

ABSTRACT

Title of Document: UNCERTAINTY ASSOCIATED WITH
TRAVEL TIME PREDICTION: ADVANCED
VOLATILITY APPROACHES AND
ENSEMBLE METHODS

YANRU ZHANG, PH.D, 2015

Directed By: ALI HAGHANI, PROFESSOR
DEPARTMENT OF CIVIL AND
ENVIRONMENTAL ENGINEERING

Travel time effectively measures freeway traffic conditions. Easy access to this information provides the potential to alleviate traffic congestion and to increase the reliability in road networks. Accurate travel time information through Advanced Traveler Information Systems (ATIS) can provide guidance for travelers' decisions on departure time, route, and mode choice, and reduce travelers' stress and anxiety. In addition, travel time information can be used to present the current or future traffic state in a network and provide assistance for transportation agencies in proactively developing Advanced Traffic Management System (ATMS) strategies. Despite its importance, it is still a challenging task to model and estimate travel time, as traffic often has irregular fluctuations. These fluctuations result from the interactions among different vehicle-driver combinations and exogenous factors such as traffic incidents,

weather, demand, and roadway conditions. Travel time is especially sensitive to the exogenous factors when operating at or near the roadway's capacity, where congestion occurs. Small changes in traffic demand or the occurrence of an incident can greatly affect the travel time. As it is impossible to take into consideration every impact of these unpredictable exogenous factors in the modeling process, travel time prediction problem is often associated with uncertainty. This research uses innovative data mining approaches such as advanced statistical and machine learning algorithms to study uncertainty associated with travel time prediction. The final objective of this research is to develop more accurate and reliable travel time prediction models.

UNCERTAINTY ASSOCIATED WITH TRAVEL TIME PREDICTION:
ADVANCED VOLATILITY APPROACHES AND ENSEMBLE METHODS

By

Yanru Zhang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

2015

Advisory Committee:
Professor Haghani, Ali, Chair
Professor Schonfeld, Paul M.
Associate Professor Cirillo, Cinzia
Assistant Professor Forman, Barton
Associate Professor Alt, Frank B.

© Copyright by
Yanru Zhang
2015

Dedication

To my parents, sister and husband,
for their endless love, support and encouragement.

Acknowledgements

First and foremost, I would like to express my greatest gratitude to my advisor, Dr. Ali Haghani, for the continuous support of my PhD study and research, for his guidance, advice, patience, motivation, enthusiasm, and immense knowledge. I cannot thank him enough for all his contributions of time, ideas, and supports throughout my research and writing the dissertation. I respect him for his attitude towards life, and his generosity to his students. One simply could not wish for a better or friendlier supervisor.

Special thanks to the members of my dissertation committee: Dr. Paul M. Schonfeld, Dr. Cinzia Cirillo, Dr. Barton Forman, and Dr. Frank B. Alt for generously giving their time, encouragement, valuable expertise and advice to improve my research. It is a great honor to have them on my committee. Their insights helped me to further improve my research work.

I am grateful for the friendship and encouragement of all members in our graduate research group, especially Masoud Hamed and Kaveh Farokhi Sadabadi for sharing their time and ideas through discussion of this topic. To all my friends, who have made my life so colorful and exciting, I cannot list all your names here, but you are always on my mind.

Finally, I would like to dedicate this dissertation to my family especially my parents and my husband. My parents made many sacrifices to ensure I get the best possible education and have provided me with unconditional love, support and understanding. My husband always supports me in my academic pursuits and is truly special to me. I am especially thankful for the wonderful life that we share together.

Table of Contents

Dedication.....	ii
Acknowledgements.....	iii
Table of Contents.....	iv
List of Tables.....	vi
List of Figures.....	viii
Chapter 1: Introduction.....	1
1.1 Problem Statement.....	1
1.2 Research Objectives.....	3
1.3 Research Contributions.....	5
1.4 Dissertation Outline.....	8
Chapter 2: Literature Review.....	10
2.1 Parametric Approaches.....	11
2.1.1 Naïve Methods.....	11
2.1.2 Autoregressive Linear Processes.....	12
2.1.3 State Space Models.....	12
2.2 Non-parametric Approaches.....	13
2.2.1 Non-parametric Regression.....	13
2.2.2 Neural Network.....	14
2.2.3 Other Artificial Intelligence Methods.....	16
2.3 Hybrid Approaches.....	16
2.3.1 Classification Based Approach.....	16
2.3.2 Kalman Filtering Based Approach.....	17
2.3.3 Decomposition Technique.....	18
2.3.4 Ensemble Trees.....	19
2.3.5 Other Combination Approaches.....	20
2.4 Prediction Interval Based Approaches.....	21
2.4.1 Ensemble Methods.....	22
2.4.2 Statistical Volatility Based Approach.....	23
2.5 Summary.....	24
Chapter 3: Statistical Volatility Models for Reliable Travel Time Prediction.....	27
3.1 Mean Prediction Models.....	29
3.1.1 Theoretical Background of ARIMA Models.....	30
3.1.2 ARIMA Model Optimization.....	32
3.1.3 Model Evaluation Criteria.....	33
3.2 Volatility Models.....	33
3.2.1 GARCH-type Models.....	34

3.2.2 Component GARCH Models.....	35
3.2.2 Stochastic Volatility Model	39
3.2.3 Prediction Interval Estimation	43
3.3 Application of Component GARCH Models in Travel Time Prediction	44
3.3.1 Modeling Conditional Mean	45
3.3.2 Testing the ARCH Effect.....	47
3.3.3 Estimating the Volatility Model.....	48
3.3.4 Construct the Mean and Prediction Intervals.....	49
3.3.5 Results and Discussion	51
3.3.6 Summary	57
3.4 Application of Stochastic Volatility Model	58
3.4.1 Model Fitting	64
3.4.2 Results and Analysis.....	65
3.4.3 Summary	70
Chapter 4: Ensemble Methods in Travel Time Prediction.....	73
4.1 Common Types of Ensembles	75
4.1.1 Bagging.....	76
4.1.2 Boosting.....	78
4.2 Ensemble Tree	79
4.2.1 Single Regression Tree	81
4.2.3 Random Forest Regression	83
4.2.4 Gradient Boosted Regression Tree	86
4.3 Application to Travel Time Prediction	90
4.3.1 Data Description and Preparation	91
4.3.2 Model Optimization.....	94
4.3.3 Model Interpretation	100
4.3.4 Model Comparison.....	104
4.3.5 Discussion and Conclusion.....	108
Chapter 5: A Travel Time Prediction Framework	110
5.1 Model Development.....	110
5.2 Data Description and Preparation	111
5.3 Model Comparison.....	115
5.4 Chapter Summary	147
Chapter 6: Conclusion and Recommendations.....	148
6.1 Summary.....	148
6.2 Conclusion	151
6.3 Future Recommendation.....	152

List of Tables

Table 1 Estimated MPIL and PICP values for GARCH, C-GARCH and MC-GARCH models.....	53
Table 2. Selected segments for this study.....	61
Table 3 Performance measures of the mean equation	68
Table 4 Selected Freeway Segments for the Study.....	92
Table 5 Basic Statistics of Travel Time Data	93
Table 6 Example of the Training/Testing Data File	93
Table 7 Relative Influence of Input Variables for GBM Models with Learning Rate of 0.001 for Multistep-ahead Prediction	102
Table 8 Comparison of 5 Minutes Ahead Prediction for ARIMA, RF and GBM....	105
Table 9 Comparison of 15 Minutes Ahead Prediction for ARIMA, RF and GBM..	106
Table 10 Comparison of 30 Minutes Ahead Prediction for ARIMA, RF and GBM	106
Table 11 Selected Freeway Segment Information	115
Table 12 Comparing the MAPE Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-08 Thursday)	122
Table 13 Comparing the RMSE Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-08 Thursday)	123
Table 14 Comparing the MPIL Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-08 Thursday)	124
Table 15 Comparing the PICP Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-08 Thursday)	125
Table 16 Comparing the PI-Ratio of the GBM-SV and the ARIMA-GARCH Models (2012-11-08 Thursday)	126
Table 17 Comparing the MAPE Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-13 Tuesday).....	127
Table 18 Comparing the RMSE Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-13 Tuesday).....	128
Table 19 Comparing the MPIL Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-13 Tuesday).....	129
Table 20 Comparing the PICP Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-13 Tuesday).....	130
Table 21 Comparing the PI-Ratio of the GBM-SV and the ARIMA-GARCH Models (2012-11-13 Tuesday).....	131
Table 22 Comparing the MAPE Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-26 Monday).....	132
Table 23 Comparing the RMSE Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-26 Monday).....	133
Table 24 Comparing the MPIL Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-26 Monday).....	134
Table 25 Comparing the PICP Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-26 Monday).....	135
Table 26 Comparing the PI-Ratio of the GBM-SV and the ARIMA-GARCH Models (2012-11-26 Monday).....	136

Table 27 Comparing the MAPE Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-05 Wednesday).....	137
Table 28 Comparing the RMSE Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-05 Wednesday).....	138
Table 29 Comparing the MPIL Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-05 Wednesday).....	139
Table 30 Comparing the PICP Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-05 Wednesday).....	140
Table 31 Comparing the PI-Ratio of the GBM-SV and the ARIMA-GARCH Models (2012-12-05 Wednesday).....	141
Table 32 Comparing the MAPE Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-19 Wednesday).....	142
Table 33 Comparing the RMSE Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-19 Wednesday).....	143
Table 34 Comparing the MPIL Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-19 Wednesday).....	144
Table 35 Comparing the PICP Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-19 Wednesday).....	145
Table 36 Comparing the PI-Ratio of the GBM-SV and the ARIMA-GARCH Models (2012-12-19 Wednesday).....	146

List of Figures

Figure 1. Concept of prediction interval [93]	28
Figure 2. Box Plot of Absolute Deviation from Predicted Mean	46
Figure 3. Multiplicative component GARCH forecasting results: decomposition of the volatility into its various components (32-33).	50
Figure 4 Predicted mean and PI for multiplicative component GARCH model	51
Figure 5. Comparing performance of GARCH, C-GARCH and MC-GARCH models during peak hours.....	55
Figure 6. Comparing performance of GARCH, C-GARCH and MC-GARCH during non-peak hours.....	55
Figure 7. Comparison of prediction intervals constructed by GARCH, C-GARCH and MC-GARCH models	56
Figure 8. Bluetooth sensor location of the study	59
Figure 9. A scatter plot of travel times on four paths.	63
Figure 10. Prediction results for peak hour travel time at four segments.	67
Figure 11. Comparison of performance measures for six dataset by using ARIMA-GARCH and ARIMA-SV model with (a) 5 minute time interval (b) 15 minute time interval	70
Figure 12 The Bagging algorithm.....	77
Figure 13. The Boosting algorithm.....	79
Figure 14. Single regression tree	82
Figure 15. Pseudo-code for random forest.....	84
Figure 16. Pseudo-code for generic gradient boosting	88
Figure 17. The Relationship between MAPE and Number of Trees for Models Fitted with Seven Learning Rates and Four Levels of Interactions	97
Figure 18. MAPE against Learning Rate for Models Fitted with Various Numbers of Trees and Different Levels of Interactions	98
Figure 19. MAPE against Tree Complexity for Models Fitted with Various Numbers of Trees and Different Learning Rate	99
Figure 20. Three-dimensional Plots for The Joint Effects of Lag One Difference and Lag One Travel Time on Predicted Travel Time Value	103
Figure 21. Sample Travel Time Prediction Results of the GBM method.....	107
Figure 22. Selected Study Segment at the I95 Southbound Direction.....	112
Figure 23. Selected Study Segment at the I495 Eastbound Direction	113
Figure 24. Selected Study Segment at the MD295 Southbound Direction.....	114

Chapter 1: Introduction

Travel time is widely acknowledged as an effective measure of highway traffic conditions, which can be easily understood by both travelers and transportation agencies [1]. Access to accurate travel time information has the potential to alleviate traffic congestion, to minimize its negative environmental and societal side effects, and to increase road network reliability [2]. Advanced Traveler Information System (ATIS) provides travelers with accurate and timely traffic information via dynamic message signs, radio, and internet. The pre-trip travel time information gives guidance for travelers' decisions for departure time, route, and mode choice, and reduces travelers' stress and anxiety. In addition, travel time information can also be used to present the current or future traffic state in a network and to provide assistance for transportation agencies in proactively developing Advanced Traffic Management System (ATMS) strategies. For example, travel time is one of the performance measures in the Freeway Performance Measurement System (PeMS) developed by the California Department of Transportation [2]. The Split Cycle Offset Optimization Technique system and the Sydney Coordinated Adaptive Traffic system are two successful traffic operation systems that use travel time information as their module input [3]. Therefore, travel time information is a critical input and output for intelligent transportation system.

1.1 Problem Statement

The success of ATIS and ATMS relies not only on the availability and accuracy of historical and real time traffic information, but also on future traffic

information. A wide range of methodologies in travel time forecasting has been proposed to model traffic characteristics and to produce short term forecasts. Most of these methods are based on the historical travel time data concurrently collected from various detection systems, such as vehicles with GPS or Bluetooth devices, electronic toll system, and video detection. Especially with the recent technology advances in vehicle tracking, direct and accurate travel time information can be easily obtained. The technology improvements make the development of an online travel time prediction algorithm more meaningful.

Despite the proliferation of traffic prediction methodologies in the existing literature, modeling and estimating travel time is still a challenging task [4]. Travel time experiences strong fluctuations across different periods and traffic conditions. These fluctuations result from the interactions among different vehicle-driver combinations, and exogenous factors such as traffic incidents, weather, demand, and roadway conditions. Travel time is especially sensitive to the exogenous factors when operating at or near the roadway's capacity, where congestion occurs. Small changes in traffic demand or the occurrence of an incident can greatly affect the travel time. As the impact of these unpredictable exogenous factors is impossible to be considered fully in the modeling process, travel time prediction problem is often associated with uncertainty.

Addressing uncertainties associated with travel time, and therefore, travel time reliability has become a topic of interest in recent years. FHWA defines travel time reliability as “the consistency or dependability in travel times, as measured from day-to-day and/or across different times of the day” [18] and proposes several travel time

reliability measures: 90th or 95th percentile travel time, buffer index, and planning time index. For prediction purpose, another efficient measure is prediction intervals (PIs) [19], based on which one can assess the reliability of the travel time forecasting results. Prediction interval is an estimated interval, which covers the expected travel time value with a predetermined probability [20]. In other words, PIs give a likely range of the predicting results to represent the uncertainties associated with travel time. Availability of PIs allows the travelers and traffic managers to quantify the level of uncertainty associated with predicted travel time and thus to make multiple strategies on route and departure time choice to deal with the worst and best conditions. Wide PIs indicate a higher uncertainty of the future traffic conditions and travelers should expect extreme delays, while narrow PIs mean the traffic condition are relatively stable [21].

In brief, due to the dynamic and stochastic nature of traffic, travel time prediction is one of the most challenging tasks in ATIS and ATMS. In order to provide meaningful traffic information to travelers and traffic managers, it is critical to develop an accurate and reliable traffic prediction algorithm that not only reduces the absolute value of prediction error but also takes into consideration the uncertainty associated with travel time prediction.

1.2 Research Objectives

The primary goal of this research is to identify uncertainties associated with travel time prediction. Both accuracy and reliability issues are addressed in terms of freeway travel time prediction. Prediction accuracy emphasizes the difference between the predicted and the actual value, or in other words, prediction error. Most

existing travel time prediction methods in the literature focus on improving travel time prediction accuracy without considering the uncertainty issue. On the other hand, reliability puts more emphasis on uncertainty associated with prediction. Instead of providing a point value (an average of travel time during a certain time interval), a prediction interval is proposed to represent how likely it will capture the observed value. To achieve both objectives, this research introduces and implements two types of data driven approaches to predict travel time and to assess uncertainty associated with predictions. These two types of methods are: statistical volatility models and ensemble methods.

A statistical volatility model is promising in terms of modeling uncertainty, as it not only provides the opportunity to develop a more accurate mean model but also produce an effective and efficient prediction interval. This research proposes and compares different types of statistical volatility-based travel time prediction models. The preliminary study results indicate that a statistical volatility model can be a promising approach to account for uncertainties associated with prediction.

In recent years, another type of prediction method—the ensemble method—has received increased interest in the prediction field. Instead of fitting a single “best” model, the ensemble method strategically combines multiple simple base models to optimize predictive performance. Drawing from insights and techniques from both statistical and machine learning methods, ensemble methods often achieve strong predictive performance. They are often less sensitive to missing data and outliers and are able to model complex relations among variables. Since traffic is a complex phenomenon, a single model may not be able to capture the complex relations.

Combining a group of individual base models potentially improves model performance. The second part of the dissertation will focus on developing ensemble-based travel time prediction models. This research involves following tasks:

- (1) Investigate the characteristics of travel time data collected from Bluetooth sensors and probe vehicles with a particular focus on the seasonality and variability of data during different time intervals.
- (2) Demonstrate the concept of prediction intervals in modeling uncertainties associated with travel time prediction.
- (3) Propose two innovative statistical volatility models that provide more reliable prediction through consideration of the changing behaviors of travel time variation. Both stochastic and seasonal (cyclical) characteristics of traffic data are addressed.
- (4) Propose an advanced ensemble algorithm for freeway travel time prediction to improve prediction accuracy.
- (5) Propose a new travel time prediction framework that is able to provide more accurate and reliable prediction, and evaluate and compare performance of different prediction algorithms comprehensively in terms of prediction accuracy and reliability.

1.3 Research Contributions

This section lists the main contributions to the state-of-the-art research offered in this dissertation.

We have developed two innovative component volatility-based travel time prediction models to better characterize long-term and short-term volatility and

cyclical patterns in travel time data. Because of the daily, weekly or even monthly recurrent traffic congestion, travel time data often show strong cyclical patterns. In the traffic prediction field, existing volatility models do not consider the possible cyclical patterns in the residual series, often referred to as a seasonal component. Conventional generalized autoregressive conditional heteroskedasticity (GARCH) models are often criticized as unsatisfactorily modeling data series that show pronounced seasonal patterns [5]. The decomposition technologies provide the potential to deal with trend and seasonal components in the data. Driven by the successful application of statistical volatility models in transportation analyses, the component GARCH models are proposed. The component GARCH models are similar to the structure of the GARCH model but include trend and seasonal elements. The component GARCH models allow for a more versatile structure with the potential to provide more accurate traffic volatility forecasting along freeway corridors.

We have introduced advanced solution technologies, namely Bayesian inference using the Markov Chain Monte Carlo (MCMC) based on ancillarity-sufficiency interweaving strategy (ASIS) proposed by Gregor & Sylvia [6], for stochastic volatility (SV) model in travel time volatility forecasting. The proposed method greatly improves the efficiency and robustness of the SV model. We have compared the proposed SV model with the GARCH model by using freeway travel time data and have demonstrated the advanced SV model as a competitive alternative in modeling the volatility nature in traffic.

We have proposed a tree-based ensemble method to predict travel time on a freeway stretch by considering all relevant variables derived from historical travel time data. Belonging to the machine learning category, the tree-based ensemble methods often have superior prediction performance over classical statistical models. Driven by the successful applications of random forest methods in traffic parameter prediction, a gradient boosting tree (GBM)-based travel time prediction method is proposed to uncover hidden patterns in travel time data to enhance the accuracy and interpretability of the model. Different from the random forest algorithm that averages a large collection of trees from random sampling [7], the gradient boosting method sequentially generates base learners from a weighted version of the training data to strategically find the optimal combination of trees. Each step of adding another base learner is aimed at correcting the mistakes made by its previous learners. Therefore, the gradient boosting method has the potential to provide more accurate predictions.

We have evaluated and tested the effect of different combinations of parameters on the gradient boosting method's performance in travel time predictions comprehensively. One issue regarding the prediction accuracy of the GBM model is related to parameter optimization. The performance of the GBM model is largely influenced by its parameters, including the number of trees, learning rate and tree complexity (variable interactions). Therefore, there is a need to test the optimal combination of variables when developing the GBM model. Computational time is another issue when tree complexity or the number of trees increases. The tradeoff

between computational cost and model accuracy should also be considered when building the model.

We have developed a new travel time prediction framework that takes advantage of both the GBM and SV models. Through the combination of these two models, the new proposed prediction framework potentially further improves the model performance. The proposed travel time prediction framework also provides an example of how to improve the overall performance of the travel time prediction model.

1.4 Dissertation Outline

The rest of this dissertation is organized as follows. The next chapter reviews previous works in freeway travel time prediction. The existing literature is categorized as parametric, non-parametric, hybrid and prediction interval based approaches. Chapter 3 describes statistical volatility models in travel time reliability prediction. The model structure and the concept of prediction interval are introduced. The statistical volatility models are composed of two parts: mean and prediction interval. The mean prediction models and the volatility models (to construct prediction intervals) are discussed in Chapter 3.1 and in Chapter 3.2. Chapter 3.3 performs a case study using volatility models in travel time prediction. Chapter 4 describes ensemble methods in travel time prediction. Chapter 4.1 summarizes common types of ensemble methods. Chapters 4.2 and 4.3 discuss in detail different ensemble tree methods and the application of these ensemble methods in travel time prediction. Chapter 5 proposes a new travel time prediction framework. Chapter 6 concludes the dissertation and also provides further recommendations for future

research in this topic.

Chapter 2: Literature Review

Short-term traffic prediction as a critical component in a real-time ITS environment has seen an explosion of interest since the 1980s. A large number of forecasting algorithms have been proposed in the literature. Because of the complexity of the traffic prediction problem, existing traffic prediction approaches are different from one another in different aspects. Vlahogianni et al. [8] suggested that the process of developing short-term traffic prediction can be divided into three essential clusters (scope, conceptual output and modeling) that involves two important issues (design and modeling parameters). The designing process mainly focuses on the objective of forecasting that includes what types of application, where to implement, and the desired output of the model. The modeling parameters procedure is the way to achieve the goal determined during the design process. This study belongs to the modeling parameters procedure.

Since this study primarily focuses on freeway travel time prediction, we will emphasize the literature on different modeling techniques in this area. There are several review or comparative studies in existing short-term traffic prediction methods [8-11] and interested readers can refer to these studies for further references. Vlahogianni et al. [8] classified traffic prediction models as consisting of three modeling approaches: parametric, non-parametric and hybrid methods. The parametric approach usually assumes a specific form for the dependent and independent variables. The modeling process involves model identification, parameter estimation, model diagnostic checking and prediction. The parametric models often have more assumptions than non-parametric methods. If these

assumptions are satisfied by the data, the parametric approach can produce accurate estimation. Otherwise, the parametric approach can be misleading. The non-parametric models are data driven approaches that usually do not assume a specific structure of the data. These algorithms heavily depend on the quality of the available data. Another traffic forecasting approach is the hybrid method that combines different models or the same type of models with different initial values or parameters to obtain better prediction performances. In addition, as uncertainties are often involved in traffic prediction, prediction interval-based approaches consider the future traffic parameter (such as volume, speed or travel time) as a distribution instead of a point value. Mostly existing prediction-interval based approaches belong to the hybrid method. Because of its importance, we discuss this approach as a separate category. The following sections will summarize the major research findings of existing literature based on the taxonomy proposed by Vlahogianni et al. [4].

2.1 Parametric Approaches

2.1.1 Naïve Methods

The naïve method can be interpreted as a simple and easy implementation method without many model assumptions. Historical average and smoothing [12] techniques received extensive attention in practical applications [13, 14]. The historical average model simply averages historical traffic data on either a certain time of the day, day of the week or other time periods based on the assumption that traffic shows similar patterns throughout the day, week or year, e.g. traffic patterns day to day often show remarkable similarities and these patterns are useful for prediction. The historical average method has already been applied to the urban

traffic control systems (UTCS) [15] and other various traveler information systems [16, 17]. However, as traffic conditions are highly dynamic, the naïve method is often a poor predictor.

2.1.2 Autoregressive Linear Processes

In the early 1990s, transportation researchers developed an alternative approach—autoregressive linear process such as the autoregressive integrated moving average (ARIMA) type models in predicting traffic. The ARIMA-type models were first introduced by Ahmed and Cook [18] and Levin and Tsao [19] in freeway traffic flow and occupancy prediction. Their studies indicated that the ARIMA-type models provide better forecasting accuracy compared with historical average and smoothing techniques. Applications of the ARIMA model in traffic parameters forecasting was also discussed in later studies [20-24]. As traffic parameters show spatial-temporal correlations, both Kamarianakis and Prastacos [22] and Min and Wynter [25] adopted a multivariate spatial-temporal autoregressive moving average model to predict traffic flow. Because of its well defined theoretical foundation and effectiveness in prediction [26], the ARIMA-type models gradually become standard methods to compare with newly developed forecasting models. However, the ARIMA-type models are sensitive to extreme values. This makes the ARIMA-type model less efficient when modeling data with large variations.

2.1.3 State Space Models

State space models belong to the multivariate forecasting category, which can be applied to multiple inputs – multiple outputs systems. It is worth noting that the

‘state space model’ and the more widely known ‘Kalman filter model’ refer to the same model structure. The term ‘state space’ refers to the model, the term ‘Kalman filter’ refers to the process of estimating and updating model parameters.

Stathopoulos and Karlaftis [27] applied a multivariate state space model to predict flow at an urban signalized arterial. Ghosh et al. [28] applied a structural time series model to forecast traffic flow in a congested urban transportation network. The structural time series model is a special form of the state-space model that represents the observed time series as a sum of different components. Their study results indicate that the proposed model is computationally more efficient and can trace the evolution of each individual component separately. The Kalman filtering algorithm allows the parameters of the model to be updated with new data available. Therefore it enables dynamic traffic prediction [29-35].

2.2 Non-parametric Approaches

2.2.1 Non-parametric Regression

Nonparametric regression is a form of regression analysis that does not predetermine a specific form for the predictor. It relies on data to describe the relationship between dependent and independent variables and is based on the principle of pattern recognition and chaotic systems [36]. Smith and Demetsky [37] demonstrated the advantages of the nonparametric regression approach when compared with the neural network. Another study by Smith et al. [36] suggested that the heuristic forecast generation method improves the performance of the nonparametric regression but does not necessarily perform better than the seasonal ARIMA model. Clark [38] proposed a nonparametric regression technique described

as a k nearest neighbor (k-NN) model to predict traffic state variables. Davis and Nihan [39] applied the k-NN model in predicting freeway traffic. They suggested that the k-NN method is comparable to the linear time-series approach. Robinson and Polak [40] proposed the use of the k-NN technique to model urban link travel time. They discussed in details the selection of a distance metric, local estimation measure and value of k. Myung et al. [41] applied the k-NN to predict travel time with data from a vehicle detector system and an automatic toll collection system. Zou et al. [42] utilized both k-NN and a multi-topology neural network model in predicting freeway travel time. Their proposed model provides reliable travel time predictions in uncongested, congested, and transition traffic conditions.

2.2.2 Neural Network

Van lint et al. [43] applied a state space neural network that is capable of dealing with the spatio-temporal relationships of traffic. Yin et al. [44] developed a fuzzy-neural model (FNM) to predict traffic flow in an urban network. Their model applied a gate network (GN) that divides input data into several clusters using a fuzzy approach, and then applied the expert network (EN) to specify the input-output relationship. Ishak et al. [45, 46] compared three different neural networks: simple recurrent networks (Jordan–Elman), partial recurrent networks (PRNs), and time-lagged feed forward networks (TLFN), with different input parameters for traffic prediction. Jiang and Adeli [47] proposed the dynamic time-delay wavelet neural network model in freeway traffic flow forecasting. The proposed model considers both the time of the day and the day of the week when predicting traffic flow. Liu et al. [48] applied the state-space neural network model to predict travel time when there

is missing data. Their proposed method is insensitive to missing data and can provide accurate forecasting. Quek et al. [49] applied a specific class of fuzzy neural network models in short term traffic flow prediction. The proposed model known as a pseudo outer-product fuzzy neural network using the truth-value-restriction method (POPFNN-TVR) was shown to outperform the conventional feed forward neural network using back propagation (BP) learning.

Zheng et al. [50] suggested that a certain model has superior performance for a particular time period and combining single neural network predictors may improve forecasting accuracy. They developed a Bayesian combined neural network model that combines the back propagation and the radial basis function neural networks in traffic flow forecasting. The credit of each individual model is estimated based on the theory of conditional probability and Bayes' rule and largely depends on the accumulative prediction performance of previous time step. Zeng & Zhang [51] applied four different neural network models and proposed one model in freeway travel time forecasting. These models include multilayer feed forward neural network, time-delay neural network, state-space neural network, and nonlinear autoregressive with exogenous inputs. In their study, they analyzed the effect of different input variables and the temporal-spatial inputs on model performance. Their study results indicate that the temporal-spatial inputs can greatly improve the model performance; the state-space neural network and the time delayed state-space neural network outperform the other models.

2.2.3 Other Artificial Intelligence Methods

Sun et al. [52] considered spatial and temporal correlations of traffic flow among adjacent road links and developed a Bayesian network approach to forecast traffic flow. In their paper, the joint probability distribution between the upstream and downstream locations is described as a Gaussian mixture model (GMM) with parameters estimated via the competitive expectation maximization (CEM) algorithm. Hong [53] proposed a traffic flow forecasting model that combines the seasonal support vector regression model with chaotic simulated annealing algorithm (SSVRCSA). In his paper, the chaotic simulated annealing algorithm is proposed to determine the value of three parameters in a SVR model; seasonal adjustment factors are then applied to deal with the cyclic trend.

2.3 Hybrid Approaches

2.3.1 Classification Based Approach

The classification-based hybrid approach classifies traffic data into different groups first and then assigns different models according to the characteristics of data in different groups. Danech-Pajouh and Aron [54] proposed an ATHENA model that groups data according to their similarities and then assigns a different linear model to each cluster. Antoniou et al. [55] developed a dynamic data-driven framework for traffic state estimation and prediction. Their model first clusters existing observations into several groups, and then predicts the future traffic state by modeling the evolution of traffic history states as a Markov process. By estimating a flexible model for each cluster, future traffic speed can be obtained. Van Der Voort et al. [56]

combined Kohonen maps with ARIMA time series. The study results were promising compared with the ATHENA model. Later, Chen et al. [57] used a self-organizing map (SOM) to initially classify traffic data into different groups and then applied the ARIMA and the multi-layer perception (MLP) model as two prediction methods. Their study results suggested that the SOM/ARIMA hybrid approach is more sensitive to missing data than the SOM/MLP hybrid approach.

2.3.2 Kalman Filtering Based Approach

The Kalman filtering method is a promising method to train and update model parameters and has been applied to different kinds of forecasting models to enable continuous parameter updating. Yu et al. [58] applied the support vector machine (SVM) method to predict baseline travel time and used the Kalman filtering technique to adjust the prediction results with updated information. Stathopoulos and Dimitriou [59] proposed a forecasting approach that utilizes a fuzzy rule based system (FRBS) that nonlinearly combines traffic flow forecasting results from an online adaptive Kalman filter (KF) and an artificial neural network (ANN) model. Their study results indicate that the combined approach improves the forecasting accuracy compared with each individual model. Liu et al. [60] proposed the extended Kalman filter (EKF) method to train the parameters of the state-space neural networks (SSNN). The proposed algorithm was tested in an urban network and the study results indicate that the proposed method is 20 times faster than the SSNNLM model with a slightly worse forecasting accuracy.

2.3.3 Decomposition Technique

Decomposition techniques decompose a complicated data set into small elements. In terms of prediction, decomposition techniques can be utilized to reduce noisy information in traffic data to improve their prediction performance [61]; they could also be used as the basis for combining different models [62]. Some popular decomposition techniques include: Fourier methods, discrete wavelet transform (DWT) and empirical mode decomposition (EMD).

Hamad et al. [63] suggested that more accurate prediction of speed data can be obtained through decomposing the time series into its basic components. They utilized the empirical model decomposition to filter out unimportant elements and applied a multilayer, feed forward neural network with BP to predict freeway travel speed. Later, Chen and Wu [64] applied empirical model decomposition and gray theory [65] in predicting bus travel time. Wei and Chen [66] forecasted metro passenger flow through empirical model decomposition and neural networks. They applied the EMD method to decompose traffic flow as several intrinsic mode function (IMF) components and selected the important information as input for back-propagation neural networks (BPNN). Their study results indicate that treating the important and non-important IMF as different inputs of the BPNN would improve the forecasting accuracy. Jiang and Adeli [67] proposed a hybrid wavelet packet-ACF method to analyze traffic flow time series and concluded that the discrete wavelet packet transform method de-noises the signal even more effectively than the conventional wavelet transform. Xie et al. [68] applied a discrete wavelet decomposition method to remove noise in original traffic data and utilized the

Kalman filter prediction model to the modified data to predict future traffic. Their study results indicate that removing noise in the original traffic data has the potential to improve the performance of a Kalman filter model in traffic volume forecasting. Daniel et al. [61] applied a wavelet based method to remove noise in the original traffic data and applied self-organizing neural networks as the prediction method. Wang and Shi [69] developed a hybrid traffic speed forecasting model based on support vector machine (SVM) regression theory. In their study, they constructed a new kernel function using a wavelet function to capture the non-stationary characteristics of the data and then used the Phase Space Reconstruction theory to identify the input space dimension. They assumed that the collected data are often accompanied with measurement errors; therefore they applied wavelet de-noising method to remove the noise in the traffic speed data.

2.3.4 Ensemble Trees

Leshem and Ritov [70] proposed a traffic flow prediction algorithm by combining Random Forests algorithm [7] into an Adaboost algorithm as a weak learner. The proposed algorithm is proved to be able to deal with missing data and is effective in predicting multiclass classification problems. Hamner [71] applied a random forest method in travel time prediction and their method is able to provide accurate travel time prediction. Wang [72] applied an ensemble bagging decision tree (ensemble BDT) to predict weather impact on airport capacity and demonstrated the superior performance of ensemble BDT compared with single SVM classifier. Ahmed and Abdel-Aty [73] utilized a stochastic gradient boosting method in identifying hazardous conditions based on traffic data collected from different

sensors. Their study results suggested that the proposed stochastic gradient boosting method has considerable advantages over classical statistical approaches. Similarly, Chung [74] applied boosted regression trees to study crash occurrence. Both studies utilized the boosting method to study classification problems.

2.3.5 Other Combination Approaches

Zheng et al. [50] proposed a freeway traffic flow prediction method that combined a back propagation neural network and a radial basis function neural network based on Bayesian model combination approach. The output of the proposed model is a weighted combination of the output of the two neural networks and the weight is estimated based on conditional probability and Bayes' rule. Hinsbergen et al. [75, 76] trained feed-forward neural networks and state-space neural networks using Bayesian inference theory. A simple average over all group members was used to combine neural networks into a group. Their study results indicate that the proposed framework provides a more accurate forecasting of both the mean and the prediction intervals. Zhang and Liu [77] predicted travel time index by utilizing six baseline individual predictors as basic combination components and combined them through four combined predictors including equal weights (EW), optimal weights (OW), minimum error (ME) and minimum variance (MV) methods. Here, travel time index is the ratio of average travel time and free flow travel time. Vlahogianni et al. [78] proposed a modular neural network prediction model that considers both spatial and temporal correlations of traffic data. In their proposed model, the spatial representation of traffic information collected from individual location is addressed through a system's modularity. Each module consists of a time delayed feed-forward

neural network (TDNN) that represents the time evolution of traffic for a corresponding location. The temporal optimization of the input windows in each TDNN is through genetic algorithms (GAs).

2.4 Prediction Interval Based Approaches

Although a wide range of approaches has been applied to the traffic prediction field and has shown promising predicting abilities, some of them have limited abilities to capture the uncertainty and variability of traffic, as they only provide a point estimate to represent future traffic conditions. Traffic condition is a complex phenomenon, as it is often affected by the interactions among different vehicles and exogenous factors such as incident, weather, demand, and roadway conditions. Small changes in current traffic conditions may greatly affect future travel time. For example, an incident during peak hour may result in extreme delays in the near future. Due to the highly dynamic nature of traffic, predicting travel time is often associated with uncertainty, especially during non-recurrent congestion when incident or bad weather occurs. A point estimate provides limited information regarding the uncertainty and unreliability of travel time. On the other hand, prediction intervals (PIs) have the potential to provide more reliable forecasting results by providing a confidence band to indicate how reliable the forecasting results are. There are few studies using prediction intervals to model uncertainties associated with travel time prediction.

2.4.1 Ensemble Methods

Khosravi et al. [79] [80] developed different neural network based approaches to provide PIs to capture uncertainties in travel time. As there is always a mismatch between the predicted and actual values, PIs provide a range that can capture the uncertainty. Van Lint [81] proposed an ensemble of state-space neural network (SSNN) models to predict prediction intervals and mean travel time. The constructed prediction interval captures the uncertainty associated with travel time prediction. Zeng and Zhang [51] employed an ensemble method (using multiple instances of the same neural network model with different initial conditions) to derive a prediction band, but the method can be computationally intense. Fei et al. [82] proposed a Bayesian inference-based dynamic linear model that considers freeway travel time as the sum of the median of historical travel times, time varying random variations in travel time, and a model evolution error. Their proposed model prediction result is a travel time distribution that can generate a mean and a prediction interval representing uncertainty associated with travel time prediction. Van Hinsbergen [76] proposed an approach that combines neural networks in a group using Bayesian inference theory to predict travel time with prediction intervals. Li and Rose [83] developed a model that models average travel time and travel time variability separately to incorporate uncertainty in travel time prediction. All these studies indicate that travel prediction is a complex problem often associated with uncertainty. Prediction interval based approaches provide the potential to capture the dynamic changes of traffic.

2.4.2 Statistical Volatility Based Approach

Besides the ensemble approaches, another popular method that was able to capture the uncertainty and variations of data is the statistical volatility approach[84]. This approach relaxes the constant variance assumption and models time changing variance as a function of its past values. As a result, the statistical volatility modeling approach can capture the dynamic changes of travel time variations and can provide more accurate PIs. The first volatility model, the ARCH model, was proposed by Engle in 1982 for financial analysis purposes [84]. Later, different variations of the ARCH model were formulated and the GARCH [85] model is one of the most widely used models. Driven by its successful applications in financial and other areas [86], transportation professionals began to apply the family of GARCH models to predict traffic volatilities. Kamarianakis et al. [87] suggested that traffic conditions are much more volatile during heavy traffic or congestion periods than at other times and effective modeling of variance can produce more accurate confidence intervals. They tested the performance of the ARIMA-GARCH model by using traffic flow data in an urban network and demonstrated that traffic flow data displayed time dependent volatilities. They also suggested that further studies should consider the asymmetric effects of positive and negative shocks.

