{1:V-1}) \qquad (7\text{-}12)$$

where,

V is the number of variables;

$1{:}V$ denotes the set $\{1,2,\ldots,V\}$.

Suppose all the variables have K discrete states, we can create $p(x_1)$ as a table of $O(K)$ numbers, representing a discrete distribution (there are actually only K-1 free parameters because of the sum-to-one constraint, but we write $O(K)$ for simplicity). Similarly, we can create $p(x_2|x_1)$ as a table of $O(K^2)$ numbers, and $p(x_3|x_2, x_1)$ as a table with $O(K^3)$ numbers, and so on. These tables are called conditional probability tables (CPTs). As can be seen, the conditional distributions $\mathrm{p}(x_t|X_{1:t-1})$ become harder to estimate as t gets larger (Murphy, 2012).

A Bayesian network is an efficient tool to overcome this problem. Specifically, a Bayesian network is a directed graphical model representing a joint distribution by making conditional independence (CI) assumptions. The nodes in the graph represent random variables, and the edges represent the CI assumptions. Based on the ordered Markov property, Bayesian network assumes that a node only depends on its immediate parents, not on all predecessors. So Equation (7-12) can be transferred into:

$$p\left(x_{1:V}|G\right) = \prod_{t=1}^{V} p(x_t \mid x_{pa(t)}) \tag{7-13}$$

where,

$pa(t)$ represent the parents of node t.

One of useful properties of Bayesian network is to perform probabilistic inference for some unobserved variables (for example, whether a traffic accident would happen or not) after the joint distribution is given. The posterior distribution can be calculated as Equation (7-14):

$$p(X_h \mid X_v, \theta) = \frac{p(X_h, X_v \mid \theta)}{p(X_v \mid \theta)} = \frac{p(X_h, X_v \mid \theta)}{\sum_{X_h^{'}} p(X_h^{'}, X_v \mid \theta)} \tag{7-14}$$

where,

X_h is the unobserved variable;

X_v are the observed variables;

θ is the known parameters in this Bayesian network.

7.2 Modeling dataset

The dataset used in this research includes the traffic accident records collected on a segment on interstate highway I-64 in Norfolk, Virginia in 2005, as marked in Figure 7-2:

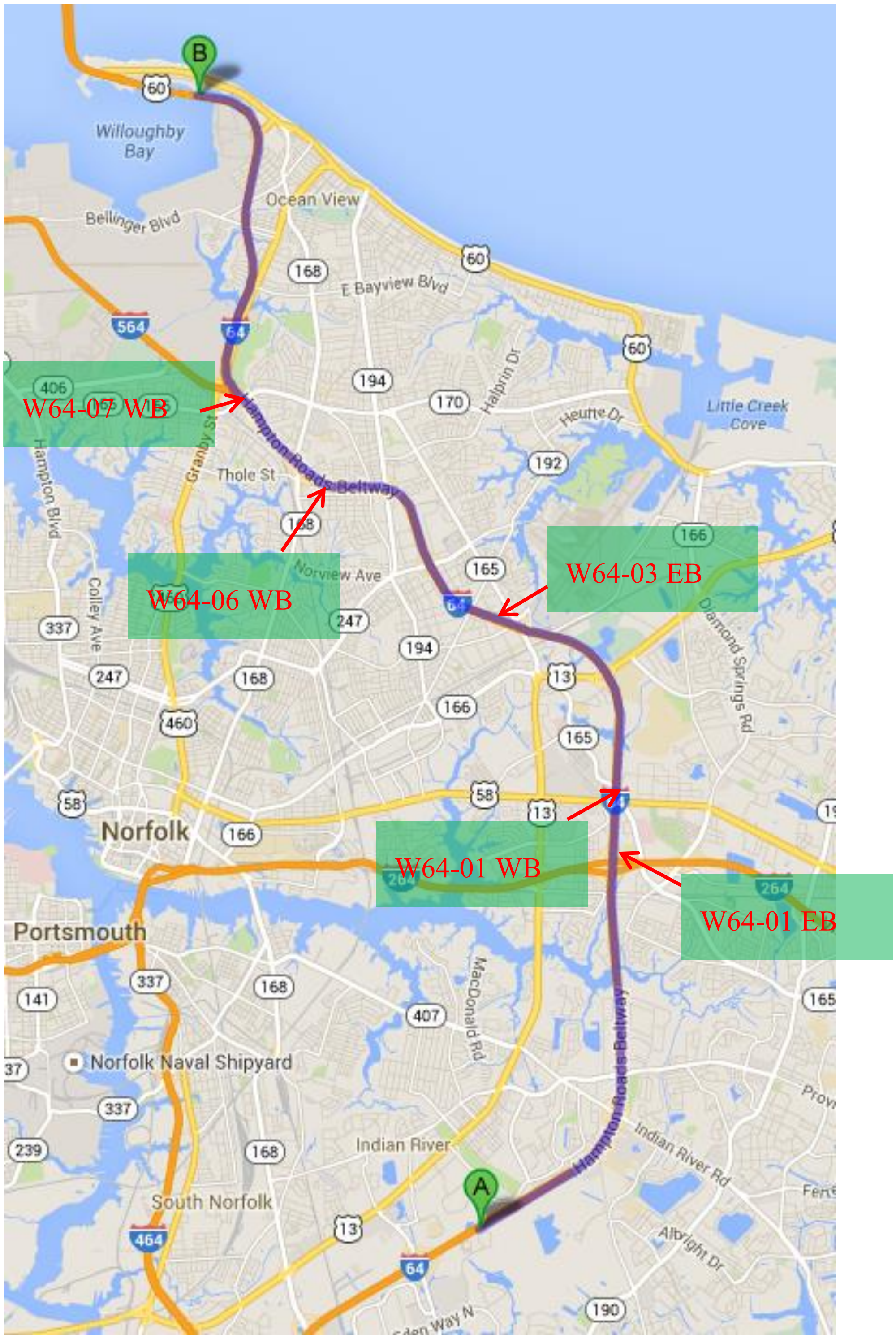


Figure 7-2 Part of I-64 in Norfolk, Virginia.

The accidents were stored in the Virginia Department of Transportation (VDOT's) Archived Data Management System (ADMS). Besides that, this dataset also contains weather, visibility, traffic volume, speed, and occupancy information, with one minute resolution.

However, this dataset by itself cannot be directly applied to predict real-time traffic risk directly. As a classification problem, the pre-crash condition and normal traffic condition have to be defined first (Hossain and Muromachi, 2012). Some studies defined the pre-crash condition as a time period starting right before an accident and extending up to 5 or 10 minutes (Oh et al., 2005; Zheng et al., 2010), while some studies defined it as a 5 minute time period starting from a close time point such as 4 or 5 minutes before the accident (Abdel-Aty et al., 2008; Hossain and Muromachi, 2012). In this research, we used two temporal settings to define the pre-crash condition: the first one is a 10-minute time period starting from 5 minute before the accident, and the other is in a 5-minute time period starting from 5 minute before the accident. The normal condition is defined as the same time period as the pre-crash condition, but taking place on the same day of the other weeks from two weeks earlier to two weeks later than the day of the week with an accident. It needs to note that a normal condition data point is excluded if there is an accident happening within one hour before or after the designated time (Hossain and Muromachi, 2012).

After the pre-crash condition and normal traffic condition are defined, the relevant data can be extracted given the number and locations of traffic detectors in place. Most of the previous studies considered more than one detector during the extraction process, such as one upstream detector and one downstream detector (Abdel-Aty et al., 2008), and two upstream detectors, two downstream detectors and one detector covering the accident location (Hossain and Muromachi, 2012). In this research, due to a huge amount of missing data, we were forced to rely on only one detector, that is to say, the one reporting an accident. There are five such detectors, labeled W64-01 EB, W64-01 WB, W64-03 EB, W64-06 WB and W64-07 WB, their approximate locations are marked in Figure 7-2.

At last, two datasets were obtained, which differ from one other in terms of the the time period used to define the pre-crash and normal traffic condition (the first DB has a time period of 10-minute long, and the second one is 5-minute long). Eight explanatory variables were contained in the data, including: the mean of the weather condition ($\mathrm{Mean}_{\mathrm{wea}}$) as defined below, the mean of visibility ($\mathrm{Mean}_{\mathrm{vis}}$), the mean and standard deviation of the traffic volume ($\mathrm{Mean}_{\mathrm{vol}}$ and $\mathrm{Std}_{\mathrm{vol}}$, unit: vehicle per hour), the mean and standard deviation of the traffic speed ($\mathrm{Mean}_{\mathrm{spe}}$ and $\mathrm{Std}_{\mathrm{spe}}$, unit: mph), and the mean and standard deviation of the occupancy ($\mathrm{Mean}_{\mathrm{ocu}}$ and $\mathrm{Std}_{\mathrm{ocu}}$). The accident response variable is defined as a binary variable with value 1 for the pre-crash situation and 0 for normal traffic. It is worth noting that the weather variable was a categorical variable originally with 26 possible different weather types. In this research we use the numbers 0 to 25 to represent these different weather types that range from fine weather like “clear” to extreme inclement weather like “thunderstorm”. Although typically, the weather condition will not change significantly within a 5- or 10- minute period, we nevertheless, take the mean value of the weather over that period. The resulting variable, therefore, may theoretically assume a non-integer value and can be assumed as a continuous (and not discrete) variable. The same applied for “visibility”, which is also a continuous variable ranging from 0 to 10 miles.

After processing, the 5-minute accident dataset included 170 pre-crash records and 555 normal traffic records, and the 10- minute accident dataset included 174 pre-crash records and 569 normal traffic records. Note that the 5-minute accident dataset has fewer records because of the higher probability of data missing for 5 minute period than the 10 minute period. For each dataset, 80% of the pre-crash records and normal traffic records were randomly chosen as the training dataset while the remaining 20% were taken as the test dataset.

7.3 Model development and results

This Chapter will calculate variable importance scores using the ROPR based variable importance calculation algorithm and the random forest algorithm discussed in Chapter 2. Then the k nearest neighbor and Bayesian network will be built and their performance will be compared based on the different variable importance calculation results.

7.3.1 Variable importance calculation

Two ***training*** datasets are generated through the random sampling with the 80% rate, including a 5-minute training dataset with 136 pre-crash records and 444 normal traffic records and a 10-minute training dataset with 139 pre-crash records and 455 normal traffic records. For each training dataset, FCM was first applied to transfer a continuous variable to a discrete cluster variable.

Table 7-1 Clustering Results for 5-minute and 10-minute Accident Training Datasets

datasets	Variable	Cluster 1 low	Cluster 2 medium	Cluster 3 high
5-minute	Mean_{wea}	[0, 5]	[6, 16]	[17, 25]
training dataset	Mean_{vis}	[0.13, 4.25]	[5, 8]	[8.8, 10]
	Mean_{vol}	[60, 564]	[576, 1164]	[1176, 1908]
	Mean_{ocu}	[1, 8.2]	[8.4, 27.6]	[31.2, 66.4]
	Mean_{spe}	[0, 33.6]	[34, 59.8]	[60, 93]
	Std_{vol}	[0, 112.24]	[115.41, 245.19]	[247.38, 642.58]
	Std_{ocu}	[0, 3.96]	[4, 15.66]	[26.62, 27.07]
	Std_{spe}	[0, 4.15]	[4.21, 11.73]	[12.19, 33.16]
10-minute	Mean_{wea}	[0, 5]	[6, 16]	[17, 25]
training dataset	Mean_{vis}	[0.25, 4.8]	[5, 8]	[8.5, 10]
	Mean_{vol}	[60, 560]	[564, 1152]	[1170, 1890]
	Mean_{ocu}	[1, 7.9]	[8, 28]	[29.7, 66.4]
	Mean_{spe}	[0, 31.6]	[34.37, 59.8]	[59.85, 94.5]
	Std_{vol}	[0, 124.73]	[124.9, 245.68]	[248.51, 699.74]
	Std_{ocu}	[0, 4.17]	[4.36, 16.06]	[17.79, 31.10]
	Std_{spe}	[0, 4.63]	[4.65, 13.48]	[13.62, 31.78]

The clustering results are shown in Table 7-1. Three clusters were generated for each continuous variable, representing: low, medium and high value ranges. The two numbers in each bracket denotes the lower bound and upper bound of a cluster. Through this process, the original eight continuous explanatory variables were transferred into 24 discrete variables (called items in the following analysis). The support of each item or the size of each cluster were obtained and sorted in a descending order as shown in Table 7-2:

Table 7-2 Supports of Items in 5-minute and 10-minute Accident Training Datasets

Index	5-minute training dataset		10-minute training dataset	
	Item	Support	Item	Support
1	Std_{ocu} low	542	Std_{ocu} low	554
2	Mean_{wea} low	435	Mean_{wea} low	436
3	Mean_{vis} high	401	Mean_{vis} high	405
4	Mean_{ocu} low	385	Mean_{ocu} low	383
5	Std_{spe} low	372	Std_{spe} low	373
6	Mean_{spe} medium	337	Mean_{spe} medium	332
7	Std_{vol} low	321	Std_{vol} low	327
8	Mean_{vol} medium	261	Mean_{vol} medium	261
9	Mean_{spe} high	216	Mean_{spe} high	231
10	Mean_{vol} low	170	Mean_{vol} low	188
11	Mean_{ocu} medium	169	Std_{spe} medium	188
12	Std_{vol} medium	169	Std_{vol} medium	184
13	Std_{spe} medium	169	Mean_{ocu} medium	178
14	Mean_{vol} high	149	Mean_{vol} high	145
15	Mean_{vis} medium	125	Mean_{vis} medium	132
16	Mean_{wea} medium	98	Mean_{wea} medium	109
17	Std_{vol} high	90	Std_{vol} high	83
18	Mean_{vis} low	54	Mean_{vis} low	57
19	Mean_{wea} high	47	Mean_{wea} high	49
20	Std_{spe} high	39	Std_{ocu} medium	34
21	Std_{ocu} medium	35	Mean_{ocu} high	33
22	Mean_{spe} low	27	Std_{spe} high	33

23	Mean$_{ocu}$ high	26	Mean$_{spe}$ low	31
24	Std$_{ocu}$ high	3	Std$_{ocu}$ high	6

When screening frequent items, we set the threshold value σ in Equation (7-1) to 0 so that all the items shown in Table 7-2 are considered. The rationale behind this is to prevent any information loss in the variable importance score calculation. Since the items have already been sorted in a support-based descending order, Table 7-2 also provides the F-List to build the FP Tree.

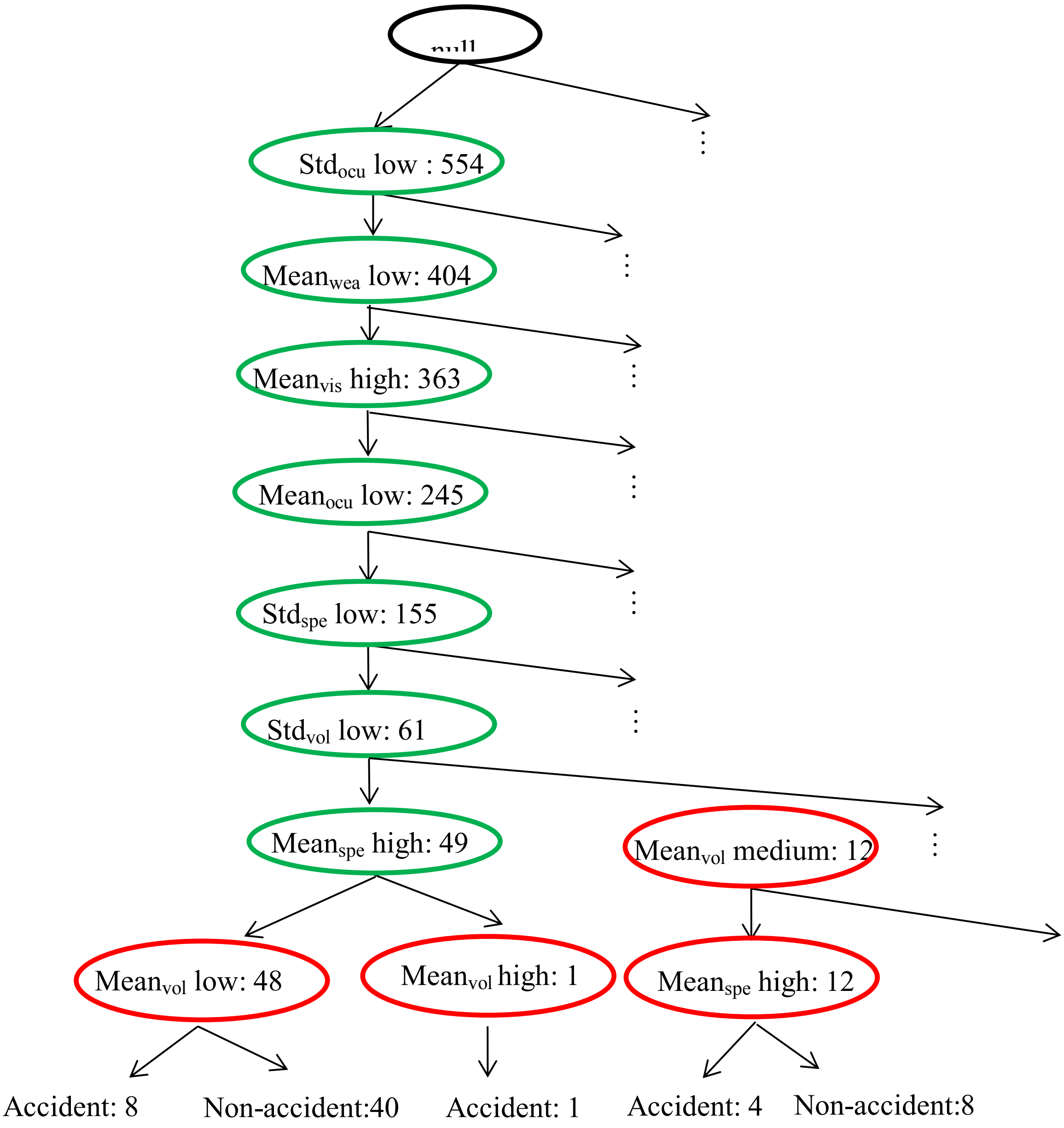


Figure 7-3 A part of the FP Tree for the 10-minute training dataset

FP trees were built for the two training datasets. Due to space limitations, only a part of the FP Tree constructed for the 10-minute training dataset is shown Figure 7-3. Three frequent patterns are demonstrated, including: frequent pattern 1 { Std$_{ocu}$ low→ Mean$_{wea}$ low→ Mean$_{vis}$ high→ Mean$_{ocu}$ low→ Std$_{spe}$ low→ Std$_{vol}$ low→ Mean$_{spe}$ high→ Mean$_{vol}$ low} with $f_{8,1} = 48$; frequent pattern 2 { Std$_{ocu}$ low→ Mean$_{wea}$ low→

Mean$_{vis}$ high→ Mean$_{ocu}$ low→ Std$_{spe}$ low→ Std$_{vol}$ low→ Mean$_{spe}$ high→ Mean$_{vol}$ high} with $f_{8,2} = 1$; and frequent pattern 3 { Std$_{ocu}$ low→ Mean$_{wea}$ low→ Mean$_{vis}$ high→ Mean$_{ocu}$ low→ Std$_{spe}$ low→ Std$_{vol}$ low→ Mean$_{vol}$ medium → Mean$_{spe}$ high} with $f_{8,3} = 12$. Figure 7-3 also marks the shared nodes and exclusive nodes for each pattern, and lists the numbers of accidents and non-accidents observed in each frequent pattern.

