ABSTRACT

The Title of Document:	ARTERIAL PROBABILISTIC TRAFFIC MODELING AND REAL-TIME TRAVEL TIME PREDICTION WITH VEHICLE PROBE DATA USING MACHINE LEARNING
	Bahar Zarin, Ph.D., 2018
Directed By:	Prof. Ali Haghani, Department of Civil and Environmental Engineering

This study proposes a probabilistic modeling framework for the estimation and prediction of link-based arterial travel time distribution using GPS data. The spatiotemporal correlations of the network are modeled using a directional acyclic graphical model, and several external variables in the prediction model are included to yield a better prediction in a variety of situations. This study also aims to investigate the effects of each factor on the travel time and the uncertainty associated with it.

In the proposed model, factors such as weather conditions, seasons, time of day, and day of the week are added as external variables in the graphical model. After determining the structure of the model, Streaming Variational Bayes (SVB) is used for training and parameter inference; this offers a valuable option when constant streaming data is utilized. SVB adaptively changes its parameters gradually with a lower computational cost, which makes the process less time-consuming and more efficient.

The analysis shows that incorporating external variables can improve the model performance.

The data used in this study is INRIX vehicle trajectory raw data from four months – February, June, July, and October of 2015 – which makes it possible to take into account the effects of seasons and weather conditions on travel time and its uncertainty.

One of the products of this study is a framework for vehicle trajectory data cleaning process including trip identification, removing outliers, and cleaning the trips data. Once the data are cleaned and ready to use, they should be mapped to the roads. The Hidden Markov Model (HMM) map matching algorithm is used to map the GPS latitude/longitude data to the Open Street Map (OSM) base map and find the traversed links between each pair of GPS points of vehicle trajectories.

Finally, a novel procedure to compare any travel time prediction model with any available commercial routing API is proposed and tested to compare the proposed model with Google API.

ARTERIAL PROBABILISTIC TRAFFIC MODELING AND REAL-TIME TRAVEL TIME PREDICTION WITH VEHICLE PROBE DATA USING MACHINE LEARNING

By

Bahar Zarin

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
2018

Advisory Committee:
Professor Ali Haghani, Chair
[Professor Larry S. Davis]
[Professor Paul M. Schonfeld]
[Dr. Mark Austin]
[Dr. Cinzia Cirillo]

Preface

This dissertation is composed as a requirement for the degree of Doctorate in Philosophy in Civil and Environmental Engineering.

Dedication

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

Acknowledgements

First and foremost I would like to express my deepest gratitude to my advisor, Dr. Ali Haghani who has been a tremendous mentor and supported me both academically and personally during my Ph.D. study. I am extremely grateful for all his contributions of time, idea, and endless support throughout my Ph.D. research and dissertation. It has been the greatest honor to learn from him and be his student.

Very special thanks to my committee members, Dr. Paul M. Schonfeld, Dr. Larry S. Davis, Dr. Cinzia Cirillo, and Dr. Mark Austin for their time, support, and their brilliant comments and suggestions to improve my research. It is an absolute honor to have them as my committee members.

I also would like to thank the Center for Transportation Technology (CATT) for giving me the opportunity to work on interesting projects. I learned a lot during my two years of research there. Additionally, I like to thank the Maryland State Highway for providing the data used in this dissertation.

I am grateful for friendship and encouragement of my graduate research group for the tremendous learning experience I gained by being in the group and learning from their research. Special thanks to Mahyar for his support and insight during my dissertation.

I would also like to thank all my friends: Kiana, Ali, Elham, Mahshid, Ali, Elham, Samira, Babak, Soroush, Marjan, Mostafa, Sepehr, Ladan, Niloofar, Pouya, Aida, Azadeh, Marjan and list goes on, for making my life beautiful and being like a family to me.

Finally, I would like to dedicate my dissertation to my parents, my brothers and my sister. Words cannot express how grateful I am for all of the sacrifices they have made

for me to get here and all the support and endless love they gave me in my entire life, thank you!

Table of Contents

Preface ii
Dedication iii
Acknowledgements iv
Table of Contents vi
List of Tables ix
List of Figures x
Chapter 1: Introduction 1
1.1 Problem Statement 3
1.2 Research Objectives 5
1.3 Research Contributions 8
1.4 Research Outline 9
Chapter 2: Literature Review 11
2.1 Parametric Models 12
2.1.1 Naïve Models 12
2.1.2 Autoregressive Models 12
2.1.3 State Space Model 13
2.1.4 Hidden Markov Model (HMMs) 14
2.1.5 Bayesian Networks (BNs) 14
2.1.6 A Feedforward Neural Network 15
2.2 Non-Parametric Models 16
2.2.1 K-nearest neighbors 16
2.3 Other Methods 16
2.4 Arterial Travel Time 17
2.5 Intersection Delay 21
2.6 Probe Data 23
2.7 Summaries and Limitations of Previous Studies 23
2.7.1 Limitations of Previous Studies 25
Chapter 3: Model Preparations 27
3.1 Data Cleaning 28
3.2 Probe Filtering 29
3.2.1 Trip Splitter 32
3.2.1.1 Time Difference Splitter 32
3.2.1.2 Way Point Sequence Splitter 33
3.2.2 Cleaning/Filtering the Trips 33
3.2.2.1 Same Time Filter 33
3.2.2.2 Same Location Filter 34
3.2.2.3 Speed Boundary Filter 34
3.2.2.4 Idle Filter 34
3.2.2.5 Location Displacement Filter 35
3.2.2.6 Trip Length Filter 35
3.2.2.7 Large Error Size Filter 36
3.2.2.8 Shaking Probe Filter 36
3.2.3 An Examples 36

3.3 Map Matching 37
3.3.1 HMM-based Map Matching 38
3.3.1.1 Measurement Probabilities 39
3.3.1.2 Transition Probabilities 39
3.3.1.3 Optimum Match 40
3.3.2 Map Matching Assumptions 40
3.4 Graphical Model 41
3.4.1 Acyclic Graphical Models 42
3.4.2 Model Assumptions 43
3.4.3 Implemented Graphical Model 44
3.4.4 Creation of the Network 46
3.5 Data 47
3.5.1 Study Area 47
3.5.2 Open Street Map (OSM) 48
3.5.3 INRIX Data 49
3.5.4 Weather Data 50
3.6 Summary 51
Chapter 4: Graphical Model 53
4.1 The External Variables. 53
4.1.1 Weather Conditions 53
4.1.2 Seasons 54
4.1.3 Day of the Week 55
4.1.4 Time of Day 55
4.1.5 The External Variables 56
4.2 Graphical Model with External Variables 57
4.3 Travel Time Allocation 59
4.4 Parameter Estimation 61
4.4.1 The Inference 61
4.4.2 Streaming Variational Bayes (SVB) 63
4.5 Summary 66
Chapter 5: Analysis 68
5.1 Case Study 68
5.1.1 Trip Summaries 70
5.2 Developed Models 71
5.2.1 The Initial Developed Graphical Model 71
5.2.2 Baseline Model 72
5.2.3 Test 73
5.3 Validation of the Proposed Model 74
5.3.1 The Model without any Variables 74
5.3.2 The Model with Day of the Week 75
5.3.3 Travel Time Distribution 76
5.4 Model Improvements 78
5.4.1 The Model with All of the External Variables 78
5.4.1 Regression Model 82
5.4.2 Comparison Between the Final Model and Improved Base Model 84
5.4.3 Comparison Between the Two Graphical Models 90

5.4.4 The Model Test on a Rainy Day 91
5.4.1 The Time of Day 93
Other Modifications 95
5.4.2 The Graphical Model with Intersection 95
5.4.3 The Graphical Model with Driver Habits Variable 99
5.5 Summary 101
Chapter 6: Comparison and Sensitivity Analysis 103
6.1 Validation 103
6.2 Shape-Matching 104
6.2.1 Hausdorff Distance 105
6.2.2 Fr´echet Distance 106
6.3 Route Similarity 107
6.4 Google Maps API 109
6.4.1 General Information 109
6.4.2 Departure Time 109
6.4.3 Travel Time with Traffic 110
6.5 Waypoint Parameter 111
6.6 Comparison 111
6.6.1 Case Studies 112
6.6.2 Results 113
6.6.2.1 Off-peak Travel Time Comparison 113
6.6.2.2 PM peak Travel Time Comparison 119
6.7 Sensitivity Analysis 121
6.7.1 Number of States 121
6.7.2 Number of Observations 124
6.8 Summary 127
Chapter 7: Summary and Conclusion 129
7.1 Summary 129
7.2 Conclusions 134
7.3 For Future Research 135
Bibliography 138

List of Tables

Table 1 Summary of trips used for each month and each day of the week in the training data set 71
Table 2 Summary of GPS points used for each month and each day of the week in the training data set 71
Table 3 Accuracy of predictions for all the weekdays 74
Table 4 Accuracy of predictions based on days of the week for the Baltimore test data 75
Table 5 Summary of winter predictions based on days of the week for the Baltimore test data 85
Table 6 Summary of summer predictions based on days of the week for the Baltimore test data 87
Table 7 Summary of fall predictions based on days of the week for the Baltimore test data 88
Table 8 Aggregated average percentage of error for different season 89
Table 9 Aggregated average percentage of error for each days of week 89
Table 10 The comparison of the model with and without weather variable 92
Table 11 Summary of winter predictions based on days of the week for daytime 94
Table 12 Google and proposed travel time prediction summary (off-peak) 117
Table 13 Sample of Google, proposed model travel time prediction and observations travel time comparison (off-peak) 118
Table 14 Google, proposed model travel time prediction and observations travel time comparison (PM peak) 120
Table 15 Sensitivity analysis on the number of states 123
Table 16 Sensitivity analysis on the amount of data 125

List of Figures

Figure 1 The change in delay over time for Baltimore, Maryland and DC (Virginia, 2017) 2
Figure 2 The change in commute time over time for Baltimore, Maryland and DC (Virginia, 2017) 2
Figure 3 The conceptual framework of modeling the arterial travel time 28
Figure 4 Sample of trips in Telenav probe data..... 30
Figure 5 Sample of trips in INRIX probe data 31
Figure 6 The data cleaning Process..... 32
Figure 7 An example of Telenav trip before and after filtering 37
Figure 8 An example of INRIX trip before and after filtering 37
Figure 9 A Map matching result instance for Baltimore city 41
Figure 10 Spatiotemporal model of arterial traffic evolution represented as a Graphical model. 46
Figure 11 Study area, downtown Baltimore..... 48
Figure 12 The graphical model with the proposed external variables 59
Figure 13 The available data on the area..... 69
Figure 14 The available data for the case study area 70
Figure 15 Comparison of prediction error between proposed model and the baseline by days of the week..... 76
Figure 16 The example link in downtown Baltimore 77
Figure 17 The travel time histogram for link 13..... 78
Figure 18 Real-time level of congestion subnetwork of Baltimore downtown..... 81
Figure 19 Real-time relative travel time of Baltimore downtown network. 82
Figure 20 Comparison of prediction error between proposed model and base model in winter..... 86
Figure 21 Comparison of prediction error between proposed model and base model in summer 86
Figure 22 Comparison of prediction error between proposed model and base model in fall 87
Figure 23 Aggregated average percentage of error for different season..... 89
Figure 24 Aggregated average percentage of error for each days of week..... 90
Figure 25 Comparison of the proposed model with seasons and weather conditions and without them..... 91
Figure 26 The comparison of the model with and without weather variable..... 93
Figure 27 Comparison of prediction error between proposed model and base model for day time in winter..... 95
Figure 28 The intersection movement demonstration..... 97
Figure 29 The proposed graphical model including the intersections 98
Figure 30 Demonstration of path travel time as the sum of links and intersections travel times 99
Figure 31 The proposed graphical model with proposed relative speed variable 100
Figure 32 Hausdorff distance as a measure of similarity 106
Figure 33 Fr´echet distance as a measure of shape similarity 107

Figure 34 Route similarity index example (two routes with similarity index of 70.05%) 108
Figure 35 The proposed framework for travel time estimation comparison between models 112
Figure 36 Proposed Model, Google API, and observed travel time comparison (off-peak) 115
Figure 37 Google and proposed model Percentage of error trend (off-peak) 117
Figure 38 Proposed Model, Google API, and observed travel time comparison (PM peak) 120
Figure 39 Sensitivity analysis on the number of states 123
Figure 40 Sensitivity analysis on the amount of data 126

Chapter 1: Introduction

Having an accurate and reliable traffic information system is the first step toward achieving active congestion control and alleviation, and can help reach a reliable network. The traffic information could be used by both travelers and agencies. One of the primary uses of travel time information is pre-trip guidance for travelers. This information impacts travelers' decisions regarding departure time, route, and mode choice. Travel time information can also be used by transportation agencies for Advanced Traffic Management System (ATMS) strategies, Emergency Transportation Operations, and Traffic Incident Management. Such applications are increasingly important in a world in which urban road transport systems experience increasing congestion year after year. Figure 1 and Figure 2 each demonstrate the increase in the congestion level for several states and cities including Washington, D.C. and Baltimore, Maryland. As shown, in these regions, average commute travel time and hours of travel delay have consistently increased over time in most cases.

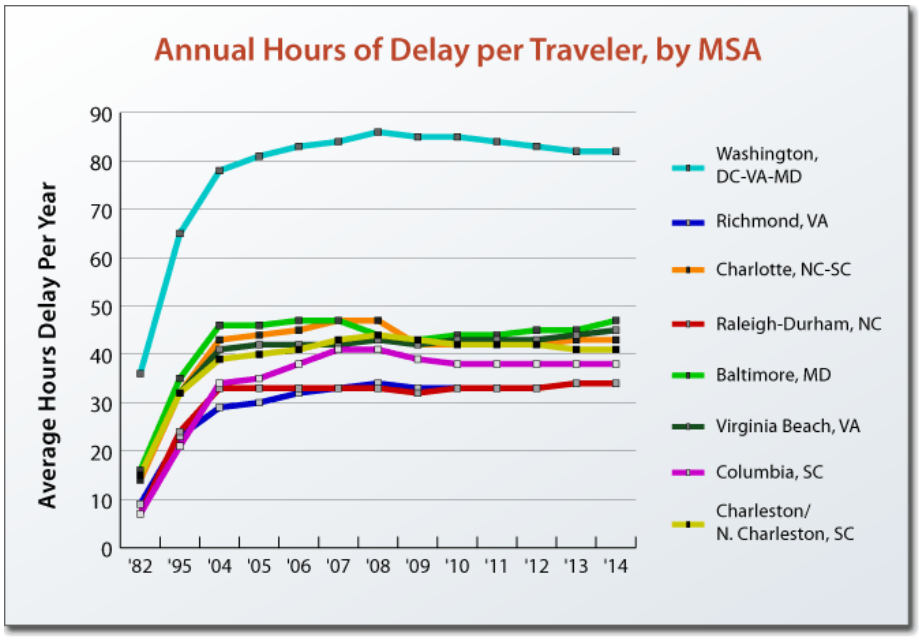

Figure 1 The change in delay over time for Baltimore, Maryland and DC (Virginia, 2017)

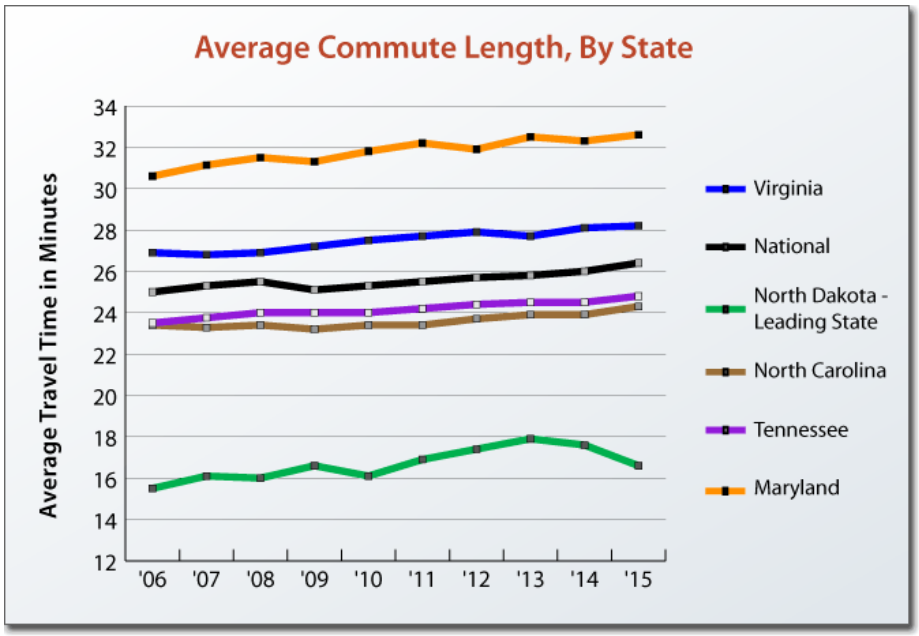

Figure 2 The change in commute time over time for Baltimore, Maryland and DC (Virginia, 2017)

Arterials are major roads that carry large volumes of traffic each day. In fact, arterials serve as the primary route of travel in major cities and as the second leading route of travel between cities. Thus, it is critical that traffic information systems offer accurate

information regarding arterial travel time. One of the most important elements for providing accurate travel time information in general and, more importantly, for arterials is travel time reliability and uncertainty associated with travel time. No matter the degree to which influential factors are considered in prediction models, there is always a degree of uncertainty associated with travel time predictions. Nevertheless, information about travel time uncertainty can be used for risk-averse routing, reporting travel time reliability to users, and for many other purposes.

1.1 Problem Statement

Historically, the use of traffic monitoring systems has been limited primarily to freeways. Arterial traffic monitoring is particularly challenging because arterials are not covered by dedicated sensing infrastructures. Probe vehicle data remains the main available source of data for arterials. Because more than 40 percent of U.S. vehicle-traveled miles occur on arterial roads – and there are not as many studies in this realm – new studies to precisely estimate and predict arterial travel time are highly needed. Many travel time forecasting models – either for freeways or arterials –provide only expected travel time. Still, by modeling statistical distributions of travel times, rather than just mean values, there is an opportunity to provide travelers with travel time reliability (or probability of on-time arrival) in addition to travel time. This probabilistic forecast of travel time is beneficial for risk-averse routing. It could also be used to report travel time reliability to users. In this research, we study the probabilistic modeling of arterial travel time.

While there have been previous efforts to establish arterial traffic estimation/prediction methodologies, which are discussed in existing literature, modeling and estimating travel time remains a challenging task. To start, there is a great deal of variation associated with travel time caused by factors such as weather, demand uncertainty, incidents, and roadway conditions, as well as varying driver behavior. Such variations are more significant in arterials as opposed to other roadways because variations associated with traffic lights, stop signs, bus stops, left turns, etc. add a higher degree of variation. Since it is impossible to fully consider the effects of all of the influential factors in the modeling process, there is always some degree of uncertainty associated with travel time estimation/prediction. It is thus essential to model this uncertainty and establish an estimation of it and how it changes due to the myriad factors.

Even though it is impossible to offer an exact estimation/prediction of travel time, incorporating as many influential factors into the modeling process as possible could produce a more accurate and more reliable estimation/prediction. Unfortunately, such factors are rarely considered in the modeling processes outlined in existing studies due to a lack of data. Low sampling frequency creates difficulties in inferring parameters. In fact, there is a direct relationship between the frequency of data and availability of data in different conditions, and the ability to include such data in the modeling process. This study thereby represents one of the only attempts to incorporate the effects of some of these aforementioned factors into the modeling process.

1.2 Research Objectives

The main objective of this study is to provide probabilistic modeling of arterial travel time using GPS data, it addresses both travel time estimation/prediction as well as uncertainty associated with it. Beyond the variability inherent in any travel time estimate due to factors such as driver behavior, there is additional uncertainty associated with arterial travel time as the result of traffic lights, and/or stop signs, and/or left turns, etc. The proposed model considers some of these variabilities and incorporates them into modeling by adding them into the modeling process in efforts to yield a more accurate estimation of travel time in the arterials. Further, the proposed model aims to capture variability associated with travel time for arterials.

Almost none of the existing travel time estimation/prediction modeling is comprehensive; in most cases, the proposed models center on normal weather conditions and specific time(s) of day and day(s) of the week. However, it is important to have an accurate travel time estimation/prediction that can be applied to a variety of scenarios, such as adverse weather conditions. Currently, even commercial estimation/prediction models lack this capability. As such, one of the primary objectives of this study is to propose a model that can provide an accurate travel time estimation/prediction in such conditions.

Another important objective of this study is to evaluate the effects of incorporating this information in modeling performance. A better performing model with this information is needed to improve mobility and network accessibility and alleviate the cost

associated with unreliability. To do so, factors such as days of the week, and time of day impact on the travel time and its variability are evaluated.

In addition to the primary objective of this study, another aim of this study is to provide a comprehensive procedure that can be used to conduct such analysis. Thus, for each step of this procedure, a separate framework is proposed and tested. The discussed procedure starts with the modeling pre-steps such as data cleaning, mapping vehicle trajectory data over a map of the observed region, and creating the city network. Then, developing a model with all of the proposed variables and testing it on a case study for several test days to compare it with other models. The final step is the creation of a framework to compare the proposed model with any commercial routing API.

This study uses raw GPS trajectory probe data in order to consider real-world situations. As previously mentioned, preparation of this data is important to this study and applications to modeling. Establishing criteria for data selection and developing a process for identifying outliers within the data pose challenges. As such, this study proposes a comprehensive framework for cleaning the raw probe data. This framework includes identifying trips from device-based probe data and applying several filters to clean trips and make them ready to be used in the model. The framework could be used for any type of vehicle trajectory data.

One of the important steps for conducting any modeling is validation in order to allow for comparisons with other models. This study uses a general method of validation that involves splitting the data into training and test datasets and validate the model on the test dataset. In addition, this study proposes a framework to compare the model with commercial navigation systems. It can be challenging to compare the proposed model

with existing commercial models, such as Google's navigation system, since neither Google's cost model nor its real-time data is available. As such, as an important follow-up step for the proposed modeling procedure, this study outlines a new method for comparing the model and the trip travel times with a commercial routing API.