Zhang et al. [88] considered the asymmetric effects of positive and negative shocks and studied two asymmetric GARCH models: EGARCH and GJR-GARCH in travel time forecasting. Their study result indicated that the GJR-GARCH model performs better. Tsekeris and Stathopoulos [89] incorporated fractionally integrated components in both the conditional mean and the conditional variance equations and

proposed the ARFIMA-FIAPARCH model. They found that the proposed model improves the accuracy of predicted volatility. Similarly, Karlaftis and Vlahogianni [26] suggested over-differentiation leads to over-inflated MA terms and applied the ARFIMA-FIGARCH model for traffic flow prediction. Tsekeris and Stathopoulos [90] predicted urban traffic variability through a stochastic volatility modeling approach. Their study results demonstrated that the stochastic volatility model outperforms the GARCH model as a latent stochastic process and can better represent the speed variability dynamics than a stochastic process with a predetermined structure concerning the decaying impact of shocks. Yang et al. [3] applied seasonal ARIMA, ANN and historical mean methods as the mean equation and the GARCH model to predict urban vehicle travel time. Their study results indicate that the proper selection of the mean equation can lead to excellent results. Xia et al. [91] considered the relationship between flow and speed and proposed a VAR-MGARCH model to predict traffic flow and speed for urban roads. Guo [92] proposed an online algorithm of the autoregressive moving average (ARMA)-GARCH model through Kalman filters in predicting traffic speed.

2.5 Summary

In summary, traffic prediction algorithms can be categorized into parametric, non-parametric and hybrid approaches. The parametric approaches usually have a clear model structure and well established theoretical foundations. Compared with nonparametric approaches, they are easier to interpret. Some of them, for example, the historical average, have been implemented in existing traffic control devices for providing traffic information. However, this type of method sets a series of strict

model assumptions. Misusing the model with the wrong data (data that does not meet certain model assumptions) will lead to inaccurate prediction. As traffic in different locations shows different characteristics, it is necessary to understand both the model and the data to select the appropriate model.

The non-parametric approach usually has fewer modeling assumptions compared with the parametric approach. Some popular methods include non-parametric regression, neural networks, SVM and other artificial intelligent methods. The structure of this type of model is usually developed based on the data. Especially, the neural network methods are analogous to a 'black-box'. The users give some simple inputs to the model and get decent predictions, but they usually are unaware of the structure of the model. During the past few decades, the non-parametric methods attracted significant attention in the traffic prediction field because of their ability in modeling complex data.

The hybrid methods consider traffic as a complex phenomenon that cannot be represented by a single model. By taking advantage of different models, it is aimed at improving model prediction performance. However, because it involves several different models, the hybrid methods are often complex.

Besides prediction accuracy, it is also critical to develop a reliable prediction model that considers uncertainty associated with travel time prediction. Therefore, the prediction interval based approach is proposed to take into consideration this uncertainty. There are generally two categories: ensemble and statistical volatility methods. The ensemble method constructs the prediction interval through developing

different base models while the statistical volatility method models the evolution of the changing behavior of the variance part of traffic data.

Although a large number of traffic prediction algorithms are proposed in the literature, prediction accuracy and reliability are still two challenging issues. In terms of prediction accuracy, most existing models can be highly accurate during non-peak hours as variations of travel time are not significant from day to day. However, prediction accuracy deteriorates during peak hour. Improving travel time prediction, especially during congested periods, is critical. To address this issue, it is important to develop an advanced prediction method that is able to model the complex relation of traffic data. As mentioned in the previous section, the ensemble methods have shown promising prediction results. This research studies and proposes a novel ensemble method for predicting travel time. At the same time, reliability is also an essential issue in travel time prediction. Since there is always a mismatch between the predicted and the actual value, there is a need to measure or predict this “mismatch” or uncertainty. Prediction interval based approaches are able to model this uncertainty. However, it is a relatively new concept and there is limited literature in this field. This research will further explore more advanced prediction interval based approaches to better model this uncertainty.

Chapter 3: Statistical Volatility Models for Reliable Travel Time Prediction

The observed travel time can be decomposed into a conditional mean (u_t) and a residual (r_t) component. The traditional time series based travel time prediction methods study correlations between travel time at different time lags or at different locations, assuming constant variance of the data (r_t is constant across different time intervals). Therefore, they only model the time variation of the data for the first order moment and predict the mean part u_t . However, uncertainty often exists in the data, especially for travel time, which can be dramatically affected by some unexpected external factors, such as bottlenecks, traffic incidents, work zones, weather and special events. The point prediction results become less reliable because of the presence of these unexpected factors. As travelers are used to the daily congestion due to regular traffic demand, it is the unexpected delays that generate the most dissatisfaction. Modeling the uncertainties, referred to as the conditional standard deviation (or residual r_t), would improve forecasting reliability.

Equation (1) is the basic structure of a travel time prediction model.

$$x_t = u_t + r_t \quad (1)$$

where x_t is the observed travel time at time t , u_t represents the estimated conditional mean, and r_t is the residual part.

Traditional prediction methods only focus on the estimated conditional mean (u_t) component and treat the residual (r_t) part as having a constant variance. However, in real situations, the variations in traffic and travel times can be different during different time periods. Therefore, prediction interval based approaches are

proposed to model this uncertainty (the residual part r_t). By providing a prediction interval, we have an idea of how likely this estimated range would capture future travel time. In other words, a prediction interval is an estimated range that captures the future observation, with a prescribed probability, given the current available observations. As illustrated in Figure 1, a prediction interval is comprised of an upper and lower prediction limit that indicates the accuracy of the model output with respect to the observed value. The traffic is a complex and uncertain system. Due to the uncertainty related with the data and the estimated model, there is often a mismatch between the model output and observed value. Models that only provide a point value (the predicted value) have limited abilities to capture the uncertainty and variability of travel time especially during congested situations. PIs provide a range to indicate how likely the travel time is during the next time interval. Therefore, PIs have the potential to capture the fluctuations and the stochastic traffic phenomena. Usually, a wider prediction interval is associated with larger variation in travel time.

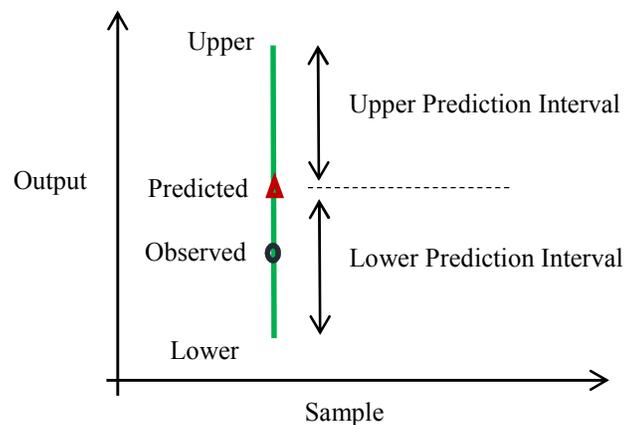


Figure 1. Concept of prediction interval [93]

The statistical volatility-based travel time prediction models include two parts: predicted mean value (the red triangle in Figure 1) and prediction interval (the green vertical lines). This section will first study efficient mean prediction models for short-term travel time prediction purposes. The second part studies the application of statistical volatility model in predicting the variance part of the travel time data, and the construction of travel time prediction intervals to account for uncertainties associated with prediction.

3.1 Mean Prediction Models

The first step of the modeling stage was to estimate the mean of the data. In the literature of traffic parameters forecasting, different mean equation models have been tested. For example, Kamarianakis et al. [87] applied the ARIMA model as the mean equation of the volatility model. Karlaftis and Vlahogianni [26] proposed using the ARFIMA model to capture the long memory in the conditional mean. Yang et al. [3] adopted three different mean equations for volatility models: SARIMA, ANN, and historical average methods. Among these existing methods that have been proposed in the literature, the ARIMA-type model becomes one of the most widely used methods due to its ease of implementation and its well-known ability in traffic parameters modeling and forecasting [26]. Therefore, for the purpose of studying the performance of different volatility models, this section applies the ARIMA model as the mean equation. However, it is worth noting that a proper mean equation model should not be restricted to the ARMA-type model. Properly choosing a mean equation regarding the structure of the data can lead to better model performance [3].

3.1.1 Theoretical Background of ARIMA Models

This section introduces the ARIMA model to capture the mean of this variation. ARIMA models, as one of the most general classes of time series models, predict the current value based on its past values. The *ARIMA* (p, d, q) model is comprised of three parts: autoregressive *AR* (p) , integrated *I* (d) , and moving average *MA* (q) . If we define B as the backshift operator with $B^k x_t = x_{t-k}$, The corresponding part of the ARIMA model for a given time series $\{x_1, x_2 \dots x_n\}$ consists of the following parts:

The autoregressive part of order p is denoted as *AR* (p) and is formulated as:

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + z_t \quad (2)$$

where $\phi_1 \dots \phi_p$ are parameters, z_t is a white noise process with zero mean and variance σ_z^2 . The equation can be rewritten in the following form by using the backshift operator:

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) x_t = z_t \quad (3)$$

Or more concisely as:

$$\phi(B) x_t = z_t \quad (4)$$

where $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$.

As many time series are non-stationary, it is necessary to transform the original data to a stationary series. There are several different ways to transfer the data to stationary, such as difference the data, remove the trend of the data (if the data contain a trend), taking the logarithm or square root of the series for data with non-constant variance. In travel time prediction, differencing the original data is most often used technique to achieve stationary. Here differencing means taking the

difference of two observations that are d periods apart. The integrated part with order d , denoted as $I(d)$, means the d th difference of the original data:

$$(1 - B)^d x_t \quad (5)$$

The moving average part with order q , denoted as $MA(q)$, is the process that the current value of x_t is a linear combination of a white noise series:

$$x_t = z_t + \theta_1 z_{t-1} + \theta_2 z_{t-2} + \dots + \theta_q z_{t-q} \quad (6)$$

where $\theta_1 \dots \theta_q$ are parameters, $z_t \dots z_{t-q}$ are white noise processes with zero mean and variance σ_z^2 . Equation (6) can be rewritten in a more concise form by using the backshift operator:

$$x_t = (1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q) z_t \quad (7)$$

$$x_t = \theta(B) z_t \quad (8)$$

where $\theta(B) = (1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q)$.

The *ARIMA* (p, d, q) model is a generalized version of the autoregressive integrated moving average process with p as the number of autoregressive terms, d as the number of differences, and q as the number of lagged forecast errors:

$$\begin{aligned} (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)(1 - B)^d x_t \\ = (1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q) z_t \end{aligned} \quad (9)$$

This can be written in a concise form as:

$$\phi(B)(1 - B)^d x_t = \theta(B) z_t \quad (10)$$

$$\{z_t\} \sim \text{WN}(0, \sigma^2)$$

Readers who are interested in theoretical foundations of the ARIMA model could refer to the book "Time series analysis and its applications: with R examples" [94] for details.

3.1.2 ARIMA Model Optimization

Order selection and parameter estimation are two major steps for the ARIMA model forecasting. Selecting a proper order of the ARIMA model is important for producing accurate forecasting results. From a prediction point of view, it is not a wise choice to select p and q arbitrarily large. Large p or q values potentially lead to over-fitting issues. To avoid over-fitting problems, a penalty factor is introduced to discourage a model with too many parameters. Some widely used criteria for model selection are the Akaike's information criterion (AIC), the corrected Akaike information criterion, (AICc) and the Bayesian information criterion (BIC). The preferred model is the one with the minimum value of one for these criteria. This study selects the order of an appropriate ARIMA model based on the Akaike's information criterion (AIC). The best model should be the one that has the smallest value of AIC:

$$\text{AIC} = -2 \log(L) + 2m \quad (11)$$

where L is the likelihood of the data for the specific model and m is the number of parameters selected for this model.

This research utilizes the method proposed by Hyndman and Khandakar [95] to select the orders of the appropriate ARIMA model automatically.

After determining the best orders of the ARIMA model, the parameters of the model are estimated through the maximum likelihood method. Detailed information on theoretical background and steps in fitting an ARIMA model can be found in [96].

3.1.3 Model Evaluation Criteria

There are several well established methods to evaluate model performance of the mean equation model. This research applies two measures of effectiveness to test the mean part: the root mean squared error (RMSE) and the mean absolute percentage error (MAPE). The RMSE is a frequently used measure of the difference between values predicted by a model and the actual observation. It is measured in the same unit as the original data. The MAPE is another commonly used measure of effectiveness. Different from the RMSE measures, the MAPE is expressed in percentage terms. Therefore, it provides us a general sense of the error even without knowledge of what constitutes a “big” error for the data set.

The equations for the RMSE and MAPE are as follows:

$$\text{RMSE} = \left(\sum_{i=1}^n \frac{(t(i) - a(i))^2}{n} \right)^{1/2} \quad (12)$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{t(i) - a(i)}{t(i)} \right| \quad (13)$$

where $t(i)$ is the actual value, $a(i)$ is the forecast value, and n is the total number of time periods.

3.2 Volatility Models

As mentioned in Equation (1), the structure of the proposed method is the sum of the mean and the variance. Most traditional models only concentrate on the mean part and assume that the variance part simply satisfies the white noise properties. Volatility models relax the assumption and characterize the changing variance of the data through time. Volatility models aim at specifying how the conditional variance r_t evolves over time. Different ways to address the variance part r_t lead to different

kinds of volatility models. In general, there are two different categories: the GARCH-type model and the stochastic volatility model.

3.2.1 GARCH-type Models

The GARCH-type models aim at capturing the changes of the variance part r_t . The first volatility model was proposed by Engle [84] in 1982, termed as the Autoregressive Conditional Heteroskedasticity (ARCH) model. This model was originally used to capture the uncertainty of financial data. This type of uncertainty refers to the variances and covariance that change over time. In his research, the discrete time stochastic process r_t is expressed as [97]:

$$r_t = \sigma_t \epsilon_t \quad (14)$$

$$\sigma_t^2 = \text{Var}(r_t | F_{t-1}) = \text{Var}(x_t | F_{t-1}) \quad (15)$$

Here ϵ_t is an independent and identically distributed (i.i.d) process with zero mean and standard deviation one and F_{t-1} denotes information available through $t - 1$. The above equation forms the foundation of the volatility model. Its various extensions are all based on this equation. Different ways of modeling σ_t lead to a wide variety of volatility models. Engle suggested in his paper that σ_t^2 can be a linear function of past squared values of the r_t process:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^m \alpha_i r_{t-i}^2 \quad (16)$$

Note that α_0 is the intercept term with $\alpha_0 > 0$, α_i represents the unknown coefficient of r_{t-i}^2 that satisfy $\alpha_i \geq 0$ to ensure the conditional variance as positive, and m denotes the number of lags selected for the model. This structure clearly captures the cluster of volatilities. The current magnitude of r_t is based on past values of r_{t-i}^2 .

Therefore, a sudden large change in the data would more likely lead to another large change. In other words, the probability of obtaining a large variance is greater than that of obtaining a small variance if the past value of the variance is large. This structure also works in traffic forecasting. For example, congestions would create unexpected delays, which lead to dramatic increase in travel time. This phenomenon would usually last for considerable time periods. Although the ARCH process has been proven useful in modeling the uncertainty in data, it often requires a relatively long lag. In order to allow both a longer memory and a more flexible lag structure, a generalized version of the ARCH model – Generalized Autoregressive Conditional Heteroskedasticity (GARCH) – was proposed by Bollerslev [85], which included the past value of σ_t^2 in the model structure:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^m \alpha_i r_{t-i}^2 + \sum_{j=1}^s \beta_j \sigma_{t-j}^2 \quad (17)$$

in which $\alpha_0 > 0$, $\alpha_i \geq 0$, $i = 1, \dots, m$, $\beta_j \geq 0$, $j = 1, \dots, s$, $\sum_{i=1}^{\max(m,s)} (\alpha_i + \beta_i) < 1$.

The appealing feature of the GARCH model is that it takes into account that σ_t^2 not only depends on past values of the variance part r_{t-1} , but also on its own past values σ_{t-j}^2 . The GARCH model can be interpreted as an ARMA process. Applications of the GARCH model in many different fields demonstrate its ability in modeling uncertainty in data. Lots of variations of the volatility model inherit this unique structure. One successful type of variant is termed as component GARCH model.

3.2.2 Component GARCH Models

Component GARCH models aim at capturing the trend and seasonal (cyclical) components in data. In the traffic prediction field, no research has yet been conducted

to study the seasonality and trend in data through the component GARCH model. One innovation of this research is introducing the component GARCH model in travel time reliability prediction. Two different component GARCH models are proposed in this research: the component GARCH (C-GARCH) and the multiplicative component GARCH (MC-GARCH) models.

As trend and seasonal components are often observed in data, it was argued that the conventional GARCH model is unable to provide adequate performance [98] if trend or seasonal components exist. Several variations of the GARCH model were developed to take into account the trend and seasonal volatility patterns. We term these models as component models that decompose the data as trend, seasonal, and random elements. The trend component represents long term changes in the level of the data series while the seasonal factor is the periodic fluctuations within the data series. Two different structures can be considered as basic component models:

The additive model, where

$$x_t = s_t + t_t + e_t \quad (18)$$

The multiplicative model, where

$$x_t = s_t \times t_t \times e_t \quad (19)$$

where s_t represents the seasonal effect, t_t represents the trend, and e_t represents the errors.

The additive model is applied to data series where the magnitude of the seasonal fluctuation does not change regardless of the level of the data series. The multiplicative model applies to situations in which the seasonal variation increases/decreases with the level of the series.

Based on the additive model structure, the component GARCH (C-GARCH) model proposed by Engle and Lee [99] decomposes the conditional variance into a long term and a transitory component. The equation of the component GARCH model is as follows.

$$\sigma_t^2 = q_t + \sum_{i=1}^m \alpha_i (r_{t-i}^2 - q_{t-i}) + \sum_{j=1}^s \beta_j (\sigma_{t-j}^2 - q_{t-j}) \quad (20)$$

$$q_t = \alpha_0 + \rho q_{t-1} + \varphi (r_{t-1}^2 - \sigma_{t-1}^2) \quad (21)$$

where α_i , β_j , ρ , φ are unknown parameters, as is α_0 .

In this model, the intercept term q_t is regarded as a time-varying process, which represents the long term component of the conditional variance. The difference of the conditional variance and the long term component $\sigma_{t-j}^2 - q_{t-j}$ is the transitory component that models the short-term volatilities.

The multiplicative component GARCH model [5] assumes the variations increase with the level of data and decomposes the variance part into three multiplicative components: daily component d_t , deterministic diurnal pattern s_i , and stochastic intraday component $q_{t,i}$.

$$x_{t,i} = u_{t,i} + r_{t,i} \quad (22)$$

$$r_{t,i} = \sqrt{d_t s_i q_{t,i}} \varepsilon_{t,i} \quad (23)$$

where the travel time at time index i in day t consists of the conditional mean $u_{t,i}$ and variance $r_{t,i}$ equations. $\varepsilon_{t,i}$ is the i.i.d (0,1) standardized innovation which can follow a normal, a student-t distribution, etc. The daily part d_t models the variance of the data across different days. It can be derived from a multifactor risk model, a daily GARCH model or a multiple indicators model [5]. The deterministic diurnal component for each time index is estimated as:

$$\hat{s}_i = \frac{1}{T} \sum_1^T \frac{r_{t,i}^2}{d_t} \quad (24)$$

where T is the total number of days, t denotes the day, i denotes the regular spaced time intervals.

As indicated in the above equation, the diurnal component at time index i is the average of variance at time index i scaled by its corresponding variance for each day. Therefore, the diurnal component represents the regular intraday variations.

After estimating the daily and deterministic intraday components, the rest component in the variance part is regarded as stochastic, and can be regarded as a GARCH (p,q) process. The normalized residual is:

$$z_{t,i} = r_{t,i} / \sqrt{d_t s_i} = \sqrt{q_{t,i}} \varepsilon_{t,i} \quad (25)$$

where the stochastic intraday component $q_{t,i}$ is assumed to follow the GARCH process:

$$q_{t,i} = \alpha_0 + \sum_{j=1}^p \alpha_j z_{t,i-j}^2 + \sum_{j=1}^q \beta_j q_{t,i-j} \quad (26)$$

From the perspective of travel time prediction, travel time exhibits both regular cyclical patterns (seasonal component) and stochastic patterns. Daily cyclical patterns distinguish travel time as peak hour and non-peak hour traffic. Stochastic patterns are the results of unexpected influential events, like bad weather conditions and traffic incidents. Capturing the time-varying features of traffic behavior is critical for travel time forecasting. In addition, decomposing data into cyclical and stochastic patterns provides a better understanding of the underlying structure of the data. The component GARCH model, a generalization of the traditional GARCH model,

considers the seasonal and trend components in the variance part through decomposition.

3.2.2 Stochastic Volatility Model

Most of the existing studies focus on the GARCH-type model. There are limited studies applying a stochastic volatility type model in traffic prediction. Part of the reason is that the estimating process for the stochastic volatility model is much more complex compared with the GARCH-type model since a new parameter estimation method is introduced to improve the estimating performance of the stochastic volatility model.

The conditional volatility of the GARCH model in Equation (14) is a deterministic function of past quantities. Provided that all relevant information is available, the model could be specified at the present time period. The stochastic volatility model is a competitive alternative to the GARCH-type model by modeling volatilities non-deterministically, which treats the volatility as a random process and evolves stochastically over time. Since traffic involves interactions among different factors such as demand, incident, and weather, this often makes future traffic conditions uncertain. Modeling the conditional variance as an unobserved stochastic process allows for a more flexible applications of the SV model and can account for the uncertainty inherent in traffic phenomena [90].

Based on the canonical model [100] of the stochastic volatility class, the volatility part r_t of Equation (1) can be expressed as follows:

$$r_t = \exp(h_t/2)\varepsilon_t \quad (27)$$

$$h_t = \mu + \phi(h_{t-1} - \mu) + \sigma v_t \quad (28)$$

$$h_1 \sim N\left(\mu, \frac{\sigma^2}{1 - \phi^2}\right) \quad (29)$$

where r_t represents volatility of travel time during time interval t . ε_t is a Gaussian white noise sequence with mean 0 and variance 1, and v_t is also a Gaussian white noise sequence with mean 0 and variance 1 which is independent of ε_t . The unobserved process h_t is interpreted as a stochastic volatility process with parameters μ , ϕ and σ to be estimated. To setup the model, a prior distribution for parameters μ , ϕ and σ should be specified. According to [6], the level μ follows a normal distribution with mean m_μ and variance M_μ , or $\mu \sim N(m_\mu, M_\mu)$. To guarantee the persistence parameter $\phi \in (-1, 1)$, $(\phi + 1)/2$ follows Beta distribution with $(\phi + 1)/2 \sim B(a_0, b_0)$, where a_0 and b_0 are positive parameters; the volatility of log variance $\sigma^2 \sim B_\sigma \cdot \chi_1^2 = \text{gamma}(1/2, 1/2B_\sigma)$, where B_σ is a single positive value that stands for the scaling of the transformed parameter σ^2 , χ_1^2 denotes a chi-squared distribution with one degree of freedom. The posterior distributions of the desired variables are estimated through Bayesian inference via the Markov Chain Monte Carlo (MCMC) method.

It should be noticed that by taking logarithms of the squared r_t in Equation (27), the SV model is transformed into a linear equation:

$$\log(r_t^2) = h_t + \log(\varepsilon_t^2) \quad (30)$$

$$\varepsilon_t \sim N(0, 1)$$

If $\log(\varepsilon_t^2)$ is approximated by a mixture of a normal distribution with m_{r_t} and $s_{r_t}^2$ being the mean and the variance of the r_t th mixture component respectively [6], the above equation reduces to the form of a conditionally Gaussian state space model:

$$\hat{r}_t = m_{r_t} + h_t + \epsilon \quad (31)$$

where $\hat{r}_t = \log(r_t^2)$, and $\epsilon \sim N(0, s_{r_t}^2)$. This linearization makes efficient MCMC sampling possible.

Model reparameterization could potentially improve simulation efficiency in the volatility model. Denoting Equation (27) to (29) as the centered parameterization models, another version of the SV model is non-centered (NC) parameterization, where parameter μ is shifted from the state Equation (28) to the observation Equation (27) by setting $\hat{h}_t = (h_t - \mu)/\sigma$. The non-centered (NC) parameterization form is given as:

$$r_t \sim N(0, \exp(\mu + \sigma \hat{h}_t)) \quad (32)$$

$$\hat{h}_t = \phi \hat{h}_{t-1} + v_t \quad (33)$$

where $\hat{h}_t = (h_t - \mu)/\sigma$, v_t is a Gaussian white noise sequence with mean 0 and variance 1.

The choice of centered parameterization(C) or non-centered parameterization (NC) would dramatically affect the sampling efficiency. Both parameterizations have their advantages and disadvantages, which heavily depends on the true parameter values of the data generating process. There exists no ‘ultimate’ parameterization. Based on the ancillarity-sufficiency interweaving strategy (ASIS) introduced by Yu and Meng [101], Kastner and Fruhwirth-Schnatter [6] proposed a strategy by interweaving C and NC to overcome this deficiency. Their study results show that interweaving C and NC leads to a robustly efficient sampler that always outperform either parameterization (C or NC) with respect to parameter costs in terms of design

and computation. Their intuitive and efficient algorithm can be briefly summarized into six steps:

Choose appropriate starting values and repeat the following steps:

- (1) Draw h from parameterization C.
- (2) Draw μ, ϕ, σ from parameterization C.
- (3) Move to non-centered parameterization NC by the simple deterministic transformation $\hat{h}_t = \frac{h_t - \mu}{\sigma}$.
- (4) Redraw μ, ϕ, σ from parameterization NC.
- (5) Move back to C by calculating $h_t = \mu + \sigma \hat{h}_t$ for all t .
- (6) Draw the indicators from parameterization C.

The detailed sampling steps in the ASIS involve extensive Bayesian approach with MCMC. In the following, a brief summary of the key ideas on how to estimate the parameters of the SV model is introduced. Readers are referred to [102] for further details. The following briefly explains the concept of Bayesian approach with MCMC.

Consider a problem that given a set of data, the posterior distribution of $P(\theta|x)$ is desired in Bayesian analysis in order to estimate parameter θ . Bayesian analysis seeks to estimate parameter θ by combining the prior knowledge about the parameters with the data. Denote $P(\theta)$ as the specified prior distribution of θ . The posterior distribution $P(\theta|x)$ is calculated through the conditional probability:

$$P(\theta|x) = \frac{P(\theta, x)}{P(x)} = \frac{P(x|\theta)P(\theta)}{P(x)} \quad (34)$$

where $P(x|\theta)$ denotes the likelihood function of the data for a given model, $P(x)$ denotes the marginal distribution of x . According to Bayes's rule:

$$P(\theta|x) \propto P(x|\theta)P(\theta) \quad (35)$$

The goal is to estimate θ through the posterior distribution $P(\theta|x)$. The Markov Chain Monte Carlo (MCMC) approach draws a sample from the posterior distribution and then calculates the estimator of θ . The Markov process is a stochastic process $\{x_t\}$ that the value of x_h does not depend on the value of x_s if the value of x_t is given and $s < t < h$. $\{x_t\}$ is a Markov process if its conditional distribution function satisfies the following criterion:

$$P(x_h|x_s, s \leq t) = P(x_h|x_t), h > t. \quad (36)$$

The basic idea of MCMC method is to simulate a Markov chain that has the desired probability distribution which is $P(\theta|x)$. The ASIS algorithm estimates the model parameters in steps two and four through Bayesian inference using MCMC.

3.2.3 Prediction Interval Estimation

Prediction intervals for the volatility model are estimated based on the idea of prediction intervals for regression models. To construct a prediction interval with $100(1 - \alpha)\%$ confidence, we assume that the error follows Gaussian distribution with zero mean and variance σ_t^2 . The prediction interval can be calculated as:

$$(u_t - z_{\alpha/2}\sigma_t, u_t + z_{\alpha/2}\sigma_t) \quad (37)$$

where u_t is the predicted mean, $z_{\alpha/2}$ denotes the standard score corresponding to the cumulative probability level of $\alpha/2$, and σ_t is the prediction variance from a volatility model.

As the concept of uncertainty or reliability is a relatively new area in traffic forecasting, there are few studies that provide criteria for PI assessment. One study by Khosravi et al. [79] suggested that two important aspects of PIs assessment should

be considered: coverage probability and length. Coverage probability measures the percentage of the targets that lie within the predicted PIs. It measures how effective the constructed prediction intervals are. The mathematical representation of PI coverage probability (PICP) is as follows:

$$\text{PICP} = \frac{1}{n} \sum_{i=1}^n c_i \quad (38)$$

where $c_i = 1$ if $y_i \in [L(x_i), U(x_i)]$; otherwise, $c_i = 0$. $L(x_i)$ and $U(x_i)$ represent upper and lower bounds of the prediction interval of x_i , n is the total number of constructed PIs.

On the other hand, another criterion called mean PI length (MPIL) measures the average length of the PIs. It measures how efficient the constructed prediction intervals are. Assume we have two models that provide PIs with the same coverage probability; the one that gives a narrower prediction band is more efficient. The following equation gives the definition of the MPIL:

$$\text{MPIL} = \frac{1}{n} \sum_{i=1}^n (U(x_i) - L(x_i)) \quad (39)$$

Therefore, both criteria should be considered when evaluating the volatility models.

3.3 Application of Component GARCH Models in Travel Time Prediction

The performance of the GARCH, the C-GARCH, and the MC-GARCH models are investigated here by using data collected from Automatic Vehicle Identification (AVI) stations located along U.S. Highway 290 (or U.S. 290) in Houston, Texas. The entire study corridor is about five miles long and covers the Northwest Freeway in the westbound direction between I-610 and the junction of Farm to Market Road 1960 (FM1960). The IDs of the selected AVI stations are 29,

30, 31, 32, 33, and 34. The travel time between each pair of consecutive detectors were collected and aggregated into five-minute time intervals. Individual segment travel time at free-flow conditions is less than four minutes. The total period of the sample was the entire year of 2008 with missing data replaced by annual medians of the missing intervals. Since travel time patterns during weekdays and weekend are quite different (congestions are more likely to occur during weekdays), weekend data were excluded from the sample. As a result, 262 weekdays of travel time data that contain 75,456 five-minute observations are selected for this study.

3.3.1 Modeling Conditional Mean

Traffic data often shows periodic patterns. Travel time increases and varies significantly during peak hours compared with travel time during non-peak hours. It is difficult to precisely predict traffic when congestion occurs. Point prediction methods are often unable to capture traffic variation in congested situations, therefore providing less reliable or accurate prediction. As the performance of the point prediction methods often decreases when congestion occurs, it is expected that the residual series (after removing the predicted mean by ARIMA model) show higher variations during peak hours.

Figure 2 provides a boxplot of absolute deviation from the predicted mean for each 20-minute time interval (outliers have been removed from this plot). Each box statistically represents interval-to-interval and day-to-day variations of the residual series. The green line indicates the mean of the residual and the lower and upper boundary of each box is the 25th and 75th percentiles of the data for corresponding time intervals. As observed from this plot, statistics of each interval are different from

each other. Both the mean and the percentiles of the data are different at different time intervals. This indicates that the residual series vary over time, and the constant variance assumption of the traditional time series models is violated. This further proves that a volatility model, which relaxes the constant variance assumption, is necessary. In addition, there is a pronounced increased variation at the beginning of 15:00 hour; subsequently, the variation decreases at 19:00 hour. Comparatively, variations during other time periods (non-peak hours) are less significant.

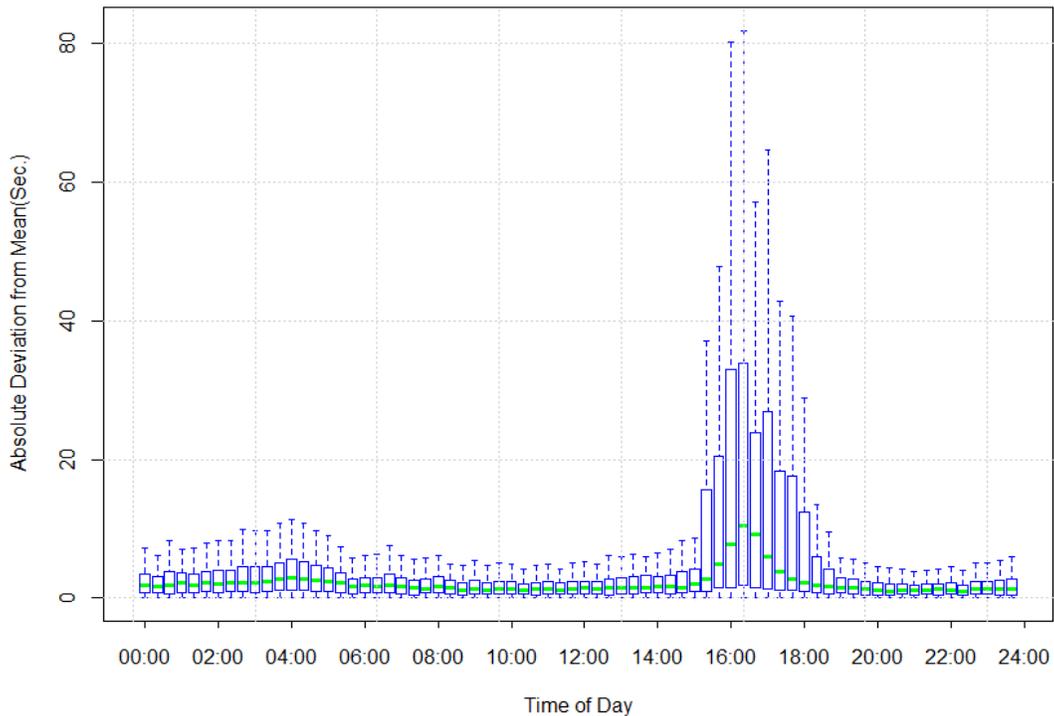


Figure 2. Box Plot of Absolute Deviation from Predicted Mean

The other four studied segments also depicted similar diurnal patterns. It is clear that seasonal components exist in the residual time series. In addition, the mean

of the absolute deviations during non-peak hours are close to zero, which means that the ARIMA model provides adequate prediction during non-peak hours. On the other hand, the prediction performance of the ARIMA model decreases during peak hours, as both the mean and the 75 percent statistics increase.

3.3.2 Testing the ARCH Effect

The basic assumption of the GARCH-type model is that square values of the residuals are correlated. Therefore, before applying different GARCH-type models to the data, there is a need to test if the data meet this assumption. Two tests are available: Ljung–Box statistics and Lagrange multiplier test [96]. This study chose the Ljung-Box statistics to test if the first lags of the squared residuals are uncorrelated. The Ljung-Box test is as follows:

$$H_0: \rho_1 = \rho_2 = \dots \rho_m = 0$$

$$Q(m) = N(N + 2) \sum_{h=1}^m \frac{\rho_h^2}{N-h} \quad (40)$$

where N is the number of data points under study, ρ_h is the sample autocorrelation at lag h, and m is the number of lags being tested.

The critical region for rejecting the null hypothesis at significance level α is:

$$Q > \chi_{1-\alpha, m}^2$$

In terms of p values, the null hypothesis will be rejected if the p value is less than α . In our study, the Ljung-Box test is applied to the residual data of all five segments. P values of the test for all studied segments are significantly less than 0.01. Therefore, the null hypothesis is rejected at the significance level of 0.01. That is to

say, correlations exist between squared values of residuals. The GARCH-type models are necessary.

3.3.3 Estimating the Volatility Model

Similar to the ARIMA type model, estimating the volatility model also involves order selection and parameter estimation. Several studies indicate that GARCH family model with order of (1, 1) was found adequate in representing the volatility dynamics [26, 86, 89]. Therefore, GARCH, C-GARCH and MC-GARCH with order of (1, 1) were adopted for ease of implementation and comparison by using the R package ‘rugarch’[103].

Since the multiplicative component GARCH model decomposes the data into a daily component, a deterministic diurnal pattern and a stochastic intraday component, the first step is to model the daily component. This study terms the average volatility for each day as the daily component. The daily component is estimated through the standard GARCH (1, 1) process. There are 262 daily data in total; the first 242 data points are used as the training data. After removing the daily component, the deterministic diurnal part is estimated as the annual average of the residual data at each time interval. The normalized residuals (22) are then used to produce the stochastic intraday component. The volatility components estimated by the MC-GARCH model are displayed in five panels as shown in Figure 3. The top panel shows the observed values of the residual data series. The second panel gives the estimated conditional variance, being the product of the following three components: deterministic intraday (panel three), daily (panel four), and stochastic intraday (panel five) components. As indicated in this figure, the MC-GARCH model

is able to model the trend, seasonal, and stochastic components of the data. This feature provides better understanding of the basic structure of the data and is easy to interpret. For example, the intraday [Deterministic] components indicate the regular cyclical patterns of travel time volatility, the intraday [Stochastic] components specify daily variations due to demand variation, incident or other abnormal traffic phenomenon.

3.3.4 Construct the Mean and Prediction Intervals

The final output of the proposed volatility models includes two measures: the predicted mean and the predicted PIs. The predicted mean part generally tells the expected value of travel time in the future, whereas the PIs tell how likely the observed value will lie within a certain range. In other words, wider PIs often indicate unreliable travel time and prediction. Thus, based on the combined information of predicted mean and PIs, travelers and operators would have a better sense of future traffic conditions. In this study, the ARIMA model provides mean values, and the prediction intervals are constructed according to Equation (37). Figure 4 plots some sample prediction results of the MC-GARCH model. The blue dot stands for observed travel-time data obtained at corresponding time intervals, and the red triangle stands for the predicted mean. The green lines represent the PIs constructed by the MC-GARCH model. It is obvious in this figure that there is always a mismatch between the predicted mean and the observed value. This partly results from the dynamic nature of traffic: travel time varies from time to time. Prediction intervals, on the other hand, are able to adequately capture this variation by covering most of

the observed values. Therefore, this model provides an effective and efficient way to measure uncertainty associated with future travel time.