With the FP Tree constructed, the variables' importance scores are calculated using Equation (7-3), (7-4) and (7-5). The results are shown in Table 7-3.

Table 7-3 Variable Importance Calculations Results based on FP Tree and Random Forest Methods

Variables	5-minute training dataset		10-minute training dataset	
	FP tree	Random Forest	FP tree	Random Forest
Mean$_{vol}$	46 (1)*	27.31 (3)	48.6 (1)	27.51 (4)
Std$_{vol}$	43.2 (2)	26.98 (4)	42.8 (2)	29.15 (2)
Mean$_{spe}$	16.6 (7)	28.56 (2)	20.8 (7)	29.02 (3)
Std$_{spe}$	35.6 (4)	29 (1)	29 (6)	30.11 (1)
Mean$_{ocu}$	21.6 (6)	25.99 (5)	35.8 (4)	24.89 (6)
Std$_{ocu}$	15.2 (8)	22.19 (6)	15 (8)	26.41 (5)
Mean$_{wea}$	40.2 (3)	8.79 (8)	37.6 (3)	9.88 (8)
Mean$_{vis}$	33.8 (5)	13.77 (7)	30.8 (5)	13.39 (7)

Notes: * The first number is the variable importance score, and the number in the following parentheses is the ranking of variable ("1" means the most important, and "8" means the least important).

We also calculated the variables importance scores based on random forest method (see Table 7-3), using the package “randomForest” within the statistics software R (Liaw and Wiener, 2002). Based on the guidance of this package’s instructions, the number of trees to grow n_{tree} should not be set to a very small number, in order to ensure that every input row gets predicted at least a few times; the default value of 500 was used in this study. The number of variables randomly sampled as candidates at each split n_{var} was set to the square root of the total number of variables (3 in this study). The size of samples $\overline{DB}^{1}, \dots, \overline{DB}^{n_{tree}}$ was set as 0.632*the total size of training dataset; for the 5-minute training dataset, the sample size was set as 366, and for the 10-minute training dataset, the sample size was set as 375. The package “randomForest” produced the mean decrease of Gini index for each variable as an output. As mentioned before, the mean decrease of the Gini index measures the contribution of a variable to the homogeneity of the nodes and leaves in the random forest (Metagenomics Statistics, 2014). The higher the mean decrease of the associated Gini index is, the more important the variable is.

Through the comparison of the variable importance scores generated from the FP tree and Random forest, we can see that the two models produce different variable importance rankings. The FP tree models tended to rank traffic volume related variables, such as Mean_{vol} and Std_{vol} as the top two most important variables while resulting in much lower scores for speed related statistics, particularly for Mean_{spe}. In contrast, traffic speed related statistics variables were deemed slightly more important by the random forest. Nevertheless, the volume related variables were judged important by both of the methods (among the top four). As for the weather related variables, Mean_{wea} was ranked as the third most important variable based on the FP method, while it was scored as the least important by the random forest method.

7.3.2 k-NN

This research tested the performance of k-NN for the 5-minute and 10-minute testing datasets. k was set as 2 and 3 separately, and each time k-NN was run for three scenarios: (1) using all the variables; (2) using all variables *except for* Mean_{spe} and Std_{ocu} which were ranked as the least important by the FP tree method; and (3) using all variables except for Mean_{wea} and Mean_{vis} that were ranked as the least important by random forest. The voting criterion of k-NN in this research is that once one of k nearest neighbors has the response variable equal to 1 (indicating the occurrence of an accident), the predicted response of the observation is set as 1. The results can be seen in Table 7-4.

Table 7-4 Performance of k-NN for Different Variable Selection

Variable selection	Criteria	5-minute testing dataset		10-minute testing dataset	
		k=2	k=3	k=2	k=3
All variables	Sensitivity	32.35%	50.00%	40.00%	48.57%
	False alarm rate	36.03%	52.25%	42.98%	56.14%
FP Tree	Sensitivity	38.23%	52.94%	45.71%	60.00%
	False alarm rate	34.23%	51.35%	42.10%	54.38%
Random Forest	Sensitivity	32.35%	44.11%	37.14%	54.28%
	False alarm rate	53.15%	64.86%	49.12%	61.40%

Note that there are two prediction performance measures used as shown in Table 7-4: (1) sensitivity, which measures the proportion of actual accidents that were accurately predicted as such; and (2) the false alarm rate that refers to the proportion of normal situations that were wrongly predicted as accidents. A good traffic accident risk prediction model should yield a high sensitivity and a low false alarm rate.

The major findings are summarized below according to Table 7-4. First of all, although k-NN doesn't perform well in general, using the FP tree to pre-select the explanatory variables significantly improved the prediction accuracy. In comparison to the all variables case, the FP tree based k-NN model always produces higher prediction sensitivity values and lower false alarm rates no matter which testing dataset is used. In contrast, there is no benefit from a random forest based variable selection with the only exception of the case of k=3 with the 10-minute testing dataset. This indicates the advantage of FP tree in sorting out important affecting factors and improving model prediction performance. Second, regardless of the type of testing datasets used, the comparison between the k-NN model with k=3 and the one with k=2 shows that adding one nearest neighbor will significantly increase the prediction sensitivity, however, as can be seen, this will also increase the false alarm rate. Lastly, the k-NN models work better for the 10-minute testing dataset than for the 5-minute testing dataset in terms of prediction sensitivity. However, the false alarm rates tend to be higher for the 10-minute testing dataset as well. This indicates that the pre-crash time period may also affect model performance.

7.3.3 Bayesian network

Bayesian network models were also built to predict accident risk for comparison. As a crucial step to perform Bayesian network modeling, the continuous variables need to be discretized. How to transform a continuous variable to discrete category variables vastly depends on the objectives set by researchers (Hossain and Muromachi, 2012). Among the discretization techniques available in the literature, we selected the normalized equal distances (NED) method, using the software Bayesialab due to its promising performance (Bayesia, 2013). The values of each variable are first normalized based on the mean and standard deviation of the variable (Han et al., 2011). Then, the normalized values are split to the user-defined number of equal width discrete intervals (Kotsiantis and Kanellopoulos, 2006). For this research, we set the number of equal width discrete intervals as 3 and 4 separately.

We considered one of the most plausible Bayesian network structures, which just let the response variable be the child node of the possible explanatory variables (Hossain and Muromachi, 2012). Three scenarios, as before, were tested under the structure: (1) using all the variables; (2) using all except of Mean_{spe} and Std_{ocu} that are ranked as the least important by FP tree; and (3) using all except Mean_{wea} and Mean_{vis} that are ranked as the least important by random forest. The software Netica was used to learn the Bayesian network parameters (Netica tutorial, 2014). As an example, the Bayesian network for the 10-minute dataset using all variables without Mean_{spe} and

Std_{ocu} and with 4 discrete intervals for each continuous variable is demonstrated in Figure 7-4.

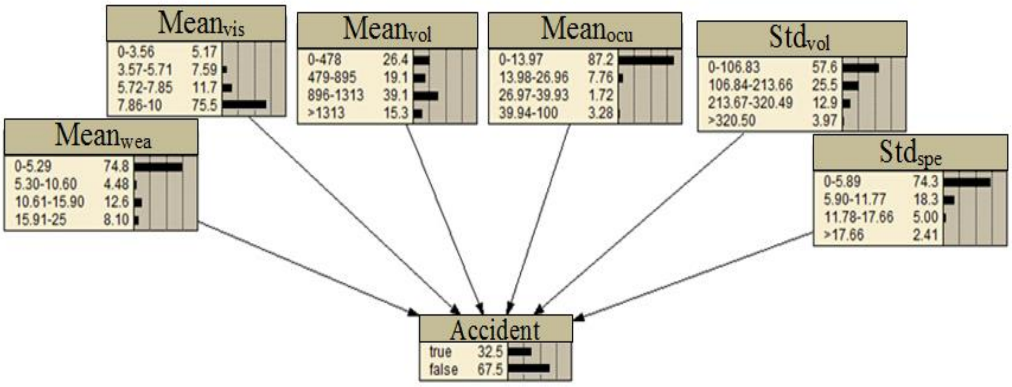


Figure 7-4 Bayesian Network for 10 min Dataset Using Variables based on FP Tree

As can be seen in Figure 7-4, each of the six explanatory variables was split into four intervals by NED. The number right next to an interval is the probability for the value of a variable to fall into that interval. Taking Mean_{vol} as an example, 26.4% of records have the values ranging between 0 and 478 vehicles/hour.

The probability threshold of Bayesian network was set to 0.2, which means that if the estimated accident probability is greater than 0.2, we predict that an accident would happen. The performance of Bayesian networks with different NED numbers (in parentheses), and for the 5-minute and 10-minute testing datasets are shown in Table 7-5.

Table 7-5 Prediction Performance of Bayesian Network with Different Variable Selection Strategies

Variable selection	Criteria	5-minute testing dataset	10-minute testing dataset

		NED (3)	NED (4)	NED (3)	NED (4)
All variables	Sensitivity	47.37%	31.57%	50.00%	61.11%
	False alarm rate	50.67%	32.00%	47.37%	47.37%
	Overall Performance	48.93%	60.63%	52.12%	54.25%
FP Tree	Sensitivity	52.63%	31.57%	44.44%	61.11%
	False alarm rate	52.00%	30.67%	38.15%	38.16%
	Overall Performance	48.93%	61.70%	58.51%	61.70%
Random Forest	Sensitivity	63.15%	47.37%	33.33%	61.11%
	False alarm rate	69.33%	60.00%	42.10%	53.95%
	Overall Performance	37.23%	41.48%	53.19%	48.93%

Several observations can be discerned from Table 7-5. First, the best Bayesian network model results in a sensitivity value as high as 61.11% and a false alarm rate as low as 38.16% when trained based on the 10-minute dataset with the NED number equal to 4. These results compare very favorably to those in the literature, especially given that only one detector was used. Second, the number of NED can affect the performance of Bayesian network. For the 5-minute dataset, the sensitivity and false alarm rate both decreased when the number of NED was set to 4 instead of 3. On the other hand, for 10-minute dataset, the sensitivity improved, but the false alarm rate remained almost the same when NED number is changed from 3 to 4, except for the situation using the variables based on random forest, for which the false alarm rate also increased. Third, for the majority of cases, the Bayesian network models using variables selected by FP tree perform better than the ones using the random forest selected variables. For example, for the 10-minute dataset, when NED number is 4, although the sensitivity values of the two types of models are somewhat similar (around 61%), the false alarm rate of the random forest based Bayesian network model is much higher than its FP tree based counterpart. For other cases, however (e.g., the models based on the 5-minute training dataset), it is hard to decisively conclude that the models based on FP tree performed better than those based on random forest because the sensitivity and false alarm rate of the former are *both* lower than those of the latter. Because of this, we introduced a third criterion, called the overall performance to measures the ratio of correct predictions (no matter whether it is accident or a non-accident) in the whole testing dataset. Based on the overall performance criterion, we can easily see that the models based on variables selected by FP tree significantly outperform those based on all the variables or based on random forest.

7.4 Conclusions

This study proposed a novel variable selection algorithm based on FP tree for real-time traffic accident risk prediction. The importance score of each explanatory variable in the dataset is calculated and ranked through the calculation of the ROPR of the corresponding frequent patterns. This variable selection algorithm was tested on the Virginia traffic accident dataset collected in 2005 in comparison to the widely used random forest variable selection. Based on the variables selected by the two methods, two traffic accident risk prediction models, the k-NN and Bayesian network models, were developed and tested for three situations: using all variables, using the important variables selected by FP tree, and using the important variables selected by random forest. The major findings are summarized as below:

(1) Generally, the accident risk prediction results are quite acceptable when using the Bayesian network model with NED number equal to 4 and based on a 10-minute dataset. This is especially true for the case using variables selected by FP tree, where the sensitivity was as high as 61.11% and the false alarm rate was as low as 38.16%. Considering that only data from one detector were available in this study, these results are very promising.

(2) In terms of the time resolution to be used in compiling the datasets, no decisive conclusions can be made regarding whether a 5-minute or a 10-minute resolution would yield better performance. For Bayesian network, the overall performances are improved by using the 10-minute dataset except the cases with NED number set as 4, using all variables and FP tree based variables.

(3) The most important finding of this research is that the accident risk prediction models based on FP tree variable selection outperform the models based on all variables and the ones based on random forest, regardless of the settings of the prediction models such as the selection of k for k-NN, the NED number selected for Bayesian network, and the pre-crash time period used in the datasets. Being insensitive to the selection of the models' parameters is a good quality that the FP tree variable selection algorithm appears to possess.

CHAPTER 8 TRAFFIC ACCIDENT DURATION PREDICTION BASED ON M5P TREE AND HBDM

This chapter introduces the integration of the data mining model M5P tree and the statistical model hazard-based duration model (HBDM) for traffic accident duration prediction. The arrangement of this chapter is as following: first, the basic methodologies of M5P tree and HBDM are introduced in Chapter 8.1, as well as the new algorithm to build the M5P-HBDM; after that, descriptive analysis of the two traffic accident datasets, I-190 and I-64 traffic accident datasets, will be presented in Chapter 8.2; then in Chapter 8.3, each traffic accident duration dataset is split into a training dataset and a testing dataset. The significant variables selected by the three models M5P, HBDM, and M5P-HBDM based on the training dataset are compared and analyzed to see which model can find more meaningful variables. The prediction performances of these models for the testing dataset are also compared in detail in Chapter 8.3; finally, the conclusions and future work will be discussed in Chapter 8.4.

8.1 Methodology

This chapter proposed a new traffic accident duration prediction model M5P-HBDM based on the decision tree model M5P tree and the statistical model HBDM. The traditional decision trees were proposed by Breiman et al. (1984), however, these trees have fixed average values at their leaves that cannot model stochastic nature of the parent-child relationship in a realistic way (Ozbay and Noyan, 2006). Considering this, Quinlan (1992) developed a new type of tree named the M5 tree which can have multivariate linear models at their leaves, so more flexible predictions are allowed. In order to handle the enumerated attributes and attribute with missing values, Wang and Witten (1997) proposed a modified M5 tree algorithm and called it M5P algorithm. M5P has the advantages like dealing with categorical and continuous variables and handling variables with missing values (Zhan et al., 2011).

M5P tree has been applied by Zhan et al. (2011) to predict lane clearance time of freeway incidents. But one problem with M5P tree is that in linear regression process to build its leaves, the residuals are assumed to be distributed normally, in another word, the accidents clearance time is assumed to follow a normal distribution too. However, the distribution for time to an event (here it is the time when the traffic returns to normal) is almost certainly nonsymmetrical (Cleves et al., 2008).

For HBDM, it is a statistical model to analyze the duration of a specific event. Different distributions of the duration can be assumed like Weibull distribution, log-normal distribution, log-logistic distribution and so on. It has been applied directly to the whole accident dataset to analyze and predict the duration (Alkaabi et al., 2011; Nam and Mannering, 2000; Chung, 2010). However, as far as we know, no researches have been conducted to test the impact of classification method to HBDM. Based on the classification of the accident dataset, it is interesting to explore whether more insights about the relationship between accident duration and the explanatory variables can be gained and whether the prediction performance can be improved with the HBDM.

Therefore simply speaking, we hope the M5P-HBDM can keep the ability of M5P tree to classify the traffic accident dataset but to build the HBDMs with the best selected distribution at its leaves instead of the linear regression model with the normal distribution assumption. This section will introduce M5P tree and HBDM first. After that, the detailed algorithm to build the M5P-HBDM will be given.

8.1.1 M5P Tree

Before introducing M5P algorithm, the constructing process of M5 tree should be understood (Quinlan, 1992). Assume there is a collection of T^n training cases at node n ($n = 0$ for the root node), each case has a fixed set of attributes, either discrete (binary or category) or numeric, and has a target value. In tree growth step, calculate the standard deviation $sd(T^n)$ of the target values of cases in T^n, suppose there is a test tree which splits T^n into N outcomes, let T_i^n denote the subset of cases that have the i^{th} outcome of the potential test, $sd(T_i^n)$ denote the standard deviation of the target values of cases in T_i^n, $|T_i^n|$ means the number of cases in T_i^n, $|T^n|$ is the number of cases in T^n. The object function is to find the potential test which maximizes the error reduction which is calculated by Equation (8-1):

$$\Delta error = sd(T^n) - \sum_{i=1}^{N} \frac{|T_i^n|}{|T^n|} \times sd(T_i^n) \qquad (8\text{-}1)$$

The same process is applied *recursively* to the subsets using the divide-and-conquer method, until the subsets at a node contain only a few instances or vary very slightly. Therefore there are two termination thresholds, the first one is $TH1$, the minimum number of cases at a node, and the second one is $TH2$, used in checking whether the standard deviation of the target values at the node is less than $TH2 * sd(T^0)$. The nodes where the split terminates are marked as leaf, otherwise the types of the nodes are marked as interior or non-leaf.

After the initial tree has been grown, a multivariate linear model is constructed for each non-leaf node of the model tree using standard regression techniques. In M5 algorithm, the linear model can only use the attributes that are referenced by sub-tree at this node. This is because M5 will compare the accuracy of a linear model with the accuracy of a sub-tree, it is fair that they use the same information. Besides that, in order to prevent the overfitting problem of the linear model, M5 uses a greedy search to remove variables that contribute little to the linear model, so sometimes the linear model can just be a constant. These linear models for non-leaf nodes will be useful in the next two steps: pruning step and smoothing step.

In pruning step, starting near the bottom, examine each non-leaf node of the model to decide whether this node should be replaced with the linear model gotten above as a new leaf node or kept the subtree unchanged. This depends on which one can bring the lower estimated error. It’s worth noting that for each leaf node, there is an associated value used for error calculation, which is the averaged target value of the training cases at that leaf. The estimated error is calculated using Equation (8-2):

$$Error = \frac{N+v}{N-v} * \frac{\sum_{i=1}^{n} abs(V_{act}-V_{pre})}{N} \tag{8-2}$$

So the estimated error is the average absolute difference between the actual target values V_{act} of the training cases and the predicted values V_{pre} given by the linear model at the current node (or the averaged target value for the leaf node), adjusted by $(N+v)/(N-v)$, where N is the number of training cases going through this current node, v is the number of the parameters in the linear model. This is also to avoid the overfitting problem. For the estimated error of the subtree, one more thing need be mentioned is that the error from each branch is combined into a single overall value for the node using a linear sum in which each branch is weighted by the proportion of the training cases that go down it (Wang and Witten, 1997).