In summary, this study includes the following tasks:

(1) Propose a comprehensive framework for working with raw vehicle GPS trajectory data. The framework includes data selection, data cleaning (trip splitters, probe filtering, and removal of outliers), and finally, preparation of the data to be used in the training and test models. Furthermore, the proposed framework is tested on two different vehicle GPS trajectory datasets.

(2) Propose a new model for real-time arterial travel time estimation/prediction utilizing Streaming Variational Bayes, which can be used with streaming data (for small-to-medium-sized networks).

(3) Investigate the effects of different factors, such as weather conditions, days of the week, time of day, and seasons on arterial travel time.

(4) Propose a new arterial travel time estimation/prediction framework that can be applied in different weather conditions, days of the week, and time of day, and provide more accurate and reliable predictions for any conditions. Then, compare the performance to a base model as well as to a base model improved by a regression model.

(5) Propose a new framework for comparing estimation/prediction resulting from the discussed model (or any real-time travel time prediction model, in general) with any available routing API, such as Google routing API.

1.3 Research Contributions

What follows is a list of the primary contributions of this dissertation to the field.

- We have proposed a framework for cleaning vehicle trajectory data which can be used for all types of data availability and is applicable to any study that utilizes real-world data.
- We have proposed a graphical model that takes into account different external variables such as weather conditions, seasons, time of day, and day of the week. We proposed the use of the Streaming Variational Bayes to make it possible to use the graphical model for larger-scale networks and when a number of variables are at play.
- We have evaluated the effects of incorporating each of these variables in the modeling and have addressed how such variables can impact the modeling process. We have done so by incrementally adding the variables into the model.
- Finally, we have proposed a new procedure with which one can compare any type of travel time prediction model (for any kind of network: freeway, arterial, or both) to any commercial routing API, such as Google API.

1.4 Research Outline

The rest of this dissertation is organized as follows. In the next chapter, we further review existing literature related to travel time estimation studies and the methods used in travel time estimation/prediction in general as well as in arterial travel time estimation/prediction. Chapter 3 mainly explains the preliminary steps taken before the modeling process. In this chapter, a new framework for a data cleaning and filtering process is proposed and tested on two different datasets. The chapter then addresses the next important step before the modeling – the map matching process used to map the cleaned data to the network. To do this, different map matching techniques and algorithms were reviewed; the map matching algorithm used in this study is discussed in greater detail. The last step before the modeling involves the creation of the roads network and capturing the roads' dependencies. The graphical model and the relationship between variables are further explained to demonstrate how the city network should be created.

Chapter 4 demonstrates the enhanced graphical model proposed in this study. All the variables considered in the model are explained in greater detail, as is the structure of the enhanced graphical model and the relationship between variables. The chapter continues by proposing using an approximate inference algorithm named Streaming Variational Bayes. This algorithm is used in order to make the model applicable for larger-sized networks and instances in which more variables need to be incorporated in the model. Chapter 5 outlines the performance of the proposed model with all the variables and compares the proposed model with an improved base model. Further, the

chapter analyzes the impact of including each of these variables in the model performance.

Chapter 6 explains the new approach for comparing the proposed model (or any other model) with a commercial routing API when the associated cost model is not available.

Finally, the last chapter includes the conclusion and possible future work for this study. In the last chapter, the summary of the research, the contributions, and the findings are discussed. The limitations of this study and suggestions for future work are also discussed in this chapter.

Chapter 2: Literature Review

In this chapter, some of the general ideas and methods related to estimating travel time (both arterial and freeway) are presented. Following that, more detailed literature discussing the arterial travel time estimation/prediction, and the associated methods, assumptions, and data used are provided. Throughout this dissertation, other specific detailed literature is referenced when appropriate.

There are different approaches that can be applied to the traffic estimation/prediction problem (both freeways and arterials). The approaches can be divided into three essential clusters: scope, output, and method (Vlahogianni et al., 2004). Regarding conceptual output, there are two main trends with different outcomes in the estimation of travel times using probe data. One trend is providing expected travel time. Most large-scale navigation systems, such as Apple Maps or Google Maps have this outcome, which can be applied to any large-traffic network. The other trend involves providing probability distributions of travel times and inferring their parameters using different statistical methods that can be used for small-to-medium-sized networks, as they are computationally intensive (Hunter, 2013). Studies on travel time estimation/prediction can also be divided based on their scope, be it microscopic or macroscopic. The microscopic modeling method mostly includes a small number of intersections and segments assuming complete data availability, including signal timing, vehicle counts, etc. In contrast, macroscopic modeling focuses on large scales, such as cities or countries. Another way to distinguish between studies on travel time estimation is by comparing the methods they use. There are different approaches to

model travel time, such as parametric, and non-parametric (Perry& Greene, 1982). This study focuses on those methods that use probe data. Given this, what follows are the methods used for travel time estimation, in general, as well as for arterial travel time estimation/prediction using probe data.

2.1 Parametric Models

The parametric models have strong assumptions about the data which should be satisfied, or they can result in inaccurate estimation. These models have a fixed number of parameters (Murphy, 2012).

2.1.1 Naïve Models

One of the simplest parametric models is the naïve model. The naïve models incorporate very simple methods such as historical averaging or smoothing and are considered the simplest forecasting models which are used in travel time estimation (Farokhi Sadabadi, Hamedi, & Haghani, 2010). Historical average models are merely averaging historical data with the assumption that the traffic repeats the same pattern over time and these patterns are used for future prediction (Smith & M. Demetsky, 1997; Farokhi Sadabadi et al., 2010).

2.1.2 Autoregressive Models

A time series is defined as a sequence of measurements of the same variable(s) made over time intervals (day, week, etc.). The model in which a value from one-time series is regressed on its own past values from that time series is called the autoregressive

model. The order of an autoregression in a time series is defined as "the number of immediately preceding values used for present value prediction" (onlinecourses.science, 2016). In Autoregressive modeling, an autoregressive integrated moving average (ARIMA) approach – also referred to as Box-Jenkins– is one of the approaches for analyzing and forecasting, and is a specific form of an autoregressive moving average (ARMA). Several early studies in freeway traffic modeling used ARIMA for volume and occupancy prediction (Ahmed & Cook 1979, Levin & Tsao1980). Later, the ARIMA model was used in all kind of traffic parameters forecasting studies or comparing the new models with it (Zhang, 2015; Davis & Nihan, 1990; Hamed et al., 1995; Kamarianakis & Prastacos, 2005; Williams et al., 1998; Cetin & Comert, 2006; Karlaftis & Vlahogiann, 2009; Min & Wynter, 2011).

SARIMA, on the other hand, is Seasonal Auto-Regressive Integrated Moving Average. In fact, SARIMA is an ARIMA model that incorporates the seasonal factors in the model as well. Some studies tried to capture the seasonal effect by using the SARIMA (Kumar &Vanajaksh 2015; Williams & Hoel 2003).

2.1.3 State Space Model

A state space model (SSM) models the probabilistic dependency between the latent state variable and the observed measurement (Koller & Friedman 2009). One specific type of SSMs is the hidden Markov models (HMMs), in which hidden states are discrete (Murphy, 2012). The model could be used to analyze both deterministic and stochastic dynamic systems that are observed through a stochastic process. Kalman filter and Kalman smoothing algorithms are among the most widely used algorithms in

SSM models. There are several studies that have used the state space model for predicting traffic both in freeway and arterial networks (Stathopoulos & Karlaftis 2003; Ghosh et al., 2009)

2.1.4 Hidden Markov Model (HMMs)

In general, in the (first-order) Markov chain or Markov model, the assumption is that the present is only dependent on the immediate past. In other words, Markov model centers on the idea that the immediate past captures everything we need to know about the entire history. Since this is a very strong assumption, higher-order Markov models are created whereby the hidden Markov model (HMM) offers another means through which to capture long-range correlations between observations. HMM assumes there is an underlying hidden process represented by a hidden variable at each time (Murphy, 2012).

2.1.5 Bayesian Networks (BNs)

A Bayesian network– also known as a causal model as well as a directed graphical model – is an approach to representing a system and the dependencies using a directed graph. Sun et al. (2006) developed a Bayesian network for the urban highways flow forecasting by considering the spatial/temporal correlations of traffic flow between adjacent links

There are several studies using this dynamic Bayesian network for arterial travel time modeling (Herring, 2010; Hunter et al., 2009; Hofleitner et al., 2012 a; Hofleitner et al., 2012 b). Within a dynamic Bayesian network, however, the network is not dynamic;

instead, it is fixed and problem-specific. There are different structures to be defined. First, the structure of the first time-slice must be defined, then, the structure between two time-slices, and finally, conditional probability distributions (CPDs) governing all the network's random variables must be defined.

2.1.6 A Feedforward Neural Network

A Feedforward Neural Network can be used for classification or regression problems. In fact, it is a series of logistic regression models on top of each other, and the information moves in a forward direction, with a final layer of logistic regression in cases where there is classification problem, or linear regression model if there is a regression problem. Neural Network models consist of hidden layers and weight matrices from input to hidden nodes and weight matrices from hidden nodes to the output. The hidden layers learn non-linear combinations of the original inputs. In instances where there is no nonlinearity, the model simply becomes a large linear regression model (Murphy, 2012).

There are several studies that have proposed different neural network for freeway traffic modeling. Zeng & Zhang proposed neural network models for freeway travel time forecasting (Zeng & Zhang 2013). Jiang and Adeli (2005) proposed a dynamic wavelet neural network model to predict freeway traffic flow and included time of the day and the day of the week as the explanatory variables. Even though most of these studies are for freeways traffic prediction, there are few studies using the neural network for arterial traffic modeling. Yin et al. (2002) used a fuzzy-neural model

(FNM) to predict the traffic flow in an urban street network. Park and Lee (2004) used a neural network as a classifier for arterial travel time estimation.

2.2 Non-Parametric Models

Unlike parametric models, the nonparametric model does not have a predetermined form for the predictor; rather the predictor form is derived from data. Nonparametric models do not have a predetermined number of parameters, and they could grow with the amount of data (Murphy, 2012).

2.2.1 K-nearest neighbors

One of the simplest examples of a non-parametric classifier is the K-nearest neighbor (KNN) classifier. This classifier consists of K points in the training set, each test input of x class is defined based on counts of members of each class that are nearest to that input (Murphy, 2012). Tiesyte and Jensen (2008) used the Nearest-Neighbor Trajectory (NNT) technique on real data from buses to predict future movement. The observed trajectory in this study is the path taken between two bus stops. Myung et al. (2011) proposed k nearest neighbor method to predict freeway travel times using vehicle detector data and an automatic toll collection data. Robinson and Polak (2005) proposed the use of the k nearest neighbors technique to estimate urban link travel time.

2.3 Other Methods

There are some other studies using parametric models, non-parametric models, or combinations of them. Zhang (2015) proposed a gradient boosting tree-based model to

predict freeway travel time. Hamner (2010) proposed the use of context-dependent random forest to predict travel time. Leshem and Ritov (2007) used Adaboost Algorithm with random forests as a weak learner to predict freeway traffic flow. Yu et al. (2010) applied the support vector machine (SVM) and Kalman filtering to predict bus travel time. Herring proposed Bayesian inference as one of the proposed models to estimate arterial travel time (2010).

Huang et al. (2014) proposed a deep learning approach for freeway traffic flow prediction. They used the combination of a deep belief network at the bottom and a multi-task regression layer at the top of the architecture to model traffic flow. Yu et al. (2017) proposed a deep neural network based on long short-term memory to forecast freeway peak-hour traffic and extreme condition traffic (for example, post-accident). Cui et al. (2018) proposed a deep bidirectional and unidirectional long short-term memory neural network architecture to predict freeway network-wide speed.

2.4 Arterial Travel Time

Thus far, this paper has outlined the general ideas of, and the methods used in travel time estimation. What follows offers a more detailed discussion of the probabilistic methods in arterial travel time estimation and prediction.

Herring (2010) compares different travel time estimation models including the regression model in which a logistic regression is used to classify the discrete states and STARMA for the travel time estimation/forecasting. Another considered model in this study is historical modeling using Bayesian inference for real-time estimation. Real-time estimation is calculated by considering the historical outcome as the prior

distribution and updating the parameters of distributions by weighting between the parameters from historical data and real-time observation. The last considered model is a probabilistic graphical model, in which the spatiotemporal conditional dependencies of the arterial traffic are modeled as a probabilistic graphical model. Then, Expectation-Maximization (EM) estimation algorithm, which is an iterative method to estimate maximum likelihood, is used to learn parameters. The used data is probe data from 500 taxis traveling in a small area (322 links) in San Francisco, from 3 p.m. to 8 p.m. on weekdays.

Hunter et al. clearly (2009) proposed a probabilistic model of travel times in the arterial network, based on low-frequency taxi GPS probes. In their study, an EM estimation algorithm is proposed that iterates between travel time allocation and parameter estimation. The study assumes that the travel times on the segments are normally distributed and travel times on different segments are independent. The data used in this study are obtained from a fleet of 50 taxis in San Francisco, however, the performance of the model and scale of the network are not mentioned. Hofleitner et al. further develop their primary in their later works (Hofleitner et al., 2012a; Hofleitner et al., 2012b).

Hofleitner et al. (2012 b) proposed a density model to scale the partial travel time on the partial links (the start GPS point and end GPS point). Also, they integrated the step of travel time allocation on the links is into the EM algorithm which means the observations are path travel times. In other words, instead of explicitly distributing the path travel time to its corresponding links, in their graphical model, the path travel time is treated as the same parent for every link inside the path. They employed a particle-

filtering inference in the E-step to optimize the model that can also learn how to distribute the total travel time to the links implicitly. The model is tested using Global Positioning System (GPS) data from a fleet of 500 probe taxis in San Francisco, and the studied network included 800 links.

Hofleitner et al. (2012 a) proposed a hybrid model. In this study, both the distribution parameters as well as static parameters of the roadways (such as free-flow velocity or traffic signal parameters) are to be estimated. Their primary approach is improved by introducing a state variable that represents the number of vehicles that stop (number of queuing vehicles) on a link per light (as opposed to binary state variable) and turn fractions at intersections. EM algorithm is used for the parameter learning. The model is tested using data from a fleet of 500 probe vehicles in a sub-network of San Francisco shown on 769 links.

Hunter et al. (2013) took a slightly different approach in using a graphical model. Their algorithm called "Stop & go" consists of three steps. First, they use maximum likelihood to get the number of stops for each link from the observation. Then, they use the Markov model (MM) to calculate the state transition probabilities – the probability of the number of stops on a link given the number of stops on the preceding links. Next, they use the Gaussian Markov Random Field (GMRF), which provides the joint distribution of the travel times of neighboring links, given their state. Their findings show that the model can be improved by incorporating the variability of travel times due to stops in the structure of the model. However, the independence or correlation of travel times does not have a significant impact on the model. It should be noted that the

data used in this study is 9.6 measurements per link, which means that the model needs to incorporate a high-frequency GPS measurement.

There are some other studies that propose probabilistic modeling for emergency vehicles, such as ambulances, and they mostly model the distribution of travel time on either the entire trip or based on links. However, this modeling might not work as well for general purposes since the data source is very low-volume and does not reflect travel time experiences for non-emergency vehicles (Budge et al., 2010; Westgate et al., 2015; Westgate et al., 2013). For instance, Westgate et al. (2016) proposed a regression model for travel time distribution for an entire trip over large networks including both freeways and arterials, using ambulance data. In another study, Westgate et al. (2013) used Bayesian data augmentation to simultaneously estimate the paths travel times as well as parameters of travel time distribution for each road segment.

Jenelius and Koutsopoulos (2013) proposed a spatial moving average (SMA) method for arterial travel time estimation on heavily traveled routes. In this study, travel time mean and variance is a function of network characteristics (speed limit, functional Class) and trip conditions (time of day, season, and weather conditions). This study is one of the few studies in which the network characteristics and trip conditions are considered in the modeling. However, according to the authors, the focus of this study is to model the impact and significance of different explanatory factors on the travel times rather than developing a high-performance model for the travel time estimation. In this study, Stockholm, Sweden probe vehicle data from a taxi fleet is used.

Woodard et al. (2017) proposed a probabilistic estimation of travel time for large networks, including both freeways and arterial networks, using Bing Maps data. In this work, the travel time for each link is considered as a Gaussian variable conditioned on some latent unobserved random variables which depend only on the same link. Assuming some probability distributions over the latent random variables, the team used the Maximum A Posteriori (MAP) method to find the set of model parameters that maximize the posterior distribution. However, the proposed model has the limiting assumption that the state of each link is independent of the others.

Yang et al. (2017) proposed a modified Gaussian mixture model to estimate arterial link travel time distributions. Their model could be used when either fixed-location sensors or mobile sensors data is available. They used simulation and a small number of real-world data to validate their model.

2.5 Intersection Delay

Intersections play a critical role in travel time estimation/prediction, but they are rarely considered in the route's travel time estimation/prediction models. In fact, the travel time of a trip consists of two parts: road section travel time and intersection delays. Intersection delay is the timestamp difference between the times of vehicles entering and leaving an intersection.

Tang et al. (2016) proposed a microscopic method to estimate intersection travel time. Their method assumes that the intersection travel pattern is different based on the different flow pattern. They divide the traffic flow patterns into four different categories: free traffic flow, minor impediment, moderate impediment, and serious

impediment. In their study, they first recognize the flow pattern, and then using the linear fuzzy regression method, they obtain the intersection travel time for each mentioned flow pattern.

Ban et al. (2009) proposed a microscopic model to estimate the individual intersection delay pattern and validated it using microsimulation. Their proposed model identifies the observed sampled travel times measured between upstream and downstream locations of a signalized intersection, and it estimates the intersection delay pattern. The group's proposed model includes two steps, the first of which detects the start and end of a cycle by considering the queue-forming and discharging process pattern, and the second of which uses a least-squares–based linear-fitting algorithm to estimate the delay pattern in each cycle. They considered two different flow conditions which are normal and oversaturated conditions.

Hao et al. (2017) proposed a model to estimate vehicle state trajectory that can potentially be used to extract travel time and delays on arterial networks. In their model, the team considered four different modal activities: acceleration, deceleration, cruising, and idling. Further, the group considered all of the possible sequences of the modal activities, and the sequence that maximizes the likelihood of the product of probabilities for multiple independent events is chosen as the vehicle speed trajectory. Finally, they compared their model with linear interpolation as a baseline using Next-Generation SIMulation dataset and showed they reached better results compared to the baseline method.

Even though there have been some studies addressing intersection-delay estimation, hardly any macroscopic level study of intersection delay estimation has been published.

In other words, there are few studies that consider both link travel times and intersection delays together and as part of the travel time estimation or prediction for a whole trip.

2.6 Probe Data

One of the most valuable types of data used in transportation studies is probe data. Probe data can be used to develop applications that can improve roadway operations, planning, and maintenance, and keep travelers informed of travel conditions. Probe technology uses location-aware and internet-enabled devices that are either installed in vehicles or carried by travelers. The providers of probe data can vary. For instance, the source could be GPS-equipped commercial fleet vehicles, such as trucks, package delivery vans, taxi vehicles, construction vehicles, or buses. The disadvantage of this data, however, that it is based on the specific spatiotemporal travel patterns of these vehicles. Another source of probe data is participatory sensing, such as INRIX, TomTom, Google, or Nokia – all of which collect location data from GPS-equipped smartphones, personal navigation devices, vehicles with embedded GPS, and other mobile consumer devices. The source of data in this study is probe data (either mobile data or vehicle embedded GPS data) provided by INRIX.

2.7 Summaries and Limitations of Previous Studies

In summary, traffic estimation/prediction models using probe data can be categorized based on three main aspects: scope, output, and method. Based on the output, there are two different potential outcomes for expected travel time and probability distributions. With regards to scope, there are two main options: microscopic (which is based on a

small number of intersections and segments, assuming comprehensive data availability) and macroscopic. Finally, based on the method, there are two main approaches: parametric, non-parametric. It should be noted that there might be some overlaps between these categories.

Since this study utilizes a macroscopic scale with an output of probabilistic distribution, most of the supporting literature reviews stem from those same categories. Parametric models are usually based, in part, on theoretical reasoning and strong assumptions about the data. Some of the frequently used parametric models applied to travel time prediction are autoregressive models, Hidden Markov Model, and Feedforward Neural Network. Non-parametric models, however, often do not carry any assumptions about the data, and their parameters grow as the direct result of the amount of data used. One of the most frequently used non-parametric models is the k-nearest neighborhood model.

When it comes to real-time arterial travel time prediction, some models are used more than the others. For example, some studies have used simple Bayesian inference or regression models to estimate the historic travel time and considered a factor deduced from real-time traffic condition that is multiplied by the historic estimation to calculate the future prediction. One of the more sophisticated models used in real-time arterial travel time prediction is the graphical model through which one considers the state of traffic as a hidden variable in the model and uses EM to solve it. Other models have implemented complex regression models in which the trip and link variables take into account some probabilistic assumptions about the variables and use the Likelihood estimator (MLE) tor Maximum A Posteriori (MAP) to solve it.

What follows discusses some of the limitations of the reviewed literature and outlines how this study is addressing those limitations.

2.7.1 Limitations of Previous Studies

Many previous studies rely on either taxi data or bus data (and in some case, ambulance data). Although these data are valuable because of their availability, they might not be representative of general travel time and, in most cases, these vehicles experience higher travel times (Jenelius & Koutsopoulos, 2013). The reason for choosing the mentioned probe data is because the other type of high frequency of probe data is usually not available for researchers to use.

Additionally, previous studies rarely consider weather conditions despite the fact that the impact of weather on the travel time is critical, however, it is one of the most needed areas to focus on since the existing models including the commercial ones (Google map, etc.), have the lowest performance in bad weather conditions. Another factor that is rarely addressed is taking into account the driver behavior. Many studies are also limited by the fact that they treat the intersection as part of the links rather than treat intersections as separate entities. Despite the fact that the link and the entering intersection are in many ways related, treating these as separate entities might result in better traffic modeling. Finally, most existing probabilistic models are applicable only to small-sized networks because many of them are not computationally efficient for large-road networks and/or large datasets.

Recognizing these challenges, this study aims to propose a probabilistic model that uses vehicle trajectory data and brings different factors, such as weather conditions and

travel behavior, into the modeling process. This study also considers the intersections and links separately in order to provide different travel time distributions for each. Last but not least, this study uses Streaming Variational Bayes for the inference which makes it applicable for large-sized networks as well.