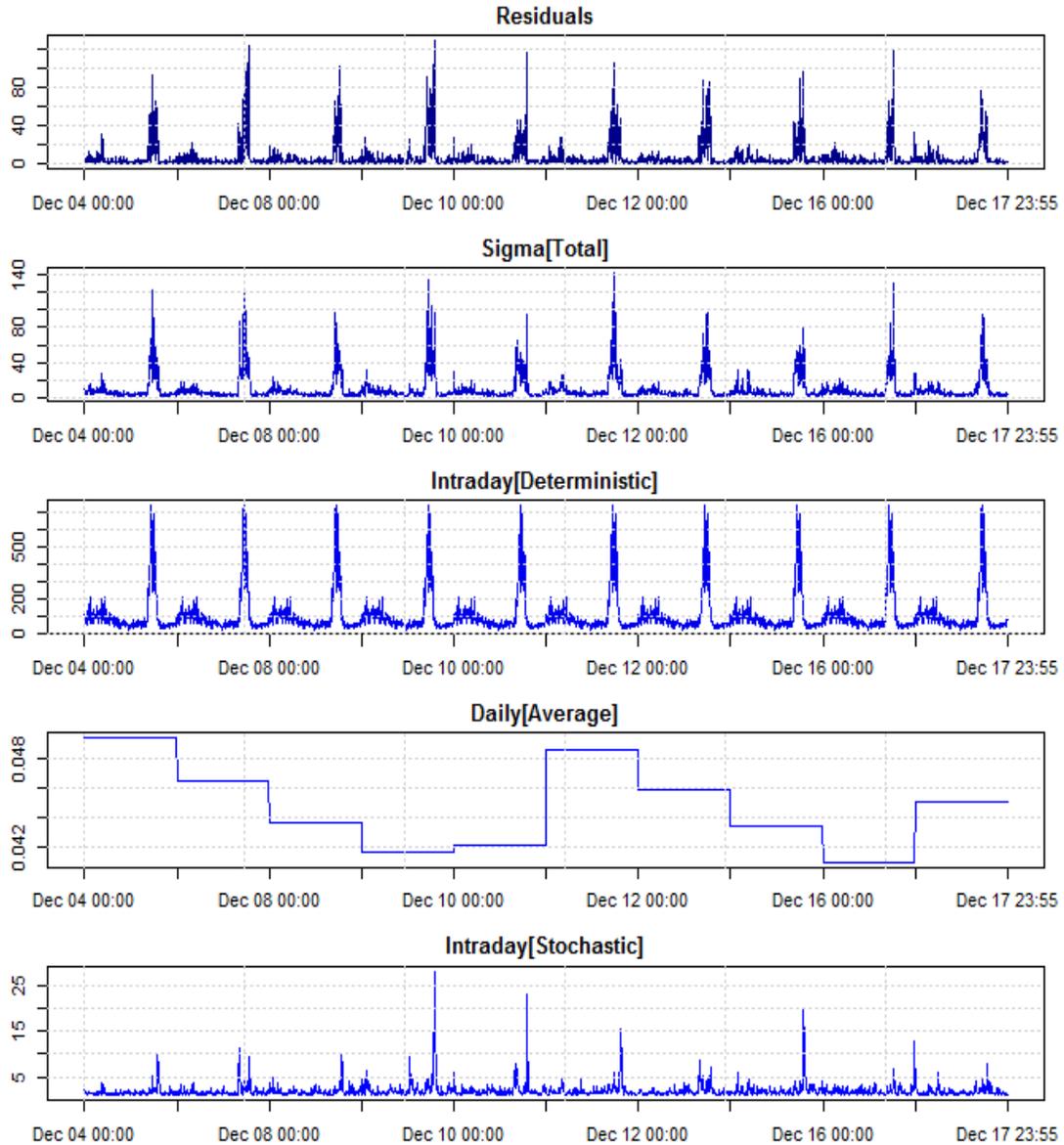


Figure 3. Multiplicative component GARCH forecasting results: decomposition of the volatility into its various components (32-33).

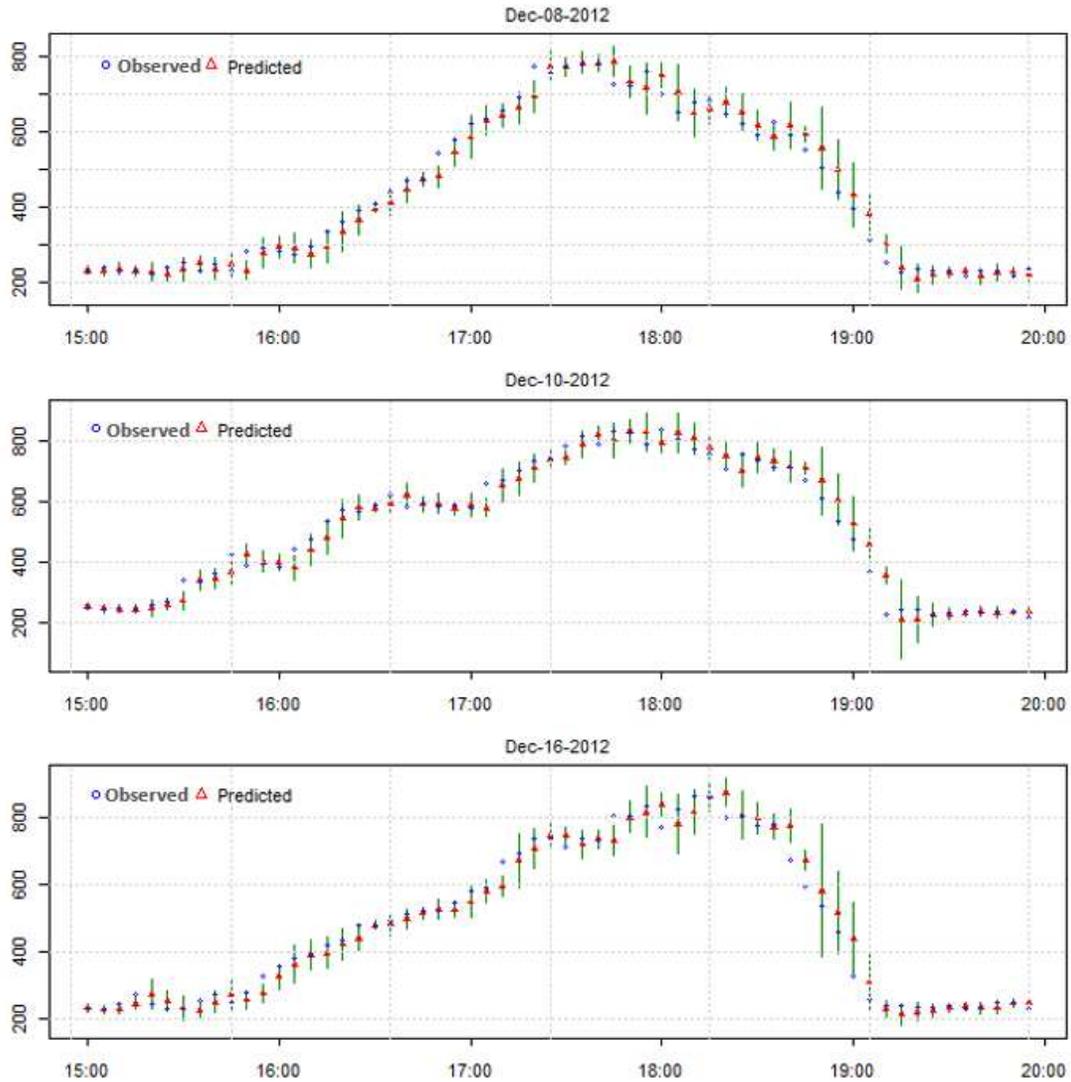


Figure 4 Predicted mean and PI for multiplicative component GARCH model

3.3.5 Results and Discussion

PIs with different confidence levels are constructed through the GARCH, the C-GARCH and the MC-GARCH models for the five studied segments. The effectiveness and efficiency of the constructed PIs is evaluated based on the criteria of coverage probability and PI length. For each of the five studied segments, thirty days of five-minute travel-time data with 8,640 observations are used as the training

data set and ten days travel time data with 2880 observations are used as the comparison (testing) dataset. We estimate individual models for each of the five segments. PIs are constructed with 95 percent, 90 percent, and 85 percent confidence levels, respectively.

Table 1 provides average MPIL and PICP values of the three models with different confidence levels during peak hours, nonpeak hours, and all day. During peak hours, the prediction interval coverage rates of the MC-GARCH are the highest compared to the C-GARCH and the GARCH model. For the 95 percent confidence level, the coverage rate of the MC-GARCH model (90.86 percent) is 2.86 percent higher than the second highest model (GARCH 88.00 percent). For the 90 percent confidence level, the coverage rate of the MC-GARCH model (87.35 percent) is 4.33 percent higher than the second highest model (GARCH 83.02 percent). For the 85 percent confidence level, the coverage rate of the MC-GARCH model (83.96 percent) is 5.18 percent higher than the second highest model (GARCH 78.78 percent). As the confidence level decreases, the advantage of the MC-GARCH model becomes obvious, proving that the MC-GARCH model is able to capture the volatility of traffic data during peak hours. In terms of mean prediction interval length, the C-GARCH is the smallest. However, on average, the C-GARCH model only reduced the length by 0.75 compared with the GARCH model.

During nonpeak hours, the MC-GARCH model also provides the highest coverage. But the advantage of MC-GARCH model is not obvious compared with the GARCH and the C-GARCH models in terms of either MPIL or PICP during non-peak hours. This is expected as travel time is relatively stable with small variations

and the trend and seasonal patterns are not obvious during this period. Therefore, performance of these three models should be similar during non-peak hours.

Investigating the all-day performance of these three models indicates that the MC-GARCH model provides the highest PICP value, whereas both the C-GARCH and the GARCH models give lower MPIL values compared with the MC-GARCH model.

Table 1 Estimated MPIL and PICP values for GARCH, C-GARCH and MC-GARCH models

Confidence Level	Model	Peak Hours		Nonpeak Hours		All Day	
		MPIL	PICP	MPIL	PICP	MPIL	PICP
95%	GARCH	115.98	88.00%	44.53	95.72%	56.69	94.41%
	C-GARCH	115.1	87.80%	45.13	95.10%	57.04	93.86%
	MC-GARCH	139.74	90.86%	45.09	95.87%	61.2	95.01%
90%	GARCH	97.04	83.02%	37.26	92.30%	47.43	90.72%
	C-GARCH	96.31	82.57%	37.77	91.42%	47.73	89.92%
	MC-GARCH	116.93	87.35%	37.73	93.08%	51.21	92.10%
85%	GARCH	85.21	78.61%	32.72	89.07%	41.65	87.29%
	C-GARCH	84.56	78.78%	33.16	87.79%	41.91	86.26%
	MC-GARCH	102.67	83.96%	33.13	90.47%	44.96	89.36%

In general, based on the estimation results, we can conclude that the MC-GARCH model tends to cover more targets compared with the C-GARCH and the GARCH model, especially during peak hours. The C-GARCH and GARCH models give a lower prediction band compared with MC-GARCH with the compromise of lower coverage rate. Since coverage rate of the C-GARCH and the GARCH model

are much lower than corresponding confidence level during peak hours (for example, PIs of both the GARCH and the C-GARCH model cover around 78 percent of the targets for 85 percent confidence level), the MC-GARCH model generates more effective PIs, although a little bit wider than others.

To check the consistency of each model's performance, Figure 5 and Figure 6 compare PICP values of the GARCH, the C-GARCH, and the MC-GARCH models at different confidence levels for individual segments. As shown in these figures, the orange, blue and green columns represent the GARCH, the C-GARCH and the MC-GARCH models; columns with different patterns represent different confidence levels. During peak hours (Figure 5), the MC-GARCH model generates PIs with the highest coverage for all segments. The advantage of the MC-GARCH model is significant with the highest difference of 6.94 percent compared with PIs (at the 85 percent confidence level) provided by the C-GARCH model for segment 33-34. On the other hand, the PICP values of the GARCH and the C-GARCH models are similar. The largest difference of PICP values between the GARCH and the C-GARCH is 1.63 percent. During non-peak hours (Figure 6), all three models provide high coverage rate at corresponding confidence levels. Differences among individual models during non-peak hours are not as significant as during peak hours. Comparing all three models' performance between peak and non-peak hours suggests that peak hour coverage is relatively low, as traffic variations increase during peak hours. In addition, PI lengths during peak hours are also longer than PI lengths during non-peak hours. This is because the uncertainties during peak hours are more evident compared with non-peak hours.

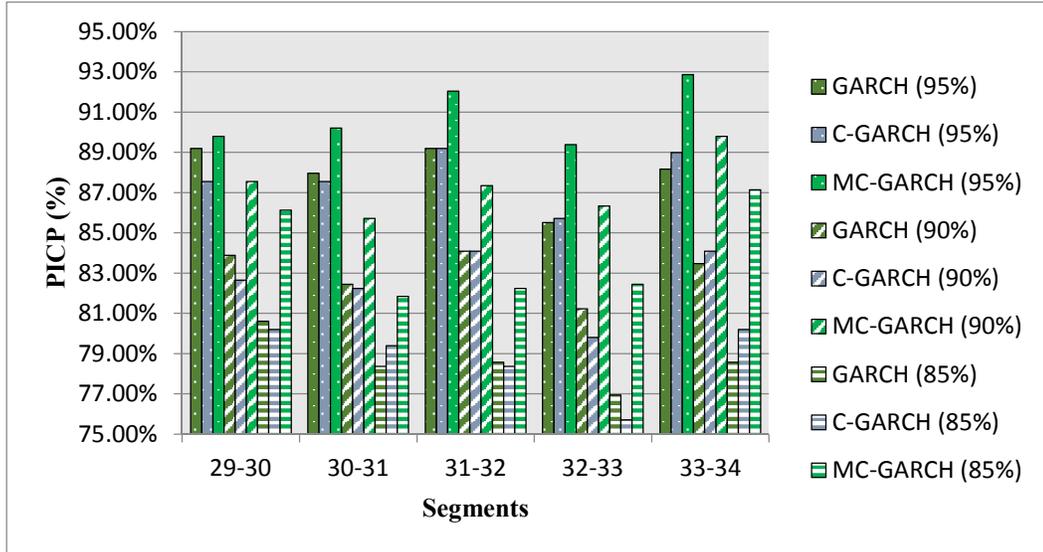


Figure 5. Comparing performance of GARCH, C-GARCH and MC-GARCH models during peak hours

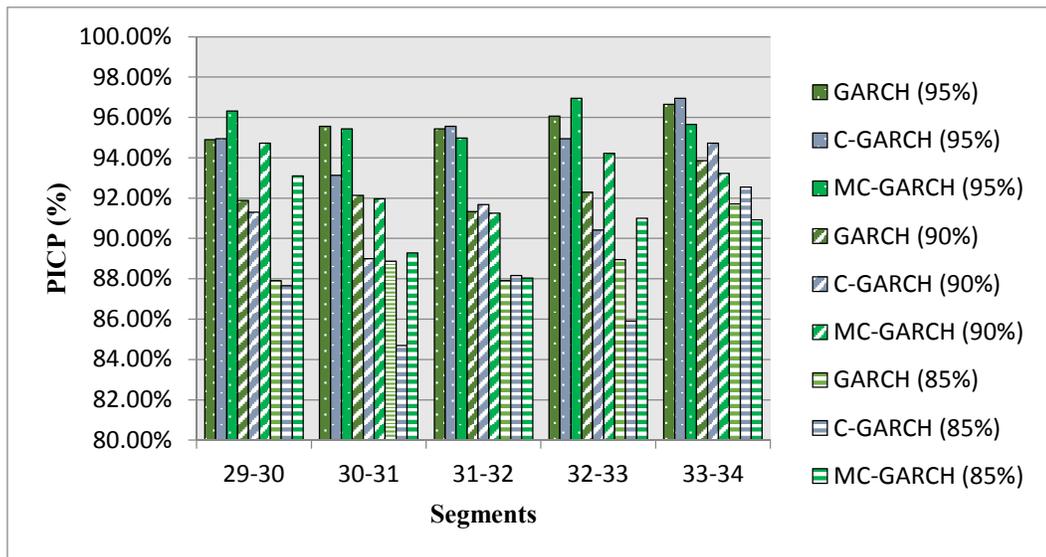


Figure 6. Comparing performance of GARCH, C-GARCH and MC-GARCH during non-peak hours

Figure 7 provides an intuitive comparison of one-day peak hour PIs at the 95 percent confidence level constructed by the MC-GARCH, the C-GARCH and the GARCH models. The green dash lines stand for PIs of the MC-GARCH model, the

yellow dash lines stand for PIs of the C-GARCH model, and the pink dash lines stand for PIs of the GARCH model. As shown in this figure, the PIs constructed by the C-GARCH and the GARCH almost overlap. It has also been depicted in Table 1 that there is no significant difference between MPIL and PICP values of the C-GARCH and the GARCH models. It seems the effect of the long-term component in the C-GARCH model is limited in this case. On the other hand, PIs of the MC-GARCH model are different from both the C-GARCH and the GARCH models. The MC-GARCH model tends to cover more targets by increasing the width of the prediction intervals at certain time intervals (Points identified by blue arrows). Overall, the C-GARCH and the GARCH models create PIs similar to each other. Compared with the C-GARCH and the GARCH models, the MC-GARCH model tends to cover more targets by increasing the length of its PIs during certain time intervals.

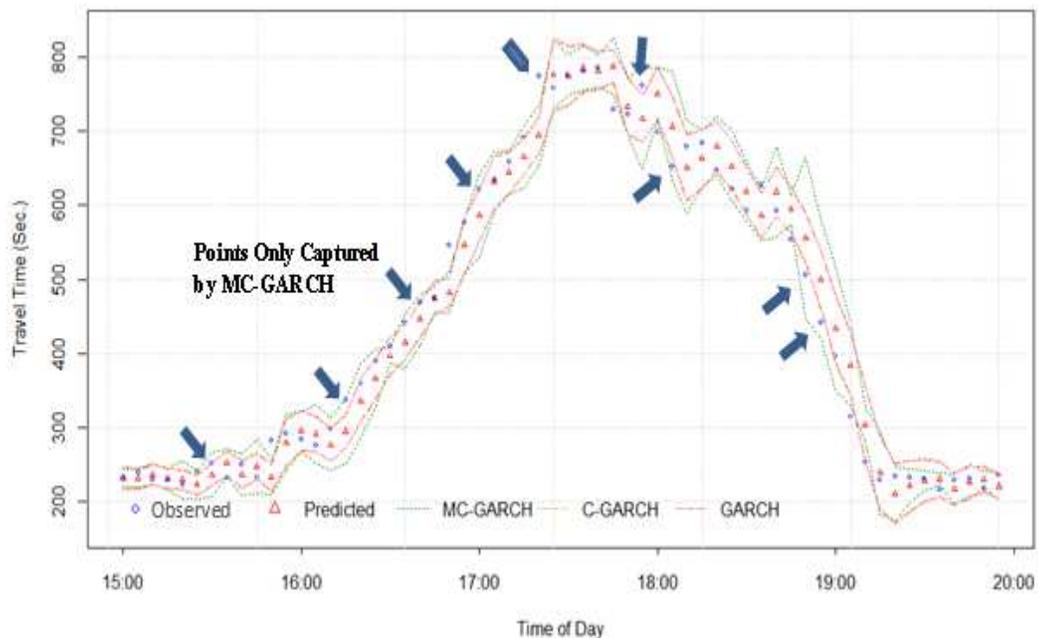


Figure 7. Comparison of prediction intervals constructed by GARCH, C-GARCH and MC-GARCH models

3.3.6 Summary

As uncertainty associated with travel time prediction becomes an important topic for implementing an intelligent transportation system, statistical volatility models provide a promising way to generate more accurate PIs that account for variability in travel time prediction. The traditional GARCH model is argued to be inadequate when modeling data that show pronounced seasonal patterns. This study developed the C-GARCH and the MC-GARCH models in travel time prediction. To empirically evaluate performances of the proposed models, this study tested the GARCH, the C-GARCH and the MC-GARCH models by using freeway travel time data collected from Automatic Vehicle Identification (AVI) stations located along U.S. Highway 290 (or U.S. 290) in Houston, Texas. The forecasting results of the proposed models are attractive, especially during peak hours. The findings of this study include:

The proposed MC-GARCH model outperforms the GARCH and the C-GARCH models during peak hour prediction. A case study of the five selected segments highlighted the strength of the MC-GARCH model in providing more effective PIs in terms of coverage rate. The idea of decomposing travel time volatility is promising when data show cyclic patterns. By decomposing travel time volatility into daily, diurnal and stochastic components, the MC-GARCH model is able to capture uniqueness of each component and captures the seasonal effect of data.

The C-GARCH model treats travel time as a long term and transitory components. It works best if there is a trend. Based on the case study, the performance of PIs constructed by the C-GARCH model and the GARCH model are

similar to each other. The effect of the long term volatility component in the C-GARCH model is not significant in this case.

During non-peak hours, there is no obvious advantage of all three models in terms of MPIL and PICP. This is partly due to the fact that travel time during nonpeak hours is relatively stable, with small variations around the mean. The trend and seasonal patterns are not obvious during this period.

Component GARCH models decompose travel time data into long term, short term and cyclical components. If there are cyclical components in the data, the MC-GARCH model has the potential to better capture uncertainties associated with travel time. In addition, the MC-GARCH model decomposes traffic volatility into several different components that can be easily interpreted and estimated. In this study, the daily component and the normalized residuals are modeled as a simple GARCH model. Besides the GARCH model, the daily component can also be estimated through multifactor risk model as suggested by Engle and Sokalska [5]. It is also worth trying different variations of GARCH models to estimate the normalized residuals. In addition, this study treats the intraday component as an average term. Further study could also explore different ways in defining the intraday component.

3.4 Application of Stochastic Volatility Model

The stochastic volatility based method is investigated here by using travel time data collected with Bluetooth sensors along an 18-mile long corridor in Connecticut. The Bluetooth sensors (Figure 8) were temporarily installed by the University of Maryland team at Interstate 95 (I-95) to collect travel time information between October 19, 2012 and October 28, 2012. Bluetooth technique enables digital

devices interconnect with each other using short-range wireless communications. Many mobile phones, car radios or other personal devices come equipped with Bluetooth wireless capability to communicate with other Bluetooth-enabled devices anywhere from 1 m to about 100 m (300 ft). In the context of travel time collection, the Bluetooth detector captures the electronic identifier, or tag, called Machine Access Control (MAC) address, in each Bluetooth enabled device and places a timestamp when the vehicle enters the detection range of the sensor. As the same vehicle passes subsequent detectors, the detected MAC can be matched allowing the calculation of travel time between these two locations. The Bluetooth detectors require at least two detectors to obtain travel time information. The Bluetooth detector has the advantage of providing more accurate traffic data with relatively low-cost installation [104].

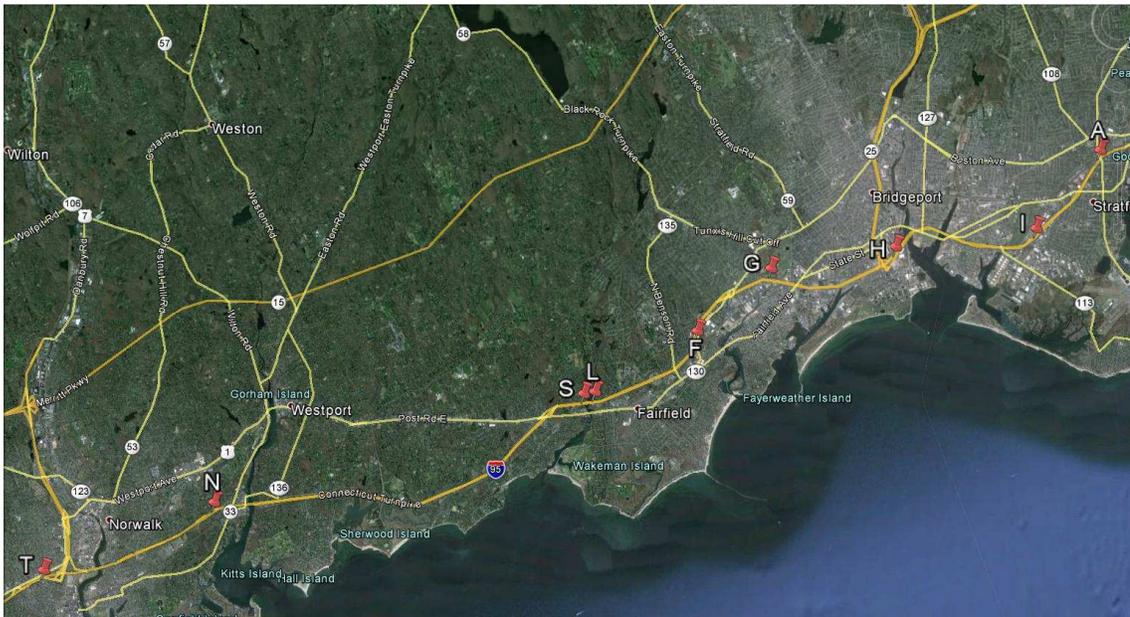


Figure 8. Bluetooth sensor location of the study

Raw data from Bluetooth sensors are MAC IDs of the detected Bluetooth devices along with their detection time stored in a removable memory card. Sample travel time for a particular freeway stretch is obtained by matching the MAC ID between two Bluetooth sensors located at the endpoint of the freeway stretch. Figure 8 depicts the location of the selected Bluetooth sensors named as “T, N, S, L, F, G, H, I, A”. This study selects six segments as indicated in Table 2. Each path is comprised of a head and a rear sensor. Individual vehicle travel time for each path is the time it takes for the same vehicle to be detected by both the head and rear sensors. Raw data are filtered through a four-step offline filtering algorithm proposed by Haghani et al. [104] to extract every five-minute ground truth travel time. Because traffic patterns during weekdays are significantly different from weekends, this study only focused on weekday travel time patterns. We separate travel time data for each pair of detectors under study as training dataset and testing dataset: the training dataset is every five minutes travel time from October 22, 2012 to October 25, 2012 with 1152 observations; the testing dataset is data obtained on October 26, 2012 with 288 observations.

Figure plots one week (excludes weekend) aggregated travel time over every five-minute time interval on four segments. The scatter plots (path one, two, five, six) illustrate considerable variations in travel time at each time interval over different days, especially during peak hours. Rush-hour traffic normally occurs between 3 pm and 8 pm on segments one and two, and between 6 am and 10 am on segments five and six. Travel time variations during non-peak and peak hours show different patterns. Variations of travel time during non-peak hours are much smaller than that

during peak hours. The considerable variations of travel time across different times of the day can be attributed to several factors: demand variations over different times and days, variations of driver behavior under different kind of weather conditions, and incidents that disrupt normal traffic. These exogenous factors are often unpredictable, which makes travel time prediction a complex problem. Therefore, it is critical to treat traffic phenomena as a stochastic process.

Table 2. Selected segments for this study

Path ID	Head Sensor	Rear Sensor	Distance (mile)		Starting at	Ending at
			Standard	Measured		
1	T	N	2.7	2.69	Fairfield Ave/Exit 14	CT-33/CT-136/Exit 17
2	N	S	5.8	5.94	CT-33/CT-136/Exit 17	Bronson Rd/Exit 20
3	A	I	1.5	1.51	Broad St/Exit 32	Surf Ave/Exit 30
4	I	H	2.4	2.36	Surf Ave/Exit 30	CT-25/CT-8/Exit 27
5	H	G	1.9	1.97	CT-25/CT-8/Exit 27	Fairfield Ave/State St/Exit 25
6	F	L	1.7	1.82	US-1/Exit 23	Bronson Rd/Exit 20

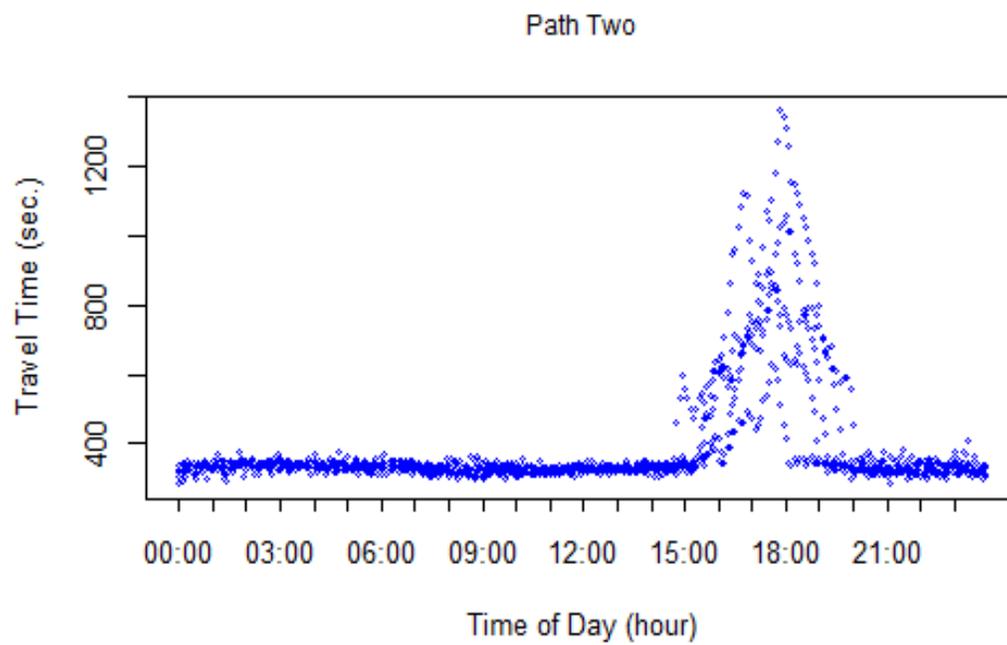
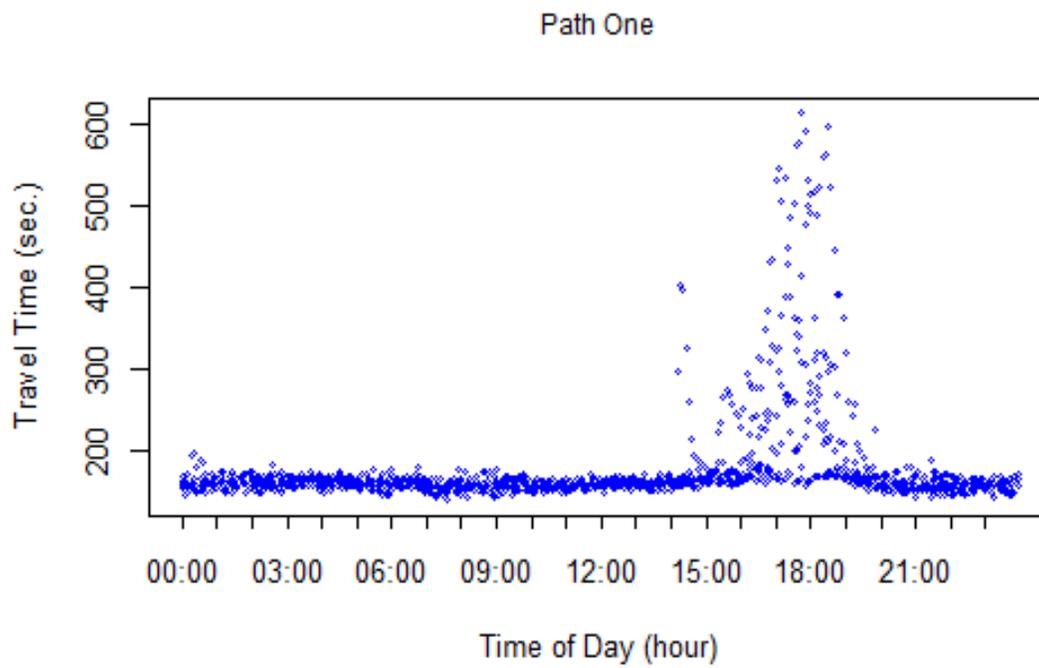


Figure 9. A scatter plot of travel times on four paths.

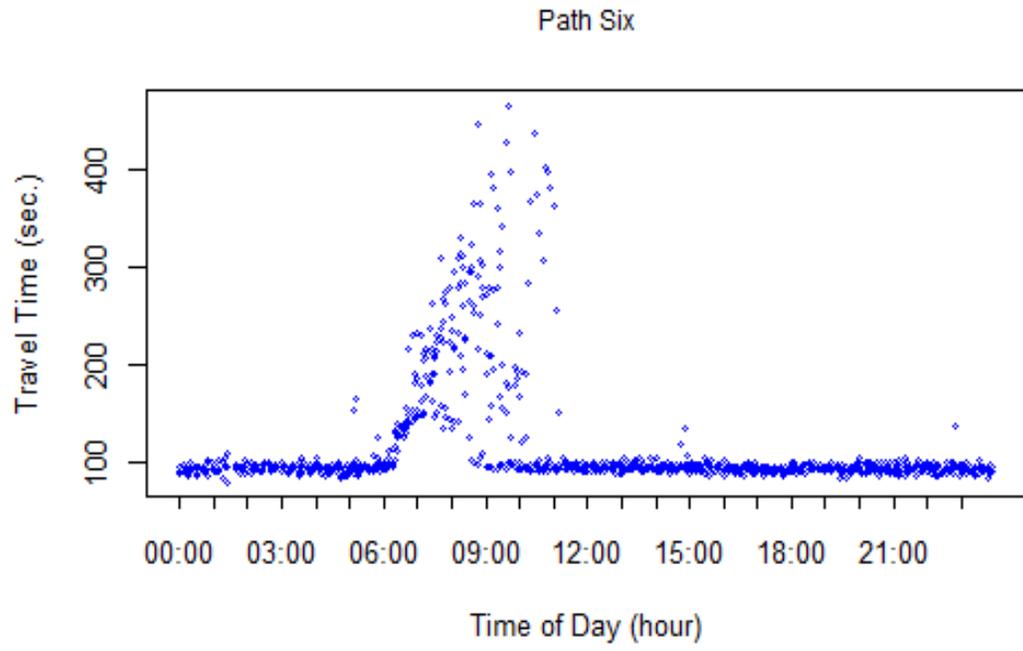
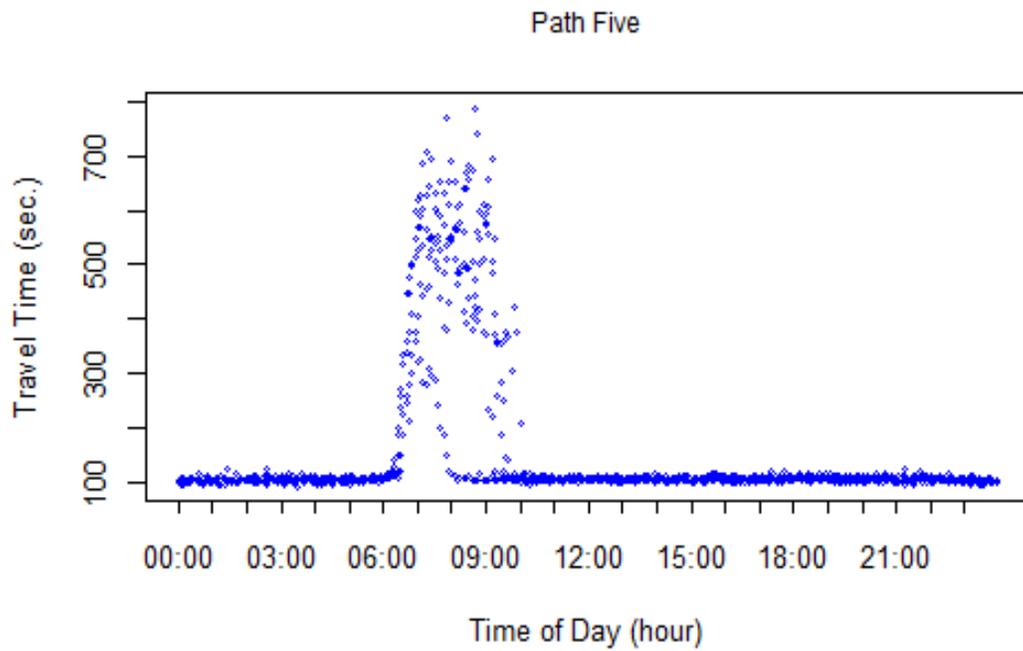


Figure 9 (Cont'd). A scatter plot of travel times on four paths.

3.4.1 Model Fitting

Both the GARCH and the SV models are tested to fit the second part of Equation 1. As the process to estimate the GARCH model has been already described in our previous study [88], the following will focus on the estimation of the SV model.

To perform Bayesian inference in the SV model, there is a need to specify parameters of the prior distribution of μ , ϕ and σ . As it already mentioned that μ follows Gaussian distribution, ϕ follows Beta distribution and σ^2 follows $B_\sigma \cdot \chi_1^2$. , we need to specify five parameters: mean m_μ and standard deviation M_μ of the normal distribution for μ ; a_0 and b_0 of the beta distribution for $(\phi + 1)/2$; and B_σ for the scaling of the transformed parameter σ^2 . After specifying parameters of the prior distributions, the SV model is ready to be estimated by applying the MCMC method. The estimation is performed by using 5000 MCMC draws after a burn-in of 100 for each data set (burn-in means throwing away some iteration at the beginning of an MCMC run). The Bayesian inference using MCMC based on ancillarity-sufficiency interweaving strategy (ASIS) is implemented through the statistical software R package developed by Gregor & Sylvia [6]. To obtain the estimated prediction intervals, we sample 5000 random variables from a normal distribution with mean zero and variance one to obtain ε_t in Equation 4. According to Equation 4, 5000 samplings of r_t could be obtained. PIs with confidence level of $(1 - \alpha)100\%$ can be derived by taking $\alpha/2$ and $(1 - \alpha/2)$ percentile of r_t .