At last, Quinlan (1992) observes that the prediction of M5 tree can be improved by a smoothing step. The smoothing step can compensate for the sharp discontinuities that will inevitable occur between adjacent linear models at the leaves of the pruned tree, particularly for some models constructed from a small number of training cases (Wang and Witten, 1997). The predicted value at the leaf is the value computed by the linear regression model or the constant averaged target value at the leaf, then the value is filtered along the path back to the root, smooth it at each node by combining it with the value predicted by the linear model for that node. The calculation is shown in Equation (8-3):

$$p' = \frac{np+kq}{n+k} \tag{8-3}$$

Where p' is the prediction passed up to the next upper node, p is the prediction passed to this current node from a lower node, q is the value predicted by the linear model at this current node, n is the number of training cases from the lower branch of this current node, and k is a user-defined value (default value 15).

The improvement of M5P algorithm to M5 is that before tree construction all enumerated attributes are transformed into binary variables to handle enumerated attributes, if the enumerated attribute has c possible values, it will be replaced by $c - 1$ synthetic binary attributes. So in M5P, all splits are binary: they involve either a continuous-valued attribute or a synthetic binary one (Wang and Witten, 1997). Besides that, in order to take account of missing values, Equation (8-1) is revised as following:

$$\Delta error = \frac{m}{|T^n|} \times \beta \times [sd(T^n) - \sum_{i=\{L,R\}}^{N} \frac{|T_i^n|}{|T^n|} \times sd(T_i^n)] \qquad (8\text{-}4)$$

where, m is the number of training cases without missing values;

β is called correction factor, for original binary and continuous attributes, $\beta = 1$; for enumerated attributes, β decays exponentially as the number of synthetic binary attributes increases, $\beta = exp(7 * (2 - max(2, c))/|T^n|)$ (Jekabsons, 2010). This is to solve the problem that the enumerated attributes having a large number of different values are automatically favored;

$i = \{L, R\}$ means in M5P the split is binary which generates the left subtree and right subtree.

8.1.2 Hazard-based Duration Model (HBDM)

Suppose the duration of a specific traffic accident is represented by a continuous random variable D. The cumulative distribution function is $F(d)$, which is also called failure function in HBDM, as shown in Equation (8-5):

$$F(d) = \int_0^d f(u)du = \mathrm{P}(D < d), 0 < d < \infty \tag{8-5}$$

$F(d)$ tells the probability that the duration D is less than time value d, and its probability density function is shown in Equation (8-6):

$$f(d) = \frac{\delta F(d)}{\delta d} = lim_{\Delta d \to 0} \frac{P(d \le D < d + \Delta d)}{\Delta d} \tag{8-6}$$

$f(d)$ describes the instantaneous failure rate in the infinitesimally small interval $[\mathrm{d}, \mathrm{d} + \Delta\mathrm{d}]$. One more function need be introduced in HBDM, which is called survival function, $S(d)$ and defined in Equation (8-7):

$$S(d) = 1 - F(d) = P(D \geq d) \tag{8-7}$$

$S(d)$ gives the probability that duration D is longer than time value d.

At last, after the probability density function $f(d)$ of failure function and survival function $S(d)$ are known, the hazard function $h(d)$ is defined in Equation (8-8) as following:

$$h(d) = \frac{f(d)}{S(d)} = \lim_{\Delta d \to 0} \frac{P(d \le D \le d + \Delta d | D \ge d)}{\Delta d} \tag{8-8}$$

$h(d)$ can be interpreted as the instantaneous failure rate at time d, given that the duration has lasted d minutes. The difference between $f(d)$ and $h(d)$ is that $h(d)$ is the normalized rate after knowing the duration will not end before d. When $\mathrm{d} = 0$, $h(d)$ is the same as $f(d)$.

There are two alternative parametric approaches to investigate the effects of explanatory variables using HBDMs: the proportional hazard metric (PH) and accelerated failure time metric (AFT). PH function is shown as following:

$$h_i(t) = h_0(t)\exp(\beta x_i) \tag{8-9}$$

where, $h_0(t)$ is called baseline hazard which can be specified a functional form;

x_i is the value vector of variables of case i;

β is the vector of estimated coefficients.

We can see that little attention is paid to the actual failure times and predictions of these failure times are seldom desired in PH. In contrast, the AFT gives a more prominent role to analysis time (Cleves et al., 2008). So as the other researches (Alkaabi et al., 2011; Chung, 2011), AFT is used in this paper. AFT assumes a distribution for

$$\tau = exp(-x_i\beta)d_i \tag{8-10}$$

where, τ may have a specified distribution like the Weibull distribution, Log-normal distribution, Log-logistic distribution and so on;

d_i is the duration of case i;

x_i is its value vector of variables;

β is the vector of estimated coefficients.

After taking logarithm for both sides, the AFT model can be framed as a linear model as shown in Equation (8-11):

$$ln(d_i) = x_i\beta + ln(\tau) \tag{8-11}$$

where, $ln(d_i)$ is the natural logarithm of the survival time. When the parameters in β and τ are estimated, given a new observation of a case, the mean and medium of the failure time distribution can be calculated as its predictions (Cleves et al., 2008).

8.1.3 M5P-HBDM Model

After introducing M5P and HBDM models, this section will talk about how to combine the two models. Table 8-1 shows the Pseudo-codes of M5P-HBDM algorithm, and compares it with the original M5P algorithm (Wang and Witten, 1997).

Table 8-1 the Pseudo-Code of M5P-HBDM Algorithm and Comparison with M5P Tree

M5P-HBDM (T^0 training cases)

```
{
  SD=sd($T^0$)
  For each c-valued category variable, convert into c-1 synthetic binary variables,
  root=new_node,
  root.trainingcases=$T^0$,
  split(root),
  prune(root),
  print_tree(root),
}

split(node)
{
  if sizeof(node.trainingcases)< $TH1$ or sd(node.trainingcases)< $TH2$*SD
    node.type=LEAF,
    node.model1=HBDM(node),
    node.model2=average of the target values of the cases at this leave node.
    if error(node.model1)<error(node.model2)
        node.model=node.model1,
    else
        node.model=node.model2,
  else
      node.type=INTERIOR,
      for each continuous and binary variable,
          for all possible split positions,
                    calculate the $\Delta error$ from the Equation (8-4),
            node.variable=variable with max $\Delta error$
            split (node.left),
            split (node.right),
}

prune(node)
{
   if node.type=INTERIOR then
            prune(node.left_child),
            prune(node.right_child),
         node.model=HBDM(node),
         (for M5P algorithm, node.model=linear_regression(node))
            if subtree_error(node)> $error$(node.model) then
                  node.type=LEAF
}

subtree_error(node)
{
   l=node.left;
   r=node.right,
   if node=INTERIOR then
              return
         (sizeof(l.trainingcases)*subtree_error(l)+sizeof(r.trainingcases)*subtree_error(r))
         /sizeof(node. trainingcases)
     else
              return
              $error$(node.model)
}

$error$(node.model)
{
   predict the target values using node.model, the model can be a constant, which is
   the average or the median of the target values for the original leave nodes; it can
   be a HBDM, the prediction is the mean or media value of AFT with a selected
   distribution shown in Equation (8-11). (for M5P algorithm, the model of the
   node can be the linear regression model, except for the original leave nodes, it
   can only be the average of the target values)
   calculate the estimated error based on Equation (8-2),
}

sizeof (node.trainingcases),
{
    returns the number of training cases that go through the current node,
}
```

As can be seen in Table 8-1, the building process of M5P-HBDM model is very similar as M5P model; two main steps tree growth and tree pruning are kept. The different parts of the two algorithms are marked as bold, italic and underlined.

First, in split step for tree growth, when the stop criteria are met, the node will be marked as a leave node, in the M5P algorithm, the average of the target values will be calculated for this leave node, but in the HBDM-M5P algorithm, we will continue to build a HBDM model using the training cases at this leave node, if the prediction performance of the HBDM model is better than the constant average value, we will use the HBDM model as the model of the leave node. This is shown in the "split(node)" function.

Second, in prune step, a model need be built for each interior/non-leaf node. In the original M5P algorithm, as mentioned in Chapter 8.1.1, the function linear_regression (node) can build a linear regression model for the current node restricted to the variables that are referenced by the subtree, and then greedily drops the variables if doing so decreases the prediction errors calculated using Equation (8-2). In another word, the linear regression models in the original M5P algorithm do not consider problems like whether the variables are significant or whether the signs of the variables are meaningful. For the M5P-HBDM algorithm, a HBDM model will be built for that node in a statistical approach based on all the variables except those that have been taken by the higher-level nodes in the path from the root to the current node. The prediction performance of HBDM model as well as the p-values of the variables, the signs of the variables and so on will all be checked. Here we also loose the limitation of the candidate variables, not just including those referenced by the subtree of the current node, because (a) in the M5P tree growth step, all the possible variables have been tested to find the best ones for splitting the nodes, in that sense, HBDM model should also be built based all the available variables, and (b) some variables that are not chosen to split the node may be useful for HBDM model. For example, in Figure 8-1, the tree growth step has been finished, the black nodes are the interior nodes, and the green ones are the leaf nodes. Suppose there are seven processed binary variables in this dataset {V1, V2, V3, V4, V5, V6, V7}, now in the tree pruning step, for the node using V2, the original M5P algorithm builds the linear regression model based on {V2, V3, V4}, while the M5P-HBDM algorithm builds the HBDM model from {V2, V3, V4, V5, V6, V7}.

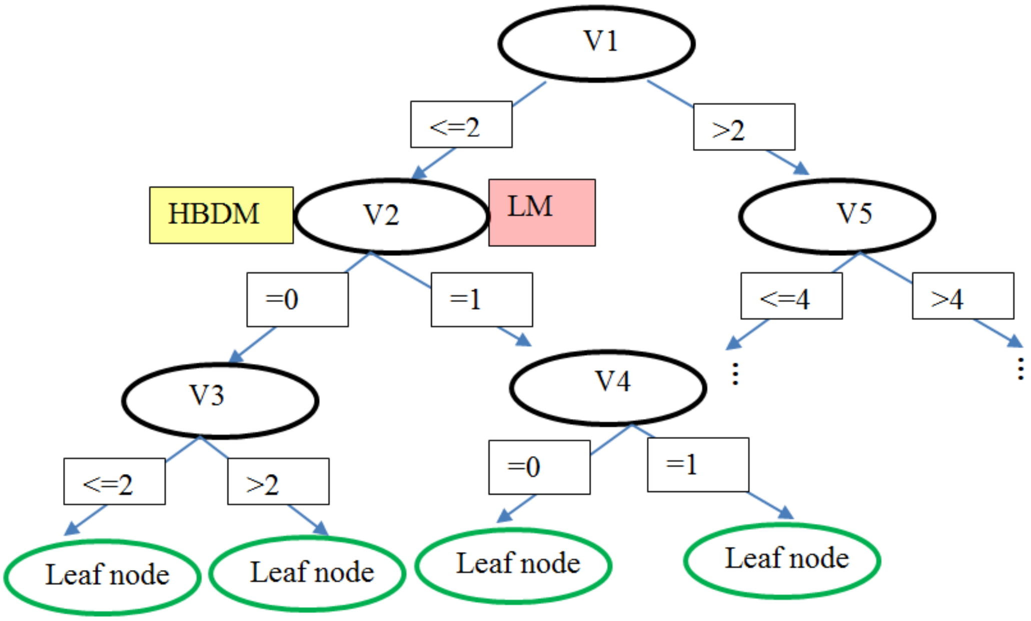


Figure 8-1 an Example of Tree Pruning Step

Third, to calculate the estimated error of the model at the node of M5P-HBDM, the model of the node can be the constant value calculated by taking the average or the median of the target values, which will be the prediction of the traffic accident duration; the model of the node can also be a HBDM, the predictions of the target values could be the mean or media value of the AFT with a selected distribution shown in Equation (8-11). It is different with the prediction calculation using the constant average values or the linear regression models in the M5P tree, the detail will be introduced later.

8.2 Modeling Datasets

8.2.1 Virginia Traffic Accident Dataset

This dataset includes the traffic accident records in 2005 and 2006 from a part of interstate highway I-64 in Norfolk, Virginia. At last, 602 accident records are picked out. For each record, 17 variables are used to describe it. These variables are summarized in Table 8-2.

Table 8-2 Traffic Accident Variables in I-64 Dataset

Variables	Values	Type
Season	Spring (March, April, May); Summer (June, July, August); Autumn (September, October, November); Winter (December, January, February)	Categorical
Weekday	Yes (Monday 2 AM-Friday 9 PM, except holidays); No	Categorical
Hour of the day	Morning (7 AM-9 AM); Early afternoon (10 AM-12 Noon); Afternoon (1 PM-3 PM); Evening rush (4 PM-6 PM); Evening (7 PM-9 PM); Night (10 PM-6 AM)	Categorical
Weather conditions	Clear; Rain; Snow	Categorical
Direction	East Bound; West Bound	Categorical
Location code	1; 2; 3; 4; 5; 6; 7; 8 ;9 (the codes mean different detectors)	Categorical

Lane number at main road	2; 3; 4	Categorical
Road structure	Highway; Ramp	Categorical
Detection source	CCTV; FIRT; Phone Call; SSP; TMS Camera; VSP CAD; VSP Radio; Other	Categorical
Accident Type	Car; Wrong Way; Truck/Tractor trailer; Motorcycle; car to facility; Others	Categorical
Moving to shoulder	Yes; No	Binary
Fire	Yes; No	Binary
Roll over	Yes; No	Binary
Number of vehicles involved	0, 1, …	Continuous
Blocked lanes	0; 1; 2; 3; 4	Categorical
Injured number	0, 1, …	Continuous
Duration	0, 1, …	Continuous

As can be seen, there are three temporal variables: season, weekday and hour of the day; one environmental variable: weather conditions; four geographic variables: direction, location code, lane number at main road and road structure; nine accident outcome variables: detection source, accident type, moving to shoulder, fire, roll over, number of vehicles involved, blocked lanes, injured number and duration. From the types of the variable values, we divide them into categorical variable, binary variable, and continuous variable.

Among these traffic accident relevant variables, "location code" can be from "1" to "9", this means the nearest traffic detector from the accident location, for example, "1" represents "W64-01". "Detection source" is included because it is interesting to know whether the accident reporting way has an impact on the accident duration. "Accident type" may affect the way and equipment of the removal work on the accident parties, which in turn affect the accident duration (Chung, 2010). At last, when the emergency team arrives at the scene, the first thing they will try to do is to move the involved vehicles to the shoulder, so “Moving to shoulder” is included because it is generally assumed that the accidents with involved vehicles moved to shoulder contribute to shorter duration.

8.2.2 Buffalo Traffic Accident Detector

This dataset has been previously described in Chapter 6.2. However, in this study, more variables are included, and some traffic accident records are deleted from this dataset because of the unclear values. At last 616 traffic accident records are available. Similarly, Table 8-3 summarizes these variables.

Table 8-3 Traffic Accident Variables in I-190 Dataset

Variables	Values	Type
Season	Spring (March, April, May); Summer (June, July, August); Autumn (September, October, November); Winter (December, January, February)	Categorical
Weekday	Yes (Monday 2 AM-Friday 9 PM, except holidays); No	Categorical
Hour of the day	Morning (7 AM-9 AM); Early afternoon (10 AM-12 Noon); Afternoon (1 PM-3 PM); Evening rush (4 PM-6 PM); Evening (7 PM-9 PM); Night (10 PM-6 AM)	Categorical
Visibility	0-10	Categorical
Wind speed	0 mph (miles per hour), …,	Continuous
Weather conditions	Clear; Rain; Snow	Categorical
Direction	North Bound; South Bound	Categorical
Location code	1; 2; …; 24; 25; 26 (the codes represent different exits at I-190)	Categorical
Lane number at main road	2; 3; >=3	Categorical
Lane number at ramp	0 (away from exit); 1; 2	Categorical
Ramp type	On ramp; off ramp; highway to highway on ramp;	Categorical

	highway to highway off ramp	
Ramp layout	On ramp, off ramp; off ramp, on ramp; only off ramp; only on ramp	Categorical
Road structure	Before the exit; at the exit; beyond the exit; highway; ramp; bridge; before the bridge; after the bridge	Categorical
Accident Type	Car; Wrong Way; Truck/Tractor trailer; Motorcycle; car to facility; Others	Categorical
Blocked lane	N/A at main road; Left lane at main road; middle lane at main road; right lane at main road; left two at main road; right two at main road; left and right lanes at main road; all lanes at main road; N/A at ramp; left lane at ramp; right lane at ramp; all lanes at ramp	Categorical
Blocked lanes number at main road	0; 1; 2; 3	Categorical
Blocked lanes number at ramp	0; 1; 2	Categorical
Injured	Yes; No	Binary
Roll over	Yes; No	Binary
Congestion	Yes; No	Binary
Fire	Yes; No	Binary
Number of vehicles involved	0, 1, greater than or equal to 2	Categorical
Duration	0, 1, …	Continuous

In this dataset, there are 23 variables in total. The three temporal variables are the same as those in the I-64 dataset: season, weekday and hour of the day. There are three environmental variables: visibility, wind speed and weather conditions; seven geographic variables: direction, location code, lane number on main road, lane number on ramp, ramp type, ramp layout and road structure; ten accident outcome variables: accident type, block lane index, blocked lanes number at main road, blocked lanes number at ramp, injured, roll over, congestion, fire, number of vehicles involved and clearance time.

Some variables in I-190 dataset also need be explained a little further. “Location code” here can be from “1” to “26”, this is the number of the nearest exit from the accident location. For example, “1” means the accident is closest to Exit 1 at I-190. “Ramp type” here can be “highway to highway on ramp” or “highway to highway off ramp”, this is because I-190 is connected to another two highways “I-290” and “I-90”. If the ramp is from the other highway to I-190, we classify the ramp as “highway to highway on ramp”. “Ramp layout” is the layout of the ramps at the exit. The relative location order of “on-ramp” and “off-ramp” may impact the accident duration. “Blocked lane” records the blocked lane at the main road or the ramp caused by the traffic accident if the relevant information can be extracted.

Comparing the two datasets, we can see that the records have different emphasis on traffic accidents characteristics. The I-64 accident dataset records detailed information about moving to shoulder and detection source. In contrast, the I-190 accident dataset include more information like which lane the accident occurs at, whether it happens at main road or ramp and so on.