Chapter 3: Model Preparations

In a perfect world, there are perfect data - and engineers simply need to find the best model that can use that data toward a final objective. In line with this thinking, many studies try to find cleaned data to use, but this limits the area of study. In fact, in the real world, there is always “dirty data” that need to be cleaned prior to any applications for modeling. Thus, it is critical to have a data cleaning framework in place, and this chapter provides such framework.

On the other hand, in this study, we have the GPS trajectory data as the observations and city roads network as the variables for which we want to predict the travel time. Thus, the next step involves mapping these observations (GPS trajectory data) on the roads network. Additionally, it is necessary to understand the structure of the city and how the road traffic could be interrelated. In order to do this, the main structure of the graphical model is explained; this model provides the primary building blocks for the model in the next chapter with all the variables.

This chapter is dedicated to all of the aforementioned preliminary steps taken prior to the modeling. Additionally, this chapter discusses input data, including geographic map and weather data. Figure 3 illustrates the modeling framework and demonstrates how the needed data and model are interrelated.

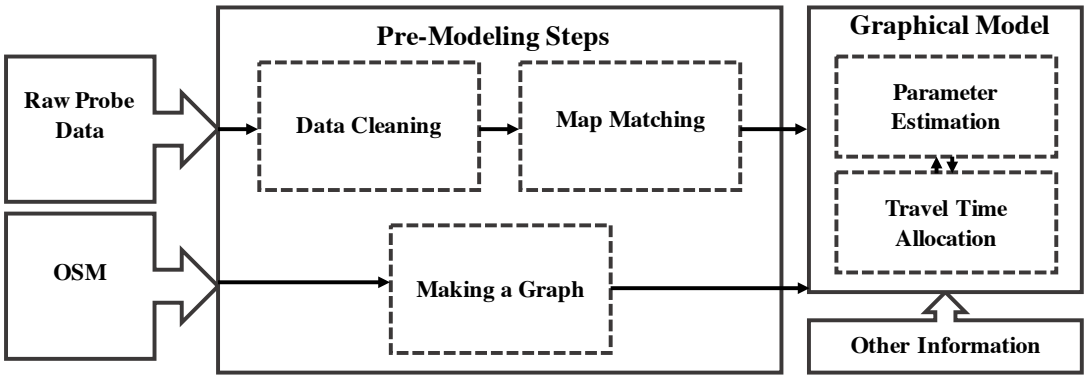

Figure 3 The conceptual framework of modeling the arterial travel time

3.1 Data Cleaning

Many studies focus on the cleaning of probe data, with different end objectives. Chung et al. (2003) proposed a framework to clean the probe data to determine the origin-destination (OD) pattern of the probes. Their proposed data cleaning framework consists of six considerations to find the correct trip destination. The six considerations are the gap between parking brake event, the long gap, the gap with unrealistic speed, the long stop, the short stop with hazard light, and the U-turn.

There are other studies that attempt to implement methods to complete an incomplete data. Hao et al. (2017) proposed a model to find the vehicle trajectory between two observations rather than just merely interpolate between them.

In this study, a data cleaning process for the vehicle trajectory data with the purpose of travel time estimation/prediction is proposed.

The vehicle trajectory data contains the GPS trace information of any vehicle traversing a path from an origin to a destination. For each vehicle along its path, there is a GPS report (including information such as location, time, speed, etc.) received for every time interval (which varies in different datasets).

However, these GPS reports are most of the time not clean and ready to use. Various external causes may degrade the quality of the reports. GPS device may not work as expected, the GPS localization may not be accurate in some regions, GPS signal may not be fixed in some time intervals, etc. Inaccurate data results in inaccurate modeling. Consequently, this proposes a framework to clean the waypoint GPS traces before modeling the travel time. This framework deals with problems such as idling, impossible speed, impossible sequence, impossible shift in location or time, etc.

What follows explains the data cleaning process in the most general form, in order to allow it to be used for any kind of data. The framework is explained in the most comprehensive manner and can be applied to any available data. The cleaning filters take into account all possible information, including latitude, longitude, timestamp, speed, headway, and GPS error. Also, the process assumes there is no order to the data. Based on the availability of data and prior cleaning processes, a subset of this filtering might be used.

3.2 Probe Filtering

If the provided probe data is raw data that is not cleaned in any way, it can be difficult to use for modeling. Use of incorrect data results in the development of an inaccurate model. As such, it is essential to have a useful framework and to develop a thorough

cleaning process. What follows is the explanation of the "probe filter" framework, which explains the procedure to find tripes, clean the probe data and eliminate noise and outliers. The proposed framework is applied to both TeleNav data and INRIX data, based on their availabilities. Figure 4 shows an example of TeleNav raw data in the Santa Clara, California area. TeleNav data is pulled from different providers; each provider has different standards of quality and uses different time intervals between the GPS reports (ranging from one second between reports to a couple of minutes). The Telenav data is only used in the data cleaning process. The data is the probe data with a one-second interval and the data is device-based. The data are from 2015 to 2017 and are for the cities in northern California, including Santa Clara, San Jose, Mountain View, and Palo Alto.

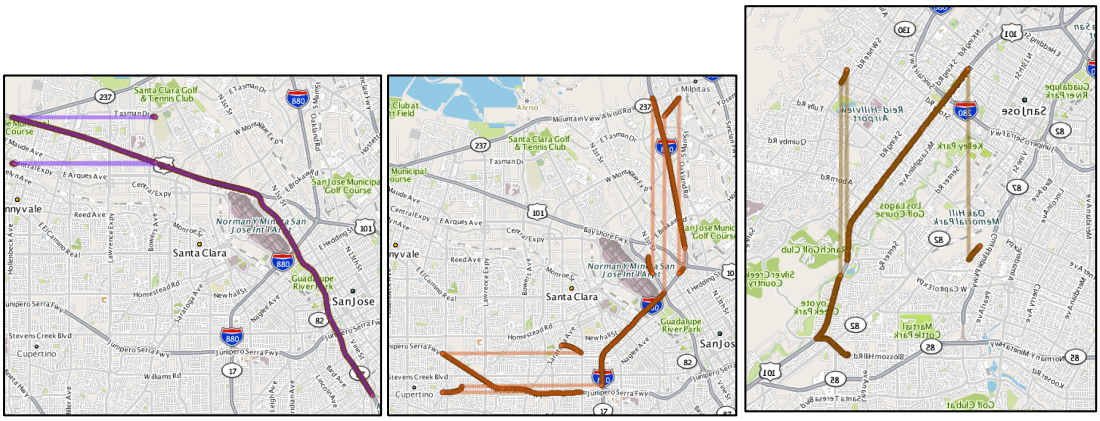

Figure 4 Sample of trips in Telenav probe data

Figure 5 shows the example of the INRIX probe data. As mentioned, from the INRIX data, the Baltimore area is selected, and the data covers four months of data with the interval of one minute between GPS reports. The INRIX data is not as raw, but it still needs to undergo a cleaning process.

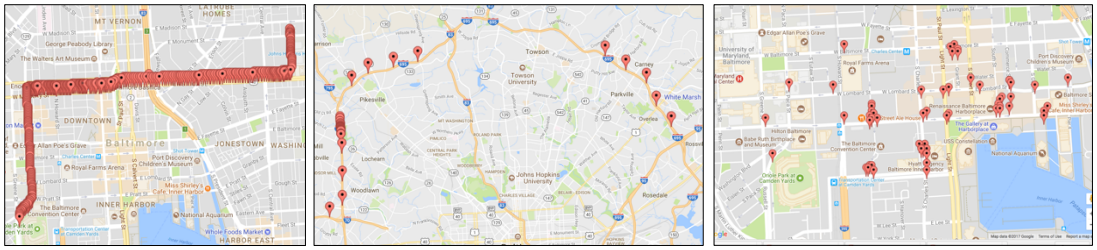

Figure 5 Sample of trips in INRIX probe data

If the data is device-based (TeleNav), then the cleaning process involves two different steps. The first step centers on finding the trips and the second step centers on cleaning the data from each trip. If the data is trip-based (INRIX) then only the second step is needed.

What follows explains each of these two steps:

1. Finding the trips from the device-based probe data (trip splitter).
2. Cleaning the trip (applying the rest of the filters).

Here, all the filters are applied consecutively to the probe data; however, one might choose only a few of the filters according to the need, data availability, and the objective of the study. Figure 6 shows the cleaning process.

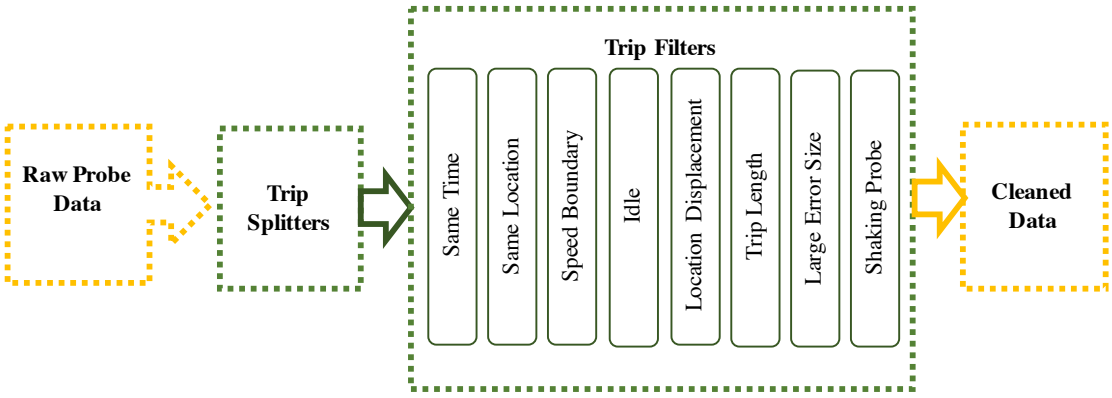

Figure 6 The data cleaning Process

3.2.1 Trip Splitter

The trip splitter might be used for one of two reasons: the waypoints' unique identifier might be device-based rather that trip-based or the trip might need to be split into more than one trip in order to use it in the modeling. The three different criteria to split the trips are time, location, and the waypoint (if available).

3.2.1.1 Time Difference Splitter

First, if the data are device-based (rather than trip-based), the data are split based on the datestamp or timestamp of the data. For instance, the first trip data might be split if it is from different days. Then, on the same day a time interval is considered above which the probe data is split into different trips (e.g. max time difference threshold = 120 sec). This time interval depends on the GPS report timestamp (for example, if it is one minute or one second between each of the two consecuitive GPS reports). This means that the probe data with a time interval over a certain threshold might be split

into two trips. The challenge with this filter, however, is that if we choose a higher time-difference threshold, we might incorrectly select two different trips as one and then we might lose the whole trip in the cleaning process ("shifted probe filter"). If we choose lower threshold, we might split one trip into more trips, but that won't cause any problem since we won't loose any information aside from the actual origin-destination of the trip. If the data is already trip-based, there is no need for this operation.

3.2.1.2 Way Point Sequence Splitter

The last step in the trip splitter process is splitting if the sequence of the waypoints is not ordered or when there is a gap in the sequence. This is due to the fact that we limited the study area to a boundary, and those trips that go out of the boundary and come back again would have a gap in their sequence. Thus, those trips are split every time they pass the boundary.

3.2.2 Cleaning/Filtering the Trips

3.2.2.1 Same Time Filter

This filter removes the probes that have the same timestamp. In a way, the first probe with a unique timestamp is kept, but all subsequent probes with the same timestamps are removed. This filter is a partial filter, which means that only the probes that have this duplicative characteristic are removed from the trip data.

3.2.2.2 Same Location Filter

This filter is also a partial filter; it removes those probes that are based on the same location. This process involves keeping the first probe and removing all the subsequent probes that have a distance lower than a minimum threshold from the first point (e.g. minimum distance between probes of 1 meter).

3.2.2.3 Speed Boundary Filter

This filter is used to filter out those trips that feature an impossible speed. The filter removes those probes that have speeds registering above the maximum speed (e.g. maximum speed = 200 mph). Similarly, if the average trip speed is above the maximum or below the minimum speed (e.g. minimum speed = 2 mph and maximum speed = 200 mph), then the whole trip is removed.This filter could be used even when speed data is not available because the speed could be calculated based on the location and timestamp data. This filter could be partial, or it could target the entire trip.

3.2.2.4 Idle Filter

This filter removes the probes that are identified as idle probes, such as when the speed is below a minimum threshold speed, from a trip. Applying this filter means that, if a car is idle in the gas station, those probes will be removed; similarly, if time is important to us the travel time from origin to destination would be much higher. To address this problem, the time difference threshold (trip splitter) could be checked once again. If removing the filter causes the time difference between the consecutive probes to be more than the time difference threshold, the trip should be split into two trips. Another

potential problem with this filter is the probes behind the traffic light, which are removed because they indicate a speed below the threshold (e.g. minimum speed of 1.0 mph). This filter could be used regardless of whether or not speed data is available since the speed could be calculated based on locations and timestamps.

3.2.2.5 Location Displacement Filter

This filter removes the whole trip if there is at least one shifted coordinate by more than max Distance threshold (e.g. maximum miles between probes = 3). This filter is not a partial filter and if the conditions are met the whole trip will be removed. The problem with this filter could be losing a lot of data if we have a higher threshold of max distance and having a lot of shifted probes if we choose the lower max distance. One solution to this problem could be to avoid losing this data by either splitting the trips to smaller ones, or removing the shifted probes and interpolating between the remaining probes or using trajectory identification techniques (Hao et al. 2017) instead of the interpolation to improve the accuracy.

3.2.2.6 Trip Length Filter

This filter removes the entire trip if the probe count is less than a minimum count. The count threshold is based on the probe timestamp intervals (if they are 1-second or 1-minute interval). Here the threshold of 60 probes is used when we have the timestamp interval of 1 second and threshold of 5 probes when we have the timestamp interval of 1 minute (e.g., Min Probes Count = 60).

3.2.2.7 Large Error Size Filter

This filter is again a partial filter; it removes those probes with GPS error larger or equal to a maximum threshold error size from the trip data. Error size/accuracy is a variable in the probe data, and it shows the accuracy in meters of the GPS recording. (e.g. maximum error size in meters = 200 m) (TeleNav, 2017). This filter could be used when the error information is available in the dataset; if it is not, this filter is omitted.

3.2.2.8 Shaking Probe Filter

A probe is identified as "shaking" if the previous and next probes have a heading angle difference outside the maximum angle difference threshold. This filter removes the shaking probes from the trip data, and if the number of shaking probes is too large, the whole trip is removed. (maximum percentage of angle difference for shaking probes = 40%) (Telenav, 2017). This filter could be used when there is a heading; if there is no heading, this filter can be omitted.

3.2.3 An Examples

The following examples (Figure 7 and Figure 8) show the before and after of the filtering. Figure 7 demonstrates how the filtering process solves the displacement probe problem by splilliting the trip to two trips. Figure 8 demonstrates how the time-difference problem is solved by splitting the trip into two trips.

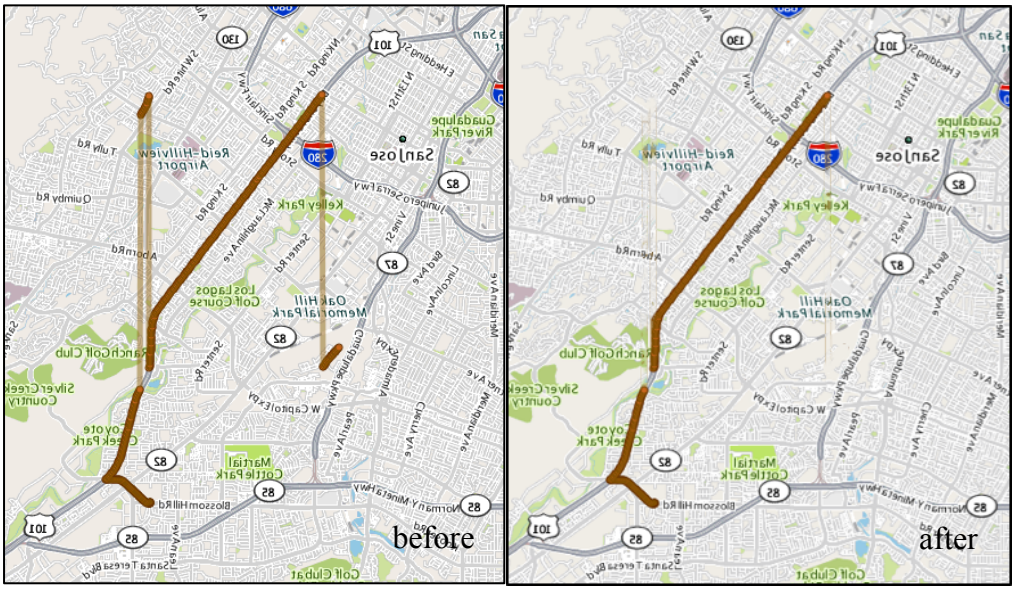

Figure 7 An example of Telenav trip before and after filtering

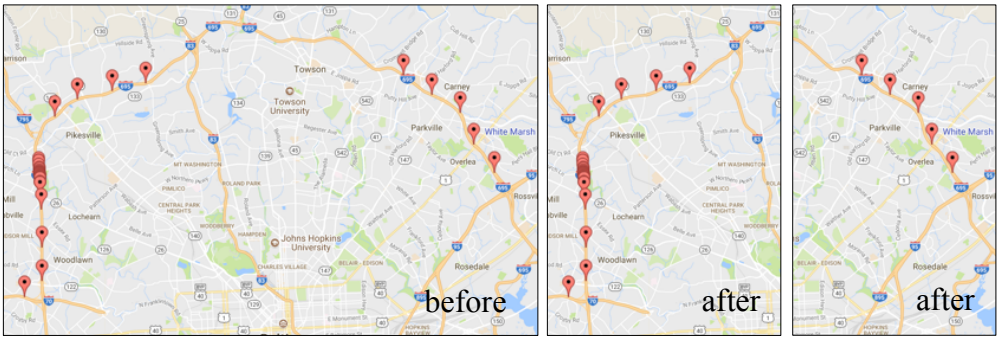

Figure 8 An example of INRIX trip before and after filtering

3.3 Map Matching

One problem with GPS data is that the data points are never precise enough, so they might be off the road. Also, we might not know the traversed links between two GPS reports. For example, in our proposed model, the observations that are used as input are link-based, however, what we collect from the data are GPS reports that are randomly reported. As such, we might have more than one GPS report in one link – which is very rare for low-frequency GPS data – or we might have only one GPS report

for several links. Therefore, it is important to find the traversed links between every two consecutive GPS reports. To address this, it is important to map these points on the road. This problem of matching measured latitude/longitude points to roads, along with the path between two points, is called map matching.

There are a lot of studies about map matching and what follows provides a very brief literature review of this concept. These approaches include a geometric approach in which the algorithm tries to match a path that has a similar curve to observed locations (Greenfeld, 2002; Brakatsoulas et al., 2005; Kim &kim 2001). Another approach uses HMM and Kalman filter (Krumm et al., 2007; Newson & Krumm 2009). Some such approaches also include the uncertainty associated with the path; however, studies have found little benefit to using this in predicting travel time distributions on routes (Westgate et al., 2015; Woodard et al., 2017).

In this study, an HMM-based map matching algorithm, according to the Newson and Krumm paper, is implemented (Newson and Krumm, 2009), and it is adapted in accordance with this study's needs. What follows explains the Newson and Krumm HMM-based map matching proposed method, as well as assumptions and considerations in this study.

3.3.1 HMM-based Map Matching

Hidden Markov Model (HMM) map matching finds the most likely point on the map for each set of latitude/longitude coordinates, as well as most likely links sequence between two consecutive latitude/longitude pairs.

3.3.1.1 Measurement Probabilities

Due to GPS noise, GPS observations rarely point exactly to a link; but, it is important to have GPS points on the links. Relatedly, measurement probability is the likelihood that location observation denoted as o_t, resulting from a given state l_i (a link) and based on that observation itself. Measurement probability is the likelihood that the observation O_t would be observed if the vehicle were actually on road segment l_i. For a given o_t and l_i, the closest point on the links is denoted as y_t, and $| o_t - y_{t,i}|$ is the distance on the surface between the observed point and the candidate match on the map. Ideally, this calculated figure should be as low as possible (Newson and Krumm, 2009).

3.3.1.2 Transition Probabilities

Once all the latitude/longitudes are mapped on the links, the next step involves finding the traveled links sequence between each consecutive latitude/longitude points pair. Transition probabilities are the probability of a vehicle traversing a set of links from the first point (i) to the second one (j). If the observation at that time is o_t and the next observation is o_{t+1}, and their candidate matches are $y_{t,}$ and $y_{t+1,\,j}$ respectively, the path distance would be $| y_{t,} - y_{t+1,j} |_{path}$, which is the network distance between $y_{t,i}$ and $y_{t+1,j}$ (sum of all the links' lengths from the first point to the second point), which should be as small as possible. Based on the Newson and Krumm study (2009), the histogram of the absolute distance differences follows an exponential distribution shown in Equation1, where β is the parameter to be estimated.

$$p(d_t) = \frac{1}{\beta} e^{d_t/\beta} \quad (1)$$

$$d_t = ||o_t - o_{t+1}| - |y_{t,i} - y_{t+1,j}|| \quad (2)$$

3.3.1.3 Optimum Match

Newson and Krumm (2009) proposed using Viterbi algorithm to find the best point matches and path between each pair of consecutive points. The Viterbi algorithm finds the match that maximizes the product of the measurement probabilities and transition probabilities through dynamic programming (Newson and Krumm, 2009).

3.3.2 Map Matching Assumptions

There are several assumptions inherent to the implemented map matching algorithm in this study and the used coding. The following are some of them.

1. When it comes to mapping each GPS latitude and longitude to the map, the priority is with intersections. Meaning the intersections are strong candidates to which we can map the GPS reports.
2. If there is more than one observation per link per vehicle, only one is considered in the modeling.

Figure 9 shows the results of one of the map matchings for the study area. The map on the left shows the observations with the reported GPS points, and the map on the right shows the result of map matching and the path that the hidden Markov map matching algorithm suggests.