Figure shows travel time prediction results of the ARIMA-SV model for four segments during peak hours. Both predicted mean and PIs with a confidence level of

95% have been provided. The ARIMA predicted value is marked as a red triangle lying at the center of each prediction interval (green vertical line segment). This plot indicates that there is always a mismatch between the observed and the predicted value, which is denoted as the prediction error. On the other hand, PIs constructed by the ARIMA-SV model cover most of the observed values. As indicated in this figure, most of the blue dots (the observed values) lie within the green vertical line segments (the PIs). Depending on the level of uncertainty in the data, the length of the PI for each time interval varies. Wider PI indicates higher uncertainty about the predicted travel time, while narrower PI indicates lower uncertainty. Providing PIs to travelers assist them to schedule their trips with more confidence.

3.4.2 Results and Analysis

To assess the effectiveness of the ARIMA-GARCH and the ARIMA-SV model, we compare these models for the six studied segments at two different time intervals: five and fifteen minute aggregation time intervals. Performance measures in terms of MAPE, RMSE, PICP and MPIL are summarized. Table 3 provides performance measure of the mean equation for the six studied segments. Since both models use the same ARIMA model to predict the mean value of travel time for each segment, each segment has only one MAPE and one RMSE value at each aggregation level. The tabulated results show that ARIMA model provides adequate prediction of future travel time in terms of MAPE, as it ranges from 3.45% to 4.47% at five minute time intervals, and ranges from 1.97% to 6.68% at fifteen minute time intervals. On the other hand, the RMSE value ranges from 3.79 up to 22.94 indicating that the absolute error is proportional to the variation of the data.

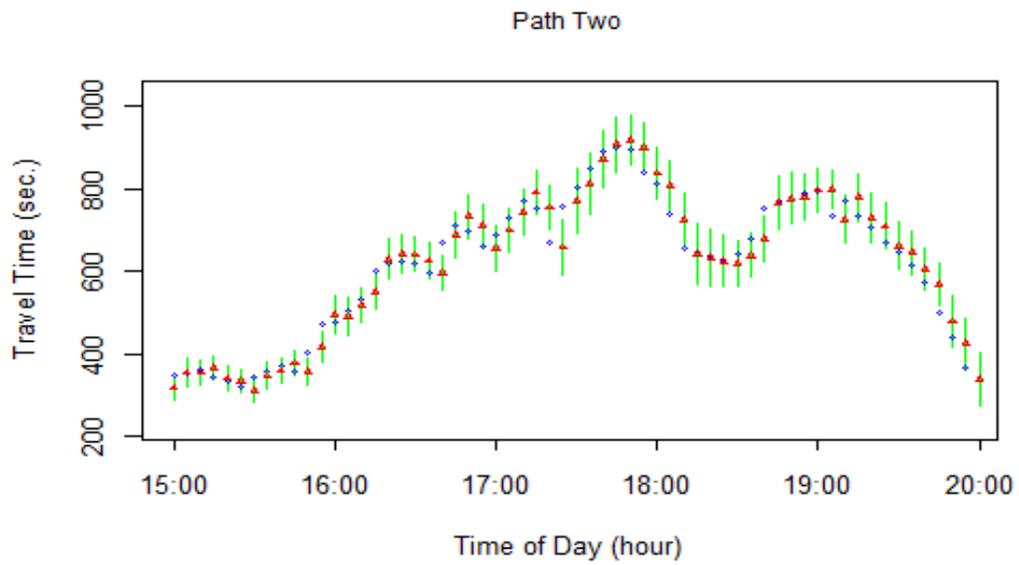
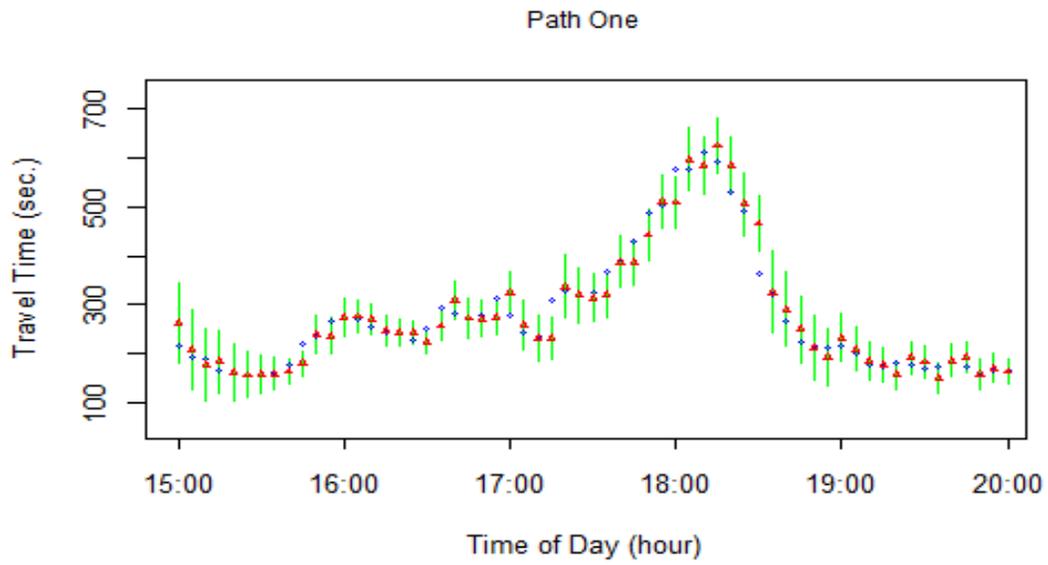


Figure 10. Prediction results for peak hour travel time at four segments.

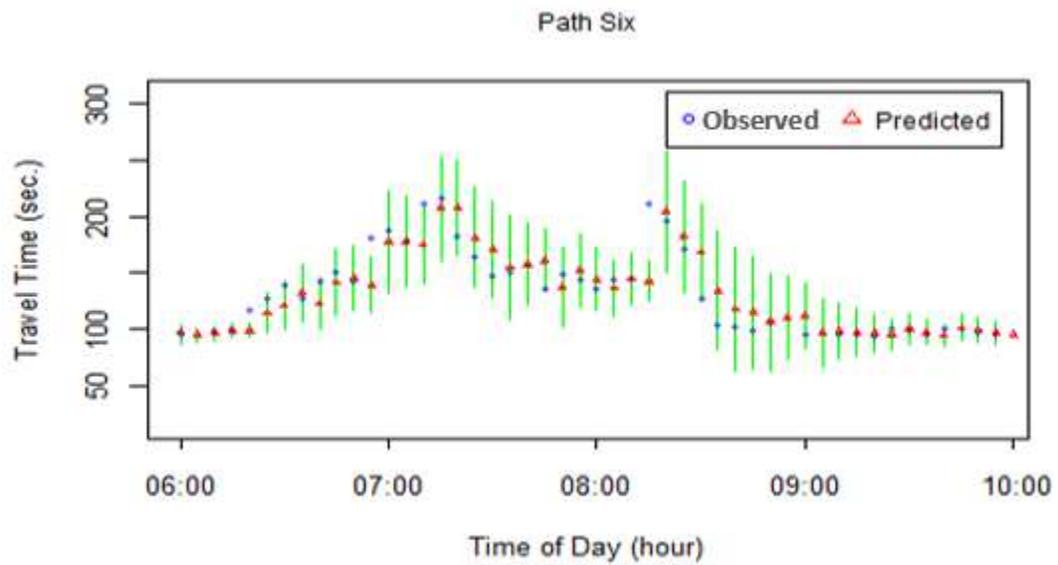
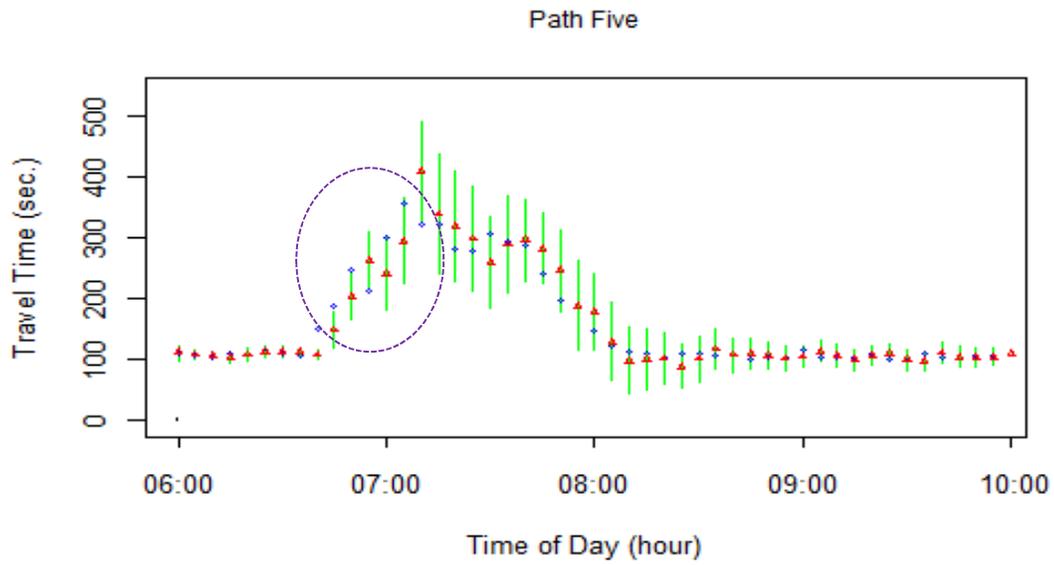


Figure 10 (Cont'd). Prediction results for peak hour travel time at four segments.

In Table 3, RMSE values for path six (8.41 at five minute time interval and 11.93 at fifteen minute time interval) are the smallest among these four segments,

while RMSE values for path two (22.94 at five minute time interval and 34.42 at fifteen minute time interval) are among the largest. The ARIMA estimations for path three and four are the best in terms of both MAPE and RMSE values. This is because travel times on these two segments remain at a certain level with slight variations during the studied time period, which proves that ARIMA model performs better when the data are relatively stable. In contrast, the forecasting accuracy of ARIMA model will decrease as the variation of the data increases.

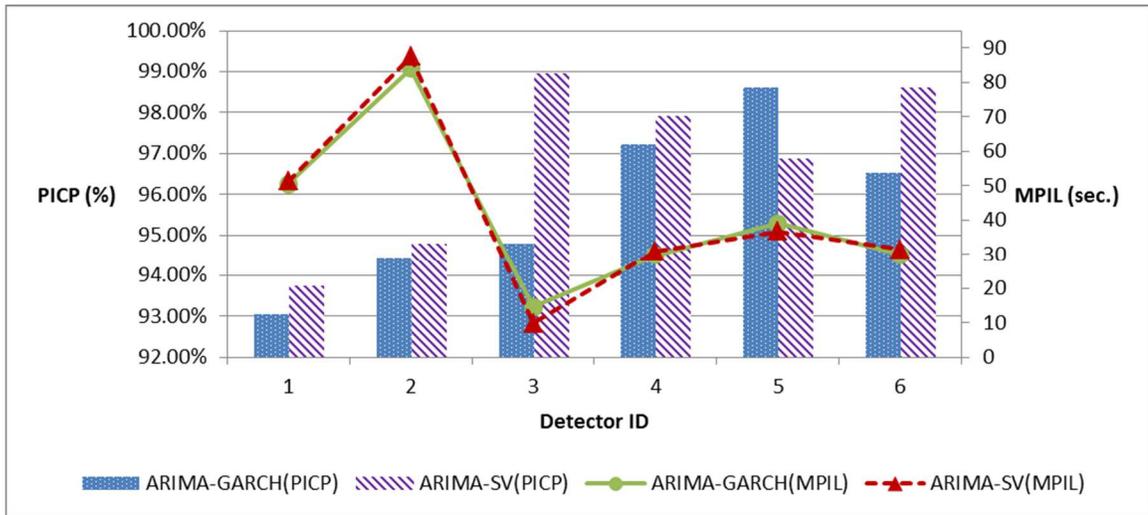
Table 3 Performance measures of the mean equation

Path ID	Time	1	2	3	4	5	6	Average
	Interval							Value
MAPE	5 min	4.47%	4.10%	3.45%	3.77%	4.72%	4.54%	4.18%
	15 min	6.68%	4.55%	1.97%	2.75%	4.79%	4.11%	4.14%
RMSE	5 min	19.67	22.94	3.79	6.82	11.82	8.41	12.24
	15 min	41.34	34.42	2.14	5.01	18.45	11.93	18.88

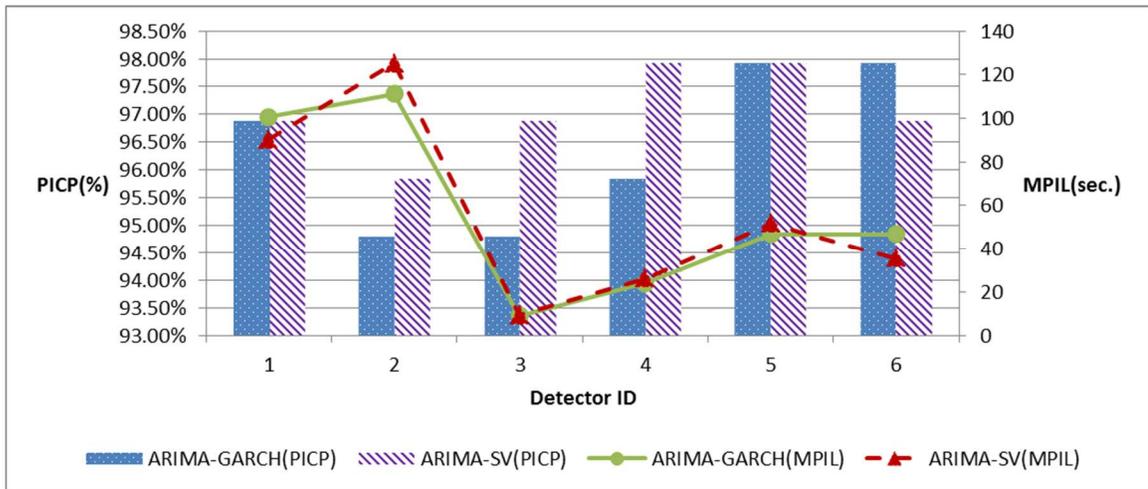
PICP and MPIL measure the prediction accuracy of the variance part. PICP measures the coverage probabilities of the prediction intervals. In our study, the prediction intervals for both ARIMA-GARCH and ARIMA-SV models are set with 95% confidence. Figure 11 compares PICP and MPIL values of both models at two different aggregation levels: five minute and fifteen minute. ARIMA-SV model provides higher coverage in most cases. At both five minute and fifteen minute aggregation levels, only in one out of five cases, the PICP value of the ARIMA-SV model is less than that of the ARIMA-GARCH model. The PICP value of the ARIMA-SV model ranges from 93.75% to 98.96% at five minute aggregation time

interval, and ranges from 95.83% to 97.92% at fifteen minute aggregation time interval. The ARIMA-SV model is capable to construct accurate PIs with predefined confidence level (95% in our study). Compared with the ARIMA-GARCH model, the ARIMA-SV model, in most cases, outperforms the ARIMA-GARCH model in terms of PCIP measure.

Regarding the MPIL value, which measures the width of the prediction interval, both models give similar performance. Comparing the MPIL values of both models for different segments indicates MPIL for segments 3 and 4 have the smallest value. This result demonstrates that the width of the prediction interval depends on the variation of traffic. As travel time for path three and four are relatively stable with minor variations, the MPIL for both models is low. Taking five minute aggregation time interval for example, the MPIL values for path three are 14.62 (ARIMA-GARCH) and 9.85 (ARIMA-SV) respectively, and 29.75 (ARIMA-GARCH) and 30.76 (ARIMA-SV) for path four. The MPIL value for segment two is the highest with 87.55 for the ARIMA-SV model. This is because the travel time varies the most on this path. It can be concluded that length of the constructed PI is sensitive to the variation of travel time on the studied segments, with higher variation leading to wider prediction interval, and vice versa. Since the width of prediction intervals provided by the ARIMA-SV and the ARIMA-GARCH models are similar and the ARIMA-SV model provides relatively higher coverage, the ARIMA-SV model outperforms the ARIMA-GARCH model in general.



(a)



(b)

Figure 11. Comparison of performance measures for six dataset by using ARIMA-GARCH and ARIMA-SV model with (a) 5 minute time interval (b) 15 minute time interval

3.4.3 Summary

This study introduced an advanced stochastic volatility model to construct a prediction interval for each prediction point to capture this uncertainty. The proposed method was tested by using travel time data collected from Bluetooth sensors located

along a freeway corridor in Connecticut. The proposed ARIMA-SV model was compared with the more widely used ARIMA-GARCH model. Different from the GARCH type models that assume deterministic nature of traffic volatility, the SV model considers this volatility as a non-deterministic process by specifying the variance follows some latent stochastic process. An advanced Monte Carlo Markov Chain estimation method for stochastic volatility model was applied to travel time reliability forecasting. The empirical experiment showed that the ARIMA-SV model outperforms the ARIMA-GARCH model in terms of coverage probability and the length of PIs, as both models construct PIs that cover most of the observed value and the ARIMA-SV model tends to provide narrower PIs. In addition, by comparing the constructed PIs across different segments and at different types of aggregation time interval, it is revealed that the length of the constructed PI is sensitive to the variation of travel time. Higher variation leads to wider prediction interval, and lower variation is associated with narrower prediction interval. The width of the prediction interval indicates the variations of the prediction results and therefore provides a measure for the reliability of predicted travel time.

In summary, the proposed ARIMA-SV method shows its advantage in constructing more accurate prediction intervals. It accurately and effectively covers most of the observed values of travel time compared with the ARIMA-GARCH model. From a practical point of view, the proposed ARIMA-SV model provides not only a mean but also an upper and lower bound of future travel time, therefore capturing the uncertainty associated with prediction. As the mean value is unable to provide information regarding the variability of future traffic, the prediction interval

can provide an upper and lower bounds that capture future travel time with predetermined confidence level. Therefore, the proposed model can be used in real time to provide more reliable and informative future traffic information for travelers. The ARIMA-SV model can be regarded as a promising algorithm to disseminate traffic information to travelers through traveler information systems to provide guidance for pre-trip planning as well as en-route navigation. Further research includes studying trend and seasonal patterns in the residuals series and to comprehensively evaluate different volatility model applications in travel time prediction field

Chapter 4: Ensemble Methods in Travel Time Prediction

Chapter 3 discussed different volatility models in capturing the uncertainty associated with travel time prediction. As discussed in equation (1), the travel time prediction can be regarded as a mean model plus a volatility model. In Chapter 3, the ARIMA model is used to predict the mean of travel time. To improve the mean model prediction accuracy, in this chapter a new ensemble based travel time prediction algorithm is proposed.

In recent years, ensemble based algorithms reached a celebrity status in solving prediction and classification problems. They have been applied to different fields and have achieved great success [105]. The \$1 Million Netflix Prize competition is a famous example: the winning team ensembles different algorithms to predict user rating for films, which produces the best accuracy among all participants [106]. Within all different ensemble methods, the tree-based ensemble method is a popular one. Instead of fitting a single “best” model, the tree-based ensemble method strategically combines multiple simple tree models to optimize predictive performance. Drawing on insights and techniques from both statistical and machine learning methods [107], the tree-based ensemble method not only achieves strong predictive performance, but also identifies and interprets relevant variables and interactions. Interpretability of the tree-based ensemble model enables transportation decision makers to better understand the output of the model and is critical in analyzing relations between traffic and their influential factors. In addition, the tree-based ensemble method can handle different types of predictor variables, requires little data preprocessing, and can fit complex nonlinear relationship[107]. These

properties make the tree-based ensemble methods good candidates in solving transportation problems, such as traffic prediction and incident classification.

However, there are limited studies on the application of tree-based ensemble methods in transportation field. To the best of our knowledge, research on gradient boosting tree in travel time prediction has not been fully documented to date.

This section first introduces the theoretical background of ensemble methods and then develops different ensemble models in predicting travel time. Within these methods, a tree-based ensemble method is proposed to predict travel time on a freeway stretch by considering all relevant variables derived from historical travel time data. Belonging to the machine learning category, the tree-based ensemble methods often have superior prediction performance over traditional prediction methods. Driven by the successful application of random forest in traffic parameter prediction, a gradient boosting tree-based travel time prediction method is proposed to uncover hidden patterns in travel time data to enhance the accuracy and interpretability of the model. Different from the random forest algorithm that averages a large collection of trees from random sampling [7], the gradient boosting method sequentially generates base learners from a weighted version of the training data to strategically find the optimal combination of trees. Each step of adding another base learner is aimed at correcting the mistakes made by its previous learners. Therefore, the gradient boosting method has the potential to provide more accurate predictions.

The following sections will first discuss two common ensemble techniques: bagging and boosting. And then introduces the single regression tree model that is

used as a base learner for ensemble tree and two different ensemble tree models: random forest regression and gradient boosted regression tree. The last section applies the random forest and gradient boosted regression tree in travel time prediction.

4.1 Common Types of Ensembles

The ensemble based algorithms consist of multiple base models (such as decision trees, neural networks), and each base model provides an alternative solution to the problem, whose predictions are combined in some way (typically by weighted or unweighted voting or averaging) to produce the final model output. Combining predictions of a group of individual base models often generates more stable and accurate prediction than the one provided by any of the individual base models included in the ensemble. The essential idea behind the ensemble methods is often used in our daily lives. We usually seek others' opinions when making decisions. Through weighted combination of these ideas, we can make more informed decisions. The success of ensemble methods largely depends on diversity of base models that compose the ensemble. Combining results from several base models is useful only if individual models provide different output, or in other words, they disagree with each other on some inputs. Ensemble methods reduce total error through correcting mistakes made by individual models. There is no advantage of combining models that make similar mistakes. Strategically combining individual base models that make different errors (or mistakes) can reduce the total error of the model. Diversity of individual models can be achieved through using different training datasets or using different training parameters for individual models. Different ways to train and combine a number of base learners lead to diverse ensemble algorithms. Bagging and

boosting are two popular ensemble techniques that utilize different re-sampling methods to create diverse training data for obtaining different base models.

4.1.1 Bagging

Bagging, or Bootstrap aggregating, was proposed by Leo Breiman in 1994 [108] to improve prediction accuracy. It is one of the earliest and most intuitive ensemble algorithms with good performance. It belongs to parallel ensemble methods. One advantage of the bagging method is that its theoretical foundation support parallel computing, thus its training speed can be accelerated through parallel computing.

The basic idea of the parallel ensemble methods is to reduce the error by combining prediction results from independent base models. Though it is practically difficult to generate independent base models, as they are trained from the same training data, dependency can be reduced by introducing randomness during model training process. Another vital element for the success of the bagging method is the instability of the individual prediction method. If a diverse set of base models is generated from perturbed training set, then bagging can improve prediction accuracy. To obtain diverse base models from similar training data set, the base learner should be weak. For example, decision tree is one of the most popular weak learners in ensemble method. There are two key steps in bagging: bootstrap sampling and aggregation.

Bootstrap techniques were originally developed to estimate sampling distribution of an estimator from limited data by sampling with replacement from the original data. In recent development of ensemble techniques, bootstrap techniques

have also been used to generate diverse subset of data for training base models. For a given training data set with sample size n , bagging generates m new training sets, each with sample size n , by sampling from the original training data set uniformly and with replacement. By sampling with replacement, some observations appear more than once in the bootstrap sample, other observations may not appear in the sample. The m basic models are trained using the newly generated m training sets and combined through averaging (regression problem) or majority voting (classification problem). Figure 12 illustrates basic steps for the bagging process.

Given a data set with total number of data sample N and p pair of input and output variables. Determine the total number of base model B as K :

For $k = 1$ to K do

 Draw a random sample S^* of size N with replacement from the training data.

 Grow a base model $B_k(x)$ using the training sample S^* .

 Output the constructed base model $B_k(x)$.

End;

Output the prediction of the ensemble trees for a given new input x :

$$\frac{1}{K} \sum_{k=1}^K B_k(x);$$

Figure 12 The Bagging algorithm

4.1.2 Boosting

Different from bagging, boosting method generates base learners sequentially. Therefore, it belongs to sequential ensemble methods. The basic idea of sequential methods is to explore the dependencies between each learner. Prediction accuracy is improved through developing multiple models in sequence by putting emphasis on these training cases that are difficult to estimate. In boosting training process, examples that are difficult to estimate by the previous learners appear more often in the training data set than examples that are correctly estimated. The boosting method attempts to develop base learners that are able to correct the mistakes made by previous learners.

The birth of the boosting method is from the answer [109] to Kearns' question [110]: Is a set of weak learners equivalent to a single strong learner? A weak learner is an algorithm that performs only slightly better than random guessing; a strong learner is more accurate prediction or classification algorithm that is arbitrarily well correlated with the problem. The answer to this question is important. It is often easier to estimate a weak learner compared with a strong learner. Schapire [110] proves that the answer is positive by applying boosting algorithms (Figure 13) to combine many weak learners into a single and high accurate learner.

The major difference between Bagging and Boosting methods is that Boosting method strategically resamples the training data to provide the most informative information for each consecutive model. The adjusted distribution during each step of creating a new base model is based on the error produced by the previous models. Unlike bagging method that each sample is uniformly selected to produce the training

dataset, the probability of selecting individual example is not equal for the Boosting algorithm. Samples that are misclassified or incorrectly estimated have more chances to be selected with a higher weight. Therefore, each newly created model focuses more on the samples that have been misclassified by its previous models.

Given a data sample distribution D and p pair of input and output variables.

Determine the total number of base model B as K :

Define the initial training sample distribution as $D_1 = D$

For $k = 1$ to K do

 Train a base model $B_k(x)$ from the training sample distribution D_k .

 Compute the error of the learner.

 Adjust the distribution D_k . to D_{k+1} . to make the mistake of the learner more evident.

 Output the constructed base model $B_k(x)$.

End;

Output the prediction of the ensemble trees for a given new input x :

$$\frac{1}{K} \sum_{k=1}^K B_k(x);$$

Figure 13. The Boosting algorithm

4.2 Ensemble Tree

The success of ensemble methods essentially depends on diversity of base models that compose the ensemble. The combination of the results of several base

models is useful only if individual models provide different output, or in other words, they disagree with each other on some inputs. There is no benefit to combine models that make similar mistakes. Strategic combination of individual base models that produce different errors can reduce the total error of the model. Ensemble method is most effective if each model's output is independent or negatively correlated.

Diversity of individual models can be achieved through using different training dataset or using different training parameters for individual models. The previously discussed bagging and boosting based ensemble methods utilize different re-sampling techniques to create diverse training data, therefore obtain different base models. In order to produce diverse base models despite similar training dataset, the base models are often forced to be weak. Therefore, perturbing the training data can generate different model outputs.

Trees are commonly used as base learners for ensembles since they can be sensitive to small perturbations in the training data and a light change can lead to different regression trees. This unique property makes them good candidates for ensemble. In addition, trees are fast and easy algorithms, which reduce computation time and complexity. Both bagging and boosting based ensemble methods can use trees as base learners. Tree based ensemble methods build a large number of de-correlated trees and then average the results from individual trees. The benefit of using ensemble tree is that through averaging, the variance can be reduced. Details will be illustrated in the later sections. In general, there are three successful tree based ensemble methods: Bagged tree, random forest, and boosted tree. The following

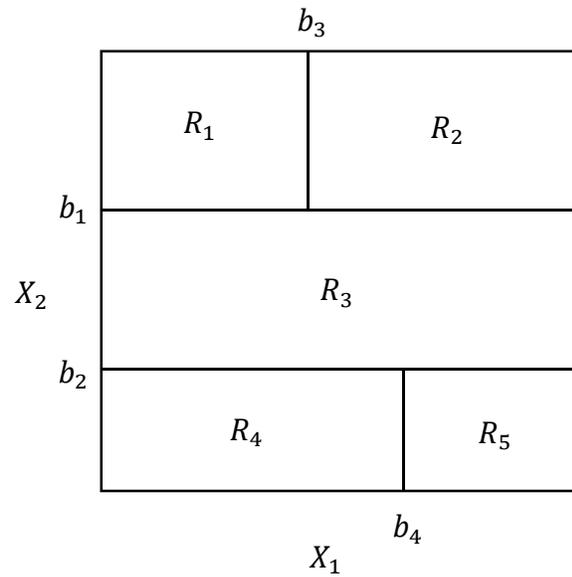
paragraphs will briefly explain theoretical background of a single regression tree and then illustrates how to construct different ensemble trees.

4.2.1 Single Regression Tree

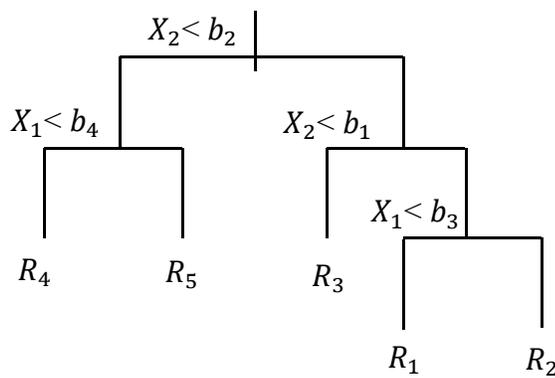
A single tree model partitions the feature space into a set of regions and fit a simple model for each region, for example fit a constant for each region. For simplicity, consider a regression problem with response variable Y and two independent variables X_1 and X_2 . We first split the space into two regions and model the response Y (mean of Y) individually in each region. Then we continue to split each individual region into two more regions and continue the process until some stopping rules are met. The upper panel of Figure 14 partitions the feature space into five regions $\{R_1, R_2, R_3, R_4, R_5\}$ according to two variables X_1 and X_2 using four split-points b_1, b_2, b_3, b_4 . During each partition process, the best fit is achieved through the selection of variables and a split-point. The lower panel of figure is a binary tree representation of the same model.

We now consider a generalized version of the above example: a regression problem consists of p inputs with one response variable. For example, we have N observations, each observation consists of $(y_i, x_{i1}, x_{i2} \dots x_{ij}, \dots x_{ip})$ for $i = 1, 2, \dots, n$, $j = 1, 2, \dots, p$. The feature space is partitioned into M regions R_1, R_2, \dots, R_m . The regression tree needs to automatically select the splitting variable and split-point and calculate the response for each region. Often the response for each region is treated as a constant C_m . If the optimization criterion is to minimize the sum of squares, the best C_m is the just the average of y_i in region R_m . To decide the best splitting variable and the split-point, a greedy algorithm is implemented. For each splitting variable, the

best split-point can be determined by scanning all possible values, which can be done quickly. By scanning through all input variables, finding the best pair of splitting variable and split-points is feasible. A single regression tree is the basic learner for bagged regression tree, random forest and gradient boosting regression tree methods.



(a)



(b)

Figure 14. Single regression tree

4.2.3 Random Forest Regression

Random forest was developed by Leo Breiman [7] in 2001. It combines two powerful machine-learning techniques: Breiman's "bagging" idea [108] and the random features selection introduced by Ho [111, 112] and Amit and Geman [113]. In bagging or bootstrap aggregation, each individual based model is trained on the bootstrap sample from the training data. The bootstrap techniques were originally developed to estimate sampling distribution of an estimator from limited data by sampling with replacement from the original data. In recent development of ensemble techniques, bootstrap has been used to generate a diverse subset of data for training base models. For a given training data set with sample size n , bagging generates k new training sets, each with sample size n , by sampling from the original training data set uniformly and with replacement. Through sampling with replacement, some observations appear more than once in the bootstrap sample, while other observations will be 'left out' of the sample. Then, k base models are trained using the newly generated k training set and combined through averaging (regression problem) or majority voting (classification problem).

Using a tree as the base learner, we can obtain perhaps the simplest and easiest ensemble tree, the bagged tree. Each tree in the ensemble is grown on data samples that were randomly drawn with replacement from the original data. The success of the bagged regression tree depends on if diverse trees are generated from different bootstrapped training dataset. However, with lots of data, we usually learn the same regression tree. Averaging output of these trees does not improve prediction accuracy.

Given a data set with total number of data sample N and p input variables.

Initially determine the total number (K) of trees to be generated and the number $m < p$ of variables used for each individual tree:

For $k = 1$ to K do

Draw a random sample S^* of size N with replacement from the training data (This is also referred as bootstrap sample. This sample will be the training data to grow the tree.);

Grow a random forest tree T_k using the training sample S^* through the following loop:

Do until (the minimum node size n_{min} is reached)

For the terminal node of the tree;

Randomly select m variables out of the p variables;

Select the best pair of split variable/point among the m variables;

Split the node into two daughter nodes;

End;

Output the constructed tree $T_k(x)$;

End;

Output the prediction of the ensemble trees for a given new input x :

$$\frac{1}{K} \sum_{k=1}^K T_k(x);$$

Figure 15. Pseudo-code for random forest

Random forest is a further development of the bagged regression tree. It is still based on the bootstrapped sampling to grow individual trees. Instead of using all features, it only allows a random subset of features at each splitting node of the tree. Therefore, it enforces diversity between base learners. Figure 15 illustrates the basic steps for random forest.

Random forest improves forecasting accuracy through variance reduction by averaging many noisy but approximately unbiased trees. According to [114], the variance of a random forest with total number (K) of trees is:

$$\rho\sigma^2 + \frac{1 - \rho}{K}\sigma^2 \quad (41)$$

Where σ^2 indicates the variance of individual tree, ρ denotes correlation between the trees, and K is the total number of trees in the ensemble. It is obvious that by increasing the total number of trees K, the second term tends to be zero. Therefore, the variance of a random forest depends on three things:

- (1) The correlation ρ between any pair of trees: Decreasing the correlation decreases the total variance. This can be achieved by: randomly selecting m out of the p variables to split at each splitting node when growing a tree on a bootstrapped dataset. Reducing m, reduces both the correlation between trees and the strength of individual tree, and vice versa. Therefore, there is a need to find the optimal value of m for certain dataset.
- (2) The variance σ^2 of each individual tree, or in other words, the strength of each individual tree: Strengthening the performance of each individual tree can decrease the total variance of the model.

(3) The total number of trees K : The second term of the equation can be reduced by increasing K . Therefore, we should train adequate number of trees to make sure the second term of the equation goes to zero.

In general, random forest is based on the idea of bagging but enforces diversity of each individual tree through random feature selection. The theoretical background of random forest supports parallel computing therefore its training speed can be accelerated through parallel computing. The prediction performance of the random forest is influenced mainly by three factors: correlation between individual trees, the performance of each tree and the total number of trees.

4.2.4 Gradient Boosted Regression Tree

Typically, a boosted method is to fit multiple base models that minimize a certain loss function averaged over the training data, such as squared-error or absolute error. The loss function measures the amount the predicted value deviates from the actual value. One of the approximate solutions to this problem is by using forward stagewise modeling approach. The forward stagewise approach sequentially adds new base models without changing parameters and coefficients of models that have already been added. The gradient boosted regression tree takes advantage of tree based methods and boosting. It can handle different types of input variables, is insensitive to outliers, can model complex nonlinear relationships, and can automatically handle interaction effects between variables. In addition, fitting multiple trees improves the prediction performance compared with single tree. Typically, the boosted regression tree performs better than the traditional prediction method.

In terms of regression problem, the boosted method is a form of “functional gradient decent”. It is an optimization technique that minimizes a certain loss function by adding a base model, at each step, that best reduces the loss function. The first step of the gradient boosted regression tree method is to fit an optimal constant model, a single terminal node tree.

The pseudo code for the generic gradient boosting method is shown in Figure 16 [114, 115].

Friedman proposes a modification to the gradient boosting method that uses a regression tree of fixed size as the base learner. The modified version improves the quality of fit of each base learner [116]. Assume that the number of leaves for each tree is J . The tree partitions the input space into J disjoint regions $R_{1k}, R_{2k}, \dots, R_{jk}$ and predict a constant value b_{jk} in each region R_{jk} . The regression tree can be formally expressed as:

$$g_k(x) = \sum_{j=1}^J b_{jk} I(x \in R_{jk}) \quad (42)$$

$$\text{where } I(x \in R_{jk}) = \begin{cases} 1, & \text{if } x \in R_{jk} \\ 0, & \text{otherwise} \end{cases}$$

Using the regression tree to replace $g_k(x_i)$ in the generic gradient boosting method, the model updating equation and gradient descent step size:

$$f_k(x) = f_{k-1}(x) + \rho_k g_k(x) \quad (43)$$

$$\rho_k = \operatorname{argmin}_{\rho} \sum_{i=1}^n L(y_i, f_{k-1}(x_i) + \rho g_k(x_i)) \quad (44)$$

become

$$f_k(x) = f_{k-1}(x) + \sum_{j=1}^J \rho_k b_{jk} I(x \in R_{jk}) \quad (45)$$

$$\rho_k = \operatorname{argmin}_\rho \sum_{i=1}^n L(y_i, f_{k-1}(x_i)) + \sum_{j=1}^J \rho b_{jk} I(x \in R_{jk}) \quad (46)$$

Initialize $f_0(x)$ to be a constant, $f_0(x) = \operatorname{argmin}_\rho \sum_{i=1}^n L(y_i, \rho)$.

For $k = 1$ to K do

 For $i = 1$ to n do

 Compute the negative gradient

$$z_{ik} = - \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{k-1}}$$

 End;

 Fit a regression tree $g_k(x)$ to predict the targets z_{ik} from covariates x_i for all training data.

 Compute a gradient descent step size as

$$\rho_k = \operatorname{argmin}_\rho \sum_{i=1}^n L(y_i, f_{k-1}(x_i) + \rho g_k(x_i))$$

 Update the model as

$$f_k(x) = f_{k-1}(x) + \rho_k g_k(x)$$

End;

Output the model $f_K(x)$

Figure 16. Pseudo-code for generic gradient boosting

Using a separate optimal ρ_{jk} for each of the tree's regions R_{jk} , b_{jk} could be discarded. The model updating rule becomes:

$$f_k(x) = f_{k-1}(x) + \sum_{j=1}^J \rho_{jk} I(x \in R_{jk}) \quad (47)$$

$$\rho_{jk} = \underset{\rho}{\operatorname{argmin}} \sum_{x_i \in R_{jk}} L(y_i, f_{k-1}(x_i) + \rho I(x \in R_{jk})) \quad (48)$$

The gradient boosting regression tree builds the model in a stage-wise fashion and updates the model by minimizing the expected value of certain loss function.