8.2.3 Accident Duration Characteristics

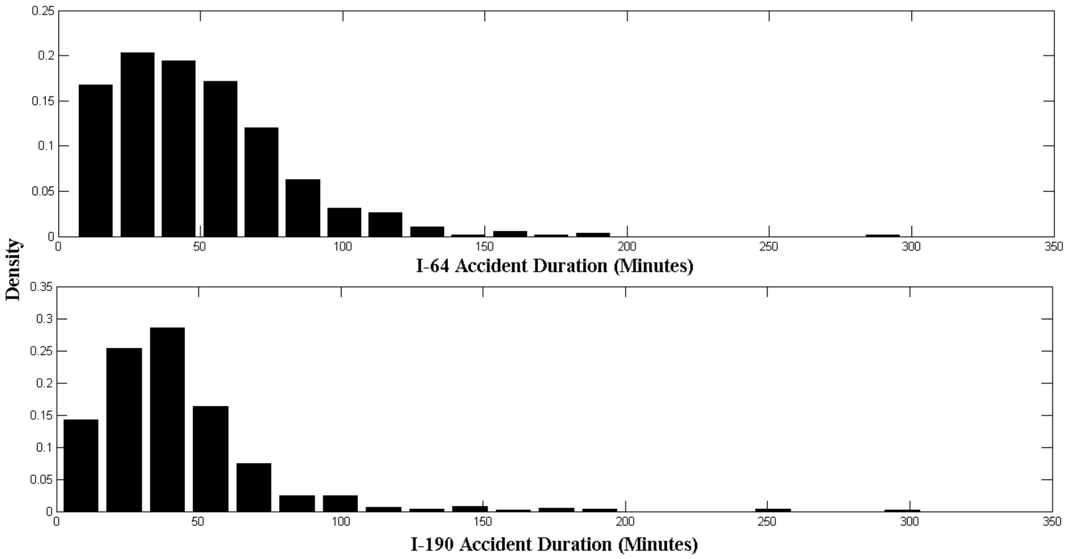


Figure 8-2 Density Distributions of Accident Duration for I-64 and I-190

In order to have a more direct understanding to the accident duration, Figure 8-2 shows the density distributions of duration for the two datasets. It can be seen that both density distributions are skewed to the right.

Table 8-4 Statistical Analysis of Accident Duration for I-64 and I-190

Datasets	Mean	Median	Minimum	Maximum	Standard Deviation
I-64	49.71	45	6	297	31.69
I-190	42.28	36	1	305	32.05

Besides that, Table 8-4 shows some basic statistical analysis for the two datasets. The mean and median traffic accident durations of I-64 dataset are a little longer than those of I-190 dataset, but its standard deviation is a little shorter than I-190.

8.3 Model Development and Comparison

As mentioned before, the I-64 dataset includes 602 traffic accident records and the I-190 dataset includes 616 traffic accident records. For each dataset, the first 500 records are used for model training, the rest records will be used for testing.

8.3.1 M5P Tree

In this study, a Matlab package called M5PrimeLab (Jekabsons, 2010) was used for M5P tree model development. This package provides different options to build the M5P tree. For both I-64 and I-190 training dataset, the smoothing parameter k in Equation (8-3) is set as the default value 15. Besides this parameter, the users also need decide two thresholds: the minimum number of training records at one node $TH1$ and the ratio of the standard deviation $TH2$ mentioned in section 8.1.1.

Although the value of $TH1$ can be set as low as 2, in order to build the linear regression models for the non-leaf nodes after the tree growth step, we don't want the node has too few records. So here we tested $TH1$ from the interval 5% to 10% of the total number of training cases, which is from 25 to 50.

After a few times of experiments, for I-64 training dataset, $TH1$ is set as 30, and $TH2$ is set as 0.95. The following Figure 8-3 shows the M5P tree model for I-64 dataset.

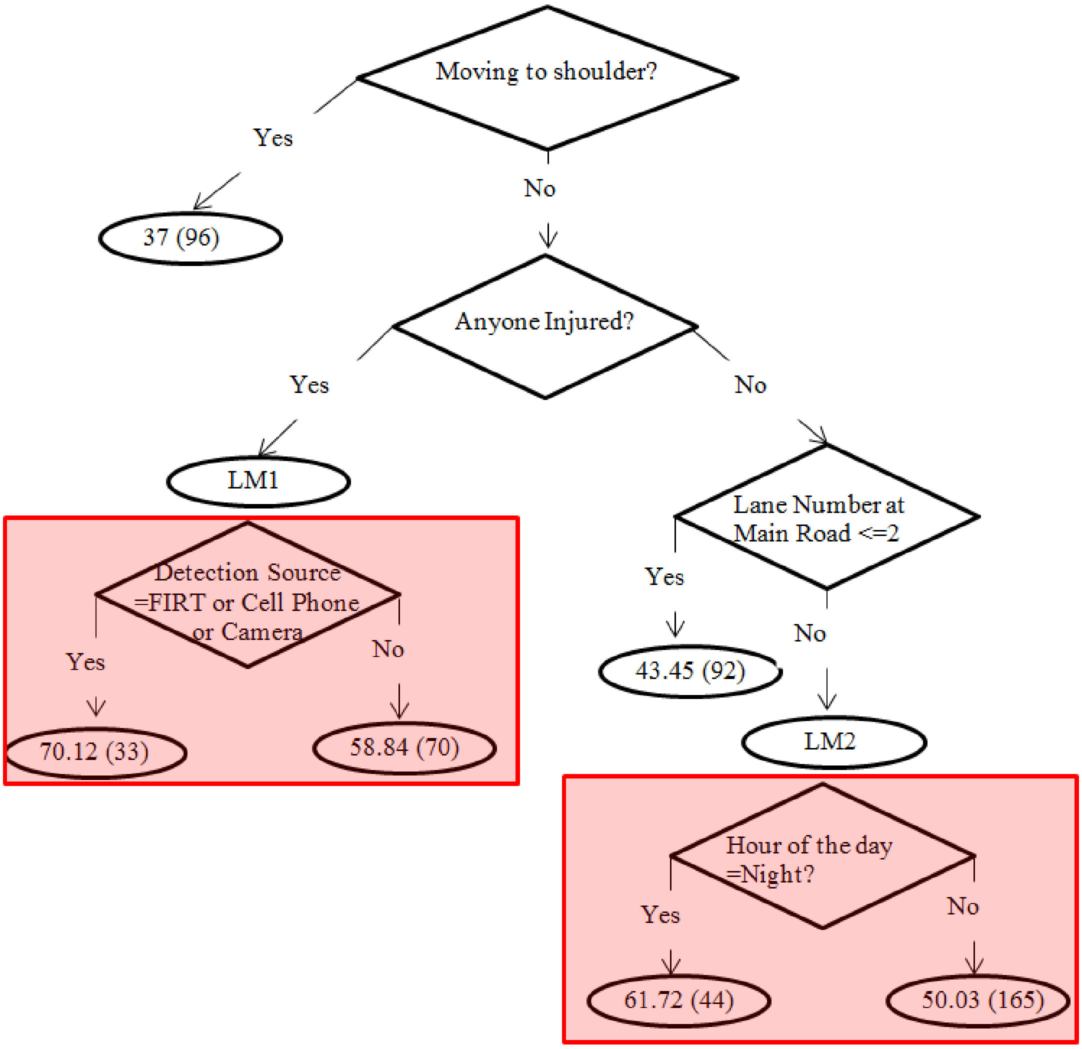


Figure 8-3 M5P Tree Model for I-64 Training Dataset

As can be seen in Figure 8-3, for some leaf nodes, there are a constant value and a number in the parenthesis, which are the averaged value and the number of the training cases in that node; there are also two linear models in two leaf nodes, LM1 and LM2. In the tree pruning step they replaced the original sub trees covered by the red rectangles. The detail of LM1 and LM2 are listed as following. Here the two linear regression models are two constants. From section 8.1.1, it's known that to build a linear regression model for an interior node, M5P uses a greedy search to remove variables that can't improve the predictions for the cases going through that node, so sometimes the linear model can just be a constant. The number in the parentheses is the number of training cases at that leave.

LM1: Duration=62.46 minutes (103 cases);

LM2: Duration=52.49 minutes (209 cases);

First, from the splitting rule at the root node, it's worth noting that if the involved vehicles can be moved to the shoulder once the accidents happen, the average accident duration is only 37 minutes, that is much lower that all the other scenarios. Second, if the involved vehicles were not moved to shoulder, and there was someone injured, LM1 shows the estimated accident duration can be as long as 62.46 minutes. If we check the subtree replaced by LM1, we know if the detection source was FIRT or Cell Phone or Camera, the estimated duration will be longer than the scenario when the accident was detected by the other ways. Third, when the involved vehicles were not moved to shoulder but nobody was injured, if the lane number at main road is less than or equal to 2, the estimated duration is 43.45 minutes, which is shorter than the cases when the accidents happen at main road with lane number greater than 2, for which the duration is 52.49 minutes shown by LM2. This may be because there is lighter traffic for the main road with less lane number. Besides that, the subtree replaced by LM2 shows that when the hour of the day is night (10 PM-6 AM), the accidents need more time to be clear, 61.72 minutes, in contract to 50.03 minutes for the other time intervals.

Similarly, for I-190 accident dataset, through a few runs of testing, $TH1$ is set as 35, and $TH2$ is set as 0.75. Figure 8-4 shows that the M5P tree model for this training dataset.

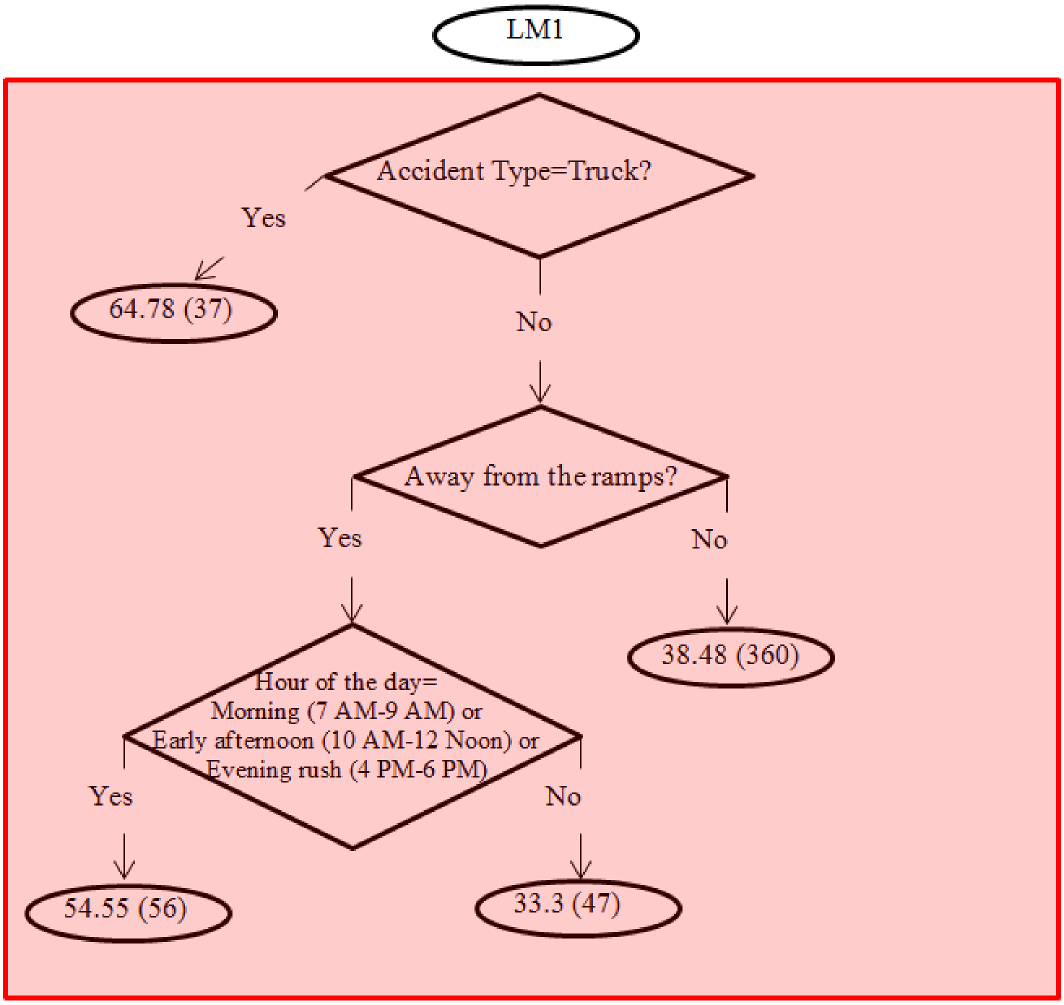


Figure 8-4 M5P Tree Model for I-190 Training Dataset

In Figure 8-4, for the original tree before pruning (red covered), first we can see that if the accident was involved with a truck, the estimated duration is 64.78 minutes, which is the longest. Second, if the accident has nothing to do with a truck, and it is near the ramps, the duration is predicted as 38.48 minutes, while if it happens away from the on-ramps and off-ramps at the highway, and the hour of the day is morning (7 AM-9 AM), early afternoon (10 AM-12 Noon) or evening rush (4 PM-6 PM), the duration is estimated as 54.55 minutes, which is much longer than 33.3 minutes for the other time intervals. This is understandable because more traffic are going through the ramps when it is rush hours like morning, early afternoon and evening rush, so the accident duration is increased a lot.

We can also see that this is an extreme situation when the whole grown M5P tree is replaced by one linear regression model LM1 in the tree pruning step, which is listed as following. The underline part is the independent variables of the linear models, for the value of the independent variable including “?”, the value is 1 if the judgment in the variable name is true, otherwise it is 0.

LM1: Duration=37.95+6.92*Hour of the day= Morning (7 AM-9 AM) or Early afternoon (10 AM-12 Noon) or Evening rush (4 PM-6 PM)? (500 cases);

LM1 shows that the estimated duration of an accident is at least 37.95 minutes, and there is only one independent variable, which is again “hour of the day=morning (7 AM-9 AM) or early afternoon (10 AM-12 Noon) or evening rush (4 PM-6 PM)”. The duration will be increased by 6.92 minutes if it is one of these three time intervals.

In conclusion, although the tree pruning step is conducted when the linear regression model can bring the lower estimated error calculated in Equation (8-2), for both the training datasets, it make the model more hard to be explained. For I-64 dataset, the two linear regression models are both constants. For I-190 dataset, the linear regression model replaces the whole original tree and only has one independent variable. So in this study one drawback of tree pruning for M5P tree is that it’s hard to tell which variable has an impact on the duration.

8.3.2 Hazard-based Duration Model

Before applying HBDM models, there are two problems, first, a distribution form needs be specified for τ in Equation (8-10); second, the significant explanatory variables x_i need be determined. This paper takes the following four steps with the help of STATA software to solve these problems (Alkaabi et al., 2011; Collett, 2003):

1. Fit the models using exponential, weibull, log-normal, log-logistic and generalized gamma models with no explanatory variables. Record the log likelihood for each model.

2. For each model, add the explanatory variable from the candidate variables one by one, test the new model, and select the one which can increase the log likelihood the most by comparing with the original model as the current model.

3. For each model, repeating Step 2 by adding the variable one by one from the rest candidate variables, stop until no variable can increase the log likelihood.

4. For each model, calculate the value of the Akaike information criterion (AIC), which can be calculated as following (Alkaabi et al., 2011; Cleves et al., 2008):

$$AIC = -2lnL + 2(k + c) \qquad (8\text{-}12)$$

where, L is the likelihood;

k is the number of model covariates;

and c is the number of model-specific distributional parameters.

Finally select the model with the lowest value of AIC as the HBDM model.

The AIC values of HBDMs for I-64 and I-190 are listed in Table 8-5, as can be seen, no matter for I-64 or I-190 dataset, the HBDM model with log-normal distribution has the lowest AIC. So it is employed to analyze the accident duration in this paper. This is consistent with the other studies (Golob et al.,1987; Chung, 2010).

Table 8-5 AIC Values of HBDMs for I-64 and I-190 Training Datasets

Model	I-64 dataset				I-190 dataset			
	-2lnL	k	c	AIC	-2lnL	k	c	AIC
Exponential	1169.42	9	1	1179.42	1223.04	2	1	1226.04
Weibull	952.92	9	2	963.92	1105.78	9	2	1116.78
Log-normal	949.08	6	2	957.08	1107.72	4	2	1113.72
Log-logistic	954.62	8	2	964.62	1186.34	9	2	1197.34
Generalized gamma	957.24	5	3	965.24	1185.3	9	3	1197.3

From Equation (8-11), we know for the log-normal regression AFT model, τ is distributed as log-normal with parameters (β_0, σ). To make it more clear, the log-normal AFT function is shown in Equation (8-13) as following (Cleves et al., 2008):

$$ln(d_i) = \beta_0 + x_i\beta + \mu \tag{8-13}$$

where, μ follows a standard normal distribution with mean 0 and standard deviation σ.

For I-64 dataset, Table 8-6 shows the estimated coefficients of explanatory variables, standard error, P-value, and percentage change (%) for the log-normal AFT model of I-64 dataset. According to Equation (8-10), percentage change (%) can be calculated by taking the exponent of the estimated coefficient of the significant variable. For example, the coefficient of variable “night” in Table 8-6 has a positive value 0.19, its exponential value is exp(0.19)=1.21, which can be interpreted as the duration will be

21% longer when it is night; if the coefficient of variable has a negative value, for example, the coefficient of “move to shoulder” in Table 8-6 is -0.36, its exponential value is exp(-0.36)=0.70, this means the duration will be about 30% shorter if the vehicles have been moved to shoulder. In one word, percentage change (%) represents the duration change due to one unit change of the variable (positive means increasing and negative means decreasing).

Table 8-6 Log-normal AFT Models on I-64 Training Dataset

Variable	Coefficient	Standard Error	P value	Percentage Change (%)
Night	0.19	0.07	0.016	21%
Move to shoulder?	-0.36	0.07	0.000	-30%
Road structure	0.26	0.10	0.017	30%
Injured Number	0.22	0.04	0.000	25%
Detection=7 (Virginia State Police Radio)	-0.16	0.08	0.025	-15%
Roll over	0.51	0.25	0.041	67%
β_0	3.41	0.11	0	
σ	0.62	0.02		

As shown in Table 8-6, the variables that can increase the traffic accident duration include the night (10 PM-6 AM), road structure (highway=0, ramp=1), injured number and roll over. If the accidents happen at night, the duration will be increased by 21%; if the accidents happen near ramps, the accident duration will be 30% longer; every time one more person gets injured, the duration will be increased by 25%; and at last if the vehicles in the accidents roll over, this will cause the maximum of increase with 67%. In contrast, moving to shoulder can decrease the accident duration by 30%, and if the detection source of the accident is 7 (Virginia State Police Radio), the duration will have a decrease of 15%.

Similarly, Table 8-7 lists the coefficients of significant variables, and the corresponding standard error, P-value, and percentage change (%) for the log-normal AFT model of I-190 training dataset.

Table 8-7 Log-normal AFT Models on I-190 Training Dataset

Variable	Coefficient	Standard Error	P value	Percentage Change (%)
afternoon	-0.16	0.10	0.007	-15%
Roll Over?	0.83	0.26	0.001	129%
Vehicle number	0.21	0.10	0.050	23%
β_0	3.06	0.20	0	
σ	0.75	0.02		

As can be seen, the only variable with negative percentage change (%) is afternoon (1 PM-3PM). If the accident happens at this time interval, the duration will be 15% shorter. Like the results for I-64 training dataset, the rolling over of the involved vehicles can also largely increase the accident duration by 129%. The duration of the accident will be increased by 23% with the increase of involved vehicle number.