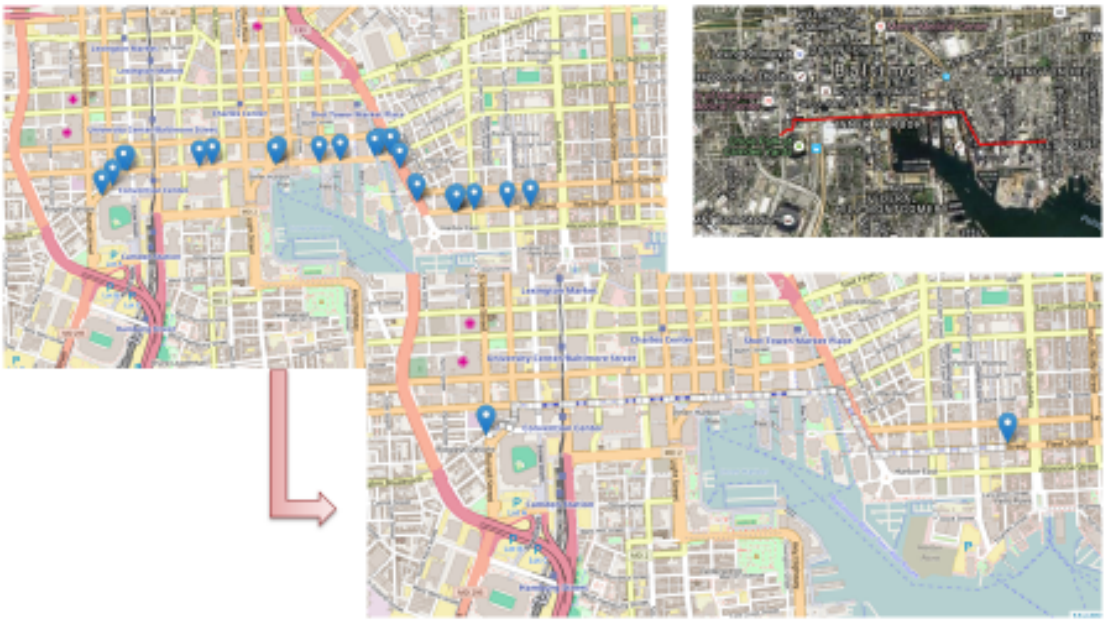

Figure 9 A Map matching result instance for Baltimore city

3.4 Graphical Model

A graphical model is a probabilistic model in which conditional dependencies between random variables are demonstrated by a graph. The vertices of this graph represent the random variables, and the edges show the statistical dependence between the variables. Graphical models have been widely applied to different problems due to their intuitive representation, the natural way to define independency between variables, and the efficient approximate methods developed to perform the inference once the graph structure is defined. These models are successfully used in probability theory, machine learning, statistics, and, in particular, Bayesian statistics. There are several types of graphical models; Bayesian networks and Markov random fields are two of the widely used branches of graphical representations of distributions (Koller & Friedman, 2009).

In arterial traffic modeling, the spatiotemporal conditional dependencies of the network traffic could be modeled as a probabilistic graphical model.

3.4.1 Acyclic Graphical Models

Bayesian Networks serve as graphical models with structures as directed acyclic graph. This model represents a factor of the joint probability of all random variables. Equation 3 is this joint probability.

$$p[Y_1, \dots, Y_n] = \prod_{i=1} P(Y_i | pa_i) \quad (3)$$

Where p_{ai} is the set of parents of node Y_i, and $Y_1, \dots, Y_n$ are the events. In other words, the joint distribution factors become a product of conditional distributions. Once the problem is formulated by means of a Bayesian network, the inference problem should be solved. In general, the inference can be defined as the problem of evaluating the distribution over a set of random variables, given another set of random variables. There are three different main classes of inference for Bayesian networks: inferring unobserved variables, inferring parameters, and structure learning (Russell & Norvig 2003). The first two are what we need to work on in this study. While it is possible to perform inference in a naïve way by computing all required probability terms, in practice this tactic is inefficient. The exact inference, however, can be solved more efficiently by means of methods like belief propagation and factor graph propagation for singly connected graphs. Moreover, for more complex problems, when the exact inference becomes intractable, approximate methods such as loopy belief propagation can be applied to reduce the computational complexity.

In parameter learning, the inference variables of interest are the model parameters. There are, in general, two cases in which parameter estimation is considered: inference when all variables are observed, and inference when variables are partially observed. Approaches toward inferring the model parameters can be divided into Maximum Likelihood and Bayesian approaches. In the Maximum Likelihood approach, the parameters of the model are estimated using the Maximum Likelihood estimator, when all variables are observed. When some of the variables are hidden (*i.e.* unobserved) the Expectation Maximization paradigm can be applied to solve the problem. On the other hand, the Bayesian approach takes the prior distribution over the parameters into account. The prior distribution can be one of any number of different distributions, but Drichlet distribution is usually deployed due to its flexibility. This leads to updating Drichlet distributions when all variables are observed. When there are hidden variables in the model, methods such as the Markov Chain Monte Carlo (MCMC), Viterbi algorithm, or Variational Bayes(VB) methods can be applied. Variational Bayes methods are among the most efficient Bayesian inference techniques and, in terms of time complexity, they are similar to the Expectation Maximization approaches.

3.4.2 Model Assumptions

Our base graphical model has the following assumptions:

1. The travel times of links are normally distributed and travel times on different links are independent of each other.
2. Discrete congestion states: a discrete value of s, indicates the level of congestion. This variable is not observable; it is a hidden variable.

3. Conditional independence of link travel times: conditioned on the state of a link, the travel time distribution of that link is independent of all other traffic variables. In words, link travel times are not correlated across links. This assumption is based on computational tractability.
4. Conditional independence of state transitions: conditioned on the states of the spatial neighbors of link l, at time t, the state of link l at time t + 1 is independent of all other current link states, all past link states, and all past travel time observations. This assumption means that each link is correlated with a few neighboring subsets of neighboring links, but remains independent of the rest of the network.

3.4.3 Implemented Graphical Model

Arterial traffic has space and time components. The spatiotemporal conditional dependencies of arterial traffic are modeled using a probabilistic graphical model, or Dynamic Bayesian Network. Each node is a representation of a random variable, and each arrow represents a dependency. The hatched nodes represent the observed variables, and non-hatched nodes represent the hidden variables. In our graph, the discrete state of each link l and time period t is denoted $X_{l,t}$. Since the state of each link for all times is not observable, these variables are considered hidden variables and shown by non-hatched nodes. The observed travel time on link l and time period t is denoted as $Y_{l,t}$, shown by hatched nodes. The index d is excluded from the variables according to assumption 1. The forward arrows show the spatial and temporal dependencies of the network. As shown in Figure 10, each $Y_{l,t}$ has the parent $X_{l,t}$ which

is a representation of assumption 2 and means that the travel time distribution on each link is conditioned solely on its hidden state. Also, each state $X_{l,t}$, has parents of $X_{l,t-1}$, and $X_{pa(l),t-1}$which is same as assumption 4, meaning a state on a link at time period t, is solely conditioned on the neighboring links' state at the previous time step. Later, this model is improved by including other information with adding observation nodes to the model.

This system is assumed to be a Markov process with unobserved states that evolve over two time steps. It is also referred to as the Coupled Hidden Markov Model (CHMM). There are three probabilities that need to be estimated to utilize this model:

- The initial probability of the state of congestions denoted $\pi_{l,s}$: It is necessary to estimate the initial state probabilities for each link.
- The discrete transition probability distribution functions, denoted $P_{l,t}$, which refer to the transition probability between the states. In other words, the element of line m and column s in the matrix of $P_{l,t}$ (m; s), represents the probability of link l to be in state s at time t + 1 given that the neighbors of l are in state m at time t.
- The distribution of travel time on a link given the state of that link, denoted $g_{l,s,t}$. As mentioned before, the travel time on each link is dependent on the state of that link.

The outcome of this model is prediction/estimation of links travel times at each time step. In the implemented model, the dependency among the neighboring links is considered as relationships between their traffic states. However, for tractability, this

study assumes the travel time distributions of the links are independent of each other given their traffic states. This independency of the links makes it possible to calculate the path travel time by summing over the estimated links travel time encountered in that path.

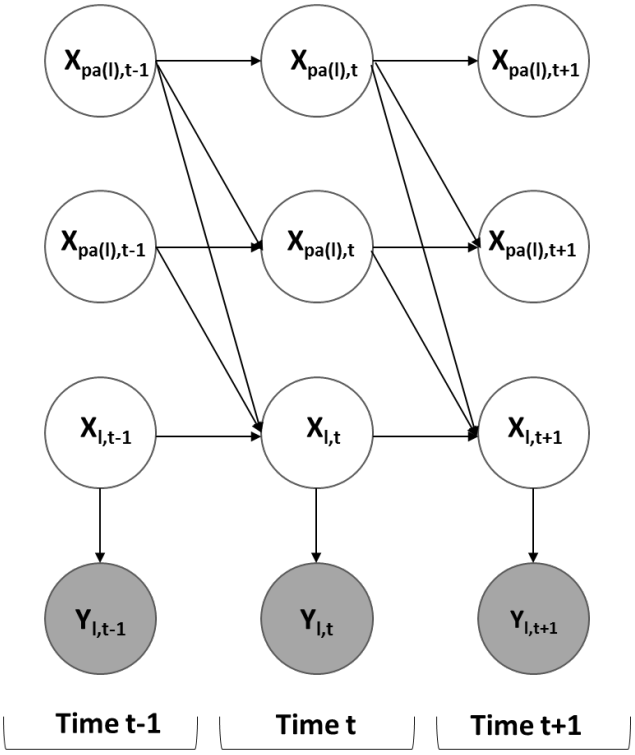

Figure 10 Spatiotemporal model of arterial traffic evolution represented as a Graphical model.

3.4.4 Creation of the Network

The network or Dynamic Acyclic Graph (DAG) of the roads is created by creating a graph of the street map. Once this graph is created, a variable is assigned to each direction of a link, in case the link runs in two directions. There are always two sets of

variables, one in time step t, and the same variable in the time step of t+1. Then, if a variable A (link and direction A) has a forward traffic direction to variable B (link and direction B), A in time step t would be the parent of B in the time step of t+1.

3.5 Data

There are different datasets used in this study: vehicle trajectory GPS probe data from INRIX, road network data from Open Street Map (OSM), and weather data and severe weather data from different sources. What follows explains each of these datasets in more detail.

3.5.1 Study Area

The study area at this stage is Baltimore, MD with more than 1,700 links and 1,250 intersections (Figure 11). The network information is extracted from Planet OSM. Planet OSM provides the data from the OSM project, which is normally updated every day (Planet OSM, 2016). Once the data is obtained, a graph is made from this area that notes network information, such as links, speed limit, and roadway type. In the graph, each edge is a link between two nodes (intersections), which means that an edge could be run forward, backward or in two directions, per the direction of traffic.

The following section explains the OSM briefly.

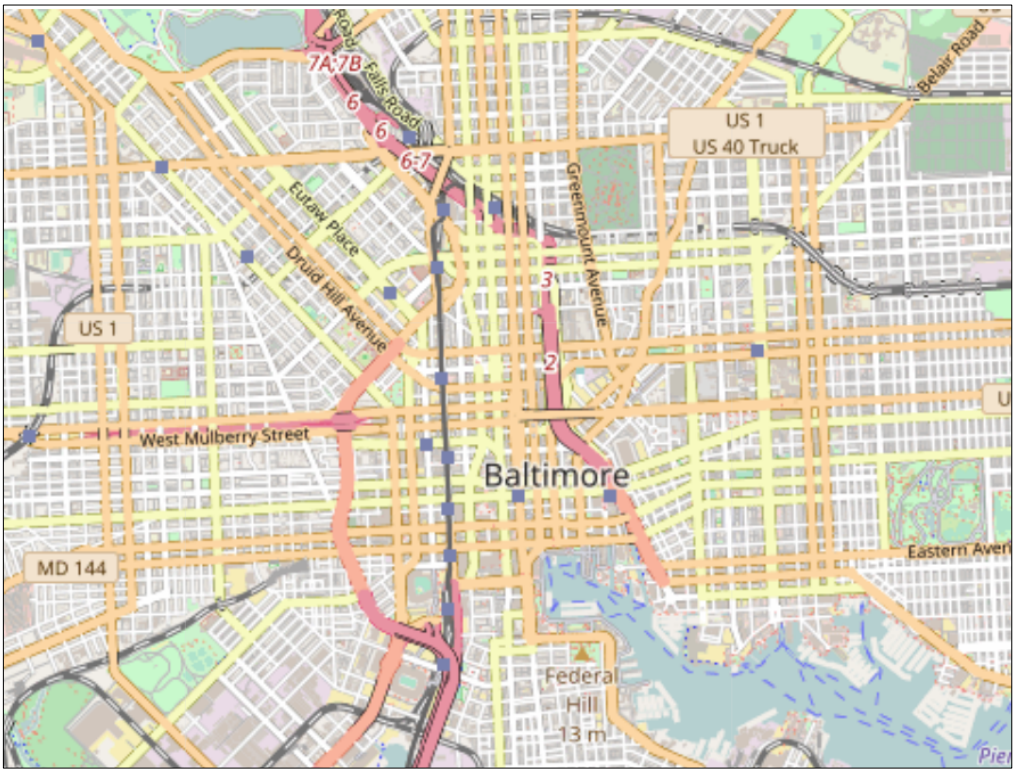

Figure 11 Study area, downtown Baltimore

3.5.2 Open Street Map (OSM)

Open Street Map is a map that features free geographic data for the world, and is created mostly by users (Open Street Map, 2016). Open Street Map represents the physical structure of the world with three main elements: nodes, ways, and relations. Elements have their specific tags which give the attributes of that element. Tags are to provide more information on a specific feature such as functional types of ways or nodes.

A node is a core element of the OSM data which has latitude, longitude, and node identification. Any point feature, ranging from a building or a traffic light to a fountain or ATM, could be represented by a node. Tags can be used to provide more information on features, such as building names or the type of an intersection. Some of the most

important nodes in arterial travel time modeling are traffic signals, stop signs, crossings, bus-stops, motorway junctions, and turning circles. In the proposed model with intersections, this data is used.

A way is another element of OSM data. It features an ordered list of nodes, and it can be open or closed. Closed ways are ways in which the first and last nodes are the same. Ways represent linear features such as roads, barriers, and railroads. Tags provide more information, such as the type of the road, the name of the road, etc. The most important ways in this study are roads that are named highways with type (tags) of motorway-link, secondary-link, primary-link, tertiary-link, trunk-link, trunk, motorway, primary, secondary, tertiary, unclassified, residential, service, road.

A relation is also a core data element. Relations are used to represent logical or geographic relationships between features. Relations consist of an ordered list of nodes, ways, and/or relations. Relations can also show all of a road's segments (Wiki Open Street Map).

3.5.3 INRIX Data

The traffic data in this study covers the full months of February, June, July and October 2015. The data includes trip information as well as trip records waypoints for each trip and trip provider details. The trip data consists of a variety of information, including start/end location and time, trip provider, vehicle weight class, source of the probe data (embedded GPS, mobile device, unknown), and an indicator in the trip data that shows the mode of transport (walk, vehicle, and rail). The data include both passenger vehicle and truck data, the information in the "vehicle weight class" is used to choose only

passenger cars or light trucks data which are the majority of the trips (Markovic et al. 2017). The waypoints data includes a trip identifier and sequence Id which notes the order of the waypoint within the trip starting with "1" and increases by one unit along with the latitude and longitude and their corresponding timestamps. The data consists of more than one billion waypoints (around 112 GB) for the available area and more than 20 million waypoints (around one GB) for the studied area in Baltimore.

It should be noted that the vehicle trajectory data is different from the Traffic Message Channel (TMC)-based Vehicle Probe Project (VPP) data provided by RITIS (2017). RITIS has TMC-based probe data for the arterial network which are provided by three vendors: INRIX, HERE, and TomTom. In fact, these providers use their own vehicle trajectory data to provide the TMC-based travel time for the arterial network. The quality of these arterial TMC-based probe data is assessed by Sharifi et al. (2017).

3.5.4 Weather Data

One type of weather data used in this study is “Quality Controlled Local Climatological Data (QCLCD),” which is high-quality data consisting of hourly, daily, and monthly summaries for more than 1,600 U.S. locations. This data is available from January 1, 2005, to the current date as of this writing (NOAA, 2015). Other weather data used in this study is from the Global Historical Climatology Network (GHCN), which collects information from stations in its database but does not offer as much quality control as the QCLCD. For those stations that are missing data due to times in which they were not functioning or are turned off, the GHCN is used (NOAA Climate, 2015). The last dataset used for this study is the adverse weather data set obtained from Storm Events

Database (NOAA Storm Events, 2015). The Storm Events Database contains data for storms and other significant weather phenomena, as well as significant meteorological events from 1950 to 2016 and features the year discussed in this study.

3.6 Summary

This section provided a comprehensive framework for the data cleaning process, which includes data splitting and trip filtering. The splitting process is more critical when the unique identifier is a device rather than a trip. The splitting process for a probe set with a unique identifier is based on the time and location and the waypoint sequence (if available). The filtering includes eight different filters and should be applied to each split trip. The filters remove the outliers or the whole trip. These filters deal with idle probes, speeds lower or higher than the possible speed, probes having the same time or same location as each other, very small-length trips, probes for trips with sudden shifts in the location, large errors, and probes with sudden or frequent changes in the heading (if the heading is available).

Once the data are cleaned, they need to be mapped to the roads. There are different map matching algorithms to do that. Map matching algorithms first find each GPS point on the map (measurement) and then find the traversed links between two GPS points (transition). HMM-based Map matching was used in this study. In HMM-based map matching, the Viterbi algorithm is used to select the mapped point and the traversed link alternative that maximizes the product of the measurement probabilities and transition probabilities.

Once the inputs are ready, the structure of the city should be created based on the graphical model assumptions. In the graphical model, each link (the roads between two intersections) is a variable, and the goal is to find the travel time distribution for each of the links (variables) at each time step. The traffic state is hidden variable, and the assumption is that travel time on each link is dependent on this state. However, in the arterial network roads, travel times are dependent on each other, and this dependency is reflected in the states; that is, each link state at time t+1 is dependent on the neighboring links at time t. The network of the roads and the adjacency of the links (to get the neighboring links) are created and the initial graph is based on them.

The data used for this study are: geographic map, which is accessed via open-source Open Street Map (OSM), the GPS trajectory probe data which is obtained from INRIX and Telenav (used only for cleaning procedure), and the weather data which was obtained from different data sources provided by NOAH.

Chapter 4: Graphical Model

In recent years, thanks in part to greater availability of data, there have been efforts to conduct probabilistic modeling of arterial travel time. It is, however, rare for any such models to consider the effects of external variables such as seasons or weather conditions. These variables are shown to be influential in both demand or driving behavior as well as travel time in freeways. Still, they are rarely considered predictive variables in arterial travel time modeling.

This chapter focuses on developing structures for enhanced graphical models, including all variables includign linkes and states variables as well as external variables into the modeling.

This chapter outlines each step of the modeling process, including travel time allocation and conducting inference using an approximate inference algorithm. Detailed algorithms for each step of the modeling are provided in the next chapter.

4.1 The External Variables.

4.1.1 Weather Conditions

Weather conditions affect travel time on both freeways and arterials network. Studies also show that weather conditions significantly affect traffic flow, demand, and travel behavior, and consequently travel time and its variations (Qiao et al., 2012; Huang & Ran, 2003; Tsirigotis et al., 2011). Qiao et al. (2012) proposed a K nearest neighbor-integrated model and incorporated the effects of weather in the freeways traffic modeling using Bluetooth data. The team demonstrated that their model outperforms

the ARIMA and KNN models in rainy weather conditions. Hranac et al. (2006) proposed weather condition adjustment factors that are multiplied by the base clear-condition variables in order to incorporate weather condition information in traffic volume estimations.

Most of these studies have been conducted for freeways, and there are rarely studies that incorporate weather conditions in arterial travel time estimation. Jenelius and Koutsopoulos (2013) considered weather conditions as an important variable in their modeling, however, their model does not provide real-time travel time estimation/prediction model.

Several types of weather information can be used to study weather impacts on traffic. The most frequent are precipitation type, precipitation intensity, visibility, average wind speed, and adverse weather. Consideration of these variables relies on data availability, and incorporation of more conditions requires the availability of data pertaining to each condition.

In this study, weather conditions are incorporated in real-time arterial travel time estimation/prediction. An external variable for weather conditions is added into the Dynamic Bayesian Network. The considered variables are precipitation intensity, visibility, and average wind speed.

4.1.2 Seasons

There are seasonal variations in demand that can affect travel times on freeways and arterials. In general, there is a seasonal pattern in traffic demand and travel time; as such, including the season as a variable factor could capture this pattern in a given

model. A lot of studies tried to use time series analysis or introducing a seasonality coefficient in the model such as, for example, Seasonal Auto Regressive Integrated Moving Average (SARIMA) to capture the seasonality effects for the travel time prediction. Most of these studies have been done more on freeway travel time modeling (Williams et al., 1998; Peng et al. ,2014) and less on arterial travel time modeling (Khoei et al., 2013).

In this study, we utilize four months of data, February in winter, June and July in the summer and October in the Fall. Thus, we have at least one month representing each season (except for spring). The summer season has the highest data and number of observations available since it has two months of data.

4.1.3 Day of the Week

Another time-varying feature is the day of the week, as it plays a key role in travel time modeling. Traffic demand and, consequently travel time varies depending on the day of the week. By including this feature, we are able to capture traffic patterns specific to each of the days of the week. This feature is critical to traffic estimation/prediction modeling. In this study, a variable for weekdays is considered in the modeling, and holidays are excluded from the dataset.

4.1.4 Time of Day

Another feature that is key to travel time modeling is the time of day. The traffic demand – and even travel behavior and signal phasing – could be different depending on the time of day. Time of day could be categorized as morning/AM peak,

afternoon/PM peak, nighttime post-afternoon peak and before morning peak, and daytime between morning peak and afternoon peak.

In this study, a variable, showing time of day, is added to the model in order to capture all of the aforementioned values. In general, time variables could capture many of the variations in travel time patterns.

4.1.5 The External Variables

As mentioned before, travel behavior and demand –and, relatedly, travel time distributions– vary based on the days of week and holidays, as well as different seasons, different time of days, and different weather conditions. As such, these variables are incorporated into the model. By including these variables in the proposed model, we are able to provide the links' travel time distributions in various combinations of these conditions. What follows are the variables and their notations.