With many trees added to the model, the fitted model may achieve an arbitrarily small training error. However, fitting the model too closely to the training data can lead to poor generalization ability. By increasing the number of iterations, the model becomes complex and minor fluctuations in data will be exaggerated. This leads to poor prediction performance on unseen data (testing data). It is necessary to determine the optimal number of iterations (or the number of trees) M to minimize future risks associated with prediction. Over-fitting can be prevented through controlling the number of gradient boosting iterations, or more effectively, scaling the contribution of each tree by a factor of $J \in (0,1]$. This implies changing the model updating equation (47) to

$$f_k(x) = f_{k-1}(x) + J \cdot \sum_{j=1}^J \rho_{jk} I(x \in R_{jk}) \quad (49)$$

Parameter J , referred to as learning rate, controls the contribution of each base model by shrinking its contribution with a factor $0 < J \leq 1$. There is a tradeoff between the number of iterations and the learning rate. With the same number of iterations, a smaller value of learning rate tends to lead to a higher training risk. Smaller value of J requires a larger number of M to obtain the same training risk. In general, a small J ($J < 0.1$) with a large M is preferable.

Another parameter, tree complexity, also influences performance of the algorithm. Tree complexity refers to the number of nodes in a tree. The optimal size of each tree can be estimated separately when building the ensembles. By simply assuming each tree is the last one in the model, we usually expect large trees, especially during the early iterations. This is a poor assumption and potentially degrades the model performance and increases computation complexity. One simple solution is to restrict all trees to be the same size C . Therefore, for the entire process, we only need to determine one value of C to optimally estimate the data. Gains from increased C are greater with larger data sets. As large data sets provide more detailed information about the problem, increasing the value of C would capture complex variable interactions in data. The tree size C constrains the interaction level of each model. Namely, $C - 1$ is the maximum level of interaction effects for a tree with size C . Therefore, the size of the trees reflects the maximum depth of variable interactions.

The GBM model strategically adds each base model to minimize a certain loss function. It uses a stage-wise sampling strategy, which put more emphasis on samples that are difficult to be estimated. This distinguishes itself from random forest that trains each base model from random sampling with replacement and equal probability. Performance of the GBM model is influenced by the number of trees, learning rate and variable interactions. Optimal performance of the model can be achieved through carefully selecting the best combination of these parameters.

4.3 Application to Travel Time Prediction

Accurate travel time prediction relies on how much information we could extract from available data. As traffic is often a complex phenomenon that involves

non-linear and chaos characteristics, it is often difficult to use an exact equation to represent this phenomenon. Data driven approach becomes a promising area in modeling and predicting traffic. In recent years, tree-based ensemble methods have shown promising results in prediction field. Developing tree-based ensemble method in travel time prediction can potentially improve prediction accuracy. This section discusses in details on how to apply the GBM model in travel time prediction.

4.3.1 Data Description and Preparation

Real-world travel time data provided by a private-sector company, INRIX, are used for this study. INRIX derives travel times from its smart driver network, which aggregates traffic data from probe vehicles and traditional sensor sources. Probe vehicles utilized include: taxis, airport shuttles, service delivery vans, long-haul trucks, consumer vehicles, and GPS enabled consumer smartphones and so on. Traffic sensors range from inductive-loop detectors, radar sensors, to toll tag readers. The data fusion methods are proprietary and travel times are reported on TMC segments. This study utilized travel time data from five TMC segments located along I-95 southbound in Maryland. Table 4 shows location information for the five selected TMCs, which includes corresponding TMC code, start and end location, and length of each segment. Travel time data observed in 2012 were downloaded from the Regional Integrated Transportation Information System (RITIS) website[117], and was aggregated into every five minutes time interval. The quality of the data is excellent with less than a 1% missing rate; 561 out of 105408 observations are missing for most segments. Given the small amount of missing values, this study simply replaced the missing values with the mean of its closest surrounding values.

Table 4 Selected Freeway Segments for the Study

Section	TMC	Start		End		Miles
		Latitude	Longitude	Latitude	Longitude	
I	110-04421	39.218482	-76.726905	39.200843	-76.760999	2.2
II	110N04421	39.200843	-76.760999	39.192756	-76.771665	0.8
III	110-04420	39.192756	-76.771665	39.182237	-76.78368	0.97
IV	110N04420	39.182237	-76.78368	39.175368	-76.794578	0.75
V	110N04419	39.160086	-76.823761	39.156238	-76.834836	0.66

Table 5 summarizes the basic statistics of collected travel time data in 2012, including: mean value, standard deviation (SD), the 25th, 50th, 75th and 95th percentiles of travel time, minimum (min), and maximum (max) observations. To prepare the input data for the model, we considered all possible variables that are relevant to future travel time. This led to ten input variables for the prediction model: three most recent travel time observations, three most recent trends of travel time (travel time growth rate over two consecutive time steps), time, day, week and month of the observation.

Table 6 is an example of the data set (training and testing set) used for our study. The first ten columns are the input variables and the last column is the corresponding output of the model. The output of the model is travel time at time lag t denoted as TT_t . The ten variables that are used as input to predict travel time at time step t are as follows: TT_{t-1} , TT_{t-2} , TT_{t-3} are three most recent travel time observations at time steps $t - 1$, $t - 2$, and $t - 3$, $\Delta TT_{t-1} = TT_{t-1} - TT_{t-2}$ is the growth rate over two consecutive time steps $t - 1$ and $t - 2$, time of day is represented by every five minute time step indexed from 1 to 288, week is indexed

from 0 to 6 to represent from Sunday to Saturday, day is the day when the vehicle is detected (from 1 to 31), and month is the month information for the observation (from 1 to 12).

Table 5 Basic Statistics of Travel Time Data

Section	Mean	SD	25th	50 th	75th	95th	Min	Max
I	2.01	0.52	1.92	1.96	2.01	2.17	1.77	26.41
II	0.73	0.19	0.69	0.71	0.73	0.78	0.65	9.66
III	0.89	0.21	0.85	0.87	0.89	1.00	0.78	22.42
IV	0.70	0.24	0.66	0.67	0.69	0.74	0.61	9.06
V	0.60	0.15	0.58	0.59	0.61	0.63	0.53	7.90

Table 6 Example of the Training/Testing Data File

TT_{t-1}	TT_{t-2}	TT_{t-3}	ΔTT_{t-1}	ΔTT_{t-2}	ΔTT_{t-3}	Time of day	Day	Week	Month	TT_t
1.88	1.86	1.94	0.02	-0.08	-0.07	283	1	0	1	1.88
1.88	1.88	1.86	0	0.02	-0.08	284	1	0	1	1.87
1.87	1.88	1.88	-0.01	0	0.02	285	1	0	1	1.89
1.89	1.87	1.88	0.02	-0.01	0	286	1	0	1	1.9
1.9	1.89	1.87	0.01	0.02	-0.01	287	1	0	1	1.93
1.93	1.9	1.89	0.03	0.01	0.02	288	1	0	1	2.01
2.01	1.93	1.9	0.08	0.03	0.01	1	2	1	1	2.01
2.01	2.01	1.93	0	0.08	0.03	2	2	1	1	2.01
2.01	2.01	2.01	0	0	0.08	3	2	1	1	1.81
1.81	2.01	2.01	-0.2	0	0	4	2	1	1	1.77
1.77	1.81	2.01	-0.04	-0.2	0	5	2	1	1	1.99
...

4.3.2 Model Optimization

To optimize the model, it is critical to know the effect of different combinations of parameters on the model's performance. Based on this information, we can then select the optimal parameters to achieve a lower prediction error. This section demonstrates how performance varies with different choices of parameters (number of trees M , learning rate J and interaction C) by using two months' traffic data as training data and the following seven days' data as testing data. Using traffic data from freeway segment one, we fitted GBM models with various numbers of trees (1 - 8000), learning rates (0.5-0.0005) and variable interactions (1-4). Figure 18, and Figure 19 show the influence of different parameters ($M, J,$ and C) on the prediction errors. In these plots, we use mean absolute percentage error (MAPE) to represent prediction error. To study the effect of parameter M on prediction accuracy, Figure 18 plots the relationship MAPE and M (with different value of J and C). The parameter M indicates how many base models are included in the ensemble. In terms of estimation, arbitrary accuracy can be achieved through increasing M . But with too many trees, over-fitting may occur, which affects prediction performance on 'unseen data' (samples not included in the training data set). In Figure (a), MAPE value decreases as M increases with $C = 1$. The slopes of the lines are different for different value of J . The line with $J = 0.0005$ has smooth slope as the contribution of each additional tree becomes limited with a small learning rate. On the other hand, a higher learning rate, such as $J = 0.5$, can be too fast so that it reaches its minimum error with $M = 600$. To continue increasing M will increase prediction error, where over-fitting occurs. This effect becomes even more obvious if we allow more variable

interactions, which makes the individual tree more complex. As shown in Figure (b), (c), and (d), higher learning rates (such as $J = 0.5, J = 0.01, J = 0.05$) reach their best prediction performances with relatively fewer numbers of trees ($M = 200$) and can easily over-fit the model if more trees are included. In general, we should guarantee enough trees to model the complexity of the data and, at the same time, prevent over-fitting with too many trees.

Figure 18 plots the effect of learning rate on MAPE value. Learning rate adjusts the contribution of each additional tree with a factor J , $0 < J < 1$. A smaller value of J limits the contribution of each tree in the model and often requires more trees to be added. Depending on the complexity and the number of trees in the ensemble, the optimal value of J can be different. Taking Figure 18 (a) for example, with $M = 1000$, prediction error (MAPE) increases if we continue to decrease J after a certain level ($J = 0.05$). This is because the contributions of individual trees become limited if J is below a certain level and the current value of M becomes insufficient. In order to obtain better prediction results, M should be increased with a decreased value of J . On the other hand using higher value of J leads to fewer trees needed to achieve better performance. But a higher J usually cannot achieve the minimum error. In Figure 18 (b), $J = 0.5$ is fitted with relatively fewer trees, but did not achieve an error as small as the one with $J = 0.1$. With a higher value of J , increasing M leads to poorer prediction performance (Fitting $M = 8000$ with $J = 0.5$), because over-fitting occurs. Therefore, a smaller learning rate with a larger number of trees is preferable, but computational time would also be increased with

many trees being fitted. There needs to be a balance between computational time and prediction accuracy.

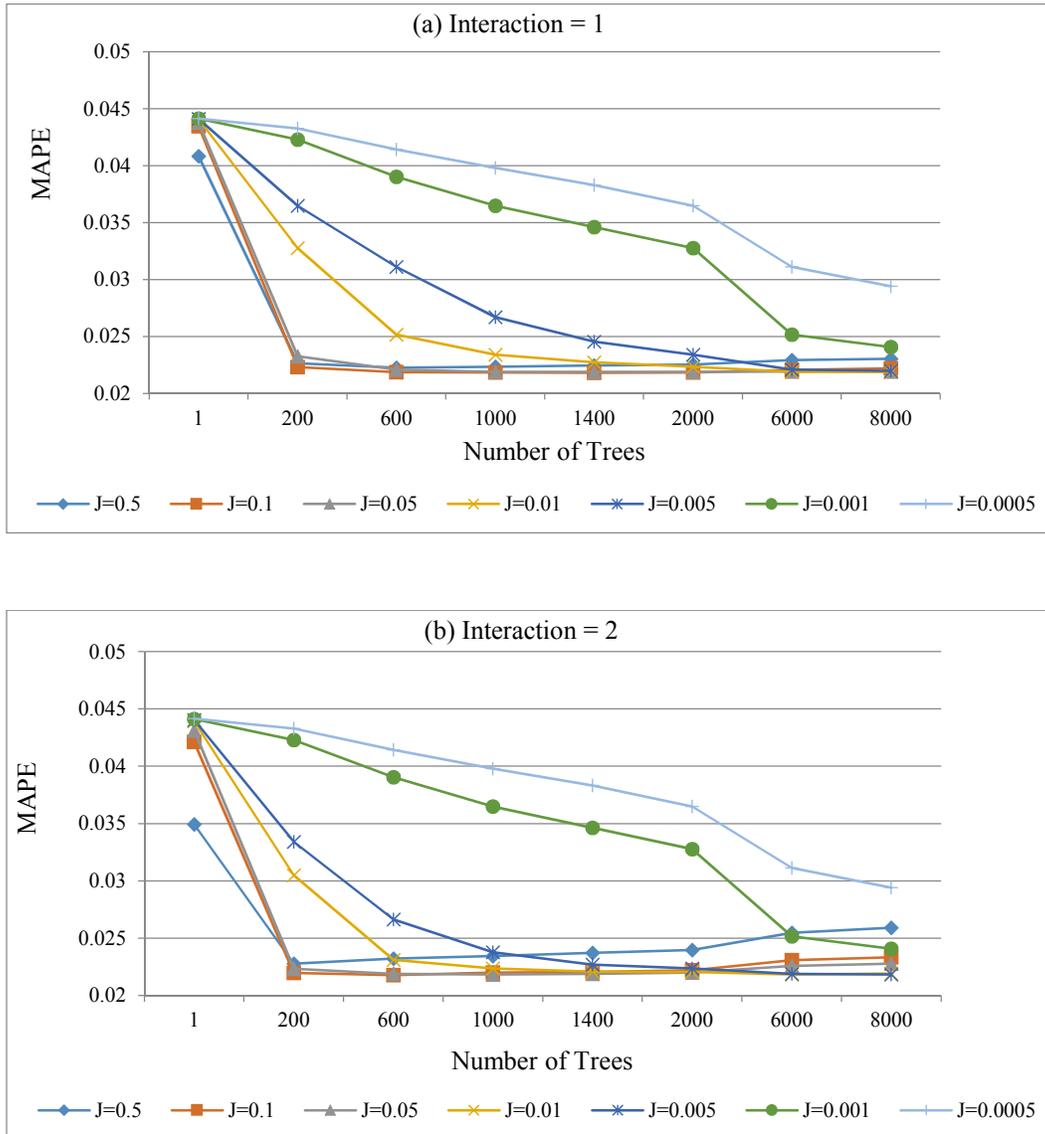


Figure 17. The Relationship between MAPE and Number of Trees for Models Fitted with Seven Learning Rates and Four Levels of Interactions

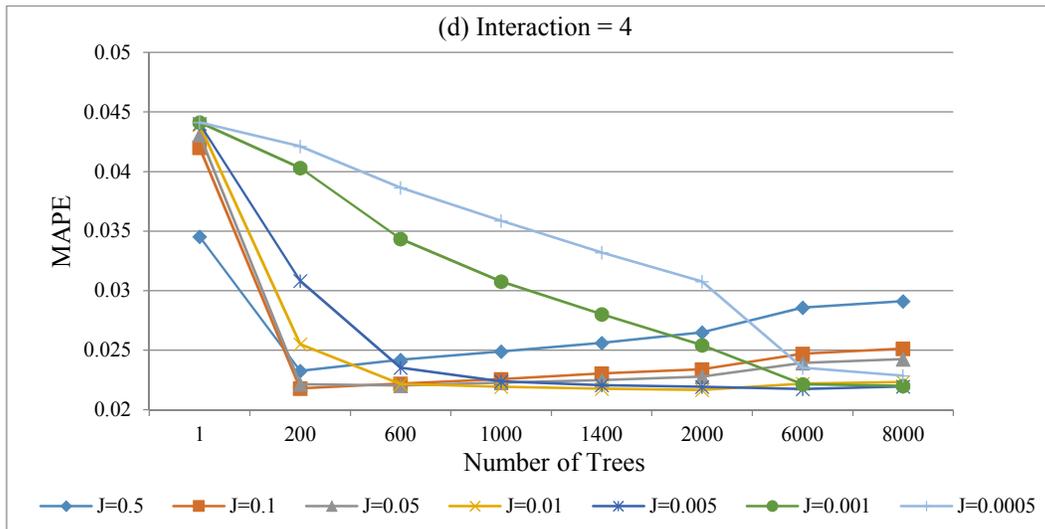
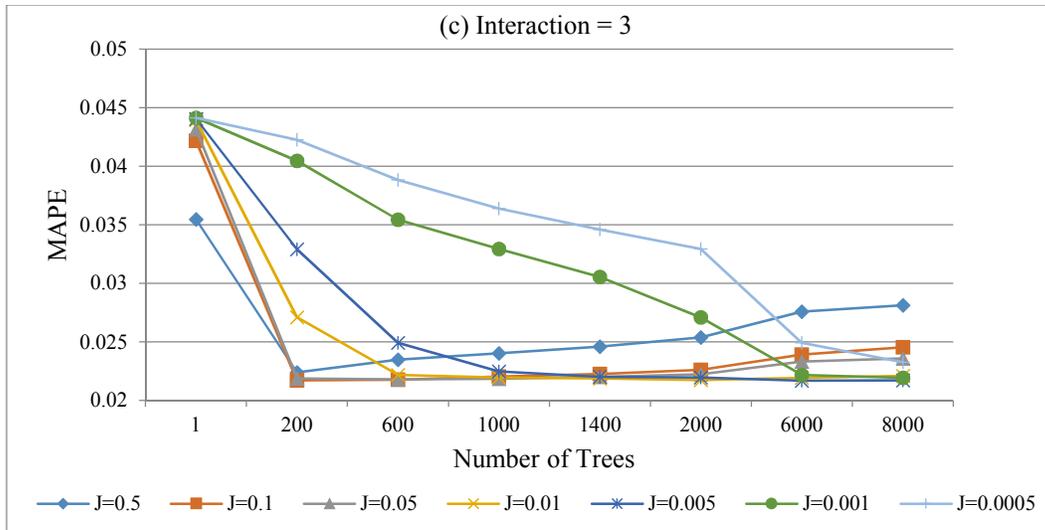


Figure 17 (Cont'd). The Relationship between MAPE and Number of Trees for Models Fitted with Seven Learning Rates and Four Levels of Interactions

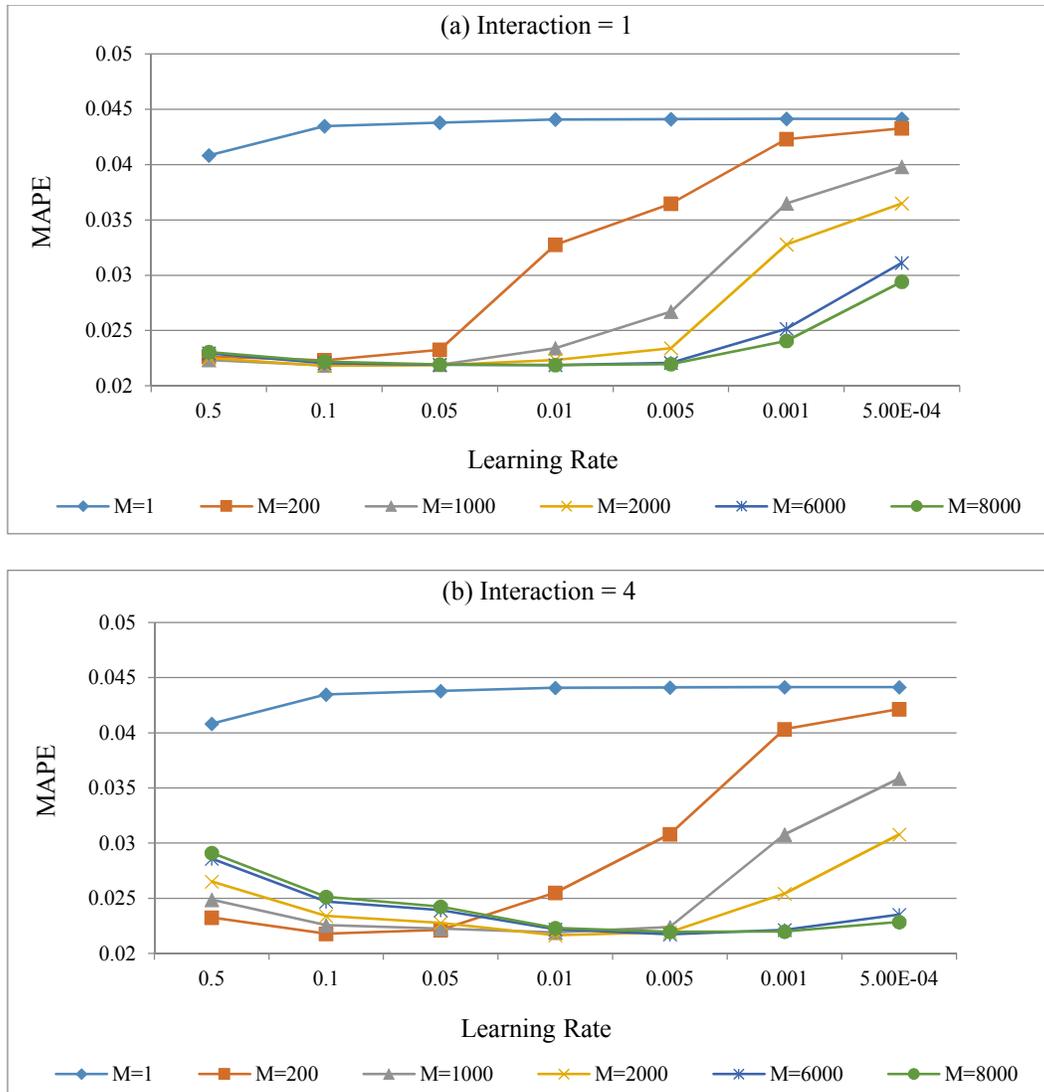


Figure 18. MAPE against Learning Rate for Models Fitted with Various Numbers of Trees and Different Levels of Interactions

Tree complexity, or variable interaction (C), also influences model performance as shown in Figure 19. With $J = 0.5$ (Figure 19 (a)) and many trees fitted, the MAPE value increases as C increases. This increasing rate of MAPE value (the slope of the lines) becomes more obvious with higher M value. This is because a higher J value makes each individual base model share a higher contribution to the

model output, with many trees fitted and more complex individual trees, over-fitting easily occurs. By reducing $J = 0.0005$ (Figure 19 (b)), the MAPE value decreases as C increases. This is partly due to the fact that the detailed information from the data can be modelled with a higher level of variable interaction. So, a smaller learning rate restricts each additional tree's contribution to the model and prevents over-fitting.

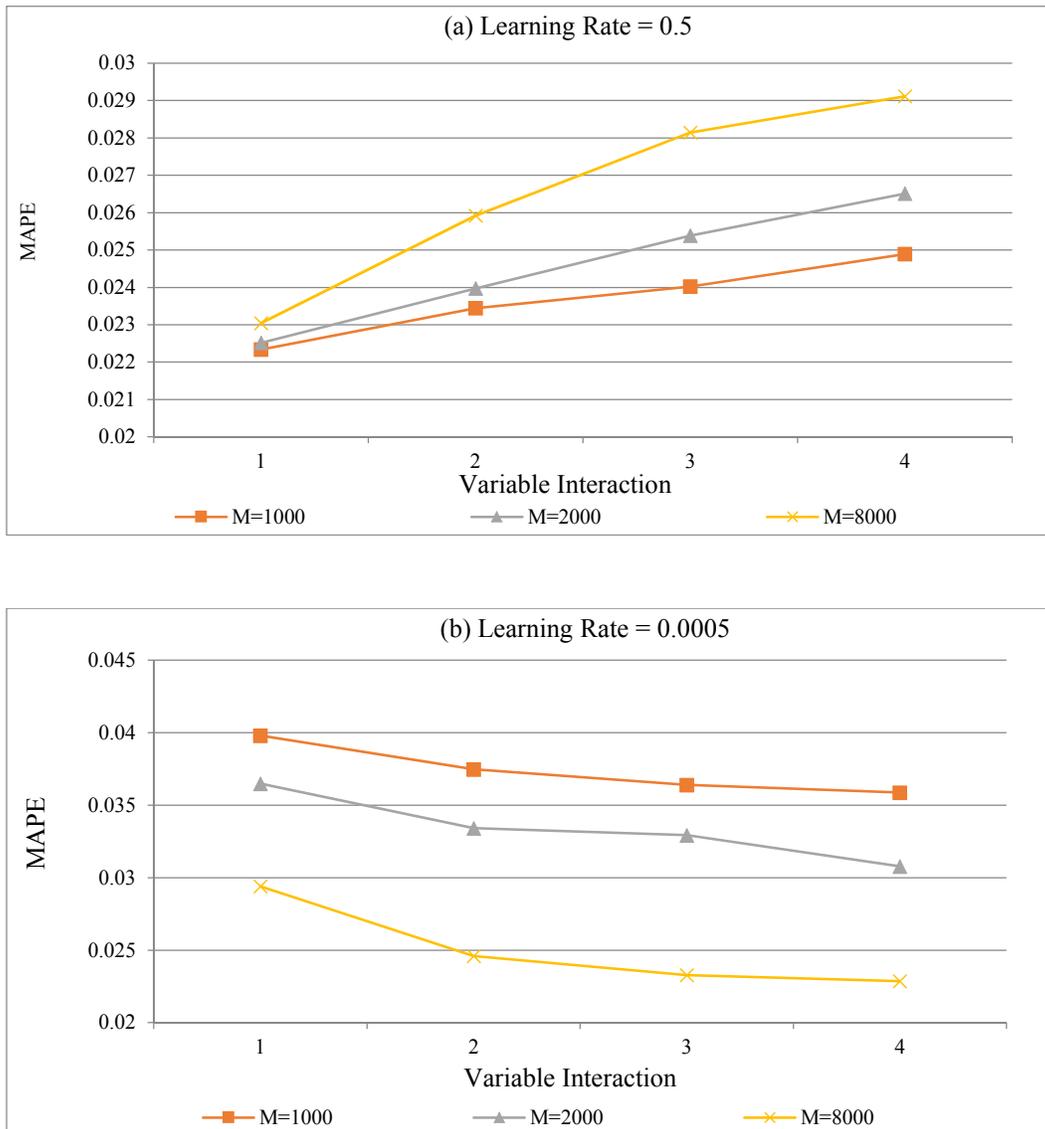


Figure 19. MAPE against Tree Complexity for Models Fitted with Various Numbers of Trees and Different Learning Rate

In general, a slower learning rate with a larger number of trees in the model is preferable to a faster learning rate with a smaller number of trees. A slower learning rate shrinks the contribution of each tree more and therefore allows smoother approach to the optimal performance and provides more reliable prediction results. However, when a large number of trees are fitted, model complexity also increases and requires more computational time. There is a need to consider the tradeoff between prediction accuracy and computational time. In addition, the level of variable interaction also affects the optimal selection of learning rate and number of trees. A higher level of variable interaction leads to a more complex model and requires fewer trees to be fitted with a given learning rate.

4.3.3 Model Interpretation

Inputs of the model, or predictor variables, usually have different influences on the output (response variable). To explore the individual input variable's influence on the response variable, we can gain better insight into the data. Breiman et al. (1984)[118] proposed a method to measure the relative influence of each predictor variable on the model output for a single decision tree. This relative influence of a predictor variable is measured based on the number of times this variable is selected to split a currently terminal region (or node) into two sub-regions, weighted by the least-squares improvement for the model as a result of this split. Friedman (2001) [115] generalized this criterion to additional tree expansions by simply averaging this criterion over all trees. The relative influence of each individual variable is scaled so that the sum of them for all the input variables equals to 100. A higher value indicates stronger influence of the input variable to the model.

Table 7 gives the relative influence of each input variable to the model output with different prediction horizons. It is obvious that each input variable contributes differently to the response (output of the model). For all three cases, the immediate previous travel time TT_{t-1} contributes the most to the predicted travel time. This is expected, as the immediate previous traffic condition will influence traffic in the near future. Therefore, it is closely related with future travel time. The changing rate of travel time ΔTT_{t-1} also has higher influence on the model output. ΔTT_{t-1} is the difference of travel time between every two consecutive time steps. It indicates the changing behavior of traffic. For example, a positive value of ΔTT_{t-1} indicates the increasing trend of travel time and higher positive values are highly correlated with congestion.

Another interesting result indicated in Table 7 is that the influence of time of day becomes more significant when we increase the prediction horizon. For a prediction 30 minutes ahead, the time of the day variable contributes 17.43% to the model output. The time of day variable is associated with the periodic feature of travel time: travel time usually increases during peak hours and maintains at a certain value during non-peak hours. With increased prediction horizon, immediate past traffic information cannot be provided and the impact of the most recent available travel time become less significant. More information can be obtained through the level of travel time during certain time interval. Therefore, contribution of the time of day variable increases with increased prediction horizon.

Table 7 Relative Influence of Input Variables for GBM Models with Learning Rate of 0.001 for Multistep-ahead Prediction

Variable	5-Min (1-Step) Ahead		15-Min (3-Step) Ahead		30-Min (6-Step) Ahead			
	Relative		Rank	Relative		Rank	Relative	
	Rank	Importance		Importance	Importance			
TT_{t-1}	1	96.34%	1	81.47%	1	66.33%		
TT_{t-2}	3	0.26%	7	0.48%	8	0.89%		
TT_{t-3}	7	0.04%	9	0.27%	7	1.63%		
ΔTT_{t-1}	2	2.95%	3	6.29%	3	7.61%		
ΔTT_{t-2}	9	0.02%	6	0.51%	9	0.1%		
ΔTT_{t-3}	6	0.09%	8	0.27%	10	0.08%		
Time of day	4	0.16%	2	7.35%	2	17.43%		
Day	10	0%	10	0.26%	6	1.65%		
Week	5	0.1%	5	1.05%	5	2.04%		
Month	8	0.04%	4	2.06%	4	2.24%		

Since TT_{t-1} and ΔTT_{t-1} are two most important variables for one-step-ahead prediction, we plot their joint influence on model output (Figure 20). The x-axis defines the lag one difference ΔTT_{t-1} , the y-axis is the lag one travel time TT_{t-1} , and the z-axis represents the predicted travel time \widehat{TT}_t . As indicated in Figure 20, there is a positive correlation between the lag one travel time TT_{t-1} and the predicted travel time \widehat{TT}_t . Traffic conditions are often consistent within a short period. If congestion occurs, it will last for several minutes or even hours. Therefore, a high value of travel time at time step $t - 1$ would more likely followed by another high value of travel time. The influence of travel time difference ΔTT_{t-1} on \widehat{TT}_t is more evident if ΔTT_{t-1}

is a positive value. In other words, if previous travel time continues to increase, the future travel time will also tend to increase. But, if the previous travel time decreases or maintains a certain value, the difference of travel time ΔTT_{t-1} will have less impact on future travel time. In general, higher values of both TT_{t-1} and ΔTT_{t-1} often indicate that congestion occurs. Therefore, they are more likely to be followed by a higher travel time value.

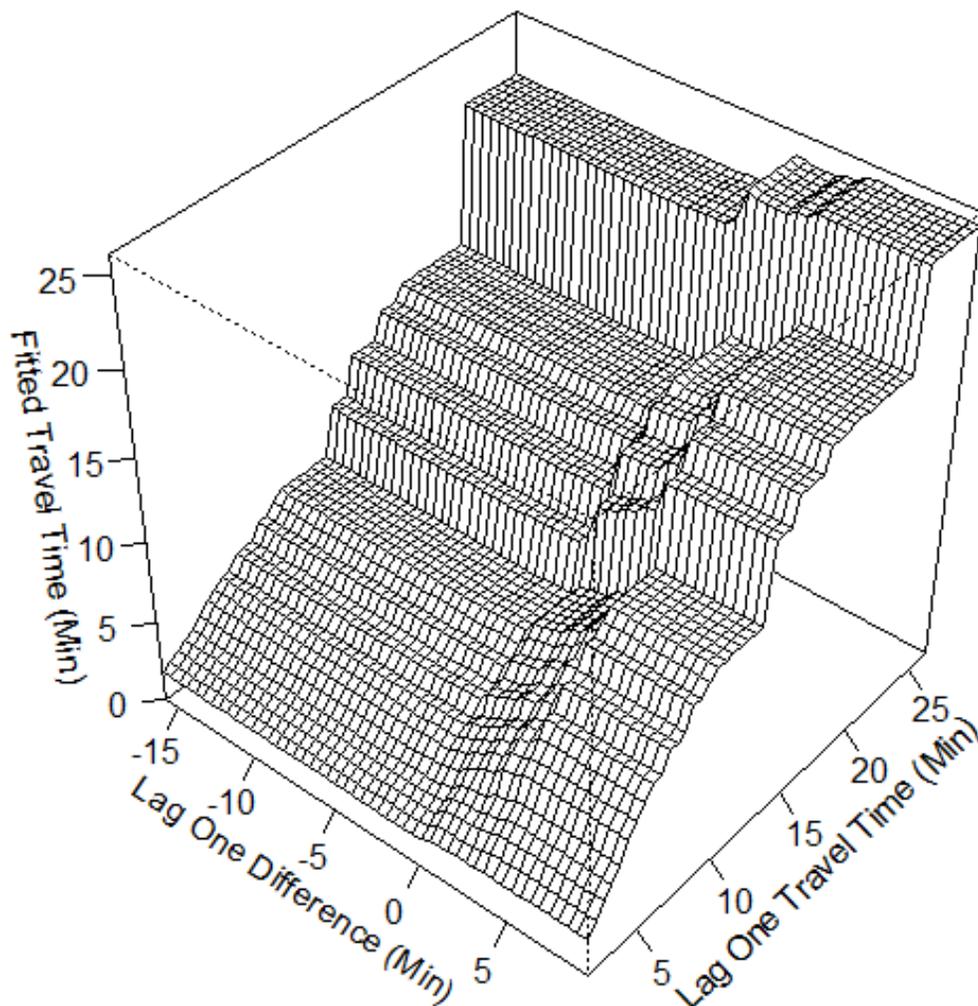


Figure 20. Three-dimensional Plots for the Joint Effects of Lag One Difference and Lag One Travel Time on Predicted Travel Time Value

4.3.4 Model Comparison

To test the effectiveness of the GBM model, this section comprehensively evaluates prediction performance of the ARIMA, the RF and the GBM models. The ARIMA model is one of the widely recognized benchmark models for traffic parameters forecasting. Prediction is based on regression of its current and past values. Optimization of the ARIMA model involves order selection and parameter estimation. Detailed information on theoretical background and steps in fitting an ARIMA model can be found in [96].

We use two months' training data and seven days' testing data to compare these three models. Prediction accuracy of the three models are compared on the basis of 5-minute (1-step-ahead), 15-minute (3-step-ahead) and 30 minute (6-step-ahead) ahead prediction. We test different combinations of variables for both the GBM and the RF models during the training process and select the best model according to their MAPE values. The best orders of the ARIMA model are selected based on the method proposed by Hyndman and Khandakar [95], and the parameters of the ARIMA model are estimated based on maximum likelihood method. Table 8 through Table 10 compare these three models' prediction performance based on MAPE value. Both the RF and the GBM model outperform the ARIMA model in the one-step-ahead prediction as shown in Table 8. The GBM and the RF perform similarly, where RF has slightly lower MAPE value. As the prediction horizon increases, all three models' performances drop. Comparatively, the GBM method is less sensitive to prediction horizon and maintains a good prediction performance. As indicated in Table 9 and Table 10, the GBM model outperforms both ARIMA and RF models, 9

out of 10 cases. In general, the RF and the GBM methods perform better than the ARIMA model in the one-step-ahead prediction. By increasing prediction horizons, the difference among these three models becomes obvious, with the GBM model being the most accurate, compared with the RF and the ARIMA models. Therefore, we conclude that both the RF and the GBM models are promising algorithms in travel time prediction as they are more accurate compared with the ARIMA model. The advantage of the GBM model becomes even obvious in multi-step-ahead predictions.

Table 8 Comparison of 5 Minutes Ahead Prediction for ARIMA, RF and GBM

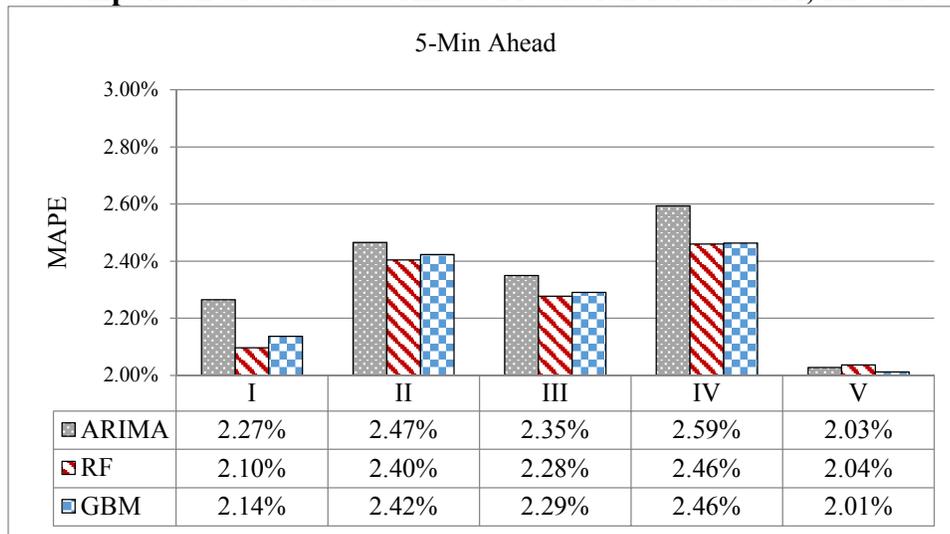


Figure 21 shows three days' forecasting results provided by the GBM model. The blue line stands for the predicted travel time by the GBM model, the red cross stands for the observed (original) value of travel time. We could see that the overall performance of the GBM model is good not only in normal traffic conditions (lower panel of Figure 21), but also is effective during traffic transitional period. On 3/1/2012 (upper panel of Figure 21), there are three recorded incidents between 6:31 and 7:16, which are possible reasons for the traffic state changing from uncongested to congested.