8.3.3 M5P-HBDM Model

With the process described in Chapter 8.1.3, for I-64 training dataset, the M5P-HBDM model is built as shown in Figure 8-5.

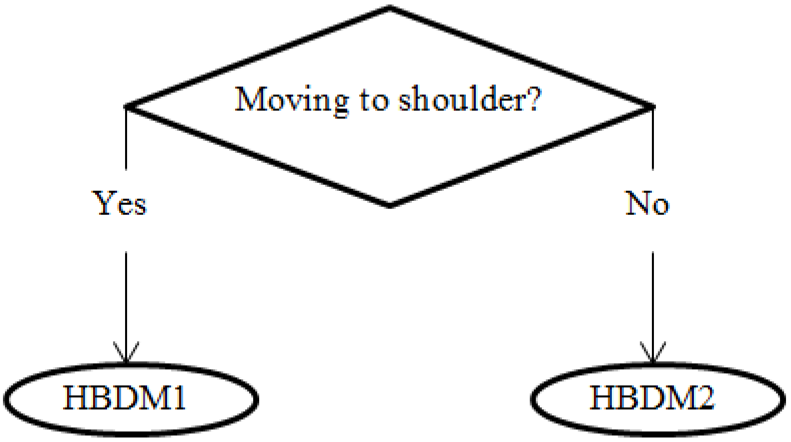


Figure 8-5 M5P-HBDM Model for I-64 Training Dataset

From Figure 8-5, in the M5P-HBDM model of I-64 training dataset, we can find only the splitting rule "moving to shoulder?" at the root node is kept. The HBDM1 replaces the subtree for the accidents when the involved vehicles are moved to the shoulder. And for the accidents when the vehicles are not moved to the shoulder, HBDM2 is built. The AIC test shows the log-normal distribution is still the best assumption for the accelerated failure time functions of HBDM1 and HBDM2. Table 8-8 shows the relevant parameters for the two models.

Table 8-8 Log-normal AFT Models in M5P-HBDM of I-64 Training Dataset

Branches	Variable	Coefficient	Standard Error	P value	Percentage Change (%)
HBDM1 (96 cases)	β_0	3.36	0.08	0	
	σ	0.74	0.05		
HBDM2 (404 cases)	night	0.14	0.07	0.06	15%
	Blocked lane number	0.06	0.04	0.007	6%
	Road structure	0.27	0.10	0.005	31%
	Injured Number	0.18	0.05	0.000	20%
	Detection= 5 (TMS Camera)?	0.06	0.07	0.007	6%
	Detection= 7 (Virginia State Police	-0.13	0.09	0.008	-12%

	Radio)?				
	Roll over?	0.54	0.27	0.05	72%
	Fire or not?	0.11	0.09	0.02	12%
	β_0	3.31	0.12	0	
	σ	0.60	0.02		

For the log-normal AFT model HBDM1 based on 96 cases, in which the involved vehicles are moved to the shoulder, no significant variables are found, only the constant β_0 and the sigma in the log-normal distribution are estimated.

For the HBDM2 based on 404 cases when the involved vehicles are not moved to the shoulder, we can see most of the significant variables are the same as the HBDM based on the whole I-64 dataset. Except that, the variables "Move to shoulder" is missing because it is a splitting rule at the root node in this M5P-HBDM. We can also find that three more variables "blocked lane number", "detection=5" (TMS camera) and "fire or not" are added. This tells that "blocked lane number" becomes a significant variable for cases when the vehicles are not moved to the shoulder, and one more lane being blocked can increase the accident duration by 6%. The detection source "detection=5" (TMS camera) verifies that the accidents detected by camera have a longer duration which has been shown in the M5P tree model before pruning in Figure 8-3. One more observation is that if the vehicle in the traffic accident is on fire, the duration will be increased by 12%.

Similarly, the M5P-HBDM of I-190 dataset is shown in Figure 8-6. Comparing with the M5P tree before pruning in Figure 8-4, HBDM1 replace the interior node "Hour of the day = Morning (7 AM-9 AM) or Early afternoon (10 AM-12 Noon) or Evening rush (4 PM-6 PM)", the leave node with 360 cases is replaced with HBDM2.

For the branch with 37 cases in which the traffic accident is involved with a truck, the leaf node is kept the same as the original M5P tree model after tree growth. This is because no HBDMs are found to bring the lower estimated error than the constant value at the leaf. It's worth noting that in Figure 8-4 this constant value is 64.78 minutes, which is the mean value of the 37 cases, here the 44 minutes is the median value of the 37 cases, which we find is more accurate based on the estimated error in the building process of M5P-HBDM.

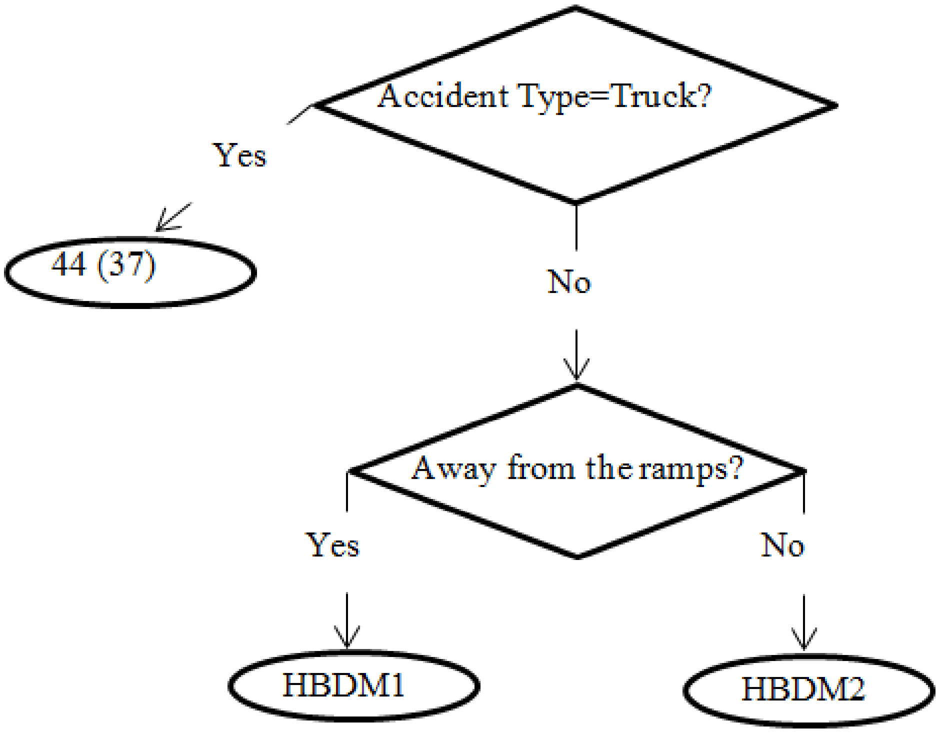


Figure 8-6 M5P-HBDM Model for I-190 Training Dataset

For both HBDM1 and HBDM2, the AIC test still shows they should choose the log-normal distribution as the best assumption for the AFT functions. The relevant parameters of HBDM1 and HBDM2 are shown in Table 8-9.

Table 8-9 Log-normal AFT Models in M5P-HBDM of I-190 Training Dataset

Branches	Variable	Coefficient	Standard Error	P value	Percentage Change (%)
HBDM1 (103 cases)	Evening Rush (4 PM-6 PM)	0.44	0.22	0.05	55%
	β_0	3.38	0.10	0	
	σ	0.87	0.06		
HBDM2 (360 cases)	Morning (7 AM-9 AM)	0.06	0.11	0.02	6%
	Afternoon (1 PM-3 PM)	-0.21	0.11	0.05	-19%
	Vehicle Number	0.32	0.14	0.007	38%
	Location=Exit 16	0.34	0.14	0.019	40%
	Main Road Lane Number=2	-0.90	0.67	0.02	-59%
	Main Road Lane Number=3	-0.96	0.67	0.01	-62%
	β_0	3.72	0.73	0	
	σ	0.67	0.02		

From Table 8-9, we can see that for HBDM1 based on 96 cases when the accidents happen away from the ramps, if it is evening rush (4 PM-6 PM), the accident duration will be increased by 55%. Recall that in the original M5P tree before pruning in Figure 8-4, the splitting rule "Hour of the day = Morning (7 AM-9 AM) or Early afternoon (10 AM-12 Noon) or Evening rush (4 PM-6 PM)?" at the interior node also shows that the accident that happens at these time intervals will experience a higher duration.

Comparing the HBDM2 based on the 360 cases with the HBDM based on the whole I-190 dataset, variable "roll over" is missing. Variable "morning" (7 am-9 am) is added, as well as the other three variables "Main Road Lane Number=2", "Main Road Lane Number=3" and "Location=Exit 16".

First, if accidents happen in the “morning” (7 AM-9 AM), the duration will be increased by 6%. Second, through comparison of the percentage changes caused by “Main Road Lane Number=2” and “Main Road Lane Number=3”, we can find an interesting conclusion: for the non-truck involved accidents that happen close to the ramps, the main road with three lanes can reduce the accident duration by 62%, which is a little larger than 59%, under the scenario when the main road close to the ramps has two lanes. Although the main road with three lanes may mean there are more traffic going through the ramp than the main road with two lanes, it’s also possible that a wider main road can provide more space for the accident clearance work and therefore the duration will be shorter. This need be verified when the traffic volume data are available.

The last thing need be noting here is that if the accident happens at the Exit 16, it can also largely increase the duration by 40%. This may be because Exit 16 is the interchange of I-190 and I-290. Our previous research in Chapter 6.3.2 also shows that Exit 16 is one of the hotspots at I-190, which means the place where the ratio of the number of accidents at that particular spot, to the number of accidents on the whole transportation system under consideration is greater than a user defined threshold.

8.3.4 Significant Variables Comparison

In the process of model building, we can find there are different significant variables found in the M5P, HBDM and M5P-HBDM models. This section will compare these significant variables among different models for both I-64 and I-190 training dataset and analyze how the traffic accident duration can be reduced.

Table 8-10 Significant Variables in M5P, HBDM and M5P-HBDM of I-64 Training Dataset

I-64 Training Dataset	M5P	HBDM	M5P-HBDM
Lane Number at Main Road <=2?	X (-)		
Move to Shoulder?	X (-)	X (-)	R
Injured Number	X (+)	X (+)	X (+)
Road Structure (0 for highway, 1 for ramp)		X (+)	X (+)
Hour of the day=night?		X (+)	X (+)
Roll Over?		X (+)	X (+)
Detection Source=Virginia State Police Radio		X (-)	X (-)
Detection Source= Camera?			X (+)
Blocked lane number at main road			X (+)
Fire or not?			X (+)

Table 8-10 lists all the significant variables of the M5P, HBDM and M5P-HBDM for I-64 training dataset. As can be seen, the significant variables for the corresponding model are marked with “X”, and the sign in the parenthesis represents whether that variable can increase or reduce the accident duration. “R” represents it’s a splitting rule at the M5P-HBDM model.

First we can see that M5P only finds three significant variables, HBDM utilizes six significant variables, while eight significant variables and one splitting rule exist in M5P-HBDM model. Two significant variables “moving to shoulder?” and “injured number” are selected by all the three models.

Second, the variable "hour of the day=night" gets selected by HBDM and M5P-HBDM, and it can increase the accident duration. This tells the clearance work need be strengthened for the traffic accidents at night, maybe more staffs are necessary. Third, besides "moving to shoulder", the detection of traffic accident by Virginia State Police Radio can also reduce the accident duration. This may be because the police can report the accident and make response more quickly, comparing with the TMS camera detection. This is useful when it can be combined with a traffic accident risk prediction model. When the risk is high like a snowy day, more police officers are needed for highway patrol.

Table 8-11 Significant Variables in M5P, HBDM and M5P-HBDM of I-190 Training Dataset

I-190 Training Dataset	M5P	HBDM	M5P-HBDM
Hour of the day= Morning (7 AM-9 AM)	X (+)		X (+)
Hour of the day= Early afternoon (10 AM-12 Noon)	X (+)		
Hour of the day= Evening rush (4 PM-6 PM)	X (+)		X (+)
Hour of the day= afternoon (1 PM-3 PM)		X (-)	X (-)
Vehicle Number		X (+)	X (+)
Roll Over?		X (+)	X (+)
Location=Exit 16			X (+)
Lane Number at Main Road =2			X (-)
Lane Number at Main Road =3			X (-)
Accident Type=Truck?			R
Away from the ramps?			R

Similarly, the significant variables in M5P, HBDM and M5P-HBDM of I-190 training dataset are summarized in Table 8-11. For I-190 training dataset, first M5P-HBDM selects eight significant variables and two splitting rules, following that, HBDM and M5P both have three significant variables. There are no common significant variables in these three models. Second, for the variables relevant with hour of the day, during the peak commute hours (7 AM-9 AM and 4 PM-6 PM) and lunch break hours (10 AM-12 Noon), the accident duration will be longer, while at the non-peak hours (1 PM-3 PM), the duration will be shorter.

So obviously, for both I-64 and I-190 training datasets, the number of significant variables in M5P-HBDM model is the maximum. Recall that in tree growth step, the splitting rule at the nodes of M5P tree is to find the variable that can bring the maximum reduction of the standard deviation of the target value (accident duration), in this way the data heterogeneity is alleviated, and more unobserved factors that impact the traffic accident duration can be captured.

8.3.5 Duration Prediction Comparison

As mentioned before, I-64 dataset has a testing dataset including 102 records, and I-190 has a testing dataset including 116 records. This section uses the previous built M5P, HBDM and M5P-HBDM models to make predictions for the two testing datasets. For the prediction performance evaluation, the Mean Absolute Percentage Error (MAPE) is a widely used measure to assess the accuracy of the models developed. MAPE can be calculated as follows:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n} | \frac{A_i - P_i}{A_i} | \qquad (8\text{-}14)$$

where, A_i is the i^{th} actual value, P_i is the i^{th} predicted value.

To calculate the predictions, for M5P tree model, each testing record will be directed into the corresponding leave and the linear functions or the mean target values at that leave will be used to estimate the accident duration. For HBDMs, the mean and the median values of the survival time (accident duration) for the log-normal AFT models can be calculated as the predictions as Equation (8-15) and (8-16) after the relevant parameters and variable coefficients in the log-normal AFT models are estimated.

$$\text{median}(d_i) = \exp(\beta_0 + x_i\beta) \tag{8-15}$$

$$\text{mean}(d_i) = \exp(\beta_0 + x_i\beta + \sigma^2/2) \tag{8-16}$$

For M5P-HBDMs, similarly as M5P, the testing record is directed into the corresponding leave. If there is no log-normal AFT model at the leave, as mentioned earlier in Chapter 8.3.3, we use the median value of the cases at that node as the prediction instead of using the mean value as M5P tree does. Otherwise, if there is a log-normal AFT model at the corresponding leave, the predictions can be gotten by using the same way in HBDMs. Note that the mean and median values can vary for different testing records when they have different values for the significant variables in the AFT model, however, if there are no significant variables found in AFT model, for example, HBDM1 in M5P-HBDM for I-64 dataset, the mean and the median values for these testing records will be the same.

It’s worth mentioning that in this study experiments show that the median values of the survival time always have better performances as the predictions than the mean values for both HBDMs and M5P-HBDMs. So the median values are used to calculate the MAPEs for HBDMs and M5P-HBDMs.

Finally, the MAPEs of M5P tree model, HBDM model and the M5P-HBDM model for the two testing datasets can be calculated in the following Table 8-12.

Table 8-12 MAPEs of M5P Tree, HBDM Model and M5P-HBDM Model for I-64 and I-190 Testing Datasets

Datasets	M5P	HBDM	M5P-HBDM
I-64	48.69%	38.32%	36.20%
I-190	38.45%	33.61%	31.87%

It can be seen that for I-64 testing dataset, the lowest MAPE is 36.20% given by M5P-HBDM, HBDM is the second best model to predict the traffic accident duration with a MAPE of 38.32%, while the MAPE of M5P is as high as 48.69%. For I-190 testing dataset, M5P-HBDM still has the best prediction performance with MAPE equal to 31.87%, which is followed by HBDM and M5P. We can also see that no matter for the I-64 or the I-190 testing dataset, the M5P-HBDM model performs the best, the MAPE of HBDM is about 2% higher than M5P-HBDM, while M5P performs the worst with an obvious highest MAPE.

Considering that the M5P tree model from the study (Zhan et al., 2011) has a MAPE of 42.70%, and the traffic accident duration prediction using HBDM from literature (Chung, 2010) has a MAPE of 47%, our results are comparable with these previous researches.

It is also interesting to know the performances of the three models for different levels of accident duration times. This paper divided the durations into four different levels: 0 minutes-20 minutes, 21 minutes-40 minutes, 41 minutes-60 minutes and more than 60 minutes. The proportions of each level and the MAPEs of each model for I-64 and I-190 testing datasets are shown in Table 8-13. First, for I-64 testing datasets, we can see M5P-HBDM has the lowest MAPE for 0 minute - 20 minutes, although this level only occupies a small portion of the whole test dataset (3.5%); for 21 minutes-40 minutes level, which occupies 46.5% of the testing dataset, M5P-HBDM still performs better than M5P and HBDM; for 41 minutes-60 minutes level and more than 60 minutes level, M5P tree performs the best with MAPE equal to 11.28% and 40.20% separately. Second, for the I-190 testing dataset, for the lower levels 0 minute - 20 minutes and 21 minutes - 40 minutes, HBDM performs the best with MAPE as 78.72% and 14.64% correspondingly. For the higher levels 41 minutes - 60 minutes and more than 60 minutes, M5P again performs the best. M5P-HBDM always performs the second best for the four levels in this testing dataset. Third, for both testing datasets, all these three models have high MAPEs for the levels 0 minute - 20 minutes and more than 60 minutes, and they have relatively better performances for the middle levels 21 minutes - 40 minutes and 41 minutes - 60 minutes.

Table 8-13 Performances of M5P, HBDM and M5P-HBDM for Different Actual Duration Intervals of I-64 and I-190 Testing Datasets

Datasets	Actual Duration levels	Proportion	M5P	HBDM	M5P-HBDM
I-64	0 min-20 min	3.5%	166.35%	98.12%	89.48%
	21 min-40 min	46.5%	59.84%	36.97%	32.43%
	41 min-60 min	17.4%	11.28%	16.25%	18.63%
	More than 60 min	32.6%	40.20%	45.67%	45.58%
I-190	0 min-20 min	7.5%	135.99%	78.72%	91.24%
	21 min-40 min	39.2%	36.73%	14.64%	23.33%
	41 min-60 min	34.6%	14.25%	31.30%	20.00%
	More than 60 min	18.7%	47.83%	59.69%	48.03%

In order to further compare the three models, another measure of prediction performance used in literature (Zhan et al., 2011; Chung, 2010; Smith and Smith, 2001) was also compared in this research. The percentages of predictions that are within a certain tolerance of actual duration times are calculated for each model and each dataset. In this study, five different tolerances are tested: 10 minutes, 20 minutes, 30 minutes, 40 minutes and 50 minutes. The mean absolute difference between the predictions and the actual values are also listed. All the results are shown in Table 8-14.