1. Variables for the summer (June–July), winter (February), and fall (October) seasons.
2. A variable for weather conditions.
3. A variable for each weekday from Monday to Friday (holiday excluded).
4. A variable for each time of day: morning peak, evening peak, between morning and evening peak, and off-peak nighttime.

As is discussed in the following section, all these variables are the parents of the link travel times in the dynamic Bayesian network. This means that, with each set of factors, we would have a different distribution for all of the links of the network.

4.2 Graphical Model with External Variables

The discussed graphical model depicted in section 3.4.3 can be generalized not only to capture the relationships between link states and observations, but also to take additional information into account.

There are additional external variables that can affect the travel time distributions of the links. As an example, we can expect to have different conditional distributions for a single like under different weather conditions (e.g., sunny or rainy). One of the advantages of modeling the travel time distributions using a graphical model is the flexibility to represent such conditional dependencies.

We introduce additional global variables to explicitly represent such information in the proposed model, where it is available. These external variables allow the model to determine different travel time distributions based on the provided additional distributions (e.g., estimating different Gaussian distributions for rainy and sunny weather conditions).

Figure 12 represents the final deployed model. In this figure, $X_{weather,t}$ represents the random variable introduced for capturing the weather information at time step t. Considering the available data sources, in this dissertation, we consider two states for the weather; that is $X_{weather,t}$is a binary random variable, which shows whether in time t the condition was rainy or not. We assume that the random variable comes from a

multinomial distribution with two states. In this study, we assume $X_{weather,t}$ is observed for all time steps.

The travel time distribution for different links and time steps also varies based on the day and season of the year. To model this distribution drift, we introduce two new random variables X_{season} and X_{day}, which denote the season of the year and the day of the week. X_{season} is modeled by a multinomial distribution with four discrete states to represent each season of the year. X_{day} is modeled as another multinomial random variable and each discrete variable represents a day of the week. These variables are observed both during training and inference. In this way, the travel time distribution for each link not only depends on the time step, but it also depends on the weather condition, the day of the week, and the season of the year

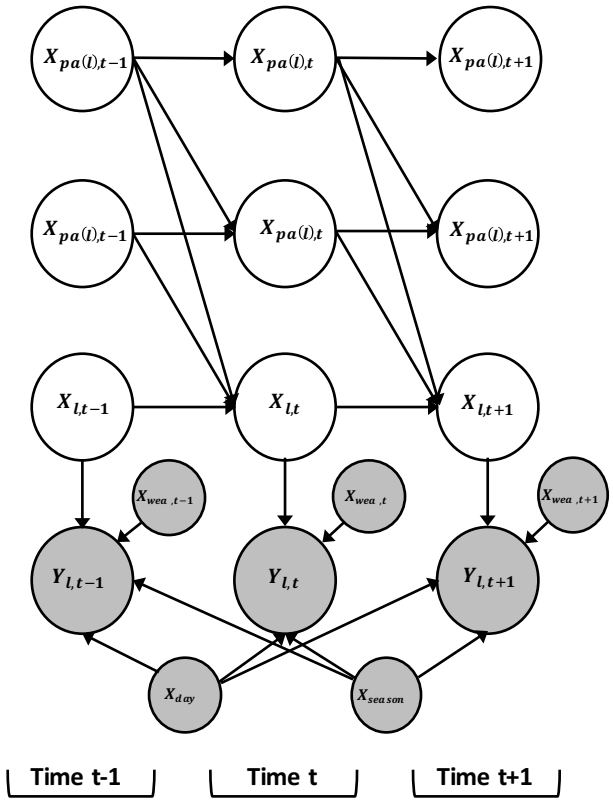

Figure 12 The graphical model with the proposed external variables

4.3 Travel Time Allocation

Once the traversed links between two consecutive GPS points are estimated from map matching, the next step is to allocate the total path travel times between the links.

This can be achieved by maximizing the log-likelihood of the link travel times for each observation using equation 4. This optimization could be calculated given the model parameters. This means that the probability of link l being in state s, and the parameters for travel time distribution of link l for state s should be known; however, we do not have this data at this stage. As such, this represents a chicken-or-egg dilemma. The optimization has two constraints, shown in equations 5 and 6. First, the sum of travel

times on the links should be equal to the path travel time (travel time between consecutive points). Second, the travel time on the links has an upper bound computed for each link by using the maximum speed that is realistically possible for the link; this maximum speed is 60 mph over the speed limit.

$$Argmax_y \left(\sum_{l\in path} ln \sum_{s=0}^{s} X_{l,s} g_{l,s}(y_l)\right) \quad (4)$$

$$S.t. \quad \sum_{l\in path} y_l = T_{path} \quad (5)$$

$$S.t. \quad y_l > y_{l\,max} \quad (6)$$

The travel time is allocated based on the speed limit and length of the links, and then, parameters of the model are estimated for the first stream. For the second stream, the estimated parameters are used to solve the optimization. The optimization and travel time allocation are updated with each stream and the iterations between travel time allocation and parameters estimation continue, as long as we have a new stream of data. The number of variables in the optimization problem in equation 4 is equal to the number of links for the path between two consecutive GPS measurements; this number is always a relatively small number, and it makes the optimization problem easy to solve using numerical methods.

4.4 Parameter Estimation

4.4.1 The Inference

To update the parameters of the model related to each of the defined hidden states, one faces a chicken or egg dilemma. To update the parameters related to each of the states, the samples – which are more likely from that state – should be determined. However, in order to determine these likely samples, the distribution parameters of the model should be known. The most frequently used approach to tackle this problem in machine-learning is the Expectation-Maximization (EM) approach.

EM consists of two stages. In the first stage, known as "expectation," the most likely state for each sample is determined based on the current estimated parameters of the model. The second stage, dubbed "maximization,"" chooses the best parameters of the model given the data and the estimated hidden state from the expectation stage. These two stages are repeatedly applied until a convergence criterion is met. Although theoretically, the EM algorithm is not guaranteed to converge to the global optima of the problem, empirically, the algorithm leads to satisfactory results and is one of the most widely used algorithms in the machine-learning community.

EM provides a practical solution for the aforementioned chicken-or-egg problem. However, it is not a truly Bayesian approach, and it follows a frequentist view in the expectation stage. In other words, during the expectation stage, it only computes a point estimate for the hidden variables (i.e. the most probable value). Another way of solving the problem is to stick with the Bayesian approach, even during the expectation stage,

to consider the full posterior distribution for describing the current estimate for hidden states.

In the proposed approach, we consider a generalization of the EM algorithm. For each batch, given the seen variables, we use the Variational Bayes approximation inference to estimate the posterior distribution of all variables (i.e. finding a set of parameters for the posterior distribution which describes the data best). For efficient inference, all probability distributions governing the variables are assumed to be a part of the exponential family. In other words, all the distributions in our model can be formulated as:

$$f_X(x|\theta) = h(x)\exp\left(\gamma(\theta)S(x) - \lambda(\theta)\right) \quad (7)$$

Where h(.) is a function of x, and γ(.) and λ(.) are only functions of θ. S (.) is a function of x, known as sufficient statistic, which contains all sufficient information for determining the posterior distribution. By these assumptions and by considering conjugate priors, one can calculate the posterior distribution in closed form which makes the inference very efficient.

After estimating the posterior of the variables in closed form, we update the parameters of the model to match the posterior using the Streaming Variational Bayes and iterate these steps through batches (sets of trips) till we visit all batch samples.

As described earlier, the SVB is used for making the inference, and the training model involves iteration between the travel time allocation for a new stream and updating the model and the parameters. The proposed algorithm iterates between

finding the most likely state of the network given the model parameters and then uses those state estimates to update the most likely model parameters.

4.4.2 Streaming Variational Bayes (SVB)

Bayesian inference is time-consuming, especially when it is used for complex models. To this end, approximate inference algorithms are usually deployed for real-world applications. There are different approaches toward the approximate training of Bayesian Networks, including Monte Carlo Markov Chain (MCMC), and Variational Inference techniques. MCMC has been used for decades as the primary technique for posterior approximation. MCMC forms a Markov Chain on the posterior variable and samples from the created chain before finally approximating the posterior with empirical estimates computed from the generated samples. However, MCMC becomes less usable when the models become more complex or the scale of the data at hand becomes large. Variational Inference methods on the other hand use optimization to find the approximate posterior. In these approaches, a family of distributions is considered to approximate the posterior. Then, an optimization problem is solved to find a specific distribution from the family which has the minimum KL-divergence with the true posterior. Since the approximate is chosen from a pre-defined family of distributions, this family should be defined to be as flexible as necessary.

The travel time estimation for arterial networks not only leads to complex inference models, but it also adds another degree of difficulty to the problem. GPS data from users continuously adds live traffic information. Thus, an effective method should not only be able to approximate the posterior distribution effectively, but it should also

change the solution adaptively with minimal computational cost. This incremental approach to the problem requires an online learning paradigm for updating the parameters of the posterior distribution.

To address these challenges, we used inference module on a family of approximate inference methods called Streaming Variational Bayes. Streaming Variational Bayes deploy the Bayes rule to convert a conventional Variational Bayes method into an incremental approach. This approach is covered with more detail in the remainder of this chapter.

Assume data is added to the system in i.i.d. batches $B_1, B_2, \dots, B_N$ where B_i contains the samples in batch "i". Also, let Θ represent the parameters of the model. Assume that batches $B_1, \dots, B_{k-1}$ have been seen previously and the new batch B_k is added. To have an effective incremental strategy, the parameters estimated from $B_1, \dots, B_{k-1}$ should be updated efficiently given the newly added batch B_k. This can be achieved by directly applying the Bayesian rule. According to the Bayes rule, one can derive the following recursive formula:

$$p(\Theta|B_1, \dots, B_k) \propto p(B_k|\Theta)p(\Theta|B_1, \dots, B_{k-1}) \quad (8)$$

That is the posterior distribution of the parameters Θ given batches $B_1 \dots B_k$ is proportional to the likelihood of the batch B_k alone and the posterior of the parameters given all batches that came prior to batch B_k.

This formula is the base idea behind the approximate Streaming Variational Bayes technique. Let $q(\Theta) = VB(B, p(\Theta))$ be the naive approximate Variational Bayes

inference method which computes the posterior over the parameters of the model Θ, given a mini-batch B and a prior over the parameters of the model $p(\Theta)$. Now, let $q_b(\Theta)$ represent the posterior estimates after seeing mini-batches $B_1 \dots B_b$ *i.e.* $q_b(\Theta) = p(\Theta|B_1 \dots B_b)$. From equation(8), we can derive a recursive update for computing $q_b(\Theta)$ based on our estimate from batches $B_1 \dots B_{b-1}$ *i.e.* $q_{b-1}(\Theta)$. More precisely, the streaming update can be written as follows:

$$q_b(\Theta) = VB(B_b, q_{b-1}(\Theta)) \tag{9}$$

This recursive nature makes efficient approximate Streaming Variational Bayes method applicable to the travel time estimation problem. Moreover, it is possible to compute the actual posterior distribution in a parallel fashion. This is achieved by reconsidering the Bayes formula. Assuming mini-batches are *i.i.d.*, we have:

$$p(\Theta|B_1, \dots, B_N) \propto p(B_1, \dots, B_N|\Theta)p(\Theta) = \prod_i p(B_i|\Theta)p(\Theta)$$

$$= [\prod_i \frac{p(\Theta|B_i)}{p(\Theta)}] \times p(\Theta) \tag{10}$$

That is, the problem of computing the posterior given all batches is converted to computing the posterior given each of the batches independently and then combining the resulting information with Equation (10). Each independent batch update can be computed in parallel (Broderick et al., 2013).

Considering Equation (10), a similar formula can be derived when an approximate inference method (*e.g.* Variational Bayes) is used. This approximate update is as follows:

$$P(\Theta|B_1 \dots B_b) \approx q_b(\Theta) \propto [\prod_i \frac{VB(B_i,p(\Theta))}{p(\Theta)}] \times p(\Theta) \quad (11)$$

That is the final approximate posterior and can be computed by distributing and computing the approximate posterior for each batch independently and then combining the results. This leads to an efficient incremental update which is essential for the problem at hand.

4.5 Summary

This chapter proposed a graphical model in which external variables such as weather conditions, season, the day of the week, and time of day are considered in the modeling. Each of these variables is shown to be influential in travel time in general.

In this study, these variables are included in the graphical model as an observed variable and as the parent of all the links. This means we would have different travel time distributions for each combination of the mentioned variables. The idea is that, by incorporating more of the influential variables, we would yield a better prediction both in general as well as in those scenarios. The next chapter demonstrates the model performance in most of these scenarios.

The process of inferring the links ravel-time distribution consists of two steps, travel time allocation, and parameter inference. For travel time allocation, an optimization model is used to allocate travel time based on the previous model parameters; this refers

to the probability of the link to be in each congestion state and the distribution parameters associated with that state.

For the inference of the parameters based on the allocated travel time, this study proposes to use the Streaming Variational Bayes to do approximate Bayesian learning. This offers an advantageous possibility of updating the model with streaming data, and it is less computationally intensive. The next chapter demonstrates the validation of the proposed model and further analysis of the considered variables.

Chapter 5: Analysis

In Chapter 4 we explained the proposed graphical model with the external variables. Any estimation/prediction model should be validated and tested to see how well the model is performing. Thus, we apply and test the proposed model on the INRIX probe data for a case study, and demonstrate and discuss the results in this chapter.

This chapter demonstrates the performance of the graphical model with no external variables and compares the result with the weighted average model. Then, the performance of the proposed model with external variables is illustrated for each day of the week, and for each day of the week in each season, and finally in bad weather conditions. For each of these scenarios, a representative test day is chosen and the trained model is tested and the performance of the model is illustrated in each scenario and compared with other models.

Finally, the chapter ends with two proposed modification of the model which were not possible to do in this study, due to lack of enough data. However, the detailed structure of the variables and how they should be included in the model is demonstrated and discussed.

5.1 Case Study

As mentioned earlier, the case study is a subnetwork of Baltimore city. The network consists of 1,700 links and 1,250 intersections. The test data are for PM weekday peak hours, 4 PM to 6 PM. The considered discrete time in this study is five minutes. Two

models (proposed model and base model) are developed for PM peak each day of the week, and the results are shown as follows.

Figure 13 demonstrates the available vehicle GPS trajectory data in the Baltimore county for a Monday PM peak. As is evident, both the important freeways and downtown Baltimore have the highest number of observations. For the purpose of this study and its focus on arterial travel time modeling, downtown Baltimore was chosen as the study area.

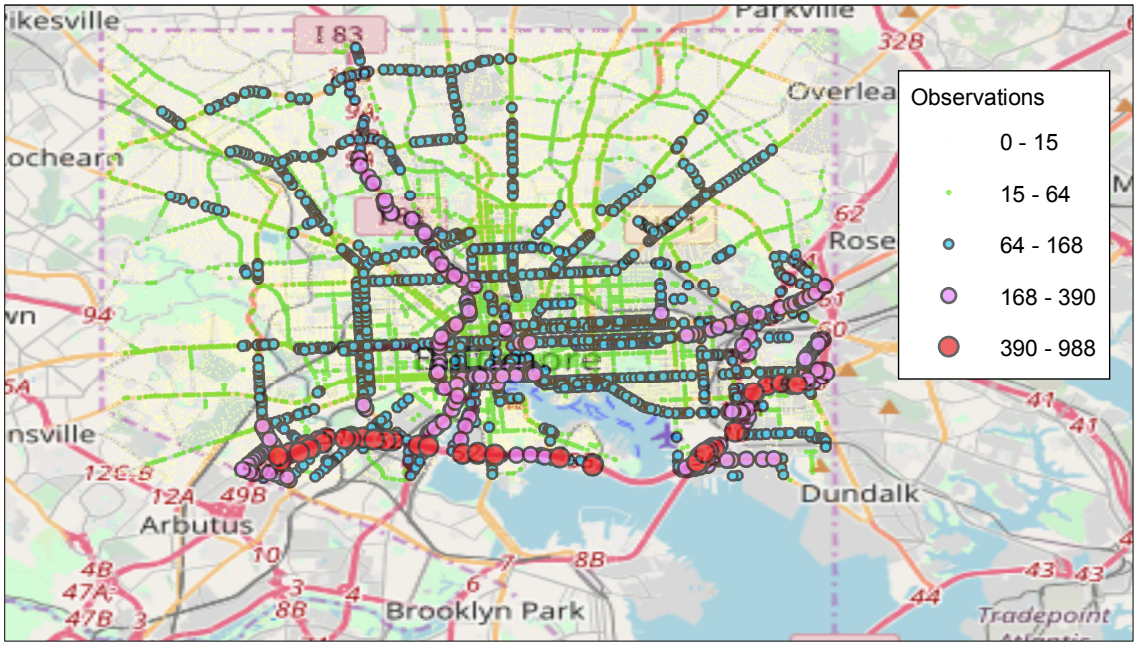

Figure 13 The available data on the area

Figure 14 demonstrates the number of observations for the Baltimore downtown, and again for a Monday PM peak. Some of the roads have quite a considerable number of observations. However, there are some roads for which there is not any observation at all.

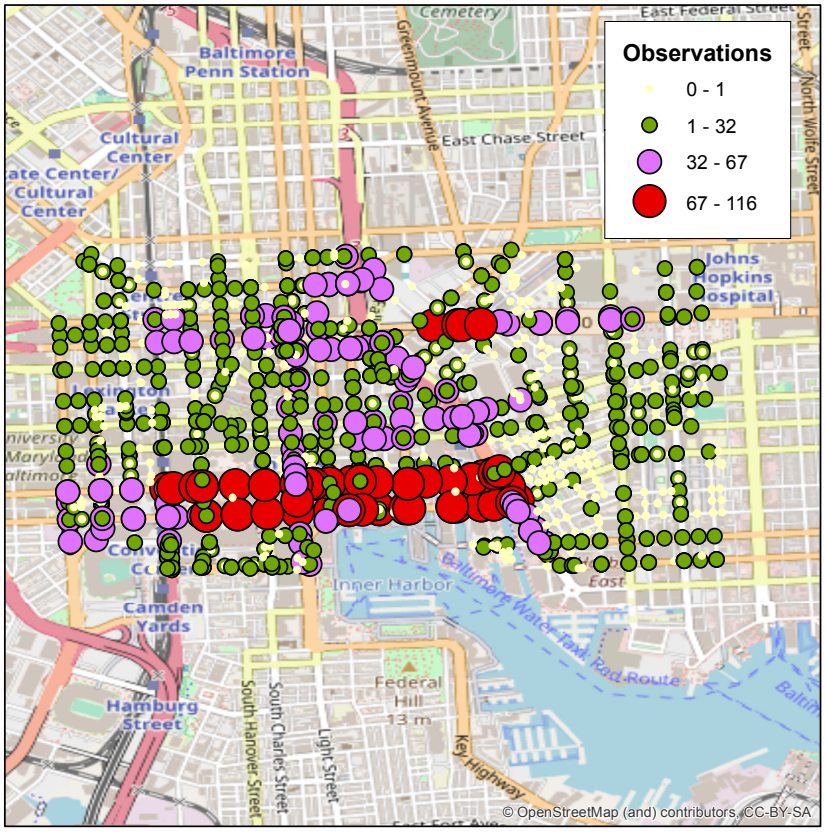

Figure 14 The available data for the case study area

5.1.1 Trip Summaries

The total number of vehicle trips used for the training purpose is 63,419 on the studied road network. Table 1 demonstrates the volume of trips used for the training purpose by days of the week and by available month. As expected, the number of trips for the weekend is much lower than the number of trips for the weekdays since the number of trips is less on weekends. The demonstrated data is for PM peak hour (4-6 PM). Table 2 demonstrates the vehicle GPS points for the training dataset. In total, more than 1 million GPS points are used to train the proposed model.

Table 1 Summary of trips used for each month and each day of the week in the training data set

Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Total
June	2,782	3,254	2,507	2,529	2,705	861	802	15,440
July	2,114	2,328	2,765	2,824	2,544	857	545	13,977
Feb	2,502	2,491	2,831	2,797	3,059	1,425	970	16,075
Oct	2,396	2,574	2,673	3,529	3,486	1,980	1,289	17,927
Total	9,794	10,647	10,776	11,679	11,794	5,123	3,606	63,419

Table 2 Summary of GPS points used for each month and each day of the week in the training data set

Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Total
June	13,510	15,102	13,434	12,537	13,978	4,246	4,047	76,854
July	9,439	11,157	13,958	13,579	12,186	4,147	2,927	67,393
Feb	52,435	52,905	80,730	66,830	89,958	70,740	53,534	467,132
Oct	67,916	77,088	76,907	111,626	113,058	106,314	70,217	623,126
Total	143,300	156,252	185,029	204,572	229,180	185,447	130,725	1,234,505

5.2 Developed Models

5.2.1 The Initial Developed Graphical Model

For the graphical model, we split the data into two data sets. One is used as a real-time data set and the other is used as the test data set to determine the performance measures. The model uses the average of observed travel time on each link as the current observation for the corresponding variable and predicts the travel time for the next time

step using the graphical model shown in the previous chapter. The test data set is used to measure the performance of the model and percentage of error for this model.

To estimate the travel time at each discrete time step, our model performs an approximate probabilistic inference on hidden traffic steps shown in equation 12.

$$s_{l,t} = argmax_L P(x_{l,t} = S^* | D_{0\ldots t}, M) \quad (12)$$

Where, $s_{l,t}$ is the state of link l at time t to be estimated, $x_{l,t}$ is hidden state variable and S^* is the state that makes the probability the highest. $D_{0\ldots t}, M$, is the learned model and real-time data.

Then the Gaussian distribution of that state is used for the estimation is shown below.

$$G(\mu_l^{s_{l,t}}, \sigma_l^{s_{l,t}}) \quad (13)$$

In the next sections, this model is improved by all the external variables, the algorithm of training procedure with all the external variables is demonstrated in Algorithm 1.

5.2.2 Baseline Model

For the baseline model, we also split the test data into two sets, the real-time data and the test data. The baseline model uses the real-time average travel for those links that are seen during real-time data. Considering the real-time average or regression as the baseline for comparison is a common practice in the literature (Herring, 2009; Hofleitner et al., 2012 a; Woodard et at., 2017; Hofleitner et al., 2012 b). Here, the baseline model uses the historical prior for those links that are not seen in real-time

data. The test data is used for measuring the performance of the model and the percentage of error.