Table 9 Comparison of 15 Minutes Ahead Prediction for ARIMA, RF and GBM

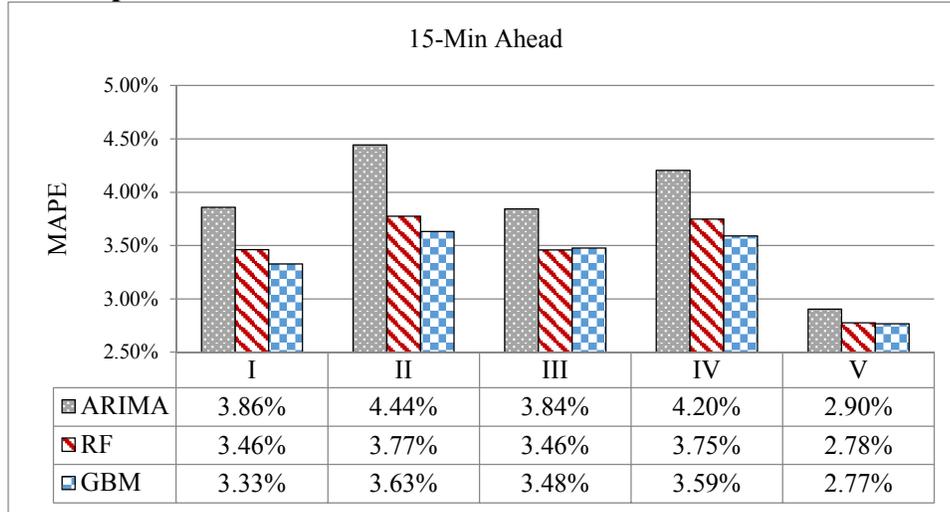
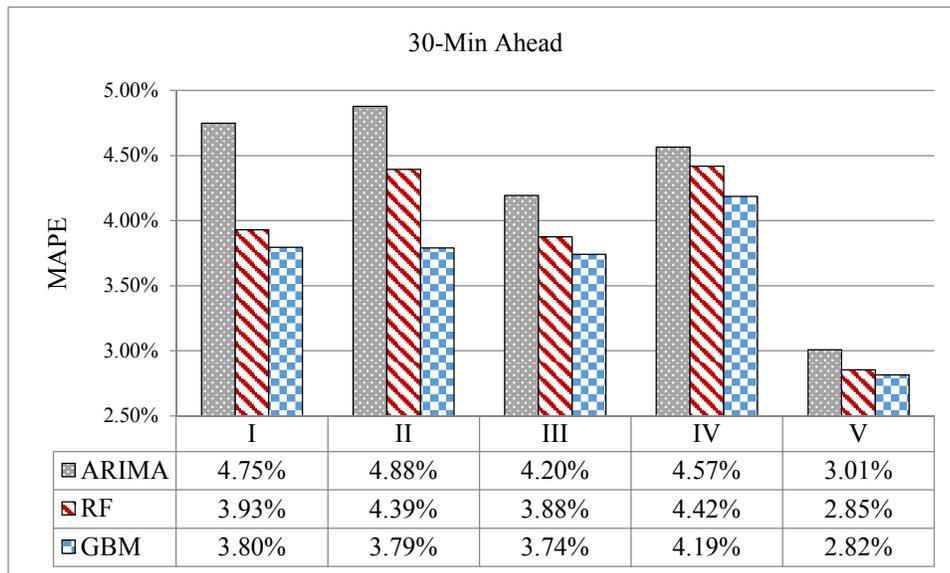


Table 10 Comparison of 30 Minutes Ahead Prediction for ARIMA, RF and GBM



The GBM model is able to capture this sudden change. On 3/2/2012 (middle panel of Figure 21), rain begins falling shortly after 16:00 and ends at 19:00. Congestion occurs during this period and travel time increases. The GBM model also adequately captures this congestion. Theoretically, the GBM model is able to handle complex interactions among input variables and can fit complex nonlinear

relationship. Therefore, the GBM model is able to model non-linear characteristics of dynamic traffic systems and leads to superior prediction performance.

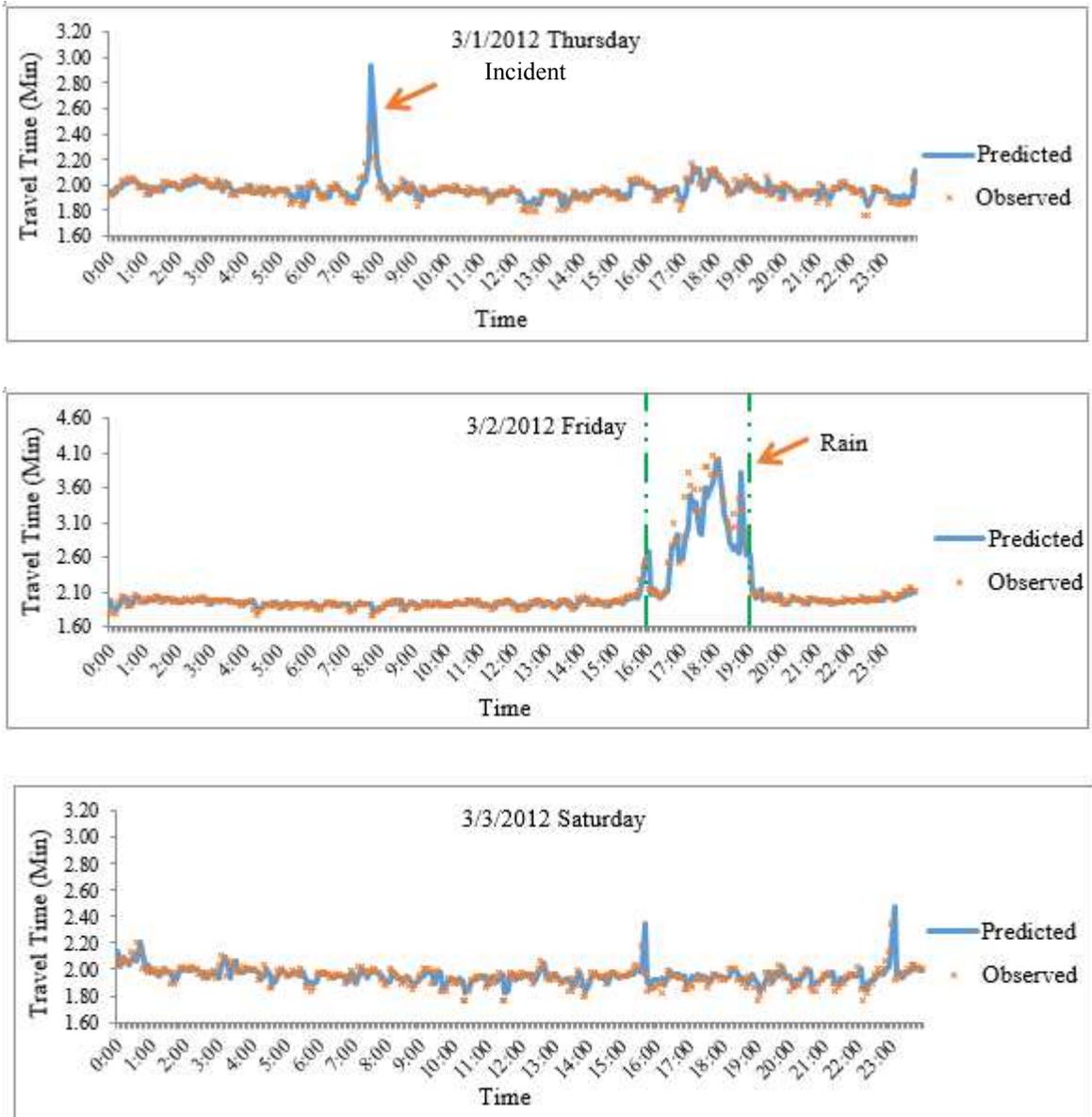


Figure 21. Sample Travel Time Prediction Results of the GBM method

4.3.5 Discussion and Conclusion

The GBM model has its unique feature that distinguishes it from other popular ensemble methods, such as bagged trees and random forests. Both bagged trees and random forests are able to reduce variance more than single trees through averaging. Random forests enhance diversity through randomly selecting a subset of variables at each splitting node. The training sample is produced from bootstrap sampling with its distribution similar to the original training set. The bias of the model cannot be reduced through averaging. On the other hand, the GBM model grows trees sequentially by adjusting the weight of the training data distribution to minimize certain loss function. It reduces both model bias through forward stage-wise modeling and reduces variance through averaging[107]. The proposed GBM-based travel time prediction method has considerable advantages over classical statistical approaches and other ensemble methods. Especially, it has superior performance in terms of prediction accuracy.

There are limited studies discussed the RF model application in traffic prediction [70, 71]. To the best knowledge of the author, we did not find any studies on the application of the GBM model with freeway travel time data. There is no comparison and discussion on the performance of the RF and the GBM models in traffic prediction. The GBM model can handle sharp discontinuities, an important feature when modeling abnormal traffic conditions where traffic states change from uncongested to congested, and vice versa. Based on data used in this study, the GBM model is able to capture the sudden changes of traffic (for example, in incidents or raining conditions). In addition, the GBM model can automatically select relevant

variables, fit accurate models, and identify and model parameter interactions. More importantly, different from other machine learning algorithms as a ‘black-box’, the relative importance or contribution of input variables are also discussed through the GBM model. This is critical for us to get powerful insight into the structure of the data. In addition, comparisons of the ARIMA, the RF and the GBM models indicate that both the RF and the GBM models outperform the conventional statistical model, ARIMA model; the advantage of the GBM becomes more evident for multi-step-ahead prediction.

One issue regarding the application of the GBM model in travel time prediction is related with parameter optimization. As addressed in the model optimization section, the performance of the GBM model is largely influenced by its parameters, including number of trees, learning rate and tree complexity (variable interactions). Therefore, there is a need to test the optimal combination of variables when developing the GBM model. Computational time is another issue when tree complexity or the number of trees increases. The tradeoff between computation cost and model accuracy should also be considered when building the model.

In short, the GBM model has its considerable advantages in freeway travel time prediction. In particular, as traffic data becomes more readily available, more information can be accessed to study traffic phenomena. The capability of the GBM model in handling different types of input variables, in modeling complex nonlinear relationship makes it a promising algorithm for travel time prediction.

Chapter 5: A Travel Time Prediction Framework

The previous two chapters discussed the application of different volatility models and ensemble methods in travel time prediction. The volatility model is aimed at improving the reliability forecasting (the variance part), while the ensemble methods are aimed at increasing the prediction accuracy of the mean part. From the discussion of Chapter 3, we know that the stochastic volatility model outperforms the traditional GARCH model in terms of constructing more efficient and effective PIs. Chapter 4 demonstrated the advantage of the GBM model in improving travel time prediction accuracy. In this chapter, we aim at developing a travel time prediction framework to improve both prediction accuracy and reliability.

5.1 Model Development

In Chapter 3, we mentioned that the observed travel time can be decomposed into a conditional mean (u_t) and a residual (r_t) component:

$$x_t = u_t + r_t \quad (50)$$

where x_t is the observed travel time at time t , u_t represents the estimated conditional mean, and r_t is the residual part. To develop an accurate travel time prediction framework, we need to carefully select both the mean model and the model for the residual part. The ARIMA mean model was used in Chapter 3 to estimate and predict the mean part of travel time for comparison purposes. This does not necessarily indicate the ARIMA model is the best model. Since we demonstrated the superior prediction performance of the GBM model (Chapter 4) in predicting the mean part of the travel time data series, we replaced the ARIMA model with the GBM model to

predict the mean part u_t of the time series and then applied the stochastic volatility model in modeling the variance part of the data. We call the newly developed prediction framework as the GBM-SV model. The ultimate goal is to further improve the prediction accuracy.

5.2 Data Description and Preparation

To comprehensively compare the proposed model's performance, this study uses travel time data from three different freeway stretches. As shown in

Figure 22, Figure 23, Figure 24, the selected freeway are the I95 southbound direction, the I495 eastbound direction and the MD295 Southbound direction. Each freeway stretch includes multiple segments, which are covered by multiple TMCs. Travel time information for individual TMC segment can be obtained from data provided by INRIX. Based on traffic information collected from these nine TMCs, path travel time for the three segments of study can be estimated. Details of calculating the path travel time for multiple TMC segments can be found in Hamed, et al. [49]. Then, the travel time information for each segment is aggregated into five minutes time intervals.

Table 11 provides detailed information for the 11 selected segments that includes: segment ID, road on which the segment is located, latitude and longitude of the start point and end point of the segment, length and the average travel time of each segment. For comparison purposes, we select the segments with similar length (most of the segments are around 2-4 miles). Average travel time of each segment is the average of five minutes travel times over the entire year.



Figure 22. Selected Study Segment at the I95 Southbound Direction

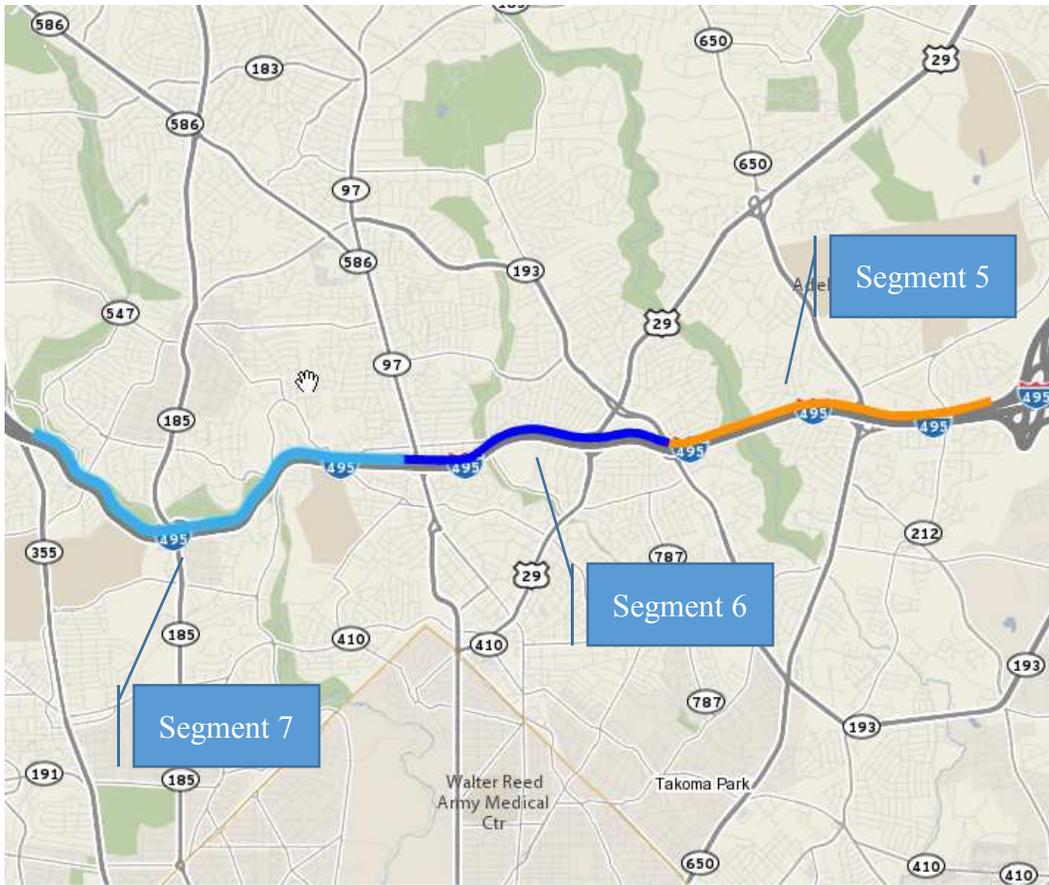


Figure 23. Selected Study Segment at the I495 Eastbound Direction

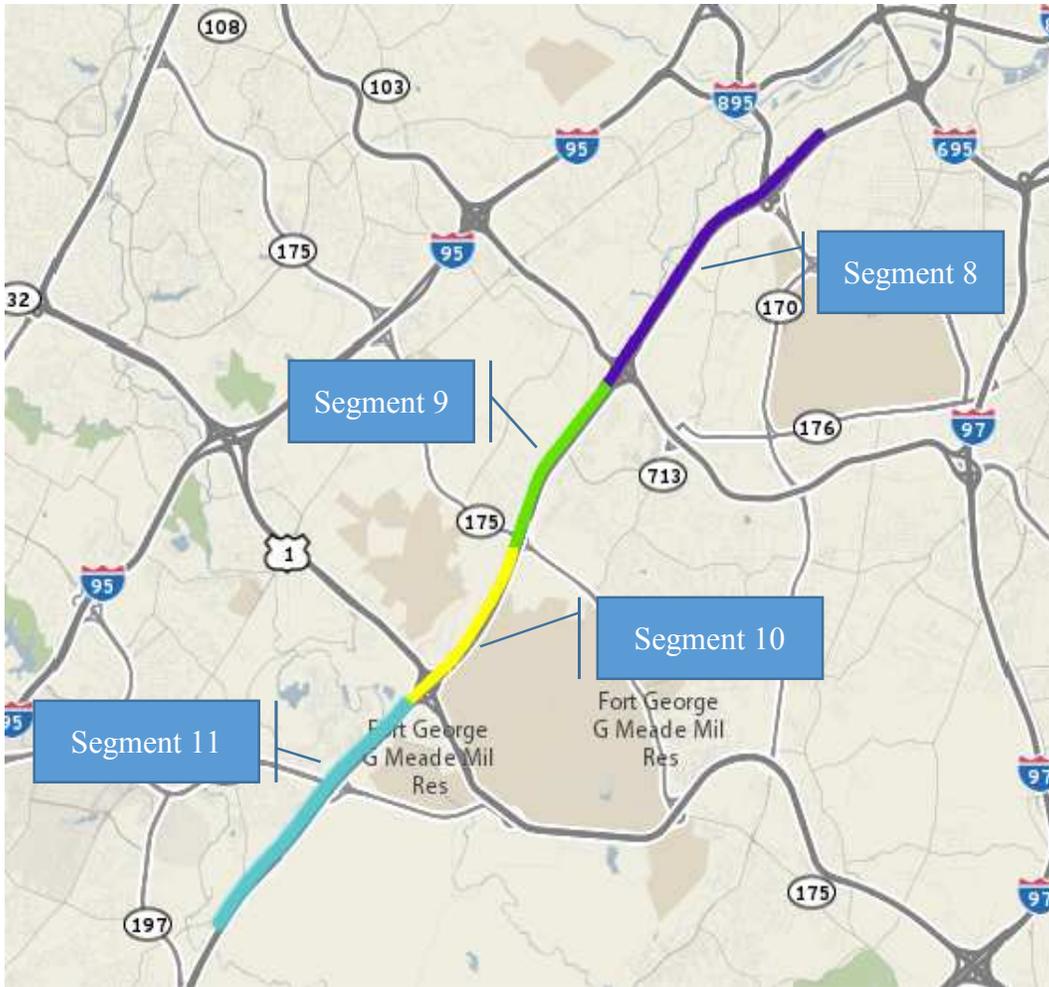


Figure 24. Selected Study Segment at the MD295 Southbound Direction

Table 11 Selected Freeway Segment Information

Segment	Road	Start		End		Length	Travel Time (min)
		Latitude	Longitude	Latitude	Longitude		
1	I95	39.21848	-76.7269	39.20084	-76.761	2.2	2.00
2	I95	39.20084	-76.761	39.17537	-76.7946	2.53	2.31
3	I95	39.17537	-76.7946	39.15624	-76.8348	2.56	2.32
4	I95	39.15624	-76.8348	39.12223	-76.8675	2.99	2.65
5	I495	39.02016	-76.9582	39.01534	-77.0051	2.59	3.43
6	I495	39.01534	-77.0051	39.01356	-77.0453	2.25	2.64
7	I495	39.01356	-77.0453	39.01631	-77.0981	3.43	3.61
8	MD295	39.21099	-76.6823	39.16517	-76.7361	4.26	4.06
9	MD295	39.16517	-76.7361	39.13714	-76.7573	2.28	2.35
10	MD295	39.13714	-76.7573	39.11021	-76.7839	2.37	2.56
11	MD295	39.11021	-76.7839	39.06913	-76.8316	3.84	4.00

5.3 Model Comparison

To test the GBM-SV model's prediction performance, this section evaluates the GBM-SV with the ARIMA-GARCH model by using travel time data from 11 freeway segments. We use two months' training data to develop the model and then predict travel time for different days. Prediction accuracy of the two models are compared on the basis of 5-minute (1-step-ahead), 10-minute (2-step-ahead), 15-minute (3-step-ahead), 20-minute (4-step-ahead), 25-minute (5-step-ahead), and 30 minute (6-step-ahead) ahead prediction using 11 freeway segments. The model selection procedure (parameter selection and optimization of the GBM and the stochastic volatility model) is the same as discussed in Chapter 3 and Chapter 4. To

compare the prediction performance of the GBM-SV model, the ARIMA-GARCH model is used as the bench mark model.

Five criterions are introduced to evaluate the model performance. The MAPE and RMSE criterion are used to measure the prediction accuracy of the mean part, while the MPIL, PICP and the PI ratio are used to measure the effective and efficiency of the PIs. Please refer to section 3.1.3 and 3.2.3 for the definition of the MAPE, RMSE, MPIL and PICP criterions. In this section, a new PIs performance measurement is introduced: PI ratio.

$$PI\ ratio = \frac{PICP}{\left(\frac{MPIL}{MTT}\right)} \quad (51)$$

where *PICP* and *MPIL* are the coverage probability and average length of the PIs of the model, *MTT* is mean of travel time. The $\frac{MPIL}{MTT}$ indicates the ratio of the average PIs length over the average value of travel time. Since we prefer a larger value of *PICP* with a smaller value of $\frac{MPIL}{MTT}$ ratio, the higher value of the PI ratio, the better the constructed PIs are.

Both the GBM-SV and the ARIMA-GARCH models are evaluated according to the above mentioned five criterions. In this section, we randomly select five days to predict their travel time: 2012-11-08, 2012-11-13, 2012-11-26, 2012-12-05, 2012-12-19. Table 12 - Table 36 provide the MAPE, RMSE, MPIL, PICP and PI-Ratio criterions for prediction results of the 11 freeway segments during these five days periods and the prediction horizons are from 1-step-ahead up to 6-step-ahead. Both the MAPE and the RMSE criterions measure the prediction accuracy of the model (the mean part). The lower values of both criterions indicate more accurate predictions of the mean values of the time series. By closely examining the MAPE

and the RMSE values for the GBM-SV and ARIMA-GARCH under different scenarios (Different freeway segment, number of steps ahead prediction and days), we could see from the following tables that the GBM-SV model produces lower MAPE and RMSE values in most of the cases. There are only a few cases that the ARIMA-GARCH model provides better performances over the GBM-SV model. In the following MAPE and RMSE tables, we highlighted (values in boldface) the cases that the ARIMA-GARCH model has a lower value compared with the GBM-SV model. There is only a small number of highlighted cases, which is to say that the GBM-SV model performs better in most cases. Therefore, we could conclude that the GBM-SV model provides more accurate predictions of the mean part of the time series.

The next step is to measure the quality of the constructed prediction intervals for each model. Here, we use the PICP, MPIL and the PI-Ratio criteria to measure the quality of the PIs. As we explained in section 3.2.3, a prediction interval of higher PICP value and lower MPIL value is desirable. When comparing PIs of two different models, we have to consider PICP and MPIL values of each model at the same time. As we know that a higher MPIL value (a wider PI) may potentially be related with a higher PICP value, and vice versa. In order to compare both measurements at the same time, the PI-Ratio criterion is introduced here. According to the definition of equation (51), the PI-Ratio considers both the coverage probability and the length of the PIs. In general, the higher value of the PI-Ratio, the better the model is in terms of the quality of the PIs. By looking at the PI-Ratio tables of the GBM-SV and the ARIMA-GARCH model during different days, we could also see that the GBM-SV

model provides higher PI-Ratio in most cases, in other words, the GBM-SV model constructs more efficient and effective PIs in most cases. Only for a few cases, the ARIMA-GARCH model provides higher PI-Ratio (We also highlighted the cases when the ARIMA-GARCH model provides better PIs).

In order to have a straight forward comparison of the GBM-SV and ARIMA-GARCH models' performance, Figure 25 through Figure 29 summarize the models' performance by averaging the performance over 11 segments and 5 days. In terms of mean performance, the average MAPE and RMSE values of the GBM-SV model are lower than the ARIMA-GARCH model. By increasing the number of steps ahead prediction, the advantages of the GBM-SV model become more significant. When comparing the PIs constructed by both models, the GBM-SV also shows its advantage as it has higher PICP and PI-ratio values while has lower MPIL value at the same time.

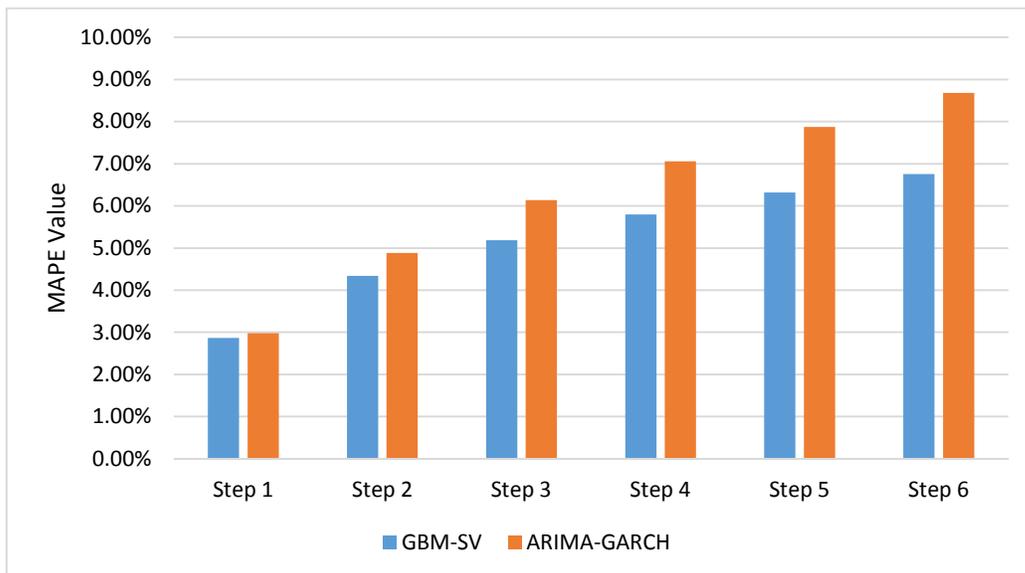


Figure 25 Average MAPE Value over the 11 Segments and 5 Days

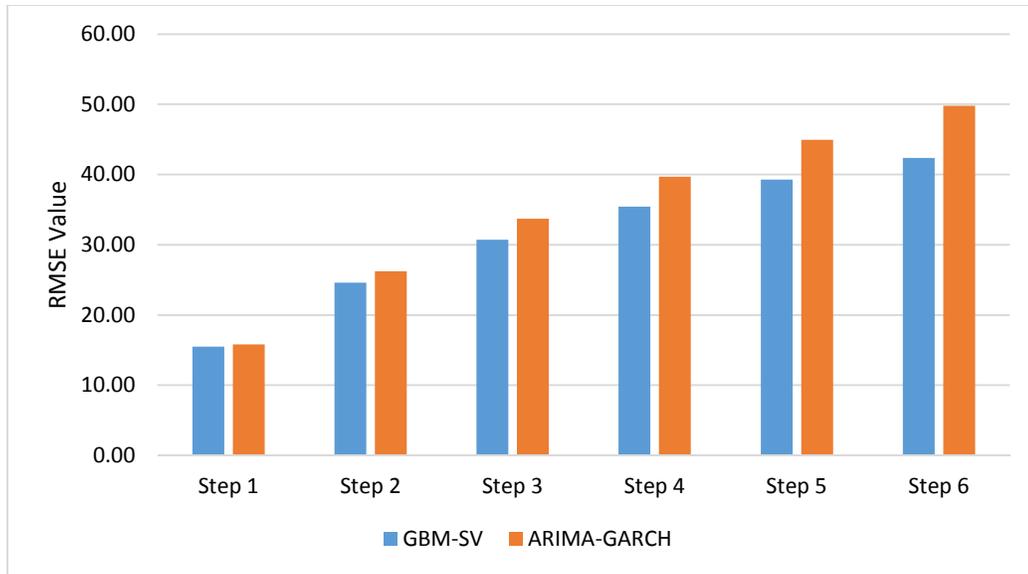


Figure 26 Average RMSE Value over the 11 Segments and 5 Days

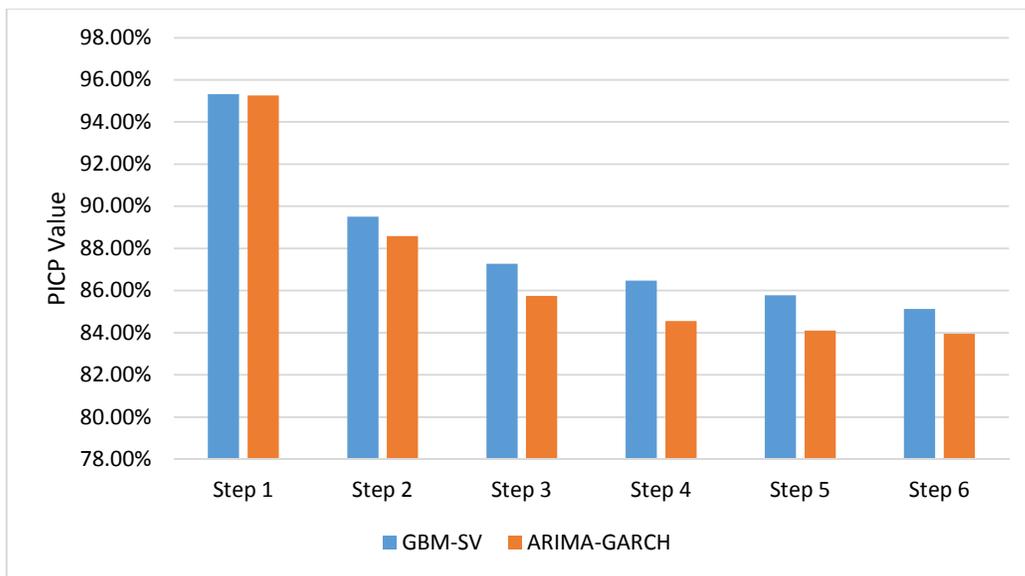


Figure 27 Average PICP Value over the 11 Segments and 5 Days

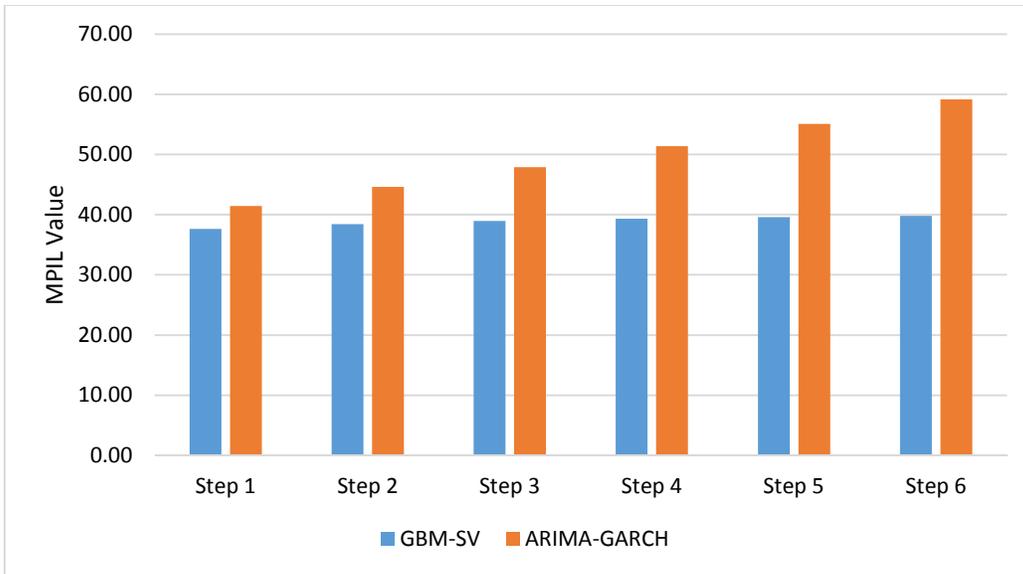


Figure 28 Average MPIL Value over the 11 Segments and 5 Days

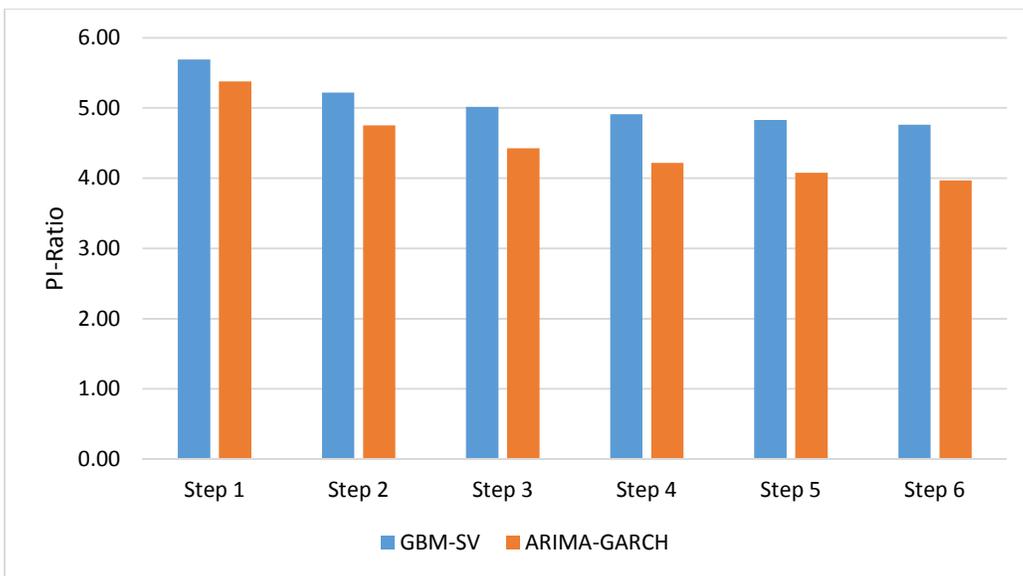


Figure 29 Average PI-Ratio over the 11 Segments and 5 Days

To sum up, by comparing the GBM-SV and the ARIMA-GARCH models' performance under different scenarios (combination of 11 freeway segments, 5 days and 6 steps ahead prediction), the GBM-SV model shows its superior prediction performance over the traditional ARIMA-GARCH model in terms of both the

prediction accuracy (prediction of the mean part) and the reliability of the prediction (the quality of the constructed PIs).