Table 8-14 Percentage of Predictions Having a Difference within a Certain Tolerance and Mean Absolute Differences of M5P Tree, HBDM Model and M5P-HBDM Model for I-64 and I-190 Testing Datasets

Datasets	Absolute Difference	M5P	HBDM	M5P-HBDM
I-64	<=10 min	27.91%	39.53%	40.70%
	<=20 min	55.81%	66.28%	68.60%
	<=30 min	72.09%	76.74%	74.42%
	<=40 min	89.53%	82.56%	83.72%
	<=50 min	93.02%	90.70%	89.53%
	Mean (min)	24.97	23.08	22.68
I-190	<=10 min	43.93%	44.86%	49.53%
	<=20 min	71.96%	73.83%	79.44%
	<=30 min	87.85%	82.24%	88.78%
	<=40 min	91.59%	87.85%	88.78%
	<=50 min	92.52%	91.56%	94.39%
	Mean (min)	17.24	18.73	16.23

Table 8-14 shows that for I-64 testing dataset, 40.70% of predictions of M5P-HBDM model have the absolute difference less than or equal to 10 minutes, and 68.60% of predictions of M5P-HBDM model have the absolute difference less than or equal to 20 minutes, which are both the highest in the corresponding group comparing with M5P and HBDM. With the tolerance threshold equal to 50 minutes, among the three models, the absolute difference of 93.02% of M5P tree predictions is within the tolerance, which is the highest. Besides that, M5P-HBDM model has the lowest mean absolute difference which is 22.68 minutes, followed by HBDM and M5P.

For I-190 testing dataset, the absolute difference of almost half of predictions (49.53%) made by M5P-HBDM is less than or equal to 10 minutes, which is the highest. The percentages of predictions of the three models that have an absolute difference within 20 minutes are all above 70%, but M5P-HBDM again has the highest percentage of 79.44%. Even when the tolerance is set as 50 minutes, M5P-HBDM still has the maximum number of predictions which have the absolute difference within 50 minutes with a percentage of 94.39%. For the mean absolute difference, the lowest one is 16.23 minutes maintained by M5P-HBDM model, M5P has the second lowest mean absolute difference, while HBDM has the largest mean absolute difference.

8.4 Conclusions

In this traffic accident duration prediction study, a new algorithm was designed to build a M5P-HBDM on the basis of the traditional M5P tree algorithm. The leaves of the M5P-HBDM can be an AFT model, which is different as the linear regression model in the M5P tree model. For the two traffic accident duration datasets in this study, I-64, Virginia and I-190, Buffalo, each of them was split into a training dataset and a testing dataset. The M5P-HBDM model, M5P tree and HBDM were built on each training dataset, and the significant variables in these models were compared and analyzed. Then for each testing dataset, the three models were used to predict the traffic accident duration. The prediction performances of the three models were compared through the following three aspects: the MAPEs of the predictions, the MAPEs for different duration levels, and the percentages of predictions with absolute difference within different tolerances. A few main conclusions are summarized as following:

(1) In tree growth step of M5P, for each interior node, all the possible splitting rules are tested and the one that can maximize the reduction of the standard deviation is selected. M5P-HBDM keeps this ability which can reduce the data heterogeneity through the splitting rules at the nodes. Through the comparison of significant variables found by the three models for each training dataset, M5P-HBDM model can reveal more factors that may affect accident durations than the M5P and HBDM model. This is also consistent with our previous research that through dataset grouping and clustering the data heterogeneity can be reduced.

(2) M5P-HBDM can build an AFT model as its leave. The AFT model does not assume the accident durations follow the normal distribution as the linear regression model in M5P model assumes. The Weibull distribution, log-normal distribution, log-logistics distribution and so on can all be tested. In this study, we found the log-normal AFT model is the best choice based on the AIC values.

(3) The comparison of the prediction performances of the three models shows that, no matter for the I-64 or I-190 testing dataset, the M5P-HBDM always has the lowest overall MAPE. M5P-HBDM also has a promising performance for different duration levels. The percentage of predictions with absolute different

less than or equal to 20 minutes given by M5P-HBDM is also the highest for each testing dataset.

CHAPTER 9 DISSERTATION CONTRIBUTION AND FUTURE RESEARCH

This doctoral research proposes an integrative approach for the emerging field data science applications in intelligent transportation systems (ITS) data analysis. The integrative approach can be conducted from the aspect of ITS data analysis process, such as the integration of data “width” decreasing and model development. It can also be conducted from the aspect of ITS modeling, combining various modeling techniques such as statistical models, data mining, machine learning, analytical models, numerical models and so on to solve one problem. With the rapid development of data science and the idea of integrative approach, novel models and processes can be developed for advanced and more efficient ITS data analysis.

The integrative approach is applied and tested on two ITS data analysis case studies to improve the efficiency and safety of transportation systems, namely the border crossing delay prediction and the traffic accident data analysis. The first case study includes two subtopics of short-term traffic volume prediction and multi-server queueing model, and the second case study includes three subtopics of traffic accident hotspots analysis, real-time traffic accident risk prediction and traffic accident duration prediction.

The border crossing delay prediction case study is described in Chapter 3 through 5, consisting of the discussions about a short-term traffic volume prediction model, a queueing model, and a border crossing android smartphone application. Chapter 3 has documented all the work relevant to short-term traffic volume prediction, such as the integration of a dataset grouping method with the model development process. The combined SARIMA and SVR model was applied to predict the border crossing traffic for Peace Bridge, Buffalo, New York. Besides this, a novel model called DTW-SPN was also proposed for border crossing traffic prediction. Chapter 3 also suggested the ITS data analysis step called the data diagnosis to be integrated with the model development process. Based on the statistical measures of multiple datasets, the performances of different short-term traffic volume prediction models are compared. Chapter 4 discussed the heuristic transient solutions for two types of multi-sever queueing models, namely the $M/E_k/n$ model and the *BMAP/PH/n* model. The $M/E_k/n$ queueing model is developed based on the real observations at the Peace Bridge border crossing, and the *BMAP/PH/n* is a more general model that can be applied to other border crossings, as well as other similar situations such as toll collection stations. The results of the analytical queueing model $M/E_k/n$ were compared with the results of VISSIM traffic simulation model for Peace Bridge, and sensitivity analysis were performed and incorporated to derive optimal management strategies for the U.S. customs and border protection agencies. Chapter 5 introduced an android smartphone app that was designed to collect, share and predict waiting times for the Niagara Frontier border crossings. It offers the users three types of waiting times, namely the current waiting times collected at the crossings, the historical waiting times, and the future waiting times predicted for the next 15 minutes and updated every five minutes.

Chapter 6 through 8 are relevant to the second case study, i.e., traffic accident data analysis. In Chapter 6, a community detection algorithm in graph and a network analysis technique called the modularity optimization method were applied to cluster the traffic accident dataset. The association rules learned from each data cluster provided interesting insight about the characteristics of accidents, such as where they tend to occur and the influential factors of the incident clearance time. Chapter 7 presented a frequent pattern (FP) tree based new variable selection method and a new variable importance score called Relative Object Purity Ratio (ROPR) for real-time traffic accident risk prediction. The k-NN and Bayesian network models were developed and tested for three data scenarios such as using all variables, using the important variables selected by FP tree, and using the important variables selected by random forest. The results of experiments show the effectiveness of the FP tree based variable selection method. Chapter 8 proposed a new combined model M5P-HBDM to predict the traffic accident durations. Instead of pruning the tree with linear regression models after tree growth in the original M5P tree algorithm, M5P-HBDM prunes the tree with HBDMs. The main advantages of the combined M5P-HBDM model are that M5P decreases the data heterogeneity by using the splitting conditions at the nodes and that HBDM does not require the accident durations to be normally distributed as assumed by linear regression models. Due to the two advantages, M5P-HBDM can identify more significant variables than the M5P tree method or HBDMs, and it outperforms both M5P and HBDM in traffic accident duration prediction.

The main contributions of this dissertation are summarized from two aspects: ITS data analysis process and ITS data analysis models. From the aspect of the ITS data analysis process shown in Chapter 1.3.1:

(1) The integration of data "width" decreasing with model development.

In Chapter 7, a novel variable selection algorithm based on FP tree for real-time traffic accident risk prediction was proposed. The importance score of each explanatory variable in the dataset is calculated and ranked through the calculation of the Relative Object Purity Ratio (ROPR) for the corresponding frequent patterns. The accident risk prediction models based on FP tree variable selection outperform the models based on all variables and the ones based on random forest, regardless of the settings of the prediction models such as the selection of k for k-NN, the NED number selected for Bayesian network, and the pre-crash time period used in the datasets. Being insensitive to the selection of the models' parameters is a good quality that the FP tree variable selection algorithm appears to possess.

(2) The integration of data "depth" decreasing with model development.

a. Grouping data and developing a separate prediction model for each data group prove to significantly improve the accuracy of border crossing short-term volume forecasts as discussed in Chapter 3.1. In the border crossing traffic volume prediction study, a convenient classification scheme involved dividing observation days into the following six groups: weekdays excluding Fridays, Fridays, Saturdays, Sundays, game days, and holidays.

b. From Chapter 6, for traffic accident dataset clustering, the community detection algorithm appears to do an excellent job in clustering the traffic accident data into well-defined clusters to decrease the data heterogeneity. Clustering the data first before running the association rule learning algorithm appears to be an important step to significantly improve the quality of the insights to be gained from the rules extracted.

(3) *Integration of Data Diagnosis and Short-term Traffic Volume Model Development*

The data diagnosis is to assess the overall predictability of a dataset through learning the complexity, nonlinearity and long range dependency of the data. As shown in Chapter 3.2, by correlating the performances of the different prediction methods to the data diagnosis measures on four different short-term traffic datasets, some guidelines can be obtained on how to choose the appropriate prediction method from the optional methods such as SARIMA, k-NN and SVR, and on how to efficiently set the parameters of the selected method.

(4) *Model Combination of SARIMA and SVR for Short-term Traffic Volume Prediction*

As found, combining SARIMA and SVR using the Fuzzy Adaptive Variable Weight method can significantly improve the performance of both SARIMA and SVR models for the border crossing traffic volume prediction problem as discussed in Chapter 3.1.

(5) *Model Application - an Android Smartphone Application for Niagara Frontier Border Crossing*

Chapter 5 introduces the Toronto Buffalo Border Waiting android smartphone app that is designed to collect, share and predict waiting time at the three Niagara Frontier border crossings. This app applies the data-level fusion to collect the current waiting times from both the official sources and the users through crowd sourcing. This app also relies on the two step border crossing delay prediction model to estimate the future waiting times. It has been downloaded close to 500 times, and is available in Google Play Store as following link https://play.google.com/store/apps/details?id=toronto.buffalo.borderwaiting&hl=en.

From the aspect of ITS data analysis Models, the main contributions of this dissertation include:

(1) *Integration of Dynamic Time Warping (DTW) and Spinning Network (SPN) for Short-term Traffic Volume Prediction.*

In Chapter 3.3, an enhanced SPN was developed to predict hourly traffic volumes at the Peace Bridge international border crossing. The enhancement is at using the DTW method to assess similarity among traffic volume data, rather than using the typical distance-based similarity index. The performance of the enhanced SPN (i.e. DTW-SPN) was then compared to three other forecasting methods, namely the original SPN algorithm (Euclidean-SPN), SARIMA, and SVR. DTW-SPN had the overall best performance when the MAPE was averaged over all the day groups. DTW-SPN also had the lowest MAPE when the whole dataset was used (i.e. the data was not broken into groups). DTW-SPN also appears to be significantly more computationally efficient in comparison with SARIMA or SVR.

(2) Integration of Data-driven Models and Analytical Methods for Border Delay Prediction.

This integration is the foundation of the two-step border crossing delay prediction model. It is also one of the most important contributions in this dissertation. For the first step, the short-term traffic volume prediction model, intensive work can be found in Chapter 3. For the second step, the multi-server queueing model, Chapter 4 gives the detailed solution. The idea of the model integration is not limited to border crossing studies. It can also be applied to other similar situations that deal with dynamic demand and queueing systems, such as like toll stations, intersections, manufacturing processes, product distribution systems, call centers, and so on.

(3) Integration of M5P tree and HBDM for Traffic Accident Duration Prediction

The research discussed in Chapter 7 improved the original M5P tree algorithm by building a combined M5P-HBDM model. Through which, the leaves of M5P tree model can be HBDMs instead of the linear regression models. This new M5P-HBDM model reduces the data heterogeneity and does not rely on the assumption of the normal distribution for traffic accident durations as the linear regression in a M5P tree does. The M5P-HBDM was found to be able to identify more significant variables and perform better than M5P tree and HBDM, based on the prediction results of two traffic accident duration datasets.

From Chapter 3 to Chapter 8, there are several future research directions which are suggested by this dissertation work.

(1) Short-term Traffic Volume Prediction

a. Border Crossing Traffic Prediction

We can investigate the benefits of incorporating traffic volumes from upstream links into the prediction process to improve accuracy and allow for extending the prediction horizon. We also need test the transferability of the approach to other border crossings. Another possible future research area involves dynamically estimating confidence bounds for the volume forecasts, which gives a measure of the reliability of the forecast.

b. Enhanced SPN

First, conduct additional testing of both the original SPN and the enhanced DTW-SPN to other data sets to further confirm the prediction accuracy and advantages of the method. Second, because the SPN models require the specification of a number of parameters such as: 1) the data item length; 2) the number of rings and ring sizes; 3) the sizes of the input, output, and TNR windows; 4) the spinning speed; 5) the threshold to next ring, and 6) the distance tolerance, guidance is needed on how to best set these parameters. The current study seems to indicate that performance is most sensitive to the data item length and the distance tolerance. However, additional research is needed in order to confirm that.

c. Data Diagnosis for Model Selection and Parameter Settings

Further testing on additional traffic volume datasets are necessary to ensure that the conclusions from this study may be generalized; this may include traffic volume datasets from arterials. Another suggested future research direction is to use the insights gained from this study to develop a decision support tool which could aid the analyst in selecting the appropriate modeling paradigm for a given data set and in setting the values of the model's parameters.

(2) Transient Multi Server Queueing Models

The accuracy of the border crossing delay, predicted by the queueing models formulated herein, is naturally dependent upon the ability to predict: (1) the future traffic volume; and (2) the number of inspection stations open. It would be quite interesting to study how robust the overall prediction system is to faulty assumptions regarding those two variables (i.e. predicted traffic volume and the number of inspection stations). Moreover, for the use of the models for online prediction (as a part of a real-time traveler information systems), future research should consider how to update the models' predictions in real-time based upon real-world observations (i.e. measurements) of delay via technologies such as blue-tooth readers at border crossings.

(3) Smartphone App for Border Crossing

First, at the moment, the TBBW app is only predicting the delay for the next 15 minutes, it would be better to make the prediction horizon a user-specified value (e.g., some users may be interested in the future delay for the next 30 minutes or one hour, if their trip origin is farther away from the border).   Second, although the app is currently designed for the Niagara International Frontier Borders, it can also be easily extended and applied to other US-Canadian borders as well as to the borders between the US and Mexico, provided that similar data are available. The app can even be extended to predict airport delay, as well as delay at many other similar queueing systems, in the future.

(4) Community Detection for Traffic Accident Hotspots and Clearance Time Analysis

For future research, the authors plan to test the community detection and association rule learning algorithms on larger and richer data sets, and to explore additional relationships between causative factors and accident attributes.

(5) Variable Selection based on FP-tree

As a novel algorithm, there are still a lot of details to be finalized in the future. For example, we may test the impact of clustering number in FCM on the FP tree variable importance calculation (in this study, we just set it as 3), and we may also try other variable discretization methods. Besides that, there are some other variable reduction/selection algorithms, such as stratified random forest (Ye, et al., 2013), and random projection (Fan, et al., 2013) that deserve to be explored. We will also test other accident risk prediction methods such as support vector machine (SVM) as our future work.

(6) M5P-HBDM for traffic accident duration prediction

For the future study, we may test the combination of M5P and random parameter HBDM which may improve the accident duration prediction. For example, in the M5P-HBDM for I-190 dataset, when the non-truck accidents happen near the ramps, the situation that the lane number at main road is three can reduce the traffic accident duration more than the situation when the lane number at main road is two. As we have analyzed, this may be true considering the three-lane main road can provide more space for the clearance work than the two-lane main road, but the three-lane main road main road may also have more traffic. In this case, we think the random parameter HBDM can capture this insight by allowing the coefficients of the variables in the model to vary across each individual observation in the dataset.

REFERENCE

Abdelwahab, H. T., & Abdel-Aty, M. A. (2002). Artificial neural networks and logit models for traffic safety analysis of toll plazas. Transportation Research Record: Journal of the Transportation Research Board, 1784(1), 115-125.

Abdel-Aty, M., Pande, A., Das, A., & Knibbe, W. J. (2008). Assessing safety on Dutch freeways with data from infrastructure-based intelligent transportation systems. Transportation Research Record: Journal of the Transportation Research Board 2083(1), 153-161.

Abdel-Aty, M., Uddin, N., Pande, A., Abdalla, F. M., & Hsia L. (2004). Predicting freeway crashes from loop detector data by matched case-control logistic regression. Transportation Research Record: Journal of the Transportation Research Board 1897(1), 88-95.

Abdel-Aty, M., & Yu, R. (2014). Integrating Safety in Developing a Variable Speed Limits System. Report submitted to National Center for Transportation System Productivity and Management.

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. ACM SIGMOD Record, Vol. 22, No. 2, pp. 207-216.

Ahmed, M. M., & Abdel-Aty, M. (2012). The viability of using automatic vehicle identification data for real-time crash prediction. Intelligent Transportation Systems, IEEE Transactions on 13(2), 459-468.

Ahmed, M. S., & Cook, A. R. (1979). Analysis of freeway traffic time-series data by using Box-Jenkins techniques. Transportation Research Record: Journal of the Transportation Research Board 722, 1-9.

Alkaabi, A. M. S., Dissanayake, D., Bird, R. (2011). Analyzing clearance time of urban traffic accidents in Abu Dhabi, United Arab Emirates, with hazard-based duration modeling method. Transportation Research Record: Journal of the Transportation Research Board, 2229(1), 46-54.

Anastasopoulos, P. C., and Mannering, F. L. (2009). A note on modeling vehicle accident frequencies with random-parameters count models. Accident Analysis & Prevention, Vol. 41, No. 1, pp. 153-159.

Anderson, T. K. (2009). Kernel density estimation and K-means clustering to profile road accident hotspots. Accident Analysis & Prevention, Vol. 41, No. 3, pp. 359-36.

Archer, K. J., & Kimes, R. V. (2008). Empirical characterization of random forest variable importance measures. Computational Statistics & Data Analysis 52(4), 2249-2260.

Arenas, A., Duch, J., Fernández, A., and Gómez, S. (2007). Size reduction of complex networks preserving modularity. New Journal of Physics, Vol. 9, No. 6, pp. 176-190.