5.2.3 Test

As mentioned before, the test data set is used for testing both models. The model can also be tested by considering the Bluetooth sensors data as the ground truth travel time as proposed by Haghani et al. (2010), for those links with available Bluetooth data. The purpose of the test is to estimate the performance of both models and calculate the percentage of error for both. To do so, the existing trips GPS point and their timestamps in the test data set are used as the ground truth. Both models are used to predict the travel time for same trips using the explained procedures. The travel time for each time is estimated using equation 14. As mentioned before, this study assumes the independency between the links travel time, so, the path travel time in the proposed model is the sum of traversed links travel time in the path which can be computed using equation 14.

$$\mathrm{TT}_{pathi} = \sum_i \mu_{l_i}^{s_{l_i,t_i}} \qquad \{l_1 \dots l_N\} \in Path_i \tag{14}$$

The average duration of test trips is above 10 minutes and with different average trip lengths for each day of the week. The result of the test is demonstrated in the next section.

5.3 Validation of the Proposed Model

5.3.1 The Model without any Variables

We implemented the model in the case study and compared the results to the base model. At first, we used all the data, without considering the days of the week and tested the model for the test data which is also from all days of the week. The performance of our model is the same as the base model, and it's very low. The reason could be attributed to the fact that when the days of the week are not considered in modeling, the variation is so high that maybe just using the average real-time data could be the best approach for the estimation/prediction. The average trip duration is 1068 seconds (18 minutes). Table 3 demonstrates the baseline error and proposed model error in seconds, as well as the percentage of errors for both models.

Table 3 Accuracy of predictions for all the weekdays

All weekdays	
Average test trip	1067.0
Proposed model error (s)	399.0
Baseline (s)	397.2
Proposed model error %	0.37
Baseline error %	0.37

5.3.2 The Model with Day of the Week

We also provided a model for each of the weekdays. The average test trip duration varies on different days, with an average of 1040 seconds (17 minutes). Table 4 demonstrates the average trip, the average error in seconds for both the baseline model and the model proposed in this study, and the percentage of error for both baseline and the model proposed in this study by days of the week. The best performance is the Monday model. Thursday and Friday have the highest percentage of errors.

Table 4 Accuracy of predictions based on days of the week for the Baltimore test data

Day of the week	Monday	Tuesday	Wednesday	Thursday	Friday
Average test trip	974.4	978.0	1347.0	1060.5	849.7
Proposed Model error (s)	210.8	248.4	330.3	279.2	221.0
Baseline error (s)	351.9	490.6	559.1	515.9	446.9
Proposed Model error %	21.6	25.4	24.5	26.3	26.0
Baseline %	36.1	50.2	41.5	48.7	52.6

Figure 15 shows the comparison between the base model and the model proposed in this study. As it is shown, there is a huge improvement in the percentage of errors when using the model proposed in this study. It should be noted that both baseline model and our model approximately have the same trend. This means that for those days on which the model proposed in this study has a lower performance and a higher percentage of

error, the baseline model also has a lower performance. This could be due to demand and variation in demand for that day of the week.

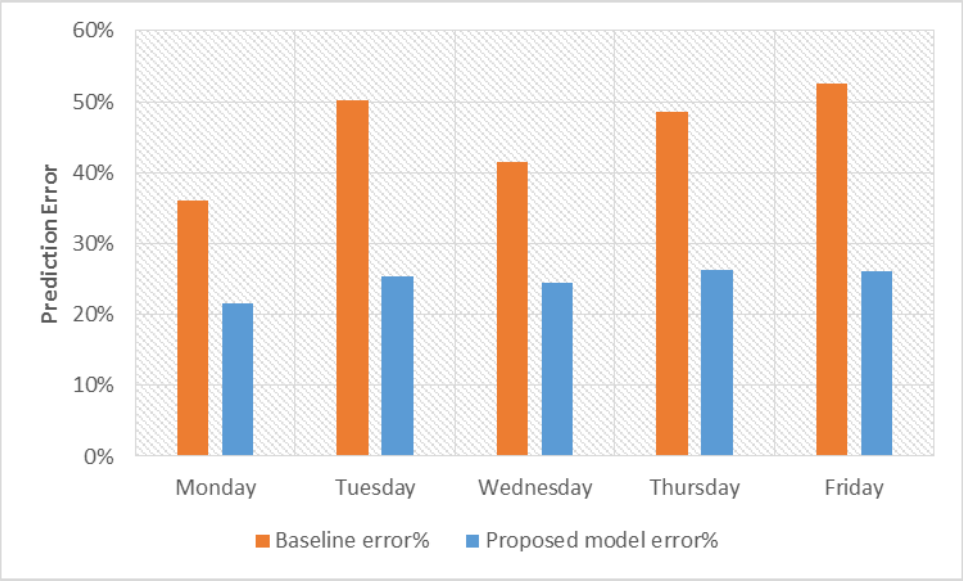

Figure 15 Comparison of prediction error between proposed model and the baseline by days of the week

Even though the model is developed in a way that it can handle all times of the day, since peak hour travel time prediction and validation is the most challenging one, here just the peak hours results are demonstrated.

5.3.3 Travel Time Distribution

To explain the proposed model, one example of link travel time is demonstrated as follows. Figure 16 shows the location of the example link for which a histogram is created from raw travel time data. The example link is, E Pratt St. in downtown

Baltimore and the length is 223 feet. Figure 17 shows the travel time frequency for the example link (link 13), for the Tuesday training data.

The histogram of the travel time on this link shows two main peaks, one around 19 seconds and one around 43 seconds. One could identify two distributions with two means. The proposed model for the same link and same days also yields two distributions, one for uncongested condition (s = 0) with the mean of 21.47 seconds, and one for congestion condition (s=1) with the mean of 45.32, which is very close to what is observed.

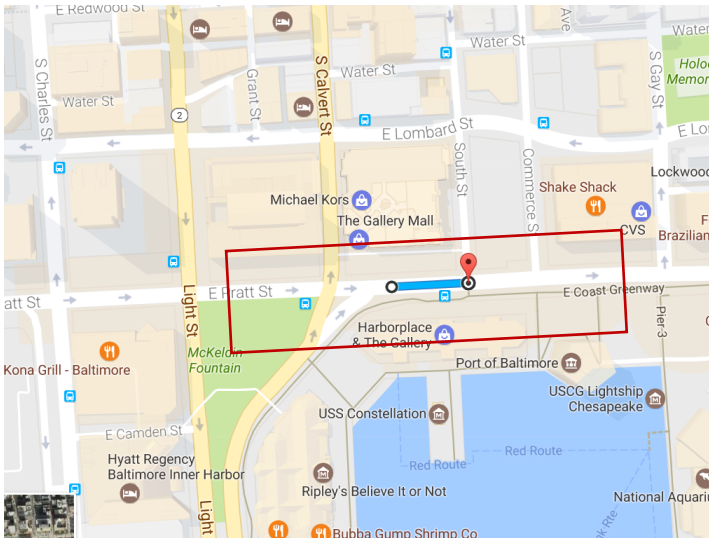

Figure 16 The example link in downtown Baltimore

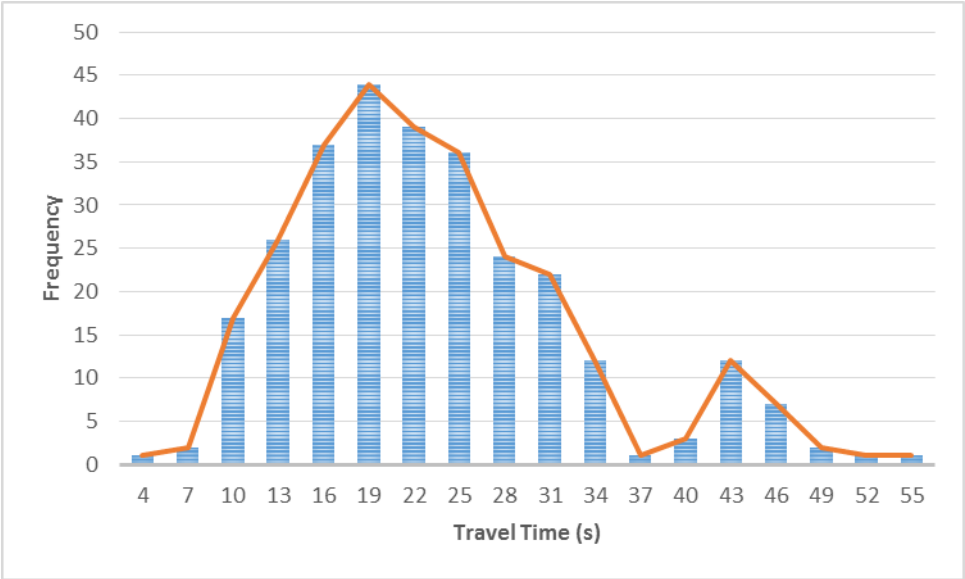

Figure 17 The travel time histogram for link 13

5.4 Model Improvements

5.4.1 The Model with All of the External Variables

In this section, we tested the model for each day of the week and each season. The developed model has the potential of being tested on any combinations of all of the scenarios (time of day, day of week, season, and weather); however, here the results of the analysis for one time of day of (PM peak) and three seasons and five weekdays (From Monday-Friday) is demonstrated. This means that we selected 15 (five weekdays* and three seasons) test days to evaluate the proposed model vs. based model improved by the regression model. Algorithm 1 illustrates the step-by-step procedure for the training of the proposed model. In this algorithm, $\mathbb{L}$ represents the set of all links in the network, $\mathbb{T}$ represents the set of time-steps, and $\mathbb{M}_i$ denotes the set of reported GPS information for the i'th training trip. $X_{ext}^{l,t}$ represents the observed external

variables for time step t and link l in the network. $X_{GPS}^{i,j}$ is the j'th reported GPS information for the i'th training trip. We denote the traffic state of link l in time step t by random variable $S^{l,t}$. $\mathcal{N}(l)$ represents the set of neighboring links connected to link l. μ_s^l, σ_s^l repsents the mean and standard deviation of the normal distribution over link l conditioned on traffic condition s.

An example of predicted traffic states and travel time associated with those states is also demonstrated as follows. Figure 18 demonstrates the predicted level of congestion for downtown Baltimore on October 26th on Monday at 4:50 PM. The green color demonstrates the level of congestion "one" and the red color demonstrates the level of congestion "two". It should be noted that the level of congestion "one" is not necessarily uncongested and level of congestion "two" is not necessarily congested. They could be congested and more congested since both belong to the same time bin of PM peak and they are just comparable to each other in that time bin, meaning we can say red is more congested than green.

Algorithm 1. *The training procedure.*

Inputs:

- $X_{ext}^{l,t}$ $\forall l \in \mathbb{L}, t \in \mathbb{T}$ observed external variables for each time step t and each link l.
- $X_{GPS}^{i,j}$ $\forall i \in \mathbb{N}, j \in \mathbb{M}_i$ the time of $j'th$ reported GPS data for trip i in the training.

Outputs:

- $P\left(S^{l,t} = s \middle| S^{\mathcal{N}(l),t-1}\right)$ $\forall l \in \mathbb{L}$, $s \in \mathbb{T}$ the probability that link l has the discrete state s in time step t given the state of its neighbors in time step $t-1$.
- μ_s^l, σ_s^l $\forall l \in \mathbb{L}, s \in \mathbb{S}$ the mean and standard deviations of the Gaussian distribution for link l for each discrete state s.

1: **while** not converged **do**

- CREATE A BATCH BY SAMPLING THE TRIPS

2: $\mathbb{B} \leftarrow sample(1 \dots \mathbb{N})$

- ALLOCATE TRAVEL TIMES BY SOLVING THE OPTIMIZATION

3: $X_{GPS}^{i,l,t} \leftarrow allocate(\{X_{GPS}^{i,j}\}_{i \in \mathbb{B}, j \in \mathbb{M}_i}, \{P\left(S^{l,t} = s \middle| S^{\mathcal{N}(l),t-1}\right)\}_{l \in \mathbb{L}, t \in \mathbb{T}})$

- EXPECTATION STEP: INFER THE SET OF HIDDEN STATES USING VARIATIONAL MESSAGE PASSING

4: $\mathcal{H}^{l,t} \leftarrow VMP(X_{GPS}^{i,l,t}, X_{ext}^{l,t}, \mu_s^l, \sigma_s^l, P\left(S^{l,t} = s \middle| S^{\mathcal{N}(l),t-1}\right))$

- MAXIMIZATION STEP: PERFORM THE MAXIMIZATION STEP BY APPLYING STREAMING VARIATIONAL BAYES

5: $\mu_s^l, \sigma_s^l, P\left(S^{l,t} = s \middle| S^{\mathcal{N}(l),t-1}\right) \leftarrow SVB(\mathcal{H}^{l,t}, X_{GPS}^{i,l,t}, X_{ext}^{l,t}, \mu_s^l, \sigma_s^l, P\left(S^{l,t} = s \middle| S^{\mathcal{N}(l),t-1}\right))$

6: **end while**

7: **return** $(\mu_s^l, \sigma_s^l, P\left(S^{l,t} = s \middle| S^{\mathcal{N}(l),t-1}\right))$ $\forall l \in \mathbb{L}, s \in \mathbb{S}$

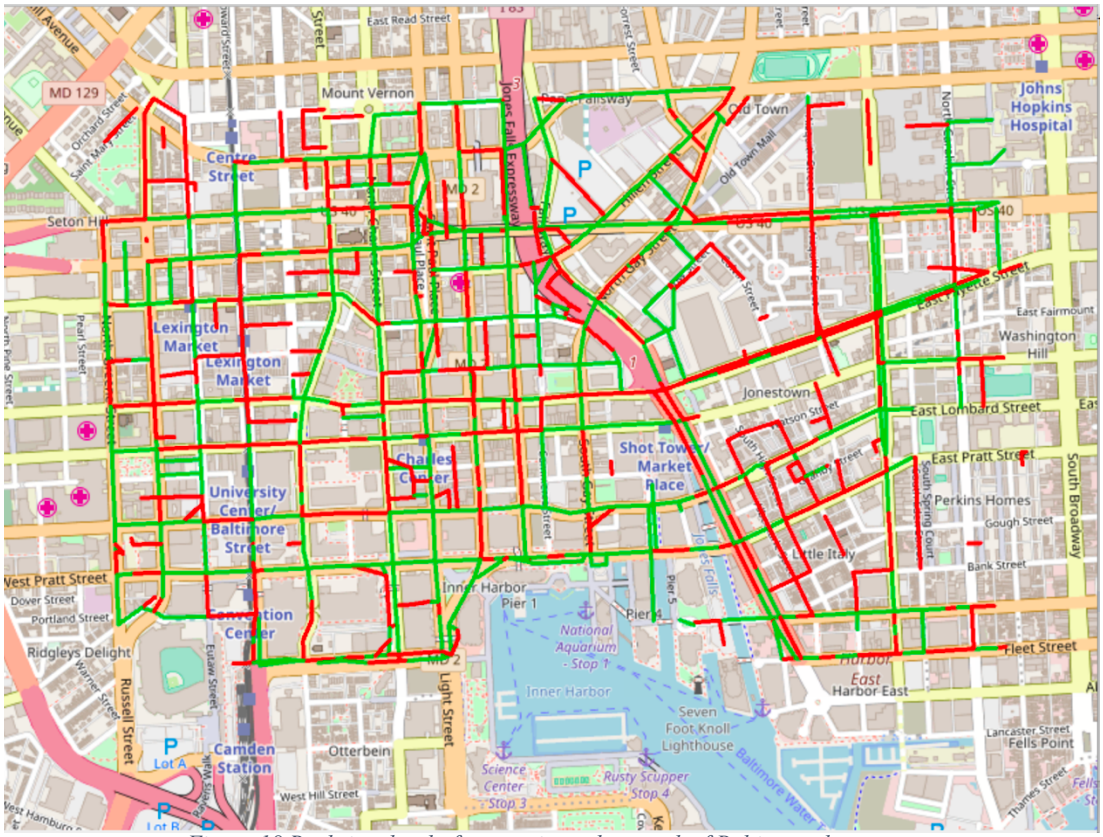

Figure 18 Real-time level of congestion subnetwork of Baltimore downtown.

Figure 19 demonstrates the predicted travel time over travel time with the speed limit. Green is for values equal or above 0.7 meaning the predicted travel time is close to the travel time driving at the speed limit. The black color indicates the relative travel time below 0.2 meaning the predicted travel time is one-fifth of the travel time at the speed limit. The red and orange are the relative travel time in between two mentioned numbers.

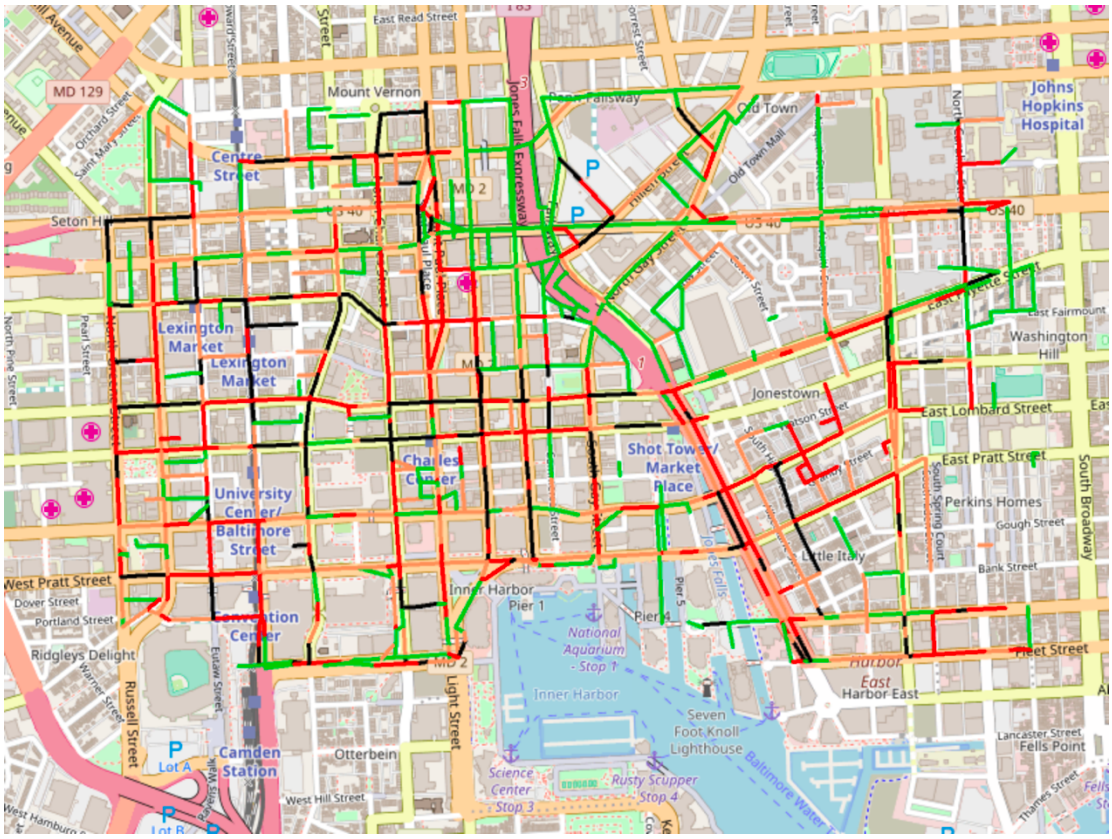

Figure 19 Real-time relative travel time of Baltimore downtown network.

What follows explains how the base model improved by the regression model. Also, an Algorithm is dedicated to demonstrate steps of developing this base model.

5.4.1 Regression Model

In order to improve the base model a regression model is developed and the proposed model is compared to the improved base model. The implemented regression model has two different sets of explanatory variables. Trip variables such as weather conditions, time of day, day of the week, and season, as well as link explanatory variables such as the speed limit, the road type, and the length of the link between the two intersections. The goal is to capture the underlying factors behind spatial and

temporal variations in the travel time. The real-time prediction in each time step would be a combination of the regression prediction and the real-time observations. Equation 15 shows the developed regression model and Algorithm 2 demonstrates the detailed procedure from travel time allocation to updating the travel time prediction for each link using real-time data and the regression model.

$$Y^{l,t} = W_{ex}^{l} X_{ex}^{l} + W_{trip}^{l,t} X_{trip}^{l,t} + b^{l,t} \quad (15)$$

In this algorithm, $\mathbb{L}$ and $\mathbb{T}$ represent the set of all links in the network, and the set of time-steps respectively. X_{ex}^{l} is the set of explanatory variables for link l, $X_{trip}^{l,t}$ is the set of trip variables for link l in time-step t, $X_{obs}^{l,t}$ is the set of real-time observations for link l in time-step t, and $Y^{l,t}$ is the travel time for link l in time-step t. We represent the size of a set by $|.|$.

Algorithm 2. *The regression model algorithm*

Inputs:

- X_{ex}^{l} $\forall l \in \mathbb{L}, t \in \mathbb{T}$ explanatory variables for each link.
- $X_{trip}^{l,t}$ $\forall l \in \mathbb{L}, t \in \mathbb{T}$ trip variables for each link at each link l and time t.
- $X_{obs}^{l,t}$ $\forall l \in \mathbb{L}, t \in \mathbb{T}$ real-time observations for each link at each link l and time t.

Outputs:

- $Y^{l,t}$ $\forall l \in \mathbb{L}, t \in \mathbb{T}$ travel time for each link l and each time step t.

---

for each $l \in \mathbb{L}$ **do**
  for each $t \in \mathbb{T}$ **do**
    Find W_{ex}^{l}, $W_{trip}^{l,t}$, and $b^{l,t}$ by solving the regression given training samples
  end for
end for

for each $l \in \mathbb{L}$ **do**
  for each $t \in \mathbb{T}$ **do**
    $n_{obs} \leftarrow |X_{obs}^{l,t}|$
    if $n_{obs} = 0$ **then**
      $Y^{l,t} \leftarrow W_{ex}^{l} X_{ex}^{l} + W_{trip}^{l,t} X_{trip}^{l,t} + b^{l,t}$
    else:
      $Y^{l,t} \leftarrow \frac{1}{n_{obs}} \sum_{n=1}^{n_{obs}} X_{obs}^{l,t}[n]$
    end if
  end for
end for
return $Y^{l,t}$ $\forall l \in \mathbb{L}, t \in \mathbb{T}$

5.4.2 Comparison Between the Final Model and Improved Base Model

The average test-trip duration varies on different days and in different seasons. Figure 20 through Figure 22 and Table 5 through Table 7 demonstrate the average trip, the average error in seconds for both the improved base model by regression and the model proposed in this study, and the percentage of error for both the improved base model and the model proposed in this study by days of the week for each season.