Table 12 Comparing the MAPE Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-08 Thursday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	2.51%	3.49%	3.92%	4.25%	4.45%	4.50%
	ARIMA-GARCH	2.41%	3.76%	4.44%	4.99%	5.49%	5.87%
2	GBM-SV	2.70%	4.12%	4.96%	5.66%	5.85%	5.93%
	ARIMA-GARCH	2.72%	4.21%	5.22%	5.92%	6.27%	6.57%
3	GBM-SV	2.34%	3.18%	3.53%	3.66%	3.82%	3.83%
	ARIMA-GARCH	2.39%	3.45%	3.87%	4.12%	4.23%	4.53%
4	GBM-SV	2.48%	3.77%	4.53%	5.19%	5.71%	6.00%
	ARIMA-GARCH	2.48%	3.98%	4.89%	5.55%	6.17%	6.68%
5	GBM-SV	4.03%	6.58%	8.42%	10.05%	11.32%	12.49%
	ARIMA-GARCH	4.01%	7.16%	9.55%	11.52%	13.44%	15.42%
6	GBM-SV	4.43%	7.40%	9.53%	10.91%	12.28%	14.00%
	ARIMA-GARCH	4.33%	7.84%	11.12%	13.78%	16.43%	19.20%
7	GBM-SV	3.57%	5.49%	6.97%	8.35%	9.66%	11.28%
	ARIMA-GARCH	3.83%	6.21%	8.40%	10.77%	13.02%	15.72%
8	GBM-SV	1.95%	2.93%	3.25%	3.37%	3.42%	3.40%
	ARIMA-GARCH	1.79%	2.64%	2.87%	2.95%	2.94%	2.91%
9	GBM-SV	4.07%	5.77%	6.87%	7.63%	8.72%	9.70%
	ARIMA-GARCH	4.32%	6.68%	8.46%	9.28%	10.00%	10.97%
10	GBM-SV	4.30%	6.97%	8.95%	10.78%	11.87%	13.03%
	ARIMA-GARCH	4.64%	8.13%	10.73%	13.16%	15.51%	17.61%
11	GBM-SV	4.93%	8.56%	11.49%	13.75%	15.71%	17.27%
	ARIMA-GARCH	5.45%	9.53%	13.30%	16.19%	18.85%	21.24%

Table 13 Comparing the RMSE Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-08 Thursday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	6.41	9.71	11.64	13.60	14.56	14.97
	ARIMA-GARCH	5.85	9.98	12.67	14.97	16.73	17.82
2	GBM-SV	7.92	13.09	16.36	19.06	19.02	19.14
	ARIMA-GARCH	7.61	12.96	16.75	19.36	21.37	22.79
3	GBM-SV	5.48	8.41	10.42	11.35	11.46	11.62
	ARIMA-GARCH	5.84	8.48	9.91	10.87	11.24	11.83
4	GBM-SV	12.06	18.80	23.32	27.27	31.16	34.16
	ARIMA-GARCH	11.47	20.99	28.58	34.06	38.52	42.47
5	GBM-SV	16.45	28.24	38.12	45.56	50.69	56.23
	ARIMA-GARCH	17.27	30.50	40.39	47.59	53.42	59.65
6	GBM-SV	26.46	47.22	62.31	69.94	77.24	85.63
	ARIMA-GARCH	24.06	43.40	57.84	67.98	78.04	87.00
7	GBM-SV	46.58	73.68	102.83	131.74	152.06	171.46
	ARIMA-GARCH	45.48	73.01	101.04	133.53	157.92	187.33
8	GBM-SV	8.56	14.99	17.76	18.77	19.17	18.36
	ARIMA-GARCH	7.68	12.01	13.53	13.76	13.69	13.52
9	GBM-SV	16.36	24.08	28.80	32.24	35.04	37.85
	ARIMA-GARCH	16.59	25.20	31.01	34.40	37.38	41.12
10	GBM-SV	31.90	50.08	63.87	73.66	80.72	85.59
	ARIMA-GARCH	33.02	54.54	71.97	84.35	94.45	100.08
11	GBM-SV	54.14	89.54	116.77	135.89	150.58	164.29
	ARIMA-GARCH	56.93	92.92	123.24	146.46	163.55	178.74

Table 14 Comparing the MPIL Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-08 Thursday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	17.35	17.34	17.32	17.29	17.25	17.20
	ARIMA-GARCH	17.42	17.81	18.06	18.21	18.29	18.33
2	GBM-SV	19.35	18.92	18.56	18.24	18.05	17.87
	ARIMA-GARCH	22.50	22.89	23.16	23.36	23.51	23.65
3	GBM-SV	16.78	16.65	16.52	16.40	16.29	16.18
	ARIMA-GARCH	17.63	17.48	17.32	17.16	17.02	16.88
4	GBM-SV	20.45	19.72	19.13	18.73	18.49	18.27
	ARIMA-GARCH	22.21	21.71	21.13	20.56	20.06	19.63
5	GBM-SV	56.70	59.66	62.17	64.26	66.16	67.95
	ARIMA-GARCH	53.39	57.60	61.86	66.21	70.69	75.34
6	GBM-SV	58.82	58.52	58.04	57.63	57.33	56.72
	ARIMA-GARCH	64.38	72.16	80.64	89.93	100.16	111.52
7	GBM-SV	75.73	73.58	71.45	69.67	67.99	66.24
	ARIMA-GARCH	88.45	92.98	97.56	102.17	107.00	112.13
8	GBM-SV	29.88	31.19	32.18	32.88	33.50	33.93
	ARIMA-GARCH	26.19	27.20	27.83	28.23	28.49	28.65
9	GBM-SV	37.96	37.65	37.24	36.79	36.38	35.83
	ARIMA-GARCH	45.88	47.32	48.75	50.19	51.62	53.05
10	GBM-SV	71.03	72.94	74.08	74.31	74.78	74.76
	ARIMA-GARCH	76.81	83.12	89.87	97.06	104.73	112.91
11	GBM-SV	123.20	124.64	124.86	123.62	122.29	119.78
	ARIMA-GARCH	155.06	186.17	222.57	265.60	316.42	376.61

**Table 15 Comparing the PICP Values of the GBM-SV and the ARIMA-GARCH
Models (2012-11-08 Thursday)**

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	95.83%	90.63%	88.54%	88.19%	87.15%	86.11%
	ARIMA-GARCH	94.10%	88.89%	87.15%	86.11%	82.99%	83.68%
2	GBM-SV	92.36%	82.99%	81.25%	78.13%	79.51%	79.17%
	ARIMA-GARCH	94.79%	86.46%	84.03%	83.33%	81.25%	80.56%
3	GBM-SV	91.67%	84.38%	84.03%	83.68%	83.68%	80.90%
	ARIMA-GARCH	92.71%	86.11%	81.25%	80.21%	79.86%	77.78%
4	GBM-SV	93.75%	85.76%	84.38%	80.21%	80.21%	79.86%
	ARIMA-GARCH	92.71%	85.76%	81.94%	82.99%	79.86%	80.90%
5	GBM-SV	94.44%	89.24%	84.72%	81.94%	81.94%	79.17%
	ARIMA-GARCH	94.44%	88.19%	80.90%	76.39%	73.26%	72.22%
6	GBM-SV	95.14%	86.11%	82.99%	80.90%	81.60%	79.17%
	ARIMA-GARCH	93.06%	85.42%	79.17%	78.13%	76.39%	74.65%
7	GBM-SV	94.44%	87.15%	85.07%	83.68%	82.64%	80.90%
	ARIMA-GARCH	93.75%	86.11%	85.07%	82.29%	84.03%	83.68%
8	GBM-SV	95.83%	90.28%	90.63%	90.28%	91.67%	90.28%
	ARIMA-GARCH	94.44%	89.24%	88.89%	89.93%	90.28%	91.67%
9	GBM-SV	93.75%	89.24%	87.15%	86.81%	84.03%	81.60%
	ARIMA-GARCH	94.10%	86.46%	84.03%	80.56%	79.17%	80.21%
10	GBM-SV	95.49%	88.89%	84.38%	82.29%	81.60%	80.56%
	ARIMA-GARCH	94.79%	89.24%	85.07%	83.68%	83.33%	83.68%
11	GBM-SV	94.79%	84.72%	80.56%	77.08%	74.65%	73.61%
	ARIMA-GARCH	95.83%	89.93%	87.85%	87.15%	86.81%	86.81%

**Table 16 Comparing the PI-Ratio of the GBM-SV and the ARIMA-GARCH
Models (2012-11-08 Thursday)**

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	6.79	6.40	6.25	6.23	6.16	6.09
	ARIMA-GARCH	6.67	6.16	5.96	5.84	5.60	5.64
2	GBM-SV	6.81	6.25	6.23	6.09	6.26	6.28
	ARIMA-GARCH	6.00	5.36	5.12	5.02	4.85	4.77
3	GBM-SV	7.69	7.12	7.15	7.17	7.20	7.01
	ARIMA-GARCH	7.43	6.96	6.62	6.60	6.63	6.51
4	GBM-SV	7.71	7.28	7.35	7.11	7.20	7.21
	ARIMA-GARCH	7.01	6.60	6.44	6.68	6.57	6.78
5	GBM-SV	3.23	2.93	2.69	2.54	2.48	2.35
	ARIMA-GARCH	3.40	2.95	2.53	2.25	2.03	1.89
6	GBM-SV	3.06	2.77	2.66	2.59	2.60	2.52
	ARIMA-GARCH	2.74	2.23	1.83	1.61	1.40	1.22
7	GBM-SV	3.56	3.28	3.21	3.16	3.13	3.09
	ARIMA-GARCH	3.07	2.67	2.50	2.30	2.24	2.12
8	GBM-SV	7.73	6.98	6.79	6.63	6.61	6.43
	ARIMA-GARCH	8.69	7.89	7.68	7.65	7.61	7.67
9	GBM-SV	3.63	3.49	3.45	3.47	3.41	3.36
	ARIMA-GARCH	2.99	2.65	2.48	2.30	2.18	2.14
10	GBM-SV	2.43	2.18	2.01	1.94	1.89	1.86
	ARIMA-GARCH	2.26	1.96	1.73	1.57	1.45	1.36
11	GBM-SV	2.39	2.08	1.95	1.85	1.78	1.76
	ARIMA-GARCH	1.94	1.50	1.22	1.01	0.84	0.70

Table 17 Comparing the MAPE Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-13 Tuesday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	2.61%	4.13%	4.55%	5.01%	5.19%	5.28%
	ARIMA-GARCH	2.54%	4.08%	4.92%	5.59%	6.18%	6.60%
2	GBM-SV	2.59%	3.95%	4.67%	5.24%	5.57%	5.71%
	ARIMA-GARCH	2.52%	4.11%	4.97%	5.39%	5.80%	6.26%
3	GBM-SV	2.73%	4.10%	5.22%	5.56%	5.67%	5.81%
	ARIMA-GARCH	2.53%	4.00%	4.99%	5.71%	6.39%	6.92%
4	GBM-SV	3.37%	5.13%	6.41%	7.37%	8.32%	8.88%
	ARIMA-GARCH	3.48%	5.24%	6.99%	8.44%	9.67%	11.12%
5	GBM-SV	3.25%	5.16%	6.39%	7.34%	8.50%	9.57%
	ARIMA-GARCH	3.49%	6.21%	8.59%	10.86%	13.25%	15.56%
6	GBM-SV	3.22%	5.31%	6.80%	8.01%	8.91%	9.53%
	ARIMA-GARCH	3.39%	6.01%	8.49%	10.94%	12.71%	14.42%
7	GBM-SV	2.54%	4.02%	4.70%	4.95%	5.49%	5.97%
	ARIMA-GARCH	2.43%	3.99%	4.98%	5.50%	6.10%	6.90%
8	GBM-SV	2.96%	4.58%	5.47%	6.35%	7.07%	7.70%
	ARIMA-GARCH	2.90%	4.90%	6.19%	7.30%	8.29%	9.50%
9	GBM-SV	4.10%	6.25%	7.45%	8.16%	8.93%	9.62%
	ARIMA-GARCH	4.29%	6.94%	8.61%	10.05%	11.33%	12.66%
10	GBM-SV	4.01%	6.25%	7.43%	8.08%	8.76%	9.24%
	ARIMA-GARCH	3.91%	7.00%	9.05%	10.44%	11.29%	12.38%
11	GBM-SV	4.40%	6.99%	8.78%	10.39%	11.50%	12.19%
	ARIMA-GARCH	4.30%	7.20%	9.65%	11.41%	13.09%	14.48%

Table 18 Comparing the RMSE Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-13 Tuesday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	5.60	12.76	13.15	13.86	13.39	12.95
	ARIMA-GARCH	5.85	10.34	12.40	14.01	15.42	16.62
2	GBM-SV	7.14	11.06	13.48	14.64	15.52	15.92
	ARIMA-GARCH	7.27	11.54	14.09	14.90	15.88	17.14
3	GBM-SV	8.05	12.74	17.71	20.14	21.85	22.67
	ARIMA-GARCH	8.42	14.62	18.96	22.12	24.63	27.23
4	GBM-SV	27.04	37.97	47.39	60.99	68.63	73.84
	ARIMA-GARCH	27.52	41.99	58.04	74.25	88.42	106.17
5	GBM-SV	35.39	65.65	93.17	117.31	140.45	157.42
	ARIMA-GARCH	39.97	74.48	106.55	137.39	166.00	189.85
6	GBM-SV	33.56	49.80	73.43	95.10	110.21	122.21
	ARIMA-GARCH	39.05	63.27	92.53	115.28	127.30	141.91
7	GBM-SV	17.94	32.02	40.08	41.45	43.16	46.06
	ARIMA-GARCH	17.69	31.55	38.88	42.36	45.08	49.44
8	GBM-SV	27.50	43.65	55.95	66.08	75.35	83.55
	ARIMA-GARCH	22.81	41.06	55.44	69.13	85.04	103.55
9	GBM-SV	23.46	33.05	39.49	43.93	48.62	53.77
	ARIMA-GARCH	24.80	35.23	43.03	49.76	56.54	63.00
10	GBM-SV	16.98	27.42	34.26	37.72	40.29	42.70
	ARIMA-GARCH	16.93	30.32	40.65	47.19	52.55	57.38
11	GBM-SV	34.57	55.46	66.83	76.40	82.30	86.60
	ARIMA-GARCH	32.34	52.70	67.08	78.63	86.64	93.29

Table 19 Comparing the MPIL Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-13 Tuesday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	20.91	21.36	21.44	21.43	21.43	21.43
	ARIMA-GARCH	24.56	25.56	26.06	26.50	26.89	27.23
2	GBM-SV	21.39	21.66	21.84	21.94	22.00	22.08
	ARIMA-GARCH	20.96	21.30	21.57	21.74	21.87	21.98
3	GBM-SV	20.93	20.78	20.59	20.41	20.24	20.10
	ARIMA-GARCH	21.70	21.76	21.76	21.73	21.68	21.62
4	GBM-SV	34.69	32.16	30.19	28.66	27.48	26.63
	ARIMA-GARCH	54.29	61.15	68.30	76.02	84.47	93.55
5	GBM-SV	79.63	81.62	82.70	83.63	84.39	85.10
	ARIMA-GARCH	89.18	94.43	99.88	105.56	111.48	117.66
6	GBM-SV	54.62	54.98	54.73	54.74	54.68	54.55
	ARIMA-GARCH	69.94	78.18	86.90	96.28	106.41	117.37
7	GBM-SV	39.22	40.36	41.48	42.31	43.11	43.88
	ARIMA-GARCH	43.08	45.70	47.71	49.29	50.56	51.58
8	GBM-SV	50.67	48.65	46.90	45.45	44.28	43.40
	ARIMA-GARCH	50.08	49.43	48.73	48.00	47.26	46.53
9	GBM-SV	54.80	55.19	55.47	55.27	54.96	54.66
	ARIMA-GARCH	70.39	76.18	82.40	89.13	96.39	104.27
10	GBM-SV	54.29	56.90	58.54	59.70	60.52	61.18
	ARIMA-GARCH	54.34	57.83	61.23	64.75	68.41	72.23
11	GBM-SV	94.44	96.08	96.21	95.68	95.22	94.31
	ARIMA-GARCH	100.92	107.24	113.88	120.87	128.20	135.93

**Table 20 Comparing the PICP Values of the GBM-SV and the ARIMA-GARCH
Models (2012-11-13 Tuesday)**

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	96.53%	90.28%	86.81%	87.50%	86.11%	85.76%
	ARIMA-GARCH	97.92%	90.63%	86.46%	86.11%	84.72%	82.99%
2	GBM-SV	95.14%	86.81%	82.29%	79.86%	77.78%	78.47%
	ARIMA-GARCH	93.06%	85.07%	79.51%	78.13%	75.00%	72.92%
3	GBM-SV	94.10%	84.38%	79.86%	77.78%	79.51%	78.47%
	ARIMA-GARCH	92.01%	84.38%	80.90%	76.04%	75.35%	75.00%
4	GBM-SV	92.36%	85.07%	80.21%	80.56%	78.82%	79.86%
	ARIMA-GARCH	95.14%	90.28%	90.28%	89.24%	88.54%	88.89%
5	GBM-SV	95.49%	89.58%	85.42%	85.42%	83.33%	81.94%
	ARIMA-GARCH	96.18%	83.68%	74.31%	67.36%	60.07%	51.39%
6	GBM-SV	96.53%	90.28%	87.15%	83.33%	82.64%	81.94%
	ARIMA-GARCH	96.88%	89.24%	86.81%	82.99%	82.29%	84.03%
7	GBM-SV	96.88%	91.32%	90.63%	90.28%	88.19%	88.54%
	ARIMA-GARCH	98.26%	95.49%	93.40%	94.44%	92.01%	89.58%
8	GBM-SV	92.71%	84.38%	80.90%	81.60%	79.17%	78.13%
	ARIMA-GARCH	92.36%	80.90%	78.13%	77.43%	76.39%	74.65%
9	GBM-SV	94.44%	87.15%	84.03%	83.33%	83.68%	81.94%
	ARIMA-GARCH	95.49%	92.36%	88.54%	88.54%	88.19%	87.85%
10	GBM-SV	93.75%	90.63%	88.19%	89.93%	88.19%	87.15%
	ARIMA-GARCH	94.79%	86.81%	83.68%	82.99%	83.68%	85.07%
11	GBM-SV	93.40%	87.85%	82.99%	78.13%	76.74%	74.65%
	ARIMA-GARCH	95.83%	87.50%	82.29%	77.78%	78.13%	76.04%

**Table 21 Comparing the PI-Ratio of the GBM-SV and the ARIMA-GARCH
Models (2012-11-13 Tuesday)**

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	5.71	5.22	4.98	5.02	4.92	4.89
	ARIMA-GARCH	4.97	4.43	4.15	4.06	3.94	3.81
2	GBM-SV	6.34	5.70	5.35	5.16	5.02	5.04
	ARIMA-GARCH	6.37	5.73	5.29	5.15	4.92	4.75
3	GBM-SV	6.61	5.94	5.67	5.54	5.70	5.65
	ARIMA-GARCH	6.25	5.71	5.47	5.14	5.11	5.10
4	GBM-SV	4.99	4.90	4.88	5.10	5.15	5.33
	ARIMA-GARCH	3.31	2.79	2.51	2.24	2.01	1.84
5	GBM-SV	3.69	3.31	3.05	2.97	2.82	2.69
	ARIMA-GARCH	3.35	2.71	2.25	1.90	1.58	1.27
6	GBM-SV	3.68	3.36	3.20	3.00	2.93	2.87
	ARIMA-GARCH	2.92	2.38	2.06	1.75	1.55	1.42
7	GBM-SV	5.55	5.07	4.90	4.77	4.58	4.51
	ARIMA-GARCH	5.16	4.74	4.45	4.37	4.16	3.97
8	GBM-SV	5.24	4.87	4.77	4.90	4.82	4.79
	ARIMA-GARCH	5.42	4.83	4.75	4.81	4.85	4.86
9	GBM-SV	3.01	2.71	2.57	2.53	2.52	2.46
	ARIMA-GARCH	2.40	2.13	1.88	1.73	1.58	1.45
10	GBM-SV	2.98	2.73	2.57	2.56	2.46	2.39
	ARIMA-GARCH	3.05	2.62	2.39	2.24	2.13	2.05
11	GBM-SV	2.93	2.68	2.50	2.34	2.28	2.22
	ARIMA-GARCH	2.86	2.46	2.18	1.94	1.84	1.69

Table 22 Comparing the MAPE Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-26 Monday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	2.29%	3.14%	3.47%	3.56%	3.62%	3.74%
	ARIMA-GARCH	2.71%	4.76%	6.09%	6.73%	7.18%	7.59%
2	GBM-SV	2.15%	3.16%	3.75%	4.08%	4.29%	4.57%
	ARIMA-GARCH	2.40%	3.60%	4.16%	4.50%	4.67%	4.87%
3	GBM-SV	1.69%	2.31%	2.50%	2.59%	2.69%	2.76%
	ARIMA-GARCH	1.77%	2.66%	3.00%	3.19%	3.31%	3.42%
4	GBM-SV	1.62%	2.22%	2.34%	2.45%	2.60%	2.69%
	ARIMA-GARCH	1.75%	2.30%	2.24%	2.22%	2.21%	2.19%
5	GBM-SV	3.58%	5.70%	6.89%	8.00%	9.20%	10.18%
	ARIMA-GARCH	4.25%	7.29%	9.21%	10.32%	11.40%	12.82%
6	GBM-SV	3.45%	5.11%	5.82%	6.36%	6.68%	6.90%
	ARIMA-GARCH	4.13%	6.81%	8.65%	9.81%	10.65%	11.37%
7	GBM-SV	2.76%	4.35%	5.27%	6.01%	6.60%	7.25%
	ARIMA-GARCH	3.25%	5.70%	6.62%	7.23%	7.64%	8.01%
8	GBM-SV	1.78%	2.53%	2.79%	2.88%	2.91%	2.98%
	ARIMA-GARCH	1.72%	2.65%	3.01%	3.17%	3.15%	3.06%
9	GBM-SV	4.69%	7.42%	8.74%	9.87%	10.49%	10.76%
	ARIMA-GARCH	4.73%	7.64%	8.95%	9.91%	10.47%	10.94%
10	GBM-SV	3.57%	5.26%	6.32%	7.39%	8.31%	9.11%
	ARIMA-GARCH	3.69%	6.11%	7.99%	9.93%	11.72%	13.35%
11	GBM-SV	3.35%	4.95%	5.65%	6.15%	6.70%	6.96%
	ARIMA-GARCH	3.22%	5.39%	6.57%	7.22%	7.98%	8.47%

Table 23 Comparing the RMSE Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-26 Monday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	5.36	7.56	8.72	9.00	9.38	9.76
	ARIMA-GARCH	5.85	10.01	13.14	14.57	15.46	15.98
2	GBM-SV	5.22	8.45	10.88	12.60	13.93	15.17
	ARIMA-GARCH	6.18	10.09	12.41	13.91	15.12	15.88
3	GBM-SV	3.22	4.46	4.90	5.22	5.52	5.70
	ARIMA-GARCH	3.35	5.09	6.00	6.55	6.88	7.16
4	GBM-SV	7.95	10.47	10.36	12.12	14.96	17.25
	ARIMA-GARCH	7.61	9.45	8.14	7.54	7.44	7.33
5	GBM-SV	41.08	65.08	69.68	73.84	90.49	101.81
	ARIMA-GARCH	42.32	77.40	85.79	93.04	116.71	135.73
6	GBM-SV	21.42	42.74	54.35	62.43	64.48	65.56
	ARIMA-GARCH	25.07	49.05	67.69	80.17	86.82	90.03
7	GBM-SV	20.96	36.69	52.01	65.65	73.00	77.64
	ARIMA-GARCH	25.31	44.72	56.90	67.39	74.19	78.98
8	GBM-SV	5.65	7.96	8.67	8.89	8.95	9.06
	ARIMA-GARCH	5.40	8.05	9.21	9.62	9.56	9.23
9	GBM-SV	16.39	26.82	31.52	35.00	37.29	38.89
	ARIMA-GARCH	16.23	27.00	32.03	35.92	39.11	41.06
10	GBM-SV	21.00	32.48	40.63	49.17	56.78	62.48
	ARIMA-GARCH	21.10	34.22	44.45	57.12	69.30	78.69
11	GBM-SV	23.39	36.34	40.29	43.06	47.02	49.60
	ARIMA-GARCH	22.45	37.44	44.94	49.10	55.27	59.87

Table 24 Comparing the MPIL Values of the GBM-SV and the ARIMA-GARCH Models (2012-11-26 Monday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	18.54	19.39	20.23	20.98	21.70	22.35
	ARIMA-GARCH	24.16	24.65	25.26	25.77	26.25	26.76
2	GBM-SV	18.42	19.09	19.64	20.08	20.46	20.82
	ARIMA-GARCH	20.81	22.40	23.85	25.24	26.62	27.95
3	GBM-SV	15.32	15.96	16.49	16.93	17.23	17.52
	ARIMA-GARCH	14.77	16.03	16.94	17.64	18.19	18.62
4	GBM-SV	16.51	17.41	18.13	18.69	19.15	19.55
	ARIMA-GARCH	19.88	22.22	23.23	23.70	23.91	24.02
5	GBM-SV	55.54	55.03	54.50	54.03	53.73	53.50
	ARIMA-GARCH	65.35	67.49	69.20	70.62	71.85	72.97
6	GBM-SV	39.81	40.21	40.41	40.57	40.67	40.96
	ARIMA-GARCH	50.53	53.63	56.66	59.67	62.69	65.71
7	GBM-SV	37.38	37.74	38.08	38.29	38.51	38.71
	ARIMA-GARCH	46.01	47.41	48.44	49.19	49.72	50.08
8	GBM-SV	28.03	29.76	31.15	32.28	33.09	33.82
	ARIMA-GARCH	28.12	28.98	29.66	30.21	30.64	30.98
9	GBM-SV	40.40	39.64	38.95	38.26	37.66	37.06
	ARIMA-GARCH	42.75	43.27	43.76	44.16	44.47	44.72
10	GBM-SV	46.97	47.46	47.63	47.85	47.80	47.85
	ARIMA-GARCH	55.41	58.18	61.14	64.24	67.45	70.80
11	GBM-SV	54.24	54.88	55.05	55.35	55.36	55.41
	ARIMA-GARCH	55.39	58.25	60.82	63.14	65.29	67.27

**Table 25 Comparing the PICP Values of the GBM-SV and the ARIMA-GARCH
Models (2012-11-26 Monday)**

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	96.88%	92.71%	90.63%	92.01%	91.32%	90.97%
	ARIMA-GARCH	94.44%	87.50%	79.51%	74.65%	75.00%	74.31%
2	GBM-SV	95.14%	89.93%	86.46%	87.15%	86.46%	86.11%
	ARIMA-GARCH	94.79%	90.28%	88.89%	88.89%	90.97%	92.71%
3	GBM-SV	98.61%	92.36%	90.63%	91.32%	90.63%	89.58%
	ARIMA-GARCH	96.88%	90.28%	88.19%	86.46%	85.76%	86.46%
4	GBM-SV	98.26%	95.14%	94.10%	95.14%	96.18%	95.83%
	ARIMA-GARCH	98.96%	96.88%	97.57%	96.53%	97.22%	97.92%
5	GBM-SV	95.49%	87.85%	85.42%	82.99%	82.64%	81.94%
	ARIMA-GARCH	95.14%	85.07%	82.29%	84.03%	84.03%	85.76%
6	GBM-SV	95.49%	90.97%	90.63%	90.63%	88.89%	87.85%
	ARIMA-GARCH	95.14%	85.07%	81.94%	83.68%	82.29%	82.99%
7	GBM-SV	95.14%	92.01%	89.24%	88.89%	86.46%	86.46%
	ARIMA-GARCH	97.22%	86.11%	85.76%	84.72%	85.42%	86.11%
8	GBM-SV	97.22%	92.36%	92.71%	92.36%	92.01%	93.40%
	ARIMA-GARCH	97.92%	93.06%	88.89%	87.15%	88.89%	92.71%
9	GBM-SV	93.06%	85.42%	82.29%	79.17%	79.17%	77.78%
	ARIMA-GARCH	93.40%	83.33%	83.33%	81.94%	81.94%	80.90%
10	GBM-SV	95.49%	89.58%	86.46%	84.72%	85.07%	85.42%
	ARIMA-GARCH	95.49%	88.19%	84.03%	80.56%	80.90%	80.90%
11	GBM-SV	95.49%	90.28%	86.11%	84.03%	84.38%	83.68%
	ARIMA-GARCH	95.14%	90.28%	86.81%	82.99%	84.72%	85.07%

**Table 26 Comparing the PI-Ratio of the GBM-SV and the ARIMA-GARCH
Models (2012-11-26 Monday)**

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	6.33	5.79	5.42	5.30	5.08	4.91
	ARIMA-GARCH	4.75	4.31	3.82	3.52	3.47	3.37
2	GBM-SV	7.16	6.54	6.12	6.04	5.88	5.77
	ARIMA-GARCH	6.29	5.56	5.13	4.84	4.69	4.55
3	GBM-SV	8.88	8.00	7.60	7.47	7.28	7.09
	ARIMA-GARCH	9.03	7.75	7.16	6.74	6.49	6.39
4	GBM-SV	9.40	8.64	8.21	8.07	7.97	7.79
	ARIMA-GARCH	7.85	6.88	6.62	6.42	6.41	6.43
5	GBM-SV	3.67	3.42	3.36	3.29	3.29	3.27
	ARIMA-GARCH	3.08	2.65	2.48	2.45	2.39	2.39
6	GBM-SV	4.07	3.83	3.77	3.74	3.63	3.55
	ARIMA-GARCH	3.17	2.66	2.40	2.31	2.14	2.04
7	GBM-SV	5.54	5.32	5.13	5.09	4.92	4.89
	ARIMA-GARCH	4.58	3.94	3.84	3.74	3.73	3.73
8	GBM-SV	8.25	7.41	7.13	6.86	6.67	6.63
	ARIMA-GARCH	8.25	7.61	7.11	6.85	6.90	7.12
9	GBM-SV	3.38	3.17	3.11	3.04	3.08	3.07
	ARIMA-GARCH	3.16	2.76	2.72	2.63	2.60	2.54
10	GBM-SV	3.62	3.35	3.20	3.10	3.10	3.09
	ARIMA-GARCH	3.10	2.73	2.48	2.26	2.16	2.06
11	GBM-SV	4.31	4.04	3.84	3.72	3.72	3.68
	ARIMA-GARCH	4.18	3.77	3.47	3.20	3.15	3.07

Table 27 Comparing the MAPE Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-05 Wednesday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	2.19%	2.96%	3.25%	3.40%	3.40%	3.43%
	ARIMA-GARCH	2.36%	3.74%	4.59%	4.76%	4.99%	5.08%
2	GBM-SV	2.04%	2.89%	3.34%	3.68%	3.92%	4.03%
	ARIMA-GARCH	2.21%	3.43%	4.11%	4.37%	4.47%	4.70%
3	GBM-SV	1.67%	2.23%	2.41%	2.55%	2.57%	2.52%
	ARIMA-GARCH	1.70%	2.43%	2.74%	2.88%	3.01%	3.07%
4	GBM-SV	1.62%	2.17%	2.40%	2.46%	2.55%	2.58%
	ARIMA-GARCH	1.66%	2.20%	2.42%	2.52%	2.56%	2.62%
5	GBM-SV	3.09%	4.67%	6.04%	7.23%	8.31%	9.40%
	ARIMA-GARCH	3.28%	5.97%	8.05%	9.61%	11.20%	12.87%
6	GBM-SV	2.98%	4.64%	5.54%	6.08%	6.41%	6.69%
	ARIMA-GARCH	3.11%	5.34%	6.89%	8.16%	9.20%	10.00%
7	GBM-SV	2.47%	3.53%	3.97%	4.31%	4.71%	4.98%
	ARIMA-GARCH	2.49%	3.80%	4.38%	4.80%	5.05%	5.25%
8	GBM-SV	1.79%	2.33%	2.48%	2.50%	2.48%	2.57%
	ARIMA-GARCH	1.92%	2.73%	3.05%	3.03%	2.98%	2.96%
9	GBM-SV	3.04%	4.90%	5.90%	6.58%	7.11%	7.62%
	ARIMA-GARCH	3.17%	5.21%	6.55%	7.27%	7.64%	8.10%
10	GBM-SV	3.21%	5.03%	6.06%	6.80%	7.35%	8.00%
	ARIMA-GARCH	3.39%	5.86%	7.72%	9.24%	10.38%	11.60%
11	GBM-SV	3.09%	4.67%	5.52%	6.20%	6.99%	7.50%
	ARIMA-GARCH	3.30%	5.79%	7.13%	8.02%	8.70%	9.49%

Table 28 Comparing the RMSE Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-05 Wednesday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	4.69	7.83	9.68	10.63	10.78	10.72
	ARIMA-GARCH	5.31	9.12	11.57	12.64	12.92	13.18
2	GBM-SV	4.69	8.12	10.00	11.36	11.84	12.28
	ARIMA-GARCH	4.90	8.18	10.09	10.99	11.61	12.19
3	GBM-SV	3.38	4.98	5.76	6.18	6.25	6.14
	ARIMA-GARCH	3.42	5.14	6.02	6.46	6.64	6.77
4	GBM-SV	4.78	6.47	7.49	7.98	8.75	8.56
	ARIMA-GARCH	4.83	6.57	7.52	8.03	8.36	8.53
5	GBM-SV	14.86	23.51	30.18	36.76	42.89	48.83
	ARIMA-GARCH	15.47	28.47	40.08	51.79	64.36	75.95
6	GBM-SV	10.44	18.45	22.76	24.38	25.54	25.74
	ARIMA-GARCH	10.80	19.56	26.07	30.76	34.64	37.62
7	GBM-SV	8.85	15.09	17.77	19.31	20.73	22.42
	ARIMA-GARCH	9.21	15.86	20.42	23.57	25.98	28.14
8	GBM-SV	6.58	8.32	8.65	8.62	8.47	8.82
	ARIMA-GARCH	7.03	9.38	9.94	9.83	9.77	9.84
9	GBM-SV	10.52	18.63	22.77	24.97	26.98	28.97
	ARIMA-GARCH	10.32	18.60	23.56	25.94	27.55	28.90
10	GBM-SV	22.26	33.15	40.41	46.60	51.98	56.52
	ARIMA-GARCH	23.68	37.37	47.79	57.30	66.44	74.13
11	GBM-SV	22.74	32.30	37.81	42.95	47.27	48.80
	ARIMA-GARCH	25.12	37.99	46.65	52.74	59.70	62.11

Table 29 Comparing the MPIL Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-05 Wednesday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	15.15	15.46	15.71	15.93	16.07	16.19
	ARIMA-GARCH	16.15	16.57	16.90	17.18	17.42	17.62
2	GBM-SV	17.03	17.57	18.04	18.38	18.72	18.96
	ARIMA-GARCH	18.77	20.58	22.22	23.67	25.05	26.33
3	GBM-SV	13.33	13.57	13.75	13.93	14.05	14.14
	ARIMA-GARCH	13.03	13.16	13.27	13.36	13.44	13.51
4	GBM-SV	15.74	16.16	16.43	16.55	16.68	16.76
	ARIMA-GARCH	18.10	18.87	19.25	19.43	19.52	19.57
5	GBM-SV	40.82	41.33	41.76	42.06	42.22	42.34
	ARIMA-GARCH	43.79	46.46	48.76	50.79	52.64	54.33
6	GBM-SV	28.49	29.11	29.65	30.03	30.41	30.69
	ARIMA-GARCH	31.33	32.63	33.84	34.98	36.05	37.06
7	GBM-SV	28.46	28.53	28.48	28.46	28.35	28.24
	ARIMA-GARCH	28.77	28.90	28.95	28.95	28.92	28.89
8	GBM-SV	27.47	28.59	29.38	29.88	30.33	30.67
	ARIMA-GARCH	25.51	26.09	26.54	26.91	27.23	27.51
9	GBM-SV	29.51	30.59	31.28	31.89	32.32	32.76
	ARIMA-GARCH	30.96	34.95	39.09	43.44	48.10	53.09
10	GBM-SV	49.02	50.01	50.44	50.39	50.43	50.39
	ARIMA-GARCH	55.59	63.74	72.69	82.62	93.70	106.09
11	GBM-SV	59.63	62.64	65.25	67.36	68.79	70.22
	ARIMA-GARCH	63.80	72.41	81.26	90.69	100.77	111.69

**Table 30 Comparing the PICP Values of the GBM-SV and the ARIMA-GARCH
Models (2012-12-05 Wednesday)**

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	96.53%	90.97%	88.89%	89.58%	89.24%	88.89%
	ARIMA-GARCH	94.10%	85.76%	81.25%	79.86%	80.90%	80.21%
2	GBM-SV	96.88%	89.24%	90.28%	89.58%	88.89%	89.24%
	ARIMA-GARCH	96.53%	89.24%	90.28%	90.28%	89.93%	90.63%
3	GBM-SV	95.83%	90.28%	90.63%	89.58%	89.58%	89.93%
	ARIMA-GARCH	96.18%	88.19%	87.50%	83.68%	84.72%	84.03%
4	GBM-SV	95.49%	94.10%	93.40%	93.75%	92.71%	91.67%
	ARIMA-GARCH	97.92%	95.83%	96.18%	94.44%	95.49%	95.14%
5	GBM-SV	94.10%	88.89%	83.68%	81.25%	77.43%	76.74%
	ARIMA-GARCH	94.79%	85.42%	78.82%	76.04%	77.08%	76.04%
6	GBM-SV	95.83%	86.46%	83.68%	84.72%	83.68%	80.21%
	ARIMA-GARCH	94.79%	86.81%	82.64%	81.60%	79.51%	77.78%
7	GBM-SV	96.53%	88.89%	84.38%	83.68%	81.60%	78.13%
	ARIMA-GARCH	95.14%	85.07%	80.90%	79.17%	79.17%	79.86%
8	GBM-SV	94.79%	89.24%	89.24%	91.32%	92.01%	90.28%
	ARIMA-GARCH	92.01%	84.38%	84.38%	85.42%	85.07%	84.72%
9	GBM-SV	93.40%	89.58%	86.81%	85.42%	85.07%	83.33%
	ARIMA-GARCH	95.83%	90.63%	88.19%	88.54%	90.63%	92.01%
10	GBM-SV	95.14%	87.85%	86.11%	85.42%	85.42%	87.15%
	ARIMA-GARCH	95.14%	88.19%	89.24%	89.58%	88.54%	90.28%
11	GBM-SV	94.44%	91.32%	89.58%	87.50%	85.76%	86.11%
	ARIMA-GARCH	96.18%	89.93%	88.89%	90.97%	90.28%	89.24%

**Table 31 Comparing the PI-Ratio of the GBM-SV and the ARIMA-GARCH
Models (2012-12-05 Wednesday)**

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	7.69	7.11	6.83	6.79	6.70	6.63
	ARIMA-GARCH	7.04	6.26	5.82	5.63	5.63	5.52
2	GBM-SV	7.83	7.00	6.91	6.74	6.58	6.52
	ARIMA-GARCH	7.09	5.98	5.60	5.26	4.96	4.76
3	GBM-SV	9.96	9.20	9.11	8.89	8.82	8.81
	ARIMA-GARCH	10.26	9.32	9.17	8.71	8.77	8.65
4	GBM-SV	9.65	9.25	9.03	9.00	8.83	8.68
	ARIMA-GARCH	8.61	8.07	7.94	7.71	7.76	7.71
5	GBM-SV	5.01	4.68	4.38	4.24	4.03	3.99
	ARIMA-GARCH	4.69	3.97	3.49	3.22	3.14	3.00
6	GBM-SV	5.37	4.75	4.53	4.54	4.44	4.23
	ARIMA-GARCH	4.83	4.24	3.89	3.71	3.50	3.32
7	GBM-SV	7.27	6.69	6.37	6.33	6.21	5.98
	ARIMA-GARCH	7.07	6.27	5.94	5.79	5.78	5.82
8	GBM-SV	8.28	7.49	7.28	7.34	7.29	7.08
	ARIMA-GARCH	8.66	7.75	7.62	7.61	7.49	7.39
9	GBM-SV	4.36	4.05	3.85	3.72	3.66	3.55
	ARIMA-GARCH	4.27	3.58	3.11	2.81	2.60	2.39
10	GBM-SV	3.44	3.09	2.98	2.94	2.93	2.98
	ARIMA-GARCH	3.06	2.47	2.19	1.94	1.69	1.52
11	GBM-SV	3.95	3.63	3.41	3.22	3.08	3.03
	ARIMA-GARCH	3.76	3.10	2.73	2.50	2.23	1.99