Asmussen, S., Nerman, O., & Olsson, M. (1996). Fitting phase-type distributions via the EM algorithm. Scandinavian Journal of Statistics 1996, 419-441.

Ausín, M.C., Lillo, R.E., Wiper, M.P. (2007). Bayesian control of the number of servers in a GI/M/c queueing system. J. Stat. Plann. Inference 137 (10), 3043–3057.

Ausín, M.C., Wiper, M.P., Lillo, R.E. (2008). Bayesian prediction of the transient behaviour and busy period in short-and long-tailed GI/G/1 queueing systems. Comput. Stat. Data Anal. 52 (3), 1615–1635.

Barbulescu, A., Serban, C., and Maftei, C. (2010). Evaluation of Hurst Exponent for precipitation time series. Proceedings of the 14th WSEAS international conference on Computers: 590-595.

Bastian M., Heymann, S., and Jacomy, M. (2014). Gephi: an open source software for exploring and manipulating networks. www.aaai.org/ocs/index.php/ICWSM/09/paper/viewFile/154Forum/ 1009.

Bayesia, S. A. S. (2013). BayesiaLab 5.1. The technology of Bayesian networks at your service. http://www.bayesia.com/en/products/bayesialab.php

Bíl, M., Andrášik, R., and Janoška, Z. (2013). Identification of hazardous road locations of traffic accidents by means of kernel density estimation and cluster significance evaluation. Accident Analysis & Prevention, Vol. 55, pp. 265-273.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, Vo. 10, P10008.

Bobbio, A., Horváth, A., & Telek, M. (2005). Matching three moments with minimal acyclic phase type distributions. Stochastic models 21(2-3), 303-326.

Box, G.E.P., and Jenkins, G.M. (2008). Time Series Analysis: Forecasting and Control (4th edition). Wiley.

Breuer, L. (2002). An EM algorithm for batch Markovian arrival processes and its comparison to a simpler estimation procedure. Annals of Operations Research 112(1-4), 123-138.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees. CRC press.

Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.

Breiman, L. (2001). Random forests. Machine learning 45(1), 5-32.

Brockfeld, E., Kühne, R. D., & Wagner, P. (2004). Calibration and validation of microscopic traffic flow models. Transportation Research Record: Journal of the Transportation Research Board, 1876(1), 62-70.

Buffalo and Fort Erie Public Bridge Authority. (2014). Daily Volumes. Retrieved October 4, 2010, from http://www.peacebridge.com

Casale, G., Zhang, E. Z., & Smirni, E. (2008). KPC-toolbox: Simple yet effective trace fitting using markovian arrival processes. In Quantitative Evaluation of Systems Fifth International Conference on, 83-92.

Castro-Neto, M., Jeong, Y.S., Jeong, M.K., Han, L.D. (2009). Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Exp.Syst. Appl. 36 (3), 6164–6173.

Chang, G.L., Su, C.C. (1995). Predicting intersection queue with neural network models. Transp. Res. Part C: Emerg. Technol. 3 (3), 175–191.

Chang, C-C., and Lin, C-J. (2001). LIBSVM: a Library for Support Vector Machines. http://www. csie. ntu. edu. tw/cjlin/libsvm.

Chen, H., Grant-Muller, S. (2001). Use of sequential learning for short-term traffic flow forecasting. Transp. Res. Part C: Emerg. Technol. 9 (5), 319–336.

Chen, K-Y, and Wang, C-H. (2007). A hybrid SARIMA and support vector machines in forecasting the production values of the machine industry in Taiwan. Expert Systems with Applications 32, pp. 254-264.

Chen, S., Wang, W., & van Zuylen, H. (2010). A comparison of outlier detection algorithms for ITS data. Expert Systems with Applications, 37(2), 1169-1178.

Chen, W. H., and Jovanis, P. P. (2000). Method for identifying factors contributing to driver-injury severity in traffic crashes. In Transportation Research Record: Journal of the Transportation Research Board, No. 1717, pp. 1-9.

Chen, Y., Yang, B., Dong, J. (2006). Time-series prediction using a local linear wavelet neural network. Neurocomputing 69 (4), 449–465.

Cheng, Y. H., & Li, Y. S. (2010). Influences of traffic emissions and meteorological conditions on ambient PM10 and PM2. 5 levels at a highway toll station. Aerosol Air Qual. Res, 10, 456-462.

Chin, S. M., Franzese, O., Greene, D. L., Hwang, H. L., & Gibson, R. C. (2004). Temporary losses of highway capacity and impacts on performance: Phase 2. United States. Department of Energy.

Chrobok, R., O. Kaumann, J. Wahle, and M. Schreckenberg. (2004). Different Methods of Traffic Forecast based on Real Data. European Journal of Operational Research 155, pp. 558-568.

Chung, E. (2004). Classification of Traffic Pattern. Proc. of the 11th World Congress on ITS.

Chung, Y. (2010). Development of an accident duration prediction model on the Korean Freeway Systems. Accident Analysis & Prevention, 42(1), 282-289.

Clemen, R. T. (1989). Combining forecasts: A review and annotated bibliography. International journal of forecasting, 5(4), 559-583.

Cleves, M. A., Gould, W. W., Gutieerez, R. G. (2008). An introduction to survival analysis using Stata. Stata Press.

Collett, D. (2003). Modelling Survival Data in Medical Research, Vol. 57. CRC press.

Cools, M., Moons, E., and Wets, G. (2009). Investigating the Variability in Daily Traffic Counts through Use of ARIMAX and SARIMAX Models. Transportation Research Record 2136, Transportation Research Board, Washington, DC, pp. 57-66.

Cordeiro, J. D., & Kharoufeh, J. P., (2010). Batch Markovian Arrival Processes (BMAP). Wiley Encyclopedia of Operations Research and Management Science.

Czachórski, T., Fourneau, J.M., Nycz, T., Pekergin, F. (2009). Diffusion approximation model of multiserver stations with losses. Electron. Notes Theor. Comput. Sci. 232, 125–143.

Daikoku, K., Masuyajna, H., Takine, T., & Takahashi, Y. (2007). Algorithmic Computation of the Transient Queue Length Distribution in the BMAP/D/c Queue. Journal of the Operations Research Society of Japan-Keiei Kagaku 50(1), 55-72.

de Oña, J., López, G., Mujalli, R., and Calvo, F. J.(2013). Analysis of traffic accidents on rural highways using Latent Class Clustering and Bayesian Networks. Accident Analysis & Prevention Vol. 51, pp. 1-10.

Depaire, B., Wets, G., and Vanhoof, K. (2008). Traffic accident segmentation by means of latent class clustering. Accident Analysis & Prevention, Vol. 40, No. 4, pp. 1257-1266.

Dimitriou, L., Tsekeris, T., Stathopoulos, A. (2008). Adaptive hybrid fuzzy rule-based system approach for modeling and predicting urban traffic flow. Transp. Res. Part C: Emerg. Technol. 16 (5), 554–573.

Disks, C., van Houwelingen, J. C., Takens, F., and DeGoede, J. (1995). Reversibility as a criterion for discriminating time series. Physical Letter A 201: 221-228.

Dlagnekov, L. (2004). License plate detection using adaboost. Computer Science and Engineering Department, San Diego.

DOMO. (2014). Data Never Sleeps 2.0. Retrieved December 5, 2014, from http://www.domo.com/blog/2014/04/data-never-sleeps-2-0/

Dong, C., Richards, S. H., Yang, Q., & Shao, C. (2014). Combining the Statistical Model and Heuristic Model to Predict Flow Rate. Journal of Transportation Engineering.

Douglas, L. (2012). The Importance of 'Big Data': A Definition. Gartner (June 2012).

Escobar, M., Odoni, A.R., Roth, E. (2002). Approximate solution for multi-server queueing systems with Erlangian service times. Comput. Oper. Res. 29 (10), 1353–1374.

Fan, J., Han, F., & Liu, H. (2013). Challenges of Big Data Analysis. arXiv preprint arXiv:1308.1479.

Faouzi, N. E. E., Leung, H., & Kurian, A. (2011). Data fusion in intelligent transportation systems: Progress and challenges–A survey. Information Fusion, 12(1), 4-10.

Farradyne, P. B. (2000). Traffic incident management handbook. Prepared for Federal Highway Administration, Office of Travel Management.

Federal Highway Administration (FHWA). How Do Weather Events Impact Roads? Retrieved July 16, 2014, from http://www.ops.fhwa.dot.gov/weather/q1_roadimpact.htm

Fernández, A., Gómez, Á., Lecumberry, F., Pardo, Á., & Ramírez, I. (2014). Pattern Recognition in Latin America in the "Big Data" Era. Pattern Recognition.

Ferrara, E. (2012). A large-scale community structure analysis in Facebook. EPJ Data Science, Vol. 1, No. 1, pp. 1-30.

Fortunato, S. (2010). Community detection in graphs. Physics Reports, Vol. 486, No. 3, pp. 75-174.

Fraser, A.M., and Swinney, H.L. (1986). Independent coordinates for strange attractors from mutual information. Physical Review A 33(2): 1134-1140.

Freitas, P. S., & Rodrigues, A. J. (2006). Model combination in neural-based forecasting. European Journal of Operational Research, 173(3), 801-814.

García-Laencina, P. J., Sancho-Gómez, J. L., & Figueiras-Vidal, A. R. (2010). Pattern classification with missing data: a review. Neural Computing and Applications, 19(2), 263-282.

Garib, A., Radwan, A. E., and Al-Deek, H.(1997). Estimating magnitude and duration of incident delays. Journal of Transportation Engineering 123(6), 459-466.

Geneva Artificial Intelligence Laboratory. Model Combination. Retrieved September 25, 2014, from http://cui.unige.ch/AI-group/teaching/dmc/09-10/cours/dm11-ensembles.pdf

Geurts, K., Wets, G., Brijs, T., and Vanhoof, K.(2003). Profiling of high-frequency accident locations by use of association rules. In Transportation Research Record: Journal of the Transportation Research Board, No. 1840, pp. 123-130.

Geurts, K., Thomas, I., & Wets, G. (2005). Understanding spatial concentrations of road accidents using frequent item sets. Accident Analysis & Prevention, 37(4), 787-799.

Ghosh, B., Basu, B., O'Mahony, M. (2009). Multivariate short-term traffic flow forecasting using time-series analysis. IEEE Trans. Intell. Transp. Syst. 10 (2), 246–254.

Giuliano, G. (1989). Incident characteristics, frequency, and duration on a high volume urban freeway. Transportation Research Part A: General 23(5), 387-396.

Golob, T. F., Recker, W. W., & Leonard, J. D. (1987). An analysis of the severity and incident duration of truck-involved freeway accidents. Accident Analysis & Prevention, 19(5), 375-395.

Gontijo, G.M., Atuncar, G.S., Cruz, F.R.B., Kerbache, L. (2011). Performance evaluation and dimensioning of $GI^X/M/c/N$ systems through kernel estimation. Mathematical Problems in Engineering.

Gregoriades, A., and Mouskos, K. C. (2013). Black spots identification through a Bayesian Networks quantification of accident risk index. Transportation Research Part C: Emerging Technologies, Vol. 28, pp. 28-43.

Gregorutti, B., Michel, B., & Saint-Pierre, P. (2013). Correlation and variable importance in random forests. arXiv preprint arXiv:1310.5726.

Gupta, S. (2011). A framework to span airport delay estimates using transient queuing models. Oper. Res. Center.

Hall, D. L., & Llinas, J. (1997). An introduction to multisensor data fusion. Proceedings of the IEEE, 85(1), 6-23.

Han, J., Kamber, M & Pei, J. (2011). Data Mining: Concepts and Techniques. Morgan kaufmann.

Han, J., Pei, J., Yin, Y., & Mao, R. (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data mining and knowledge discovery 8(1), 53-87.

Haughton, M., & Sapna Isotupa, K. P. (2012). Scheduling commercial vehicle queues at a Canada–US border crossing. Transportation Research Part E: Logistics and Transportation Review, 48(1), 190-201.

He, Q., Kamarianakis, Y., Jintanakul, K., & Wynter, L (2013). Incident Duration Prediction with Hybrid Tree-based Quantile Regression. In Advances in Dynamic Network Modeling in Complex Transportation Systems (pp. 287-305). Springer New York.

Ho, K.K., Moody, G.B., Peng, C.K., Mietus, J.E., Larson, M.G., Levy, D., and Goldberger, A.L. (1997). Predicting survival in heart failure case and control subjects by use of fully automated methods for deriving nonlinear and conventional indices of heart rate dynamics. Circulation 96(3):842-848.

Hong, S., Kim, J., Oh, C., & Ulfarsson, G. F. (2014). The Effect of Road Environment Factors on Freeway Traffic Crash Frequency during Daylight, Twilight, and Night Conditions. In Transportation Research Board 93rd Annual Meeting (No. 14-2418).

Hong, W.C., Dong, Y., Zheng, F., Lai, C.Y. (2011). Forecasting urban traffic flow by SVR with continuous ACO. Appl. Math. Model. 35 (3), 1282–1291.

Hossain, M., & Muromachi, Y. (2012). A Bayesian network based framework for real-time crash prediction on the basic freeway segments of urban expressways. Accident Analysis & Prevention, 45, 373-381.

Huang, S. (2011). Next-generation transportation simulation and modeling tools. State University of New York at Buffalo.

Huang, S., & Sadek, A. W. (2009). A novel forecasting approach inspired by human memory: The example of short-term traffic volume forecasting. Transportation Research Part C: Emerging Technologies 17(5), 510-525.

Hung, M. C., & Yang, D. L. (2001). An efficient fuzzy c-means clustering algorithm. In Data Mining, Proceedings IEEE International Conference on, 225-232.

Hurst, H. E. (1951). Long-term storage of reservoirs: an experimental study. Transactions of the American society of civil engineers 116:770-799.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters 31(8), 651-666.

Jayawardena, A.W., Li, W.K., and Xu, P. (2002). Neighborhood selection for local modeling and prediction of hydrological time series. Journal of Hydrology, 258: 40-57.

Jekabsons G. (2010). M5PrimeLab: M5' regression tree and model tree toolbox for Matlab. http://www.cs.rtu.lv/jekabsons/

Joint Directors of Laboratories (JDL). (1991). Data Fusion Lexicon. Technical Panel For C3, F.E. White, San Diego, California, USA.

Juri, N. R., Unnikrishnan, A., & Waller, S. T. (2007). Integrated traffic simulation-statistical analysis framework for online prediction of freeway travel time. Transportation Research Record: Journal of the Transportation Research Board, 2039(1), 24-31.

Kam, A., Kaysi I., Abdulhai B., Peng J. (2005). Motorists guidance concept for Niagara Peninsula border crossings: delay prediction using ANN. In Transportation Research Board 84th Annual Meeting.

Karlaftis, M. G., and Tarko, A. P. (1998). Heterogeneity considerations in accident modeling. Accident Analysis & Prevention, Vol. 30, No. 4, pp. 425-433.

Karlaftis, M. G., & Vlahogianni, E. I. (2011). Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. Transportation Research Part C: Emerging Technologies, 19(3), 387-399.

Ke, J.C., Chu, Y.K. (2006). Nonparametric and simulated analysis of intensity for a queueing system. Appl. Math. Comput. 183 (2), 1280–1291.

Ke, J.C., Chu, Y.K. (2009). Comparison on five estimation approaches of intensity for a queueing system with short run. Comput. Stat. 24 (4), 567–582.

Kennel, M. B., Brown, R., and Abarbanel, H. D. I. (1992). Determining embedding dimension for phase-space reconstruction using a geometrical construction. Physical Review A 45(6): 3403-3411.

Keogh, E., & Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. Knowledge and information systems 7(3), 358-386.

Kim, K-J. Financial Time Series Forecasting Using Support Vector Machines. (2003). Neural Computation 55. pp. 307-319.

Kim, S. (2009). The toll plaza optimization problem: design, operations, and strategies. Transport. Res. Part E: Logist. Transport. Rev. 45 (1), 125–137.

Koppelman, F. S., & Bhat, C. (2006). A self instructing course in mode choice modeling: multinomial and nested logit models. US Department of Transportation, Federal Transit Administration, 31.

Kotsiantis, S., & Kanellopoulos, D. (2006). Discretization techniques: A recent survey. GESTS International Transactions on Computer Science and Engineering 32(1), 47-58.

Kouzani, A. Z. (2007). Road-sign identification using ensemble learning. In Intelligent Vehicles Symposium, 2007 IEEE (pp. 438-443). IEEE.

Lam, L., & Suen, C. Y. (1997). Application of majority voting to pattern recognition: an analysis of its behavior and performance. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 27(5), 553-568.

Ledoux, C. (1997). An urban traffic flow model integrating neural networks. Transp. Res. Part C: Emerg. Technol. 5 (5), 287–300.

Lee, C., Hellinga, B., & Saccomanno, F. (2003). Real-time crash prediction model for application to crash prevention in freeway traffic. Transportation Research Record: Journal of the Transportation Research Board 1840(1), 67-77.

Lee, Y., & Wei, C. H. (2010). A computerized feature selection method using genetic algorithms to forecast freeway accident duration times. Computer-Aided Civil and Infrastructure Engineering 25(2), 132-148.

Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. R News 2(3), 18-22.

Lin, F.B., Lin, M.W. (2001). Modeling traffic delays at northern New York border crossings. J. Transp. Eng. 127 (6), 540–545.

Liu, S., Zhang, Y., Zhu, S. and He, Q. (2011). Study on Prediction of Chaotic Time Series based on Phase Space Reconstruction. Proceedings of the 2011 International Conference on Informatics, Cybernetics, and Computer Engineering: 87-97.

Lord, D., & Mannering, F. (2010). The statistical analysis of crash-frequency data: a review and assessment of methodological alternatives. Transportation Research Part A: Policy and Practice, 44(5), 291-305.

Lu, J., Valois, F., Dohler, M., & Wu, M. Y. (2010). Optimized data aggregation in wsns using adaptive arma. The Fourth International Conference on Sensor Technologies and Applications, 115-120.

Lwebuga-Mukasa, J.S., Ayirookuzhi, S.J. and Hyland, A. (2002). Traffic Volumes and Respiratory Health Care Utilization among Residents in Close Proximity to the Peace Bridge before and after September 11, 2001. Journal of Asthma, Vol. 40, No. 8, pp. 855-864.

Mannering, F. L., & Bhat, C. R. (2014). Analytic methods in accident research: Methodological frontier and future directions. Analytic methods in accident research, 1, 1-22.

Ma, Yongchi and Liu, Zhibin. Combination Forecasting Petroleum Production based on Function of Fresh Degree. Acta Petrolei Sinica 26(1), pp. 87-95, 2005.

McCullagh, P. (2002). What is a statistical model?. Annals of statistics, 1225-1267.

Medhi, J., 2003. Stochastic Models in Queueing Theory. Academic Press.

Merkwirth, C., Parlitz, U., Wedekind, I., Engster, D. and Lauterborn, W. (2009). TSTOOL User Manual. http://www.physik3.gwdg.de/tstool/index.html.