Figure 20 and Table 5 demonstrate the analysis results for winter season and all of the weekdays. As it is shown, there is a significant improvement in the proposed model with all the variables compared with the previous model across nearly all weekdays. Also, there is a huge improvement in the percentage of errors for the base model (regression and real-time average). Still, the proposed model significantly outperforms the improved base model.

Table 5 Summary of winter predictions based on days of the week for the Baltimore test data

Day of the week	Monday	Tuesday	Wednesday	Thursday	Friday
Average test trip	544	511	546	518	592
Proposed Model error (s)	83	120	99	111	117
Baseline error (s)	138	166	179	152	157
Proposed Model error %	15.3	24.1	18.5	22.2	19.6
Baseline %	25.3	31.8	32.3	28.2	26.9

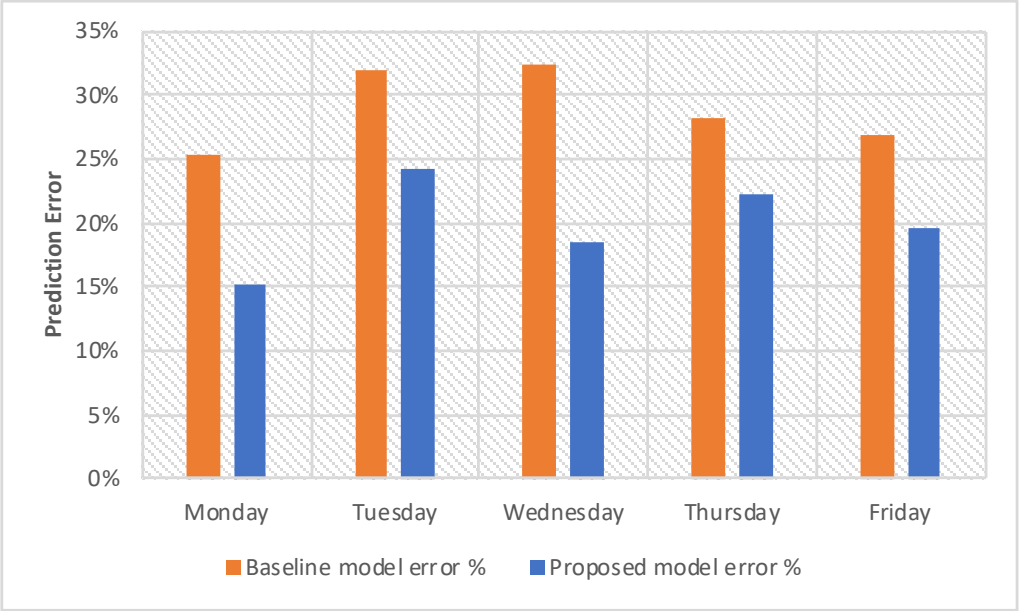

Figure 20 Comparison of prediction error between proposed model and base model in winter

Figure 21 and Table 6 demonstrate the same analysis for the summer and Figure 22 and Table 7 demonstrate it for the fall. As it is shown there is a general improvement both in the base model and the proposed model.

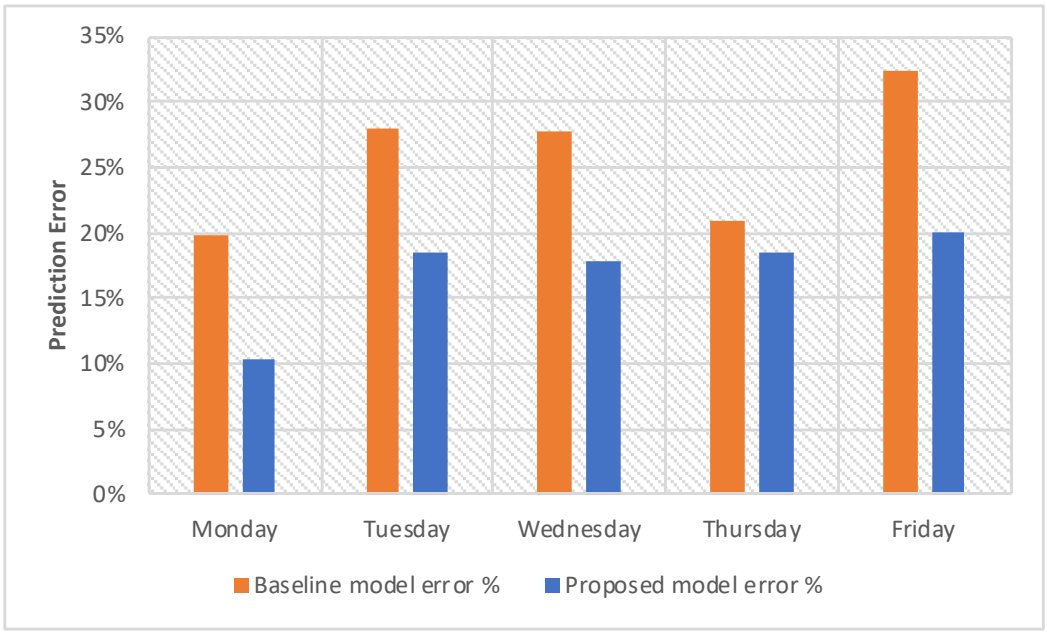

Figure 21 Comparison of prediction error between proposed model and base model in summer

Table 6 Summary of summer predictions based on days of the week for the Baltimore test data

Day of the week	Monday	Tuesday	Wednesday	Thursday	Friday
Average test trip	570	529	553	583	602
Proposed Model error (s)	59	95	95	107	123
Baseline error (s)	112	152	159	122	191
Proposed Model error %	10.4	18.5	17.8	18.4	20.1
Baseline %	19.8	28.0	27.8	20.8	32.2

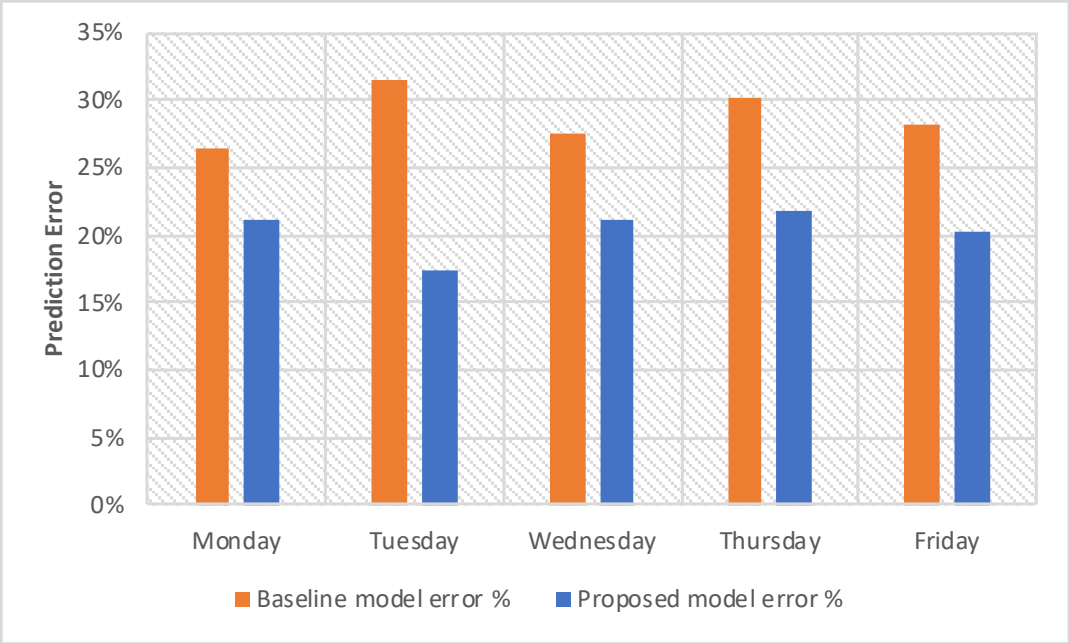

Figure 22 Comparison of prediction error between proposed model and base model in fall

Table 7 Summary of fall predictions based on days of the week for the Baltimore test data

Day of the week	Monday	Tuesday	Wednesday	Thursday	Friday
Average test trip	502	542	503	550	537
Proposed Model error (s)	107	89	104	121	108
Baseline error (s)	132	180	141	163	152
Proposed Model error %	21.2	17.4	21.1	21.8	20.2
Baseline %	26.4	31.4	27.6	30.1	28.1

On average, the model performance in the summer is better, with an average percentage of error of 17% for all days of the week as shown in Table 8 and Figure 23. This could be attributed to the higher number of observation for the summer season. We had two months of data for summer, June and July. This also could be attributed to the fact that summer has less traffic demand and better weather conditions, so traffic patterns are more predictable. The fall and winter average performance for different days of weeks are the same, with an average percentage of error of 20%.

On average, the performance of the proposed model is better on Mondays, with an average percentage of error of 15.5% for all the seasons (Table 9 and Figure 24). This is followed by Friday and Wednesday. The highest percentage of error is for Thursday with the average percentage error of 21 %. This again could be attributed to the lower traffic demand on Mondays and better predictions of the traffic pattern for them.

Table 8 Aggregated average percentage of error for different season

Season	Average Percentage of Error %
Fall	20.3
Summer	17.0
Winter	19.9

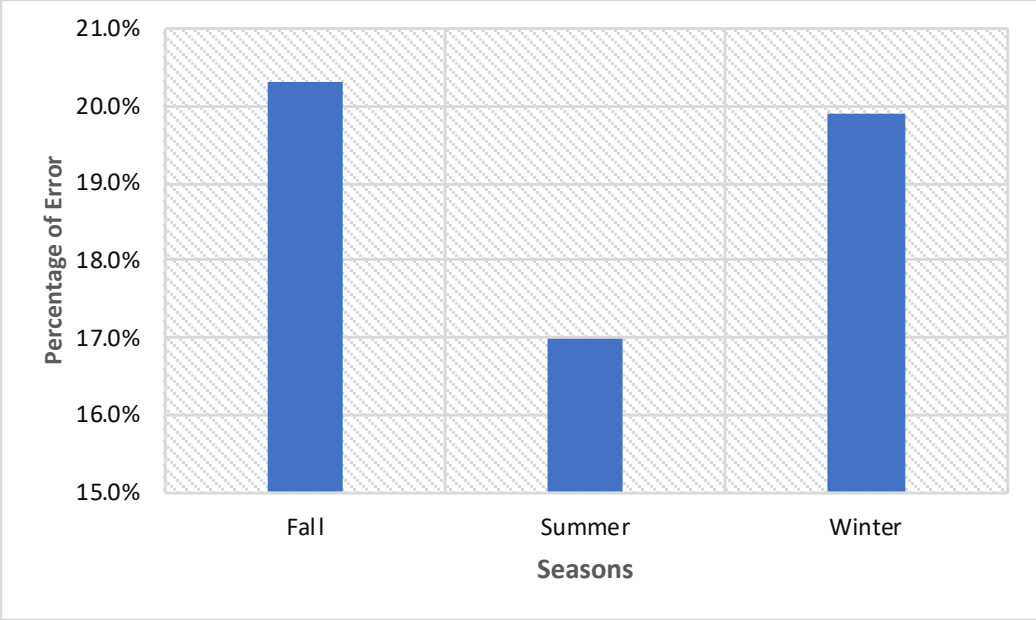

Figure 23 Aggregated average percentage of error for different season

Table 9 Aggregated average percentage of error for each days of week

Season	Average Percentage of Error %
Monday	15.6
Tuesday	20.0
Wednesday	19.1
Thursday	20.8
Friday	20.0

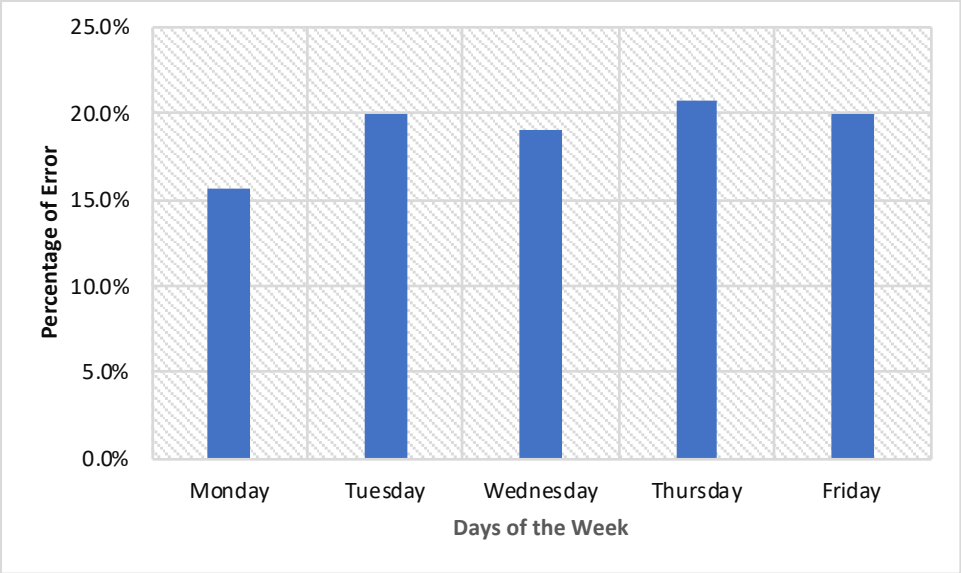

Figure 24 Aggregated average percentage of error for each days of week

5.4.3 Comparison Between the Two Graphical Models

To demonstrate how incorporating other variables into the graphical model has led to better travel time prediction models, the performance of the two graphical models are compared. As discussed before, one of these models is a model including the season and weather conditions (discussed in section 5.4.1), and the other model is the one with only the day of week and time of day variables and not the other two (discussed in section 5.2.1).

The problem is that we have different parentages of error for different days and seasons; to overcome this challenge, the average of the percentage of error for all of the weekdays are estimated, and a comparison of the two models' performance is done for each day of weekdays. As it is shown in Figure 25, there is an evident improvement for

the weekdays. For most of the weekdays, there is more than 5% improvement in the percentage of error.

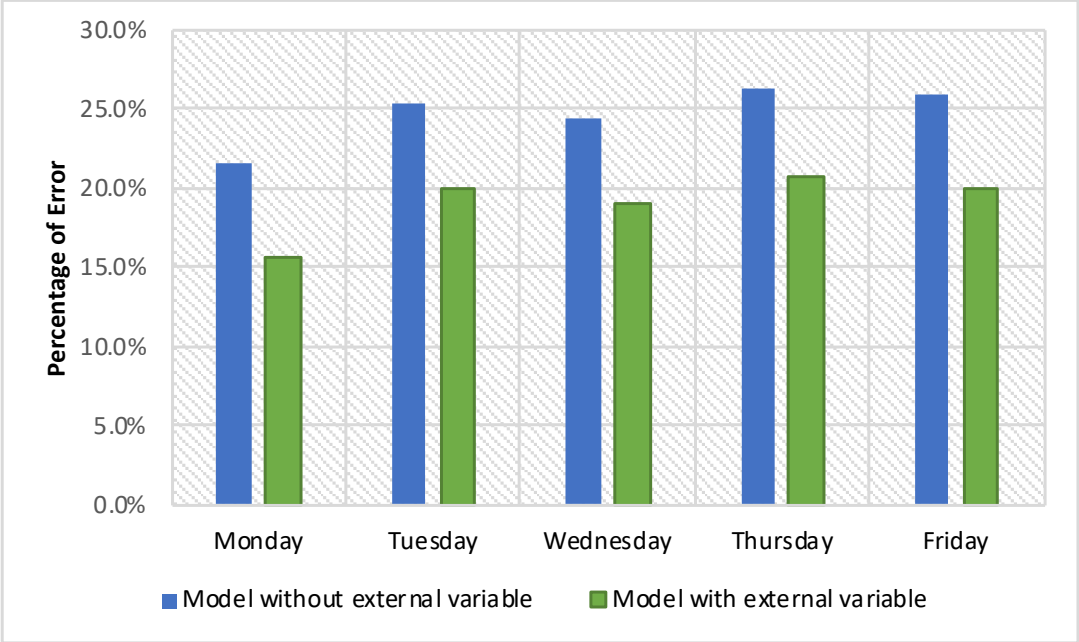

Figure 25 Comparison of the proposed model with seasons and weather conditions and without them

It should be noted here that some of the test datasets might be in the bad weather conditions or good weather conditions. In the next section, in order to specifically evaluate the effects of weather conditions variables on the performance of the model, another model is trained with all the variables, except for the weather conditions, and it is tested on a rainy day.

5.4.4 The Model Test on a Rainy Day

As mentioned before, one of the objectives of this study is to develop an integrated model that could predict travel times under various weather conditions. Thus, in this study, various weather data sources were used, and the variables were included in the

graphical model as a trip (global) variable. The test day is Friday, October 9th which was rainy during the PM peak hour. To evaluate the proposed model in bad weather conditions, a test day in rainy weather conditions was selected and results of the proposed model with weather variable inclusion vs without this variable and also base model improved by the regression model is demonstrated in Table 10 and Figure 26.

The average test trip duration is 491 seconds (more than eight minutes). Table 10 demonstrates the average trip, the average error in seconds for the graphical model without weather conditions variable, and including the weather condition variable – as well as the base model with regression.

Figure 26 shows the comparison between the aforementioned models. As it is shown, there is an improvement in the percentage of errors when using the proposed graphical model with weather conditions variable compared to the graphical model without it as well as compared to the base model improved by a regression model that includes weather conditions variable.

Table 10 The comparison of the model with and without weather variable

Model	Percentage of Error %	Model error (s)	Average test trip
Model with Rain	22.1	110	491
Model without Rain	24.0	117	491
Base model	29.0	141	491

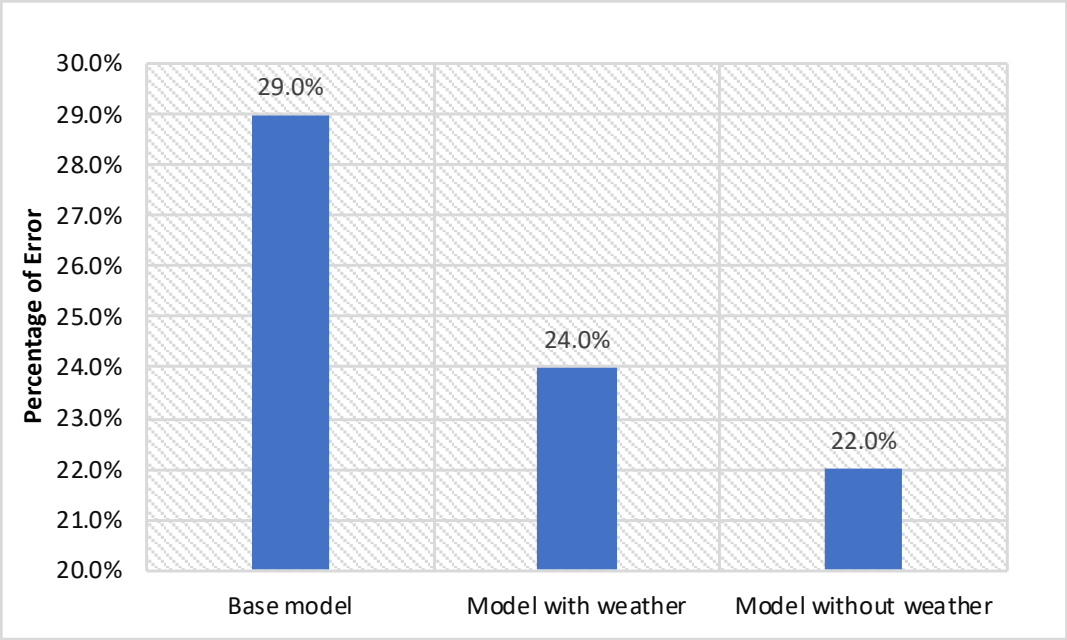

Figure 26 The comparison of the model with and without weather variable

5.4.1 The Time of Day

Even though the analysis is shown for the most challenging time of day, PM peak, the proposed graphical model has the capability of providing travel time estimation/prediction in any time of day because it includes the time of day variable in the model. In this study, the time of day is categorized into bins based on the characteristics of traffic. The bins times are chosen in a way that the traffic pattern is nearly consistent in each bin, and different from others.

The time period bins include the following bins:

- "AM Rush Hour": weekdays 7 – 9 AM
- "PM Rush Hour": weekdays 3 – 6 PM
- "Nighttime": Monday-Sunday 7 PM – 6 AM
- "Weekday Daytime": remaining times during weekdays

- "Weekend"

Table 11 and Figure 27 demonstrate the performance of the proposed model with all the variables for one of the other time bins, weekday daytime bin and winter season, as an example. As it is shown in Figure 27, the percentage of error for the proposed model is less than the base model. However, there is an improvement in the base model performance compare to PM peak base model. This could be attributed to the fact that the average or regression model might be able to capture the pattern to some extent better for the time bins with less traffic congestion since there is less variability in the traffic compared to time bins with more traffic congestion.