Table 32 Comparing the MAPE Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-19 Wednesday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	2.10%	2.68%	2.84%	2.78%	2.90%	2.92%
	ARIMA-GARCH	2.41%	3.62%	4.23%	4.24%	4.42%	4.68%
2	GBM-SV	1.81%	2.34%	2.51%	2.56%	2.64%	2.65%
	ARIMA-GARCH	2.01%	2.85%	3.08%	3.18%	3.25%	3.32%
3	GBM-SV	1.61%	2.21%	2.37%	2.41%	2.34%	2.29%
	ARIMA-GARCH	1.57%	2.27%	2.47%	2.56%	2.58%	2.62%
4	GBM-SV	1.39%	1.92%	2.10%	2.12%	2.13%	2.12%
	ARIMA-GARCH	1.41%	1.91%	2.06%	2.13%	2.17%	2.20%
5	GBM-SV	3.46%	5.68%	7.37%	8.29%	9.06%	9.50%
	ARIMA-GARCH	3.68%	6.42%	8.65%	10.02%	11.25%	12.35%
6	GBM-SV	3.56%	5.56%	6.75%	7.85%	8.75%	9.43%
	ARIMA-GARCH	4.04%	7.23%	9.59%	11.95%	14.27%	16.43%
7	GBM-SV	2.53%	3.88%	4.56%	5.03%	5.32%	5.64%
	ARIMA-GARCH	2.57%	4.39%	5.42%	6.31%	7.21%	7.99%
8	GBM-SV	1.72%	2.25%	2.42%	2.46%	2.46%	2.49%
	ARIMA-GARCH	1.70%	2.40%	2.59%	2.71%	2.76%	2.77%
9	GBM-SV	2.52%	3.73%	4.20%	4.59%	5.03%	5.32%
	ARIMA-GARCH	2.66%	4.27%	5.24%	5.99%	6.83%	7.50%
10	GBM-SV	2.85%	3.87%	4.26%	4.55%	5.00%	5.44%
	ARIMA-GARCH	2.71%	4.17%	4.84%	5.17%	5.56%	6.08%
11	GBM-SV	2.98%	4.42%	4.94%	5.00%	5.42%	5.59%
	ARIMA-GARCH	2.82%	4.41%	4.94%	5.20%	5.67%	6.05%

Table 33 Comparing the RMSE Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-19 Wednesday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	3.88	5.00	5.46	5.16	5.34	5.32
	ARIMA-GARCH	4.36	6.16	7.09	7.20	7.79	8.22
2	GBM-SV	3.40	4.48	4.98	5.23	5.31	5.29
	ARIMA-GARCH	3.72	5.09	5.58	5.73	5.78	5.85
3	GBM-SV	2.98	4.04	4.39	4.56	4.42	4.28
	ARIMA-GARCH	2.90	4.09	4.54	4.70	4.75	4.74
4	GBM-SV	3.14	4.40	4.81	4.91	4.88	4.87
	ARIMA-GARCH	3.07	4.31	4.68	4.80	4.85	4.87
5	GBM-SV	19.17	26.25	31.77	32.85	34.29	36.46
	ARIMA-GARCH	17.87	26.85	33.78	36.15	36.61	38.45
6	GBM-SV	13.16	20.15	22.38	26.19	28.93	30.44
	ARIMA-GARCH	13.60	21.01	24.98	29.29	32.53	35.13
7	GBM-SV	11.03	17.53	21.08	23.24	24.20	25.72
	ARIMA-GARCH	10.92	18.44	23.14	27.26	30.50	33.43
8	GBM-SV	5.67	7.28	7.77	7.89	7.90	7.99
	ARIMA-GARCH	5.75	7.63	8.11	8.39	8.54	8.61
9	GBM-SV	8.59	12.53	13.61	15.19	17.17	17.96
	ARIMA-GARCH	8.68	13.44	15.89	18.59	21.35	23.48
10	GBM-SV	7.90	10.01	10.27	11.49	13.08	14.53
	ARIMA-GARCH	7.62	11.13	12.92	15.14	16.86	18.64
11	GBM-SV	17.49	26.69	29.42	27.79	29.79	30.12
	ARIMA-GARCH	16.35	25.44	28.40	28.12	29.71	31.43

Table 34 Comparing the MPIL Values of the GBM-SV and the ARIMA-GARCH Models (2012-12-19 Wednesday)

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	16.25	17.04	17.66	18.17	18.61	18.99
	ARIMA-GARCH	17.59	18.98	20.19	21.20	22.13	22.96
2	GBM-SV	16.66	17.58	18.43	19.08	19.61	20.14
	ARIMA-GARCH	17.79	19.98	22.15	24.28	26.49	28.77
3	GBM-SV	14.08	14.64	15.09	15.45	15.77	16.04
	ARIMA-GARCH	13.58	14.22	14.79	15.30	15.78	16.22
4	GBM-SV	13.22	13.48	13.69	13.89	14.01	14.13
	ARIMA-GARCH	12.95	13.42	13.73	13.95	14.11	14.22
5	GBM-SV	41.06	42.78	44.34	45.56	46.62	47.83
	ARIMA-GARCH	51.06	54.19	57.38	60.73	64.15	67.70
6	GBM-SV	38.31	39.92	41.29	42.41	43.27	44.12
	ARIMA-GARCH	42.00	44.25	46.43	48.58	50.69	52.78
7	GBM-SV	37.76	39.62	41.26	42.61	43.77	44.96
	ARIMA-GARCH	40.05	49.10	59.10	70.25	82.83	97.38
8	GBM-SV	30.03	32.63	34.77	36.37	37.53	38.59
	ARIMA-GARCH	24.81	25.78	26.64	27.43	28.15	28.82
9	GBM-SV	27.91	30.87	33.35	35.72	37.59	39.53
	ARIMA-GARCH	28.56	34.73	41.90	50.31	60.22	71.95
10	GBM-SV	32.63	36.37	39.48	42.26	44.70	46.89
	ARIMA-GARCH	25.64	28.08	30.53	32.98	35.53	38.13
11	GBM-SV	54.56	58.14	61.36	63.81	65.85	67.78
	ARIMA-GARCH	46.86	49.78	52.62	55.40	58.13	60.83

**Table 35 Comparing the PICP Values of the GBM-SV and the ARIMA-GARCH
Models (2012-12-19 Wednesday)**

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	95.83%	89.58%	89.93%	91.32%	91.32%	92.01%
	ARIMA-GARCH	94.10%	88.89%	87.85%	88.54%	89.93%	90.63%
2	GBM-SV	97.92%	93.40%	94.10%	93.75%	93.40%	94.44%
	ARIMA-GARCH	97.22%	92.71%	93.75%	96.18%	96.88%	98.96%
3	GBM-SV	97.92%	92.36%	90.97%	89.93%	91.32%	92.71%
	ARIMA-GARCH	97.22%	90.97%	89.58%	88.89%	89.93%	90.97%
4	GBM-SV	95.49%	91.67%	88.89%	88.89%	89.24%	90.97%
	ARIMA-GARCH	95.14%	92.36%	90.28%	90.63%	90.97%	92.01%
5	GBM-SV	95.14%	89.93%	83.68%	84.38%	81.94%	83.33%
	ARIMA-GARCH	95.49%	88.89%	80.56%	78.82%	78.82%	74.31%
6	GBM-SV	94.44%	89.24%	84.72%	84.03%	81.60%	79.86%
	ARIMA-GARCH	96.53%	87.50%	78.47%	70.49%	60.42%	53.82%
7	GBM-SV	95.14%	88.89%	86.11%	84.72%	83.68%	84.38%
	ARIMA-GARCH	95.49%	92.01%	90.63%	91.32%	91.32%	93.40%
8	GBM-SV	97.22%	94.10%	94.10%	95.83%	96.53%	96.53%
	ARIMA-GARCH	94.79%	89.58%	88.89%	89.93%	91.32%	92.01%
9	GBM-SV	97.22%	94.44%	95.83%	95.49%	95.49%	94.44%
	ARIMA-GARCH	96.88%	93.75%	94.10%	94.44%	95.83%	96.88%
10	GBM-SV	96.18%	92.71%	94.10%	93.40%	92.71%	90.97%
	ARIMA-GARCH	94.44%	89.93%	87.50%	88.89%	91.67%	93.40%
11	GBM-SV	96.53%	94.44%	94.10%	92.71%	93.06%	93.75%
	ARIMA-GARCH	96.53%	91.67%	89.58%	89.58%	88.54%	90.97%

**Table 36 Comparing the PI-Ratio of the GBM-SV and the ARIMA-GARCH
Models (2012-12-19 Wednesday)**

Data Set	Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
1	GBM-SV	7.04	6.28	6.09	6.01	5.88	5.81
	ARIMA-GARCH	6.40	5.61	5.21	5.01	4.88	4.75
2	GBM-SV	7.98	7.22	6.95	6.70	6.50	6.41
	ARIMA-GARCH	7.44	6.33	5.79	5.43	5.02	4.73
3	GBM-SV	9.59	8.69	8.30	8.02	7.98	7.98
	ARIMA-GARCH	9.90	8.85	8.38	8.04	7.89	7.77
4	GBM-SV	11.43	10.76	10.27	10.13	10.09	10.20
	ARIMA-GARCH	11.65	10.91	10.42	10.29	10.22	10.25
5	GBM-SV	3.92	3.61	3.29	3.25	3.11	3.10
	ARIMA-GARCH	3.14	2.77	2.39	2.23	2.14	1.93
6	GBM-SV	3.77	3.45	3.20	3.11	2.97	2.87
	ARIMA-GARCH	3.51	3.04	2.63	2.28	1.89	1.63
7	GBM-SV	5.49	4.89	4.56	4.35	4.19	4.12
	ARIMA-GARCH	5.22	4.12	3.38	2.88	2.45	2.14
8	GBM-SV	7.75	6.91	6.49	6.32	6.17	6.01
	ARIMA-GARCH	9.15	8.33	8.01	7.87	7.79	7.67
9	GBM-SV	4.73	4.18	3.93	3.67	3.50	3.30
	ARIMA-GARCH	4.62	3.69	3.09	2.59	2.20	1.87
10	GBM-SV	4.07	3.54	3.33	3.10	2.92	2.74
	ARIMA-GARCH	5.07	4.41	3.94	3.71	3.55	3.37
11	GBM-SV	4.08	3.76	3.57	3.39	3.31	3.25
	ARIMA-GARCH	4.74	4.24	3.92	3.72	3.51	3.45

5.4 Chapter Summary

This chapter presented a new travel time prediction framework that takes advantage of both the GBM and SV models. Travel time data from 11 freeway segments and the prediction horizons from 1-step-ahead up to 6-step-ahead prediction were used for examining the performance of the models. In order to test the performance of the proposed GBM-SV model, the ARIMA-GARCH model is used as the benchmark model. Through comparing these two modeling approaches under different scenarios, the GBM-SV model shows its considerable advantage over the ARIMA-GARCH model in terms of prediction accuracy and reliability. This conclusion also supports the conclusions of Chapters 3 and 4. In Chapter 3, we demonstrated that the SV model provides more efficient and effective PIs compared with the GARCH model, while the experiment in Chapter 4 indicates that the GBM model provides more accurate predictions compared with the ARIMA model. Therefore, the combination of these two models potentially further improves model performance. The proposed travel time prediction framework in this chapter is an example of how to improve the overall performance of the travel time prediction models. In future research, we could explore more accurate models to estimate the mean part and more efficient and effective models in determining the residual part (construct high quality PIs).

Chapter 6: Conclusion and Recommendations

6.1 Summary

Travel time prediction is a critical topic in the development of ITS systems. Especially with the rapid development of the Advanced Traveler Information Systems and Advanced Traffic Management Systems, more accurate and reliable travel time information is needed to enable the success of these systems' development. Apart from its importance, travel time estimation and prediction are complex and challenging tasks. Resulting from the interactions among different vehicle-driver combinations, and exogenous factors such as weather, demand, and roadway conditions, travel time often experiences strong fluctuations across different periods and traffic conditions. These rapid fluctuations are often complex and difficult to predict. Fully understanding these fluctuations and developing accurate travel time prediction algorithms is critical.

Inspired by the need of travel time predictions, a wide range of methodologies have been proposed in the literature. As discussed in Chapter 2, existing travel time prediction algorithms can be divided into four major categories: parametric, non-parametric, hybrid and prediction interval based approaches. The parametric methods usually have a well-established theoretical foundation but with lots of strict model assumptions. Comparatively, the non-parametric methods require less model assumptions but some of them may be difficult to interpret. The hybrid methods take advantage of different prediction models but some models may be too complex when

making predictions. The travel time interval based algorithms belongs to the category of hybrid methods and it provides not only the mean but also a prediction bound to capture both prediction accuracy and reliability of the model. As it is a relatively new area in travel time prediction, there are limited studies in the literature.

In this research, both prediction accuracy and reliability issues have been addressed in freeway travel time prediction. Although most existing travel time prediction models are able to provide accurate predictions during non-peak hours, peak hour travel time prediction is still a challenging topic. Investigating travel time patterns during both non-peak and peak hours and developing a more accurate travel time prediction algorithm is critical. On the other hand, because of the difficulties in predicting travel time, especially during peak hours, another issue that needs to be considered is the reliability issues of the model. The model should consider situations when traffic is highly volatile when a point prediction becomes 'less accurate'. In this case, the prediction interval based approach provides a prediction bound to indicate how likely it will capture the observed travel time value and therefore is able to indicate how reliable the prediction is.

To capture the uncertainty and variations of travel time data, this study proposed two different statistical volatility models: component GARCH and stochastic volatility models. In general, the statistical volatility model predicts future traffic volatility based on its previous volatility values. In a transportation system, travelers respond differently to unexpected changes in travel time. The presence of this volatility in traffic may lead to changes in driving behavior in order to compensate for the resulting changes in expected arrival time. These changes lead to

increased traffic volatility with a decreasing rate over time in order to restore the past stability of the system. The volatility models capture this changed traffic patterns over time and make further prediction.

The component GARCH models consider situations when seasonal (cyclical) patterns or trends exist in data. For some road segments, when commuters account for a large percentage of the total traffic volume, travel time of these types of segments may show strong cyclical patterns. In this case, the seasonal component should be considered when modeling the data. Through decomposition, the component GARCH models potentially improve the prediction accuracy. Another type of the volatility model, the stochastic volatility model considers the conditional variance of travel time data as an unobserved stochastic process therefore allows for a more flexible application and can account for uncertainties inherent in traffic phenomena.

In term of prediction accuracy, Chapter 4 proposed the application of tree based ensemble methods in travel time prediction. The gradient boosted regression tree method was developed to model and make more accurate prediction of travel time. The basic idea of the gradient boosting method is to sequentially generate base learners from a weighted version of the training data to strategically find the optimal combination of trees. In contrast to other machine learning methods that have been treated as black-boxes, tree based ensemble methods provide interpretable results, while requiring little data preprocessing, are able to handle different types of predictor variables, and can fit complex nonlinear relationship. These properties make the tree based ensemble methods good candidates for solving travel time prediction problems.

To both consider the prediction accuracy and reliably, Chapter 5 proposed a new travel time prediction framework that combines the gradient boosting tree and statistical volatility model. The new proposed method is able to take advantage of these two models and provide better performance.

6.2 Conclusion

The following list provides the conclusions/findings of this research:

- Due to the complex nature of travel time prediction problem, the traditional point based prediction approach is unable to perfectly account for uncertainties in traffic. There is often a mismatch between the predicted mean and the observed value. A prediction interval based approach as an alternative way to represent uncertainties associated with travel time prediction has the potential to provide more reliable prediction information.
- Volatility-based travel time prediction models relax the constant variation assumption. This kind of method treats the current volatility as a function of its past values and can be used to construct more accurate PIs to capture travel time uncertainty.
- The component GARCH models are able to capture the seasonal patterns in travel time volatility. When seasonal (cyclical) patterns exist, the component GARCH model could be a better choice compared with the traditional GARCH models.
- The stochastic volatility models consider part of the change in travel time volatility are due to random shocks, while the GARCH type model treat the volatility as time changing but not stochastic process. Through using

advanced Monte Carlo Markov Chain estimation method to fit the stochastic volatility model, the model is able to provide more accurate PIs.

- The GBM model has considerable advantages in freeway travel time prediction. The capability of the GBM model in handling different types of input variables in modeling complex nonlinear relationship makes it a promising algorithm for travel time prediction.
- The new proposed travel time prediction framework GBM-SV model improves both the model accuracy and reliability. The GBM-SV model also provides a framework for future development of travel time prediction models.
- As traffic at different freeway segments may show different patterns or characteristics, it is necessary to study the traffic patterns in order to select the appropriate model to predict travel time.

6.3 Future Recommendation

Although this research provided contributions to the existing literature in the area of freeway travel time prediction, there are some other research avenues that can be pursued. Future directions of the research are provided below:

- In this research, we only utilize travel time information. But with the advanced technology development, more and more data are available to use, such as incident, weather, work zone and so on. Since these events may have significant influence on travel time, utilizing this information could help improve prediction accuracy. For example, traffic congestion is more likely to occur due to inclement weather conditions as the freeway capacity drops while demand does not drop. In this case, if we can include weather

conditions as explanatory variables, then the model would potentially capture the weather impact on travel time. As the GBM model is capable of handling different types of input information, it gives the advantage of utilizing different input information to further improve prediction accuracy. When using weather information to predict travel time, the weather forecast information will be used, therefore, how to utilize the weather forecast information and considering its reliability could be another research topic. To sum up, future research should use the information of external impact factors when predicting travel time.

- Uncertainties associated with travel time prediction is relatively a new area in travel time prediction, few literature focuses on travel time uncertainty prediction. As indicated in the literature review section, there are generally two types of PIs based approach: ensemble methods and statistical volatility based approach. This research mainly focuses on using statistical volatility methods to model the uncertainty associated with travel time prediction. Two different types of volatility models have been proposed. The study results show that the PIs based approach shows its promising abilities in indicating the uncertainty associated with prediction. While in the future, we can also use the ensemble based algorithms to construct PIs when predicting travel time. It is also beneficial to have a comparison of the PIs constructed by volatility based and ensemble based method and discuss the advantages and disadvantages of each model in addressing uncertainty associated with travel time prediction.

- In this study, we only predict travel time for one segment. While, how to utilize the segment travel time information to derive dynamic path travel times can also be studied in the future. As for the purpose of providing travel time information through Advanced Traveler Information Systems, dynamic path information would help the traveler to find the minimum travel time path. By utilizing the method proposed in this study, we can obtain individual segment travel time information and therefore further derive the dynamic path travel time information based on individual segment travel time information.

References

- [1] J. Yeon, L. Elefteriadou, and S. Lawphongpanich, "Travel time estimation on a freeway using Discrete Time Markov Chains," *Transportation Research Part B: Methodological*, vol. 42, pp. 325-338, 2008.
- [2] T. Choe, A. Skabardonis, and P. Varaiya, "Freeway performance measurement system: operational analysis tool," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1811, pp. 67-75, 2002.
- [3] M. Yang, Y. Liu, and Z. You, "The reliability of travel time forecasting," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, pp. 162-171, 2010.
- [4] M. Yildirimoglu and N. Geroliminis, "Experienced travel time prediction for congested freeways," *Transportation Research Part B: Methodological*, vol. 53, pp. 45-63, 2013.
- [5] R. F. Engle and M. E. Sokalska, "Forecasting intraday volatility in the us equity market. multiplicative component garch," *Journal of Financial Econometrics*, vol. 10, pp. 54-83, 2012.
- [6] G. Kastner and S. Frühwirth-Schnatter, "Ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC estimation of stochastic volatility models," *Computational Statistics & Data Analysis*, 2013.
- [7] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5-32, 2001.

- [8] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short - term traffic forecasting: Overview of objectives and methods," *Transport reviews*, vol. 24, pp. 533-557, 2004.
- [9] B. Van Arem, H. R. Kirby, M. J. Van Der Vlist, and J. C. Whittaker, "Recent advances and applications in the field of short-term traffic forecasting," *International Journal of Forecasting*, vol. 13, pp. 1-12, 1997.
- [10] S. Ishak and H. Al-Deek, "Performance evaluation of short-term time-series traffic prediction model," *Journal of Transportation Engineering*, vol. 128, pp. 490-498, 2002.
- [11] J. W. C. Van Lint and C. P. I. J. Van Hinsbergen, "Short-Term Traffic and Travel Time Prediction Models," *Artificial Intelligence Applications to Critical Transportation Issues*, p. 22, 2012.
- [12] K. Farokhi Sadabadi, M. Hamed, and A. Haghani, "Evaluating moving average techniques in short-term travel time prediction using an AVI data set," in *Transportation Research Board 89th Annual Meeting*, 2010.
- [13] B. Smith and M. Demetsky, "Traffic Flow Forecasting: Comparison of Modeling Approaches," *Journal of Transportation Engineering*, vol. 123, pp. 261-266, 1997.
- [14] B. Williams, P. Durvasula, and D. Brown, "Urban Freeway Traffic Flow Prediction: Application of Seasonal Autoregressive Integrated Moving Average and Exponential Smoothing Models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1644, pp. 132-141, 1998.

- [15] V. Stephanedes, P. G. Michalopoulos, and R. A. Plum, "Improved estimation of traffic flow for Real-Time control (Discussion and closure)," *Transportation Research Record*, 1981.
- [16] D. Jeffery, K. Russam, and D. Robertson, "Electronic route guidance by AUTOGUIDE: the research background," *Traffic engineering & control*, vol. 28, pp. 525-529, 1987.
- [17] I. Kaysi, M. Ben-Akiva, and H. Koutsopoulos, *Integrated approach to vehicle routing and congestion prediction for real-time driver guidance*, 1993.
- [18] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using Box-Jenkins techniques," *Transportation Research Record*, 1979.
- [19] M. Levin and Y.-D. Tsao, "On Forecasting Freeway Occupancies and Volumes (Abridgment)," *Transportation Research Record*, 1980.
- [20] G. A. Davis, N. L. Nihan, M. M. Hamed, and L. N. Jacobson, "Adaptive forecasting of freeway traffic congestion," *Transportation Research Record*, 1990.
- [21] M. Hamed, H. Al-Masaeid, and Z. Said, "Short-Term Prediction of Traffic Volume in Urban Arterials," *Journal of Transportation Engineering*, vol. 121, pp. 249-254, 1995.
- [22] Y. Kamarianakis and P. Prastacos, "Space-time modeling of traffic flow," *Computers & Geosciences*, vol. 31, pp. 119-133, 2005.
- [23] B. M. Williams, P. K. Durvasula, and D. E. Brown, "Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transportation Research*

- Record: Journal of the Transportation Research Board*, vol. 1644, pp. 132-141, 1998.
- [24] M. Cetin and G. Comert, "Short-term traffic flow prediction with regime switching models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1965, pp. 23-31, 2006.
- [25] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, vol. 19, pp. 606-616, 2011.
- [26] M. G. Karlaftis and E. I. Vlahogianni, "Memory properties and fractional integration in transportation time-series," *Transportation Research Part C: Emerging Technologies*, vol. 17, pp. 444-453, 2009.
- [27] A. Stathopoulos and M. G. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transportation Research Part C: Emerging Technologies*, vol. 11, pp. 121-135, 2003.
- [28] B. Ghosh, B. Basu, and M. O'Mahony, "Multivariate Short-Term Traffic Flow Forecasting Using Time-Series Analysis," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, pp. 246-254, 2009.
- [29] J. Whittaker, S. Garside, and K. Lindveld, "Tracking and predicting a network traffic process," *International Journal of Forecasting*, vol. 13, pp. 51-61, 1997.
- [30] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transportation Research Part B: Methodological*, vol. 18, pp. 1-11, 1984.

- [31] S. Chien and C. Kuchipudi, "Dynamic Travel Time Prediction with Real-Time and Historic Data," *Journal of Transportation Engineering*, vol. 129, pp. 608-616, 2003.
- [32] C. Nanthawichit, T. Nakatsuji, and H. Suzuki, "Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1855, pp. 49-59, 2003.
- [33] J. W. C. van Lint, "Online learning solutions for freeway travel time prediction," *Ieee Transactions on Intelligent Transportation Systems*, vol. 9, pp. 38-47, Mar 2008.
- [34] Y. Wang, M. Papageorgiou, and A. Messmer, "Real-time freeway traffic state estimation based on extended Kalman filter: Adaptive capabilities and real data testing," *Transportation Research Part A: Policy and Practice*, vol. 42, pp. 1340-1358, 2008.
- [35] F. Yang, Z. Yin, H. X. Liu, and B. Ran, "Online recursive algorithm for short-term traffic prediction," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1879, pp. 1-8, 2004.
- [36] B. L. Smith, B. M. Williams, and R. Keith Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 10, pp. 303-321, 2002.

- [37] B. Smith and M. Demetsky, "Multiple-Interval Freeway Traffic Flow Forecasting," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1554, pp. 136-141, 1996.
- [38] S. Clark, "Traffic Prediction Using Multivariate Nonparametric Regression," *Journal of Transportation Engineering*, vol. 129, pp. 161-168, 2003.
- [39] G. Davis and N. Nihan, "Nonparametric Regression and Short - Term Freeway Traffic Forecasting," *Journal of Transportation Engineering*, vol. 117, pp. 178-188, 1991.
- [40] S. Robinson and J. Polak, "Modeling Urban Link Travel Time with Inductive Loop Detector Data by Using the k-NN Method," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1935, pp. 47-56, 2005.
- [41] J. Myung, D. K. Kim, S. Y. Kho, and C. H. Park, "Travel Time Prediction Using k Nearest Neighbor Method with Combined Data from Vehicle Detector System and Automatic Toll Collection System," *Transportation Research Record*, pp. 51-59, 2011.
- [42] N. Zou, J. Wang, G.-L. Chang, and J. Paracha, "Application of Advanced Traffic Information Systems," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2129, pp. 62-72, 2009.
- [43] J. W. van Lint, S. Hoogendoorn, and H. J. van Zuylen, "Freeway travel time prediction with state-space neural networks: Modeling state-space dynamics with recurrent neural networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1811, pp. 30-39, 2002.

- [44] H. Yin, S. Wong, J. Xu, and C. Wong, "Urban traffic flow prediction using a fuzzy-neural approach," *Transportation Research Part C: Emerging Technologies*, vol. 10, pp. 85-98, 2002.
- [45] S. Ishak, P. Kotha, and C. Alecsandru, "Optimization of dynamic neural network performance for short-term traffic prediction," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1836, pp. 45-56, 2003.
- [46] S. Ishak and C. Alecsandru, "Optimizing traffic prediction performance of neural networks under various topological, input, and traffic condition settings," *Journal of Transportation Engineering*, vol. 130, pp. 452-465, 2004.
- [47] X. Jiang and H. Adeli, "Dynamic wavelet neural network model for traffic flow forecasting," *Journal of Transportation Engineering*, vol. 131, pp. 771-779, 2005.
- [48] J. Van Lint, S. Hoogendoorn, and H. J. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transportation Research Part C: Emerging Technologies*, vol. 13, pp. 347-369, 2005.
- [49] C. Quek, M. Pasquier, and B. B. S. Lim, "POP-TRAFFIC: A novel fuzzy neural approach to road traffic analysis and prediction," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, pp. 133-146, 2006.
- [50] W. Zheng, D.-H. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," *Journal of Transportation Engineering*, vol. 132, pp. 114-121, 2006.

- [51] X. Zeng and Y. Zhang, "Development of Recurrent Neural Network Considering Temporal - Spatial Input Dynamics for Freeway Travel Time Modeling," *Computer - Aided Civil and Infrastructure Engineering*, 2013.
- [52] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, pp. 124-132, 2006.
- [53] W.-C. Hong, "Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm," *Neurocomputing*, vol. 74, pp. 2096-2107, 2011.
- [54] M. Danech-Pajouh and M. Aron, "ATHENA: a method for short-term inter-urban motorway traffic forecasting," *Recherche Transports Sécurité*, 1991.
- [55] C. Antoniou, H. N. Koutsopoulos, and G. Yannis, "Dynamic data-driven local traffic state estimation and prediction," *Transportation Research Part C: Emerging Technologies*, vol. 34, pp. 89-107, 2013.
- [56] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, pp. 307-318, 1996.
- [57] H. Chen, S. Grant-Muller, L. Mussone, and F. Montgomery, "A Study of Hybrid Neural Network Approaches and the Effects of Missing Data on Traffic Forecasting," *Neural computing & applications*, vol. 10, pp. 277-286, 2001/12/01 2001.

- [58] B. Yu, Z. Z. Yang, and K. Chen, "Hybrid model for prediction of bus arrival times at next station," *Journal of Advanced Transportation*, vol. 44, pp. 193-204, Jul 2010.
- [59] A. Stathopoulos, L. Dimitriou, and T. Tsekeris, "Fuzzy modeling approach for combined forecasting of urban traffic flow," *Computer - Aided Civil and Infrastructure Engineering*, vol. 23, pp. 521-535, 2008.
- [60] H. Liu, H. van Zuylen, H. van Lint, and M. Salomons, "Predicting urban arterial travel time with state-space neural networks and Kalman filters," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1968, pp. 99-108, 2006.
- [61] D. Boto-Giralda, F. J. Díaz-Pernas, D. González-Ortega, J. F. Díez-Higuera, M. Antón-Rodríguez, M. Martínez-Zarzuela, and I. Torre-Díez, "Wavelet-Based Denoising for Traffic Volume Time Series Forecasting with Self-Organizing Neural Networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 25, pp. 530-545, 2010.
- [62] Y. Peng, M. Lei, J.-B. Li, and X.-Y. Peng, "A novel hybridization of echo state networks and multiplicative seasonal ARIMA model for mobile communication traffic series forecasting," *Neural Computing and Applications*, pp. 1-8, 2012/12/01 2012.
- [63] K. Hamad, M. T. Shourijeh, E. Lee, and A. Faghri, "Near-Term Travel Speed Prediction Utilizing Hilbert-Huang Transform," *Computer-Aided Civil and Infrastructure Engineering*, vol. 24, pp. 551-576, 2009.

- [64] H. K. Chen and C. J. Wu, "Travel Time Prediction Using Empirical Mode Decomposition and Gray Theory Example of National Central University Bus in Taiwan," *Transportation Research Record*, pp. 11-19, 2012.
- [65] J.-L. Deng, "Introduction to grey system theory," *The Journal of grey system*, vol. 1, pp. 1-24, 1989.
- [66] Y. Wei and M.-C. Chen, "Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 21, pp. 148-162, 2012.
- [67] X. Jiang and H. Adeli, "Wavelet Packet-Autocorrelation Function Method for Traffic Flow Pattern Analysis," *Computer-Aided Civil and Infrastructure Engineering*, vol. 19, pp. 324-337, 2004.
- [68] Y. Xie, Y. Zhang, and Z. Ye, "Short - Term Traffic Volume Forecasting Using Kalman Filter with Discrete Wavelet Decomposition," *Computer - Aided Civil and Infrastructure Engineering*, vol. 22, pp. 326-334, 2007.
- [69] J. Wang and Q. Shi, "Short-term traffic speed forecasting hybrid model based on Chaos-Wavelet Analysis-Support Vector Machine theory," *Transportation Research Part C: Emerging Technologies*, 2012.
- [70] G. Leshem and Y. a. Ritov, "Traffic Flow Prediction using Adaboost Algorithm with Random Forests as a Weak Learner," *International Journal of Intelligent Technology*, vol. 2, 2007.
- [71] B. Hamner, "Predicting travel times with context-dependent random forests by modeling local and aggregate traffic flow," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, 2010, pp. 1357-1359.

- [72] Y. Wang, "Prediction of weather impacted airport capacity using ensemble learning," in *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*, 2011, pp. 2D6-1-2D6-11.
- [73] M. M. Ahmed and M. Abdel-Aty, "Application of Stochastic Gradient Boosting Technique to Enhance Reliability of Real-Time Risk Assessment," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2386, pp. 26-34, 2013.
- [74] Y.-S. Chung, "Factor complexity of crash occurrence: An empirical demonstration using boosted regression trees," *Accident Analysis & Prevention*, vol. 61, pp. 107-118, 2013.
- [75] C. P. I. van Hinsbergen, J. W. van Lint, and H. Van Zuylen, "Bayesian training and committees of state-space neural networks for online travel time prediction," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2105, pp. 118-126, 2009.
- [76] C. Van Hinsbergen, J. Van Lint, and H. Van Zuylen, "Bayesian committee of neural networks to predict travel times with confidence intervals," *Transportation Research Part C: Emerging Technologies*, vol. 17, pp. 498-509, 2009.
- [77] Y. Zhang and Y. C. Liu, "Analysis of peak and non-peak traffic forecasts using combined models," *Journal of Advanced Transportation*, vol. 45, pp. 21-37, Jan 2011.
- [78] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Spatio - Temporal Short - Term Urban Traffic Volume Forecasting Using Genetically Optimized

- Modular Networks," *Computer - Aided Civil and Infrastructure Engineering*, vol. 22, pp. 317-325, 2007.
- [79] A. Khosravi, E. Mazloumi, S. Nahavandi, D. Creighton, and J. Van Lint, "Prediction intervals to account for uncertainties in travel time prediction," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, pp. 537-547, 2011.
- [80] A. Khosravi, E. Mazloumi, S. Nahavandi, D. Creighton, and J. Van Lint, "A genetic algorithm-based method for improving quality of travel time prediction intervals," *Transportation Research Part C: Emerging Technologies*, vol. 19, pp. 1364-1376, 2011.
- [81] J. Van Lint, "Reliable real-time framework for short-term freeway travel time prediction," *Journal of Transportation Engineering*, vol. 132, pp. 921-932, 2006.
- [82] X. Fei, C.-C. Lu, and K. Liu, "A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 19, pp. 1306-1318, 2011.
- [83] R. Li and G. Rose, "Incorporating uncertainty into short-term travel time predictions," *Transportation Research Part C: Emerging Technologies*, vol. 19, pp. 1006-1018, 2011.
- [84] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica: Journal of the Econometric Society*, pp. 987-1007, 1982.

- [85] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, pp. 307-327, 1986.
- [86] C. Chen, J. Hu, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with ARIMA-GARCH model," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, 2011, pp. 607-612.
- [87] Y. Kamarianakis, A. Kanas, and P. Prastacos, "Modeling traffic volatility dynamics in an urban network," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1923, pp. 18-27, 2005.
- [88] Y. Zhang, R. Sun, A. Haghani, and X. Zeng, "Univariate Volatility-Based Models for Improving Quality of Travel Time Reliability Forecasting," in *Transportation Research Board 92nd Annual Meeting*, 2013.
- [89] T. Tsekeris and A. Stathopoulos, "Real-time traffic volatility forecasting in urban arterial networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1964, pp. 146-156, 2006.
- [90] T. Tsekeris and A. Stathopoulos, "Short-term prediction of urban traffic variability: Stochastic volatility modeling approach," *Journal of Transportation Engineering*, vol. 136, pp. 606-613, 2009.
- [91] J. Xia, Q. Nie, W. Huang, and Z. Qian, "Reliable Short-Term Traffic Flow Forecasting for Urban Roads Using a Multivariate GARCH Model," in *Transportation Research Board 92nd Annual Meeting*, 2013.
- [92] J. Guo and B. M. Williams, "Real-Time Short-Term Traffic Speed Level Forecasting and Uncertainty Quantification Using Layered Kalman Filters,"

Transportation Research Record: Journal of the Transportation Research Board, vol. 2175, pp. 28-37, 2010.

- [93] D. L. Shrestha and D. P. Solomatine, "Machine learning approaches for estimation of prediction interval for the model output," *Neural Networks*, vol. 19, pp. 225-235, 2006.
- [94] R. H. Shumway and D. S. Stoffer, *Time series analysis and its applications : with R examples*, 2nd [updated] ed. New York: Springer, 2006.
- [95] R. J. Hyndman and Y. Khandakar, "Automatic time series for forecasting: the forecast package for R," 2007.
- [96] R. S. Tsay, *Analysis of financial time series*, 3rd ed. Cambridge, Mass.: Wiley, 2010.
- [97] T. Bollerslev, R. Y. Chou, and K. F. Kroner, "ARCH modeling in finance: a review of the theory and empirical evidence," *Journal of Econometrics*, vol. 52, pp. 5-59, 1992.
- [98] T. G. Andersen and T. Bollerslev, "Intraday periodicity and volatility persistence in financial markets," *Journal of empirical finance*, vol. 4, pp. 115-158, 1997.
- [99] G. Lee and R. Engle, "A Permanent and Transitory Component Model of Stock Return Volatility," *Available at SSRN 5848*, 1993.
- [100] S. Kim, N. Shephard, and S. Chib, "Stochastic volatility: likelihood inference and comparison with ARCH models," *The Review of Economic Studies*, vol. 65, pp. 361-393, 1998.

- [101] Y. Yu and X.-L. Meng, "To center or not to center: That is not the question— an Ancillarity–Sufficiency Interweaving Strategy (ASIS) for boosting MCMC efficiency," *Journal of Computational and Graphical Statistics*, vol. 20, pp. 531-570, 2011.
- [102] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis*: CRC press, 2003.
- [103] A. Ghalanos, "rugarch: Univariate GARCH models," in *R package* version 1.2-7 ed, 2013.
- [104] A. Haghani, M. Hamed, K. F. Sadabadi, S. Young, and P. Tarnoff, "Data collection of freeway travel time ground truth with bluetooth sensors," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2160, pp. 60-68, 2010.
- [105] Z.-H. Zhou, *Ensemble methods : foundations and algorithms*. Boca Raton, FL: Taylor & Francis, 2012.
- [106] Y. Koren, "The bellkor solution to the netflix grand prize," *Netflix prize documentation*, 2009.
- [107] J. Elith, J. R. Leathwick, and T. Hastie, "A working guide to boosted regression trees," *Journal of Animal Ecology*, vol. 77, pp. 802-813, 2008.
- [108] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, pp. 123-140, 1996.
- [109] R. E. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, pp. 197-227, 1990.

- [110] M. Kearns, "Thoughts on hypothesis boosting," *Unpublished manuscript*, December, 1988.
- [111] T. K. Ho, "The random subspace method for constructing decision forests," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, pp. 832-844, 1998.
- [112] T. K. Ho, "Random decision forests," in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, 1995, pp. 278-282.
- [113] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural computation*, vol. 9, pp. 1545-1588, 1997.
- [114] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The elements of statistical learning* vol. 2: Springer, 2009.
- [115] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189-1232, 2001.
- [116] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, pp. 367-378, 2002.
- [117] T. C. f. A. T. T. Laboratory. Available:
<http://www.cattlab.umd.edu/?portfolio=ritis>
- [118] L. Breiman, *Classification and regression trees*. Belmont, Calif.: Wadsworth International Group, 1984.