Metagenomics Statistics. (2014). Random Forecasts. Retrieved December 14, 2014, from http://dinsdalelab.sdsu.edu/metag.stats/index.html

Miaou, S. P., Song, J. J., and Mallick, B. K. (2003). Roadway traffic crash mapping: A space-time modeling approach. Journal of Transportation and Statistics, Vol. 6, pp. 33-58.

Min, W., Wynter, L. (2011). Real-time road traffic prediction with spatio-temporal correlations. Transportation Research Part C: Emerging Technology 19 (4), 606–616.

Mohamed, M. G., Saunier, N., Miranda-Moreno, L. F., and Ukkusuri, S. V. (2013). A clustering regression approach: A comprehensive injury severity analysis of pedestrian–vehicle crashes in New York, US and Montreal, Canada. Safety Science, Vol. 54, pp. 27-37.

Montella, A. (2010). A comparative analysis of hotspot identification methods. Accident Analysis & Prevention, 42(2), 571-581.

Morgul, E. F., Yang, H., Kurkcu, A., Ozbay, K., Bartin, B., Kamga, C., & Salloum, R. (2014). Virtual Sensors: A Web-based Real-Time Data Collection Methodology for Transportation Operation Performance Analysis. Transportation Research Record: Journal of the Transportation Research Board, 2014.

Mukherjee, Rima. (2012). Travel and Transportation in the Age of Big Data. Retrieved September 20, 2014, from http://www.ibmbigdatahub.com/blog/travel-and-transportation-age-big-data

Müller, M. (2007). Information retrieval for music and motion. Springer.

Murphy, K. P., (2012). Machine learning: a probabilistic perspective. MIT Press.

Nam, D., & Mannering, F. (2000). An exploratory hazard-based analysis of highway incident duration. Transportation Research Part A: Policy and Practice 34(2), 85-102.

Nanni, L., & Lumini, A. (2008). Ensemble of multiple pedestrian representations. Intelligent Transportation Systems, IEEE Transactions on, 9(2), 365-369.

Nayak, R., Emerson, D., Weligamage, J., & Piyatrapoomi, N. (2011). Road crash proneness prediction using data mining. In Proceedings of the 14th International Conference on Extending Database Technology (pp. 521-526). ACM.

Netica tutorial. (2014). Retrieved December 12, 2014, from https://norsys.com/netica.html

Neuts, M. F. (1979). A versatile Markovian point process. Journal of Applied Probability 16, 764-779.

Newman, M. E., and Girvan, M. (2004). Finding and evaluating community structure in networks. Physical review E, Vol. 69, No. 2, 15 pages.

New York Times. (2013). Data Science: the Number of Our Lives. Retrieved December 5, 2014, from http://www.nytimes.com/2013/04/14/education/edlife/universities-offer-courses-in-a-hot-new-field-data-science.html?pagewanted=all&_r=0.

Niagara Falls Bridge Commission. Retrieved April 12, 2014, from http://niagarafallsbridges.com

Nguyen, L. N., & Scherer, W. T. (2003). Imputation techniques to account for missing data in support of intelligent transportation systems applications (No. UVACTS-13-0-78,). Charlottesville, VA: Center for Transportation Studies, University of Virginia.

Oh, C., Oh, J. S., & Ritchie, S. G. (2005). Real-time hazardous traffic condition warning system: framework and evaluation. Intelligent Transportation Systems, IEEE Transactions on 6(3), 265-272.

Ontario Chamber of Commerce (OCC). (2005). Cost of Border Delays to the United States Economy. Retrieved October 18, 2013, from http://www.thetbwg.org/downloads/Cost%20of%20Border%20Delays%20to%20the%20United%20States%20Economy%20 - %20April%202005.pdf

Okutani, I., Stephanedes, Y.J., 1984. Dynamic prediction of traffic volume through Kalman filtering theory. Transp. Res. Part B: Methodol. 18 (1), 1–11.

Ozbay, K., & Kachroo, P. (1999). Incident management in intelligent transportation systems. Artech House, Bonston.

Ozbay, K., & Noyan, N. (2006). Estimation of incident clearance times using Bayesian networks approach. Accident Analysis & Prevention 38(3), 542-555.

Pai, P-F., and Hong, W-C. (2005). Forecasting Regional Electricity Load based on Recurrent Support Vector Machines with Genetic Algorithms. Electric Power Systems Research 74. pp. 417-425.

Pande, A., Abdel-Aty, M. (2006). Comprehensive analysis of the relationship between real-time traffic surveillance data and rear-end crashes on freeways. Transportation Research Record: Journal of the Transportation Research Board 1953, 31–40.

Papadokonstantakis, S., Machefer, S., Schnitzlein, K., & Lygeros, A. I. (2005). Variable selection and data pre-processing in NN modelling of complex chemical processes. Computers & chemical engineering, 29(7), 1647-1659.

Park, B., Messer, C.J., Urbanik II, T. (1998). Short-term freeway traffic volume forecasting using radial basis function neural network. Transp. Res. Rec.: J. Transp. Res. Board 1651 (1), 39–47.

Park, B. B. (2002). Hybrid neuro-fuzzy application in short-term freeway traffic volume forecasting. Transportation Research Record: Journal of the Transportation Research Board, 1802(1), 190-196.

Parlar, M., Sharafali, M. (2008). Dynamic allocation of airline check-in counters: a queueing optimization approach. Manage. Sci. 8, 1410–1424.

Paselk, T.A., Mannering, F.L. (1994). Use of duration models for predicting vehicular delay at a US/Canadian border crossing. Transportation 21 (3), 249–270.

PTV (2010). VISSIM COM, User Manual for the VISSIM COM Interface, VISSIM 5.30. PTV Planung Transport Verkehr AG: Karlsruhe, Germany.

Qian, B., and Rasheed, K. (2004). Hurst Exponent and financial market predictability. IASTED conference on Financial Engineering and Applications: 203–209.

Quinlan, J. R. (1992). Learning with continuous classes, in Proc. 5th Australasian Joint Conf. Artif. Intell., Singapore, pp. 343–348.

Savolainen, P. T., Mannering, F. L., Lord, D., & Quddus, M. A. (2011). The statistical analysis of highway crash-injury severities: A review and assessment of methodological alternatives. Accident Analysis & Prevention, 43(5), 1666-1676.

Sawalha, Z., & Sayed, T. (2006). Traffic accident modeling: some statistical issues. Canadian Journal of Civil Engineering 33(9), 1115-1124.

Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. Machine learning, 37(3), 297-336.

Schreiber, Th. and Schmitz, A. (2000). Surrogate time series. Physical D 142: 346-382.

Schutt, R., & O'Neil, C. (2013). Doing Data Science: Straight Talk from the Frontline. O'Reilly Media, Inc.

Shang, P., Li, X. and Kamae, S. (2005). Chaotic analysis of traffic time series. Chaos, Solitons and Fractals 25: 121-128.

Shekhar, S. and Williams, B. M. (2007). Adaptive Seasonal Time Series Models for Forecasting Short-Term Traffic Flow. Transportation Research Record: Journal of the Transportation Research Board, No. 2024, pp. 116-125.

Smith, B.L., Demetsky, M.J., 1994. Short-term traffic flow prediction: neural network approach. Transp. Res. Rec.: J. Transp. Res. Board 1453, 98–104.

Smith, B.L. and Demetsky, M.J. (1997). Traffic flow forecasting: Comparison of modeling approaches. ASCE Journal of Transportation Engineering 123(4): 261-266.

Smith, B.L., Williams, B.M. and Oswald, R.K. (2002). Comparison of Parametric and Nonparametric Models for Traffic Flow Forecasting. Transportation Research Part C, pp. 303-321.

Smith, B.L., Scherer, W. T. and Conklin, J. H. (2003). Exploring Imputation Techniques for Missing Data in Transportation Management Systems. Transportation Research Record: Journal of the Transportation Research Board, No. 1836, pp. 132-142.

Smith, K. W., & Smith, B. (2001). Forecasting the clearance time of freeway accidents, University of Virginia, Charlottesville, http://cts.virginia.edu/docs/UVACTS-15-0-35.pdf

Sohn, S. Y., & Lee, S. H. (2003). Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents in Korea. Safety Science, 41(1), 1-14.

Srinivasan, D., Wai Chan, C., & Balaji, P. G. (2009). Computational intelligence-based congestion prediction for a dynamic urban street network. Neurocomputing, 72(10), 2710-2716.

Stathopoulos, A., Karlaftis, M.G. (2003). A multivariate state space approach for urban traffic flow modeling and prediction. Transportation Research Part C: Emerging Technology 11 (2), 121–135.

Stathopoulos, A., Dimitriou, L., & Tsekeris, T. (2008). Fuzzy modeling approach for combined forecasting of urban traffic flow. Computer−Aided Civil and Infrastructure Engineering, 23(7), 521-535.

Steinfeld, A., Zimmerman, J., Tomasic, A., Yoo, D., & Aziz, R. D. (2011). Mobile transit information from universal design and crowdsourcing. Transportation Research Record: Journal of the Transportation Research Board, 2217(1), 95-102.

Stone, M. (1974). Cross-validatory Choice and Assessment of Statistical Predictions. Journal of the Royal Statistical Society 36(2), pp. 111-147.

Sun, H., Henry X. Liu, H. Xiao, R.R. He and B. Ran. (2007). Use of local linear regression model for short term traffic forecasting. Transportation Research Record: Journal of the Transportation Research Board 1836/2003: 143-150.

Tan, H., Feng, G., Feng, J., Wang, W., Zhang, Y. J., & Li, F. (2013). A tensor-based method for missing traffic data completion. Transportation Research Part C: Emerging Technologies, 28, 15-27.

Tan, M. C., Wong, S. C., Xu, J. M., Guan, Z. R., & Zhang, P. (2009). An aggregation approach to short-term traffic flow prediction. Intelligent Transportation Systems, IEEE Transactions on, 10(1), 60-69.

Tang, Xiaowo. (1997). A New Fuzzy Adaptive Variable Weighting Algorithm for Combination Forecasting. Journal of the University of Electronic Science and Technology of China, pp. 289-292.

The Economist. (2010). Data, data everywhere. Retrieved December 14, 2014, from http://www.economist.com/node/15557443

Tostes, A. I. J., de LP Duarte-Figueiredo, F., Assunção, R., Salles, J., & Loureiro, A. A. (2013, August). From data to knowledge: city-wide traffic flows analysis and prediction using bing maps. In Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing (p. 12). ACM.

Tseng, F. M., Yu, H. C., & Tzeng, G. H. (2002). Combining neural network model with seasonal time series ARIMA model. Technological Forecasting and Social Change, 69(1), 71-87.

Turner, S. M., Eisele, W. L., Benz, R. J., & Holdener, D. J. (1998). Travel time data collection handbook (No. FHWA-PL-98-035,).

US Census Bureau. (2012). Motor Vehicle Accidents-Number and Deaths: 1990 to 2009. Retrieved December 13, 2014, from http://www.census.gov/compendia/statab/2012/tables/12s1103.pdf

U.S. Customs and Border Protection. Border Wait Times. Retrieved March 6, 2014, from http://bwt.cbp.gov/?com=1&pas=1&ped=1&plist=0901

U.S. Department of Transportation (U.S.DOT). (2008). US Department of Transportation Unveils New Program to Fight Border Congestion. Retrieved on October 18, 2013, from http://www.dot.gov/affairs/fhwa1208.htm

U.S. Department of Transportation (U.S.DOT), Research and Innovative Technology Administration. Connected Vehicle Research in the United States. Retrieved September 21, 2014, from http://www.its.dot.gov/connected_vehicle/connected_vehicle_research.htm

U.S. Department of Transportation (U.S.DOT). Research and Innovative Technology Administration. What is ITS. Retrieved September 21, 2014, from http://www.its.dot.gov/faqs.htm

Valent, F., Schiava, F., Savonitto, C., Gallo, T., Brusaferro, S., and Barbone, F. (2002). Risk factors for fatal road traffic accidents in Udine, Italy. Accident Analysis & Prevention, Vol. 34, No. 1, pp. 71-84.

Valenti, G., Lelli, M., & Cucina, D. (2010). A comparative study of models for the incident duration prediction. European Transport Research Review, 2(2), 103-111.

Van Der Voort, M., Dougherty, M., Watson, S. (1996). Combining Kohonen maps with ARIMA time series models to forecast traffic flow. Transportation Research Part C: Emerging Technology 4 (5), 307–318.

Van Hinsbergen, C. P. I., & van Lint, J. W. (2008). Bayesian combination of travel time prediction models. Transportation Research Record: Journal of the Transportation Research Board, 2064(1), 73-80.

Van Woensel, T., & Vandaele, N. (2006). Empirical validation of a queueing approach to uninterrupted traffic flows. 4OR, 4(1), 59-72.

Van Woensel, T., & Vandaele, N. (2007). Modeling traffic flows with queueing models: a review. Asia-Pacific Journal of Operational Research, 24(04), 435-461.

Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer, New York.

Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C. (2005). Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. Transp. Res. Part C: Emerg. Technol. 13 (3), 211–234.

Vlahogianni, E. I., Karlaftis, M.G. and Golias, J.C. (2006). Statistical methods for detecting nonlinearity and non-stationarity in univariate short-term time-series of traffic volume. Transportation Research Part C 14: 351-367.

Wang, H., Fan, W., Yu, P. S., & Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 226-235). ACM.

Wang, Y., & Witten, I. H. (1997). Inducing model trees for continuous classes. In Proceedings of the Ninth European Conference on Machine Learning, pp. 128-137.

Waze. Retrieved April 6, 2014, from https://www.waze.com/

Weather Underground. Information about Our Data. Retrieved September 22, 2014, from http://www.wunderground.com/about/data.asp

Wei, C. H., & Lee, Y. (2007). Sequential forecast of incident duration using Artificial Neural Network models. Accident Analysis & Prevention 39(5), 944-954.

Wei, Y., Chen, M.C. (2012). Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. Transp. Res. Part C: Emerg. Technol. 21 (1), 148–162.

Wen, F. and Wan, Q. (2009). Time Delay Estimation based on Mutual information. Image and Signal Processing.

Wikipedia. (2014). Data science. Retrieved December 5, 2014, from http://en.wikipedia.org/w/index.php?title=Data_science&oldid=636340200.

Wikipedia. (2014). Grabage in, Garbage out. Retrieved September 24, 2014, from http://en.wikipedia.org/wiki/Garbage_in,_garbage_out

Wikipedia. (2014). Machine Learning. Retrieved September 27, 2014, from http://en.wikipedia.org/wiki/Machine_learning.

Wikipedia. (2014). Android. Retrieved April 2, 2014, from http://en.wikipedia.org/wiki/Android_(operating_system)#Applications

Wild, D. (1997). Short-term Forecasting based on a Transformation and Classification of Traffic Volume Time Series. International Journal of Forecasting 13, pp. 63-72.

William, H. LAM, K., Tang,Y.F., Chan, K.S. and Tam, M-L. (2006). Short-term hourly traffic forecasting using Hong Kong Annual Traffic Census. Transportation 33: 291-310.

Williams, B.M. (2001). Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. Transportation Research Record: Journal of Transportation Research Board 1776 (1), 194–200.

Williams, B.M. and Hoel, L.A. (2003). Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results. Journal of Transportation Engineering 129(6), pp. 664-672.

Xie, Y., Zhang, Y. (2006). A wavelet network model for short-term traffic volume forecasting. J. Intell. Transp. Syst. 10 (3), 141–150.

Xie, Y., Zhang, Y., Ye, Z. (2007). Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition. Computer-Aided Civil and Infrastructure Engineering. 22 (5), 326–334.

Xie, Z., and Yan, J. (2008). Kernel density estimation of traffic accidents in a network space. Computers, Environment and Urban Systems, Vol. 32, No. 5, pp 396-406.

Xu, C., Tarko, A. P., Wang, W., & Liu, P. (2013). Predicting crash likelihood and severity on freeways with real-time loop detector data. Accident Analysis & Prevention 57, 30-39.

Ye, Y., Wu, Q., Zhexue Huang, J., Ng, M. K., & Li, X. (2013). Stratified sampling for feature subspace selection in random forests for high dimensional data. Pattern Recognition 46(3), 769-787.

Yeh, Y. R., Lin, T. C., Chung, Y. Y., & Wang, Y. C. (2012). A novel multiple kernel learning framework for heterogeneous feature fusion and variable selection. Multimedia, IEEE Transactions on, 14(3), 563-574.

Yin, H., Wong, S.C., Xu, J., Wong, C.K. (2002). Urban traffic flow prediction using a fuzzy-neural approach. Transportation Research Part C: Emerging Technology 10 (2), 85–98.

Yu, R., & Abdel-Aty, M. (2013). Utilizing support vector machine in real-time crash risk evaluation. Accident Analysis & Prevention 51, 252-259.

Zhan, C., Gan, A, Hadi, M. (2011). Prediction of lane clearance time of freeway incidents using the m5p tree algorithm. Intelligent Transportation Systems, IEEE Transactions on, 12(4), 1549-1557.

Zhang, B. (2013). Reliable classification of vehicle types based on cascade classifier ensembles. Intelligent Transportation Systems, IEEE Transactions on, 14(1), 322-332.

Zhang, G. P. (2003). Time Series Forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50. pp. 159-175.

Zhang, J., Wang, F. Y., Wang, K., Lin, W. H., Xu, X., & Chen, C. (2011). Data-driven intelligent transportation systems: A survey. Intelligent Transportation Systems, IEEE Transactions on, 12(4), 1624-1639.

Zhang, Y., & Liu, Y. (2009). Data Imputation Using Least Squares Support Vector Machines in Urban Arterial Streets. Signal Processing Letters, IEEE, 16(5), 414-417.

Zhang, Y. and Xie, Y. (2008). Forecasting of short-term freeway volume with v-support vector machines. Transp. Res. Rec.: J. Transp. Res. Board 2024 (1), 92–99.

Zhang, Y., & Ye, Z. (2008). Short-term traffic flow forecasting using fuzzy logic system methods. Journal of Intelligent Transportation Systems, 12(3), 102-112.

Zhang, Z.G., Luh, H.P., Wang, C.H. (2011). Modeling security-check queues. Manage. Sci. 57 (11), 1979–1995.

Zhao, Y., & Sun, J. (2010). Improved scheme to accelerate sparse least squares support vector regression. Journal of Systems Engineering and Electronics 21(2), 312-317.

Zheng, Z., Ahn, S., & Monsere, C. M. (2010). Impact of traffic oscillations on freeway crash occurrences. Accident Analysis & Prevention 42(2), 626-636.

Zheng, W., Lee, D. H., & Shi, Q. (2006). Short-term freeway traffic flow prediction: Bayesian combined neural network approach. Journal of Transportation Engineering, 132(2), 114-121.

Zimmerman, J., Tomasic, A., Garrod, C., Yoo, D., Hiruncharoenvate, C., Aziz, R., & Steinfeld, A. (2011). Field trial of tiramisu: crowd-sourcing bus arrival times to spur co-design. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1677-1686). ACM.