Table 11 Summary of winter predictions based on days of the week for daytime

Day of the week	Monday	Tuesday	Wednesday	Thursday	Friday
Average test trip	524	549	537	537	545
Proposed Model error (s)	96	105	99	130	107
Baseline error (s)	142	136	135	165	156
Proposed Model error %	18.7	19.8	18.7	20.3	19.8
Baseline %	26.6	23.9	24.8	26.5	28.5

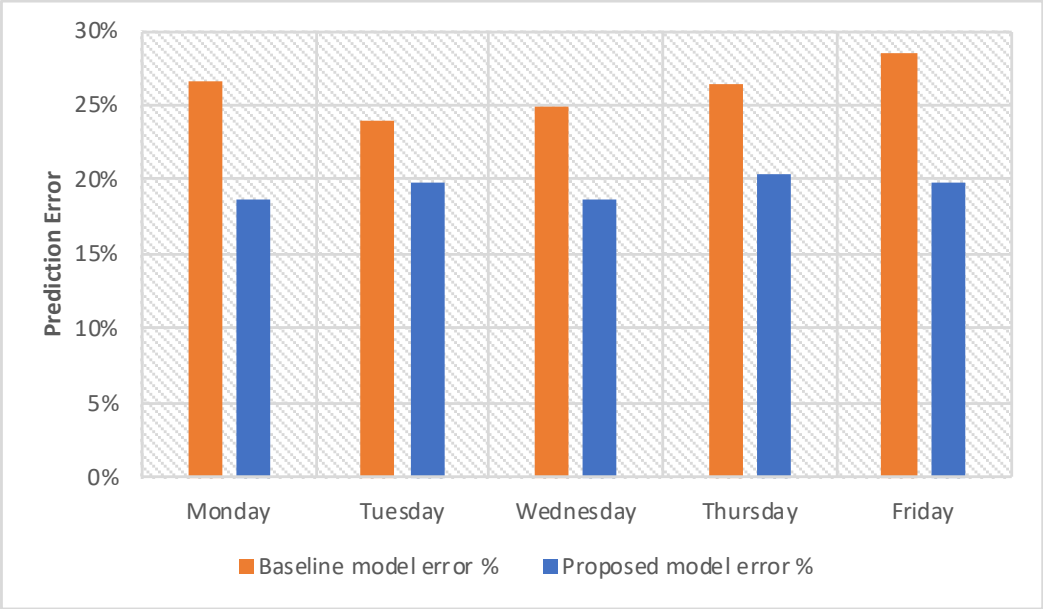

Figure 27 Comparison of prediction error between proposed model and base model for day time in winter

Other Modifications

This section focuses on the possible improvements of the graphical model by adding as much as information as possible. What follows demonstrates the proposed model which includes intersections in the modeling process and the associated graphical mode. The section after that proposes an approach by which driver habits and traffic conditions could be modeled by introducing a new variable and adding it to the model and demonstrating it in the graphical model.

5.4.2 The Graphical Model with Intersection

In most of the arterial travel time studies, travel time (expected value or the distribution) on the links is provided and the trip travel time is calculated as the sum of the link travel times. The alternative, however, is to include travel time on links and delays in the

intersections, and calculate trip travel time by adding the link travel times and delays at intersections.

In general, the turn delays can be considered as deterministic or stochastic variables. The variation in the delay comes from unobserved changes such as in traffic flows, signal phasing, driver behavior, etc. Considering intersection delay as a separate entity is very important when we want to distinguish between right turn, left turn, and forward movements. If not for these three different movements, the delays at intersections could easily be added to the link travel times and there is no need to consider them separately. Including the intersection variable in the modeling requires much more data. Because there were not enough samples for each of the movements, the proposed model could not be tested. We propose this model for future work when and if there is enough data for each movement.

In this modeling, there are three main challenges to overcome. The first would be to determine what variable should be considered in the modeling to represent the reality the best. We propose the movement variable. The movement variable is any movement from a link to another connected link.

It could be noted that the movement variable for an intersection is not just one variable. Any combination of movement from any incoming link to any outgoing links could be considered separately, as shown below. However, one might be interested in just some of the movements such as the left turn and through movements. In Figure 28, there are two incoming links and two outgoing links, which make the total number of movements 4 (2*2). The number would be much higher when there are four incoming and four outgoing links (16 movements). For this reason, having the appropriate number of

observations is key because all of these movements should be backed by enough observations to ensure we have a good model.

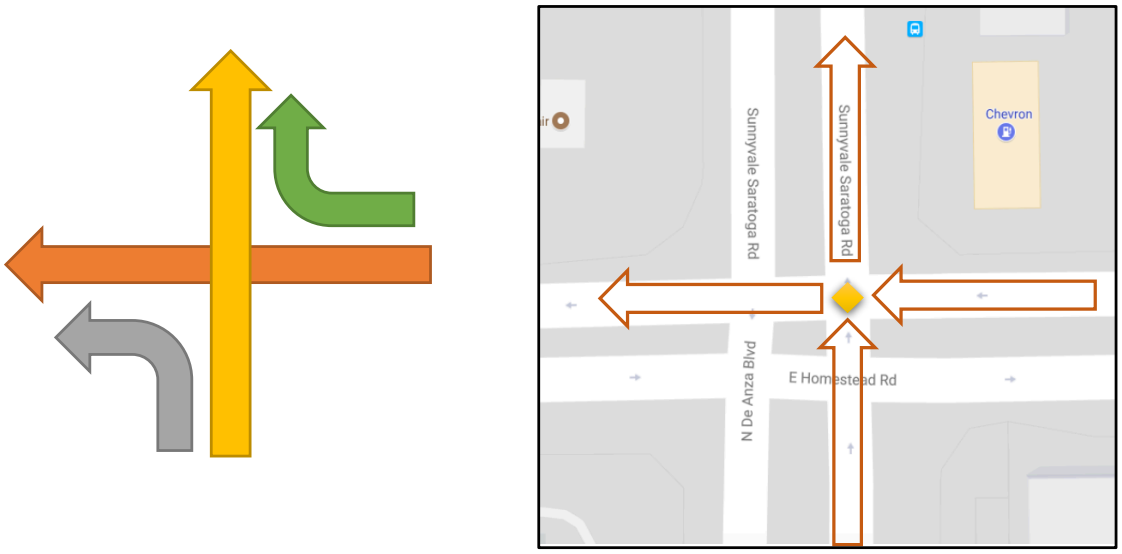

Figure 28 The intersection movement demonstration

The second challenge is to create a new network of the city in which all the intersections (movements) are added and the graphical model is made to include the movement variables. To ensure this, a dummy link variable is proposed instead of each movement, and new parents and children are created based on this new proposed model structure. The proceeding nodes $X_{pa(l,t)}$ (links) are now parents of dummy node $D_{l,t}$(intersection) and $D_{l,t}$ is the parent of $X_{l,t}$ as shown in Figure 29.

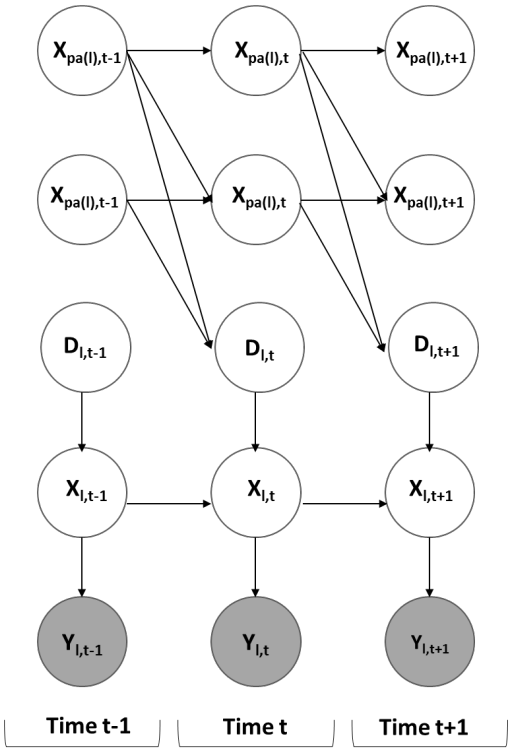

Figure 29 The proposed graphical model including the intersections

The third challenge is how to allocate the observation travel time to the movements. For the link-based travel time, the travel time is allocated only on the links. When considering the intersections, the travel time should be allocated to the combination of links and intersection (movements). For that, the same optimization model explained in section 4.3 can be used. The optimization needs to be solved to decompose the path travel time to links' travel times and intersections' (dummy variables) travel times. The new bound for lower-bound and upper-bound for the intersections' travel times is proposed based on the intersection type.

The final proposed model provides the travel time distribution on the links as well as travel time (delays) distribution for the intersections, and a trip travel time is the sum

of the traversed link travel times and traversed intersection travel times, as demonstrated in Figure 30

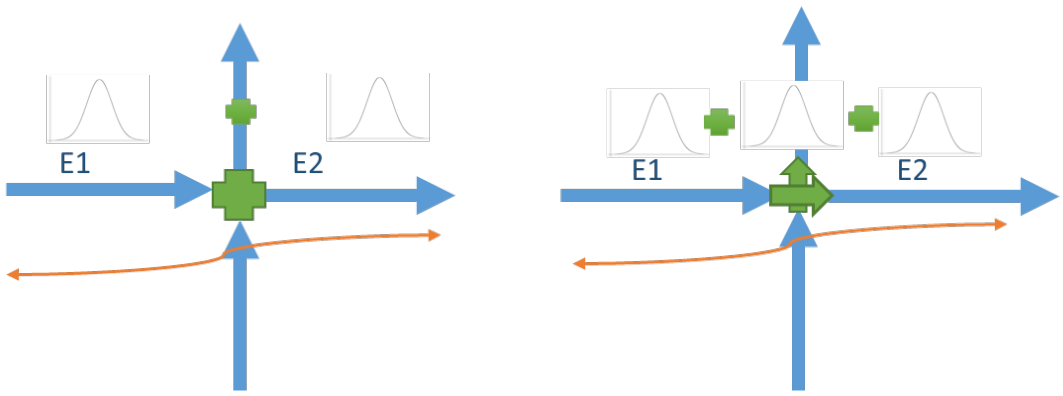

Figure 30 Demonstration of path travel time as the sum of links and intersections travel times

5.4.3 The Graphical Model with Driver Habits Variable

Another variable that is assumed to have an influence on arterial travel time modeling is the driver behavior variable. This study suggests a new variable that can capture driver habits and traffic conditions. This is particularly applicable when the vehicles' speeds are also available. If it they are not available, the speed can still be estimated based on the time and the link lengths, but might not be as accurate. This variable is an observed variable and it provides a way to capture dependency of travel times across links. The new suggested variable is shown in equation 16.

$$V_{up\ to\ now} = \text{Avg. } V_{driver} / \text{Avg. } V_{speed\ limit} \quad (16)$$

Figure 31 demonstrates how to include this variable in the model structure. The new variable could be used for both the initial graphical model as well as the graphical model with intersections. In this study, due to the low number of observations, this variable effect could not be tested. This study suggests these variables for future studies for which a higher number of observations is available. This variable also could potentially be used to provide the individual travel time prediction based on individual driver behavior.

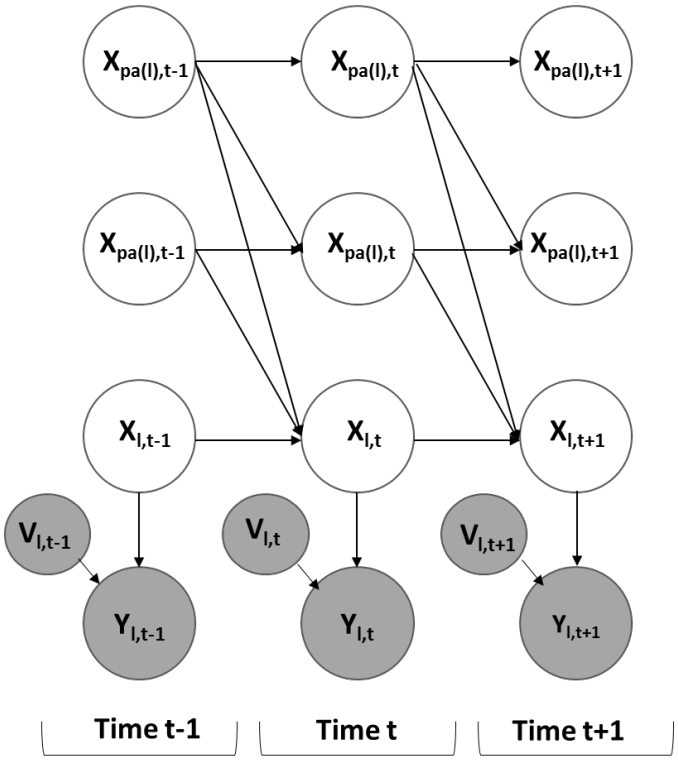

Figure 31 The proposed graphical model with proposed relative speed variable

5.5 Summary

This chapter demonstrated the validation and comparison of the model with a base model. It also demonstrated that the graphical model performance would improve with the addition of the explanatory variables. There is an improvement by adding the days of the week variable, seasons, as well as weather conditions, all of which are presented by testing the model on a suitable test day. Model percentage of error in most scenarios is around or less than 20 %. Testing the model with weather conditions variable on a rainy day showed a lower percentage of errors when compared to the graphical model without this variable.

The model is also compared with a base model with regression and a real-time average. The regression model includes the link explanatory variables such as speed limit, length, etc., and the trip explanatory variables such as seasons, weather conditions, etc. Even though there is a significant improvement in the performance of the base model when switching from real-time average to the combination of real-time average and regression model, the proposed graphical model with all the variables still outperforms the base model.

There were two model modifications that this study aimed to explore but, due to the low number of observations and lack of data, the modifications could not be tested to see their impact on the performance of the model. Given this, we have left the implementation and testing of these modifications for future research. The first important modification is the inclusion of the intersection movements, which would

cause the model to have a different delay time for straight movement vs. left turn movement.

Chapter 6: Comparison and Sensitivity Analysis

Validation of the model via separate test data is a common way of validating any prediction model; however, in reality, one may want to compare a model to existing commercial models. To the best of the author's knowledge, none of the proposed models in existing literature have been compared to any commercial prediction model. As such, there is not a framework to do so. A comprehensive framework for comparing the model proposed in this study (or any model) to commercial travel time providers is proposed. The comparison of the proposed model in this study with the Google API, and the original observations are demonstrated in this chapter.

The chapter continues with the sensitivity analysis on a parameter of the proposed model as well as available data, and the results of the sensitivity analysis are illustrated and discussed.

6.1 Validation

Another way to validate the proposed model is to compare the outcome of this study to Google estimates. There are two challenges with this approach. First, we do not have the current probe data and Google API only provides the travel times for current or future time. This poses a big challenge, and there is no way to avoid this obstacle without access to current time vehicle trajectory data. However, we can develop a general estimate on how the model performs in off-peak hours, when the traffic has a lesser effect on travel time. As such, the travel times for the roads network are compared to each other. This just gives us an estimate of how our model is performing

in general and without the effect of traffic. It should be noted that we may want to compare only those trips' travel times that have common route with the observed GPS trajectory. As such, the second challenge is that the Google-proposed route should be the same as the observed trajectory. To address this, we have proposed to use a shape similarity algorithm to only select those trips that have the same observed routes as the Google routes. Here, the probe vehicle trajectory and Google routes should be compared to each other to determine they are the same routes and comparison is meaningful.

6.2 Shape-Matching

Shape-Matching is an important area of research that has many applications in computer vision, and it is the area of arbitrary target object detection. In computer graphics, the Hausdorff distance is used to measure the difference between two different representations of the same 3D object. It is also used in biological sciences for the analysis of protein structures. Considering we have point sets $A \in R^2$ and $\in R^2$, the goal is to find the level of similarity between A and B sets. There are several algorithms for finding the shape similarity including Hausdorff, Fr´echet distances, and Minkowski distance (Veltkamp, 2001). In this section, we try to use these algorithms to find the similarities between two routes in three dimensions (Veltkamp, 2001). This mathematically means that we want to find a match that has one of the following characteristics:

- Minimizing maximum distance between mapped points
- Minimizing sum of distances between mapped points

- Minimizing sum of squared distances between mapped points

6.2.1 Hausdorff Distance

The Hausdroff distance is a measure of how far two subsets are. It is calculated by determining the maximum distance from a set to another set in which each point in a set A is matched to the nearest point in set B.

The Hausdorff distance from A to B, h(A, B), is $max_{a\in A} min_{b\in B} ||a-b||$,
The Hausdorff distance from B to A, h(B, A), is $max_{b\in B} min_{a\in A} ||b-a||$,
The Hausdorff distance between A and B, H(A, B), is max {h(A, B), h(B, A)}.
(Hausdorff and Frechet distance Lecture, 2017)

Basically, h(A, B), for each point on dataset A, first finds the nearest neighbor point to it on dataset B, and finally reports the farthest of these distances. In this way, Hausdorff distance measures the mismatch between these two datasets. The idea is shown in Figure 32. (Alt & Guibas 1999).

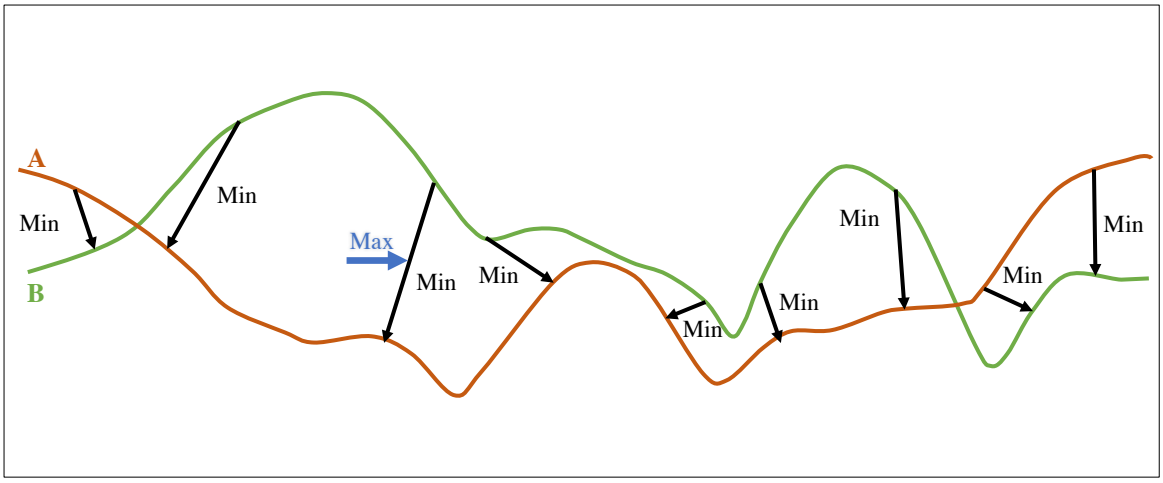

Figure 32 Hausdorff distance as a measure of similarity

6.2.2 Fr´echet Distance

The Fr´echet distance is explained with the following example. Assume a man is walking a dog where the man is walking on one curve and the dog on another curve. The minimum length of a leash allowing a dog and its owner to walk along the two curves without backtracking is defined as the Fr´echet distance (Figure 33). The Fr´echet distance is known to be a better measure of shape-matching for curve or surface matching.

The Fr´echet distance between two curves is calculated as follows:

$$F(A,B) = inf_{\alpha,\beta} max_{t\in[0\alpha,1]} ||A((t)) - B(\beta(t)|| \quad (17)$$

where A, B : [0, 1] R^2 are parametrizations of the two curves

and α, β: [0, 1] → [0, 1] range overall continuous and monotone increasing functions. (Alt & Godau, 1995; Alt & Godau, 1999)

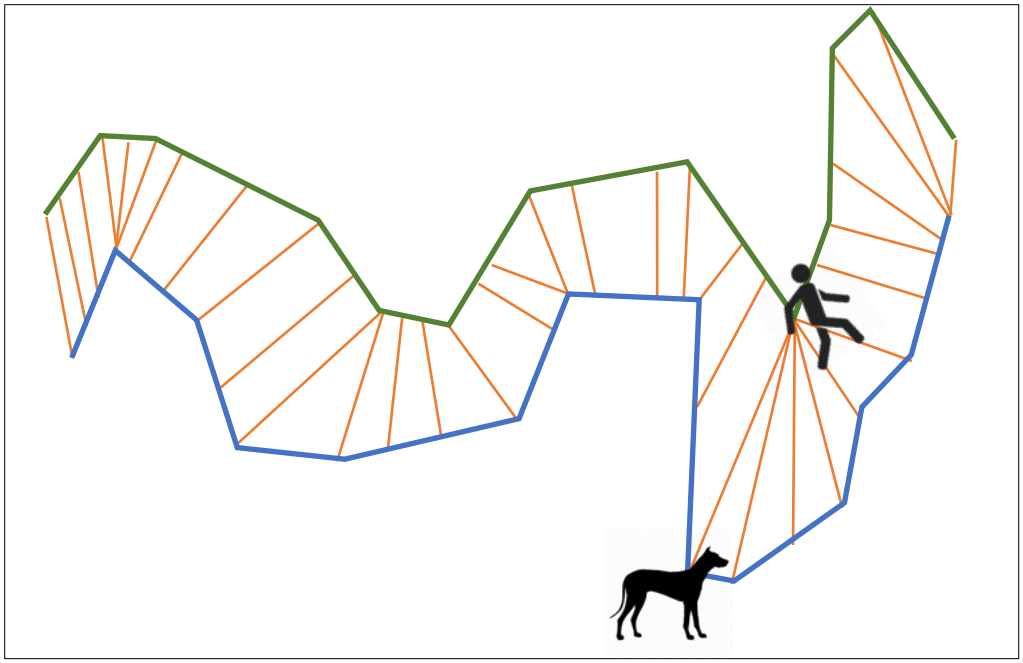

Figure 33 Fr´echet distance as a measure of shape similarity

6.3 Route Similarity

As mentioned before, this study proposes using shape-matching to find the similarities between two routes in order to compare the travel times of two similar routes. In this study, two algorithms of Fr´echet distance and Hausdorff distance were implemented. The Hausdorff distance seems to be a better indication of route similarity and is chosen for this aim.

The implemented algorithm is an approximation to the Hausdorff Distance for two geometries. This algorithm measures the degree of similarity between two geometries. The computed measure is in the range of [0, 1]. Higher measures indicate a great degree of similarity, which means that a measure of 1.0 shows two exact matches and a

measure of 0.0 indicates that the two have essentially no similarity. The algorithm takes the geometry of the two routes of interest and returns the normalized Hausdorff metric. When comparing travel times, it is important to use the same route; otherwise, the comparison is meaningless. Knowing this, we calculate this metric for probe data trajectory vs. Google routes. The travel time estimation comparison can be calculated between routes that have a similarity score higher than 0.99.

Figure 34 shows an implementation of the proposed algorithm for a GPS route (in blue) vs. Google route (in red) with a 0.7005 (70.05%) similarity index. The origin-destination is the same for both GPS and Google routes, and it is evident the routes have the same path up to a point and they bifurcate after that and meet again at the destination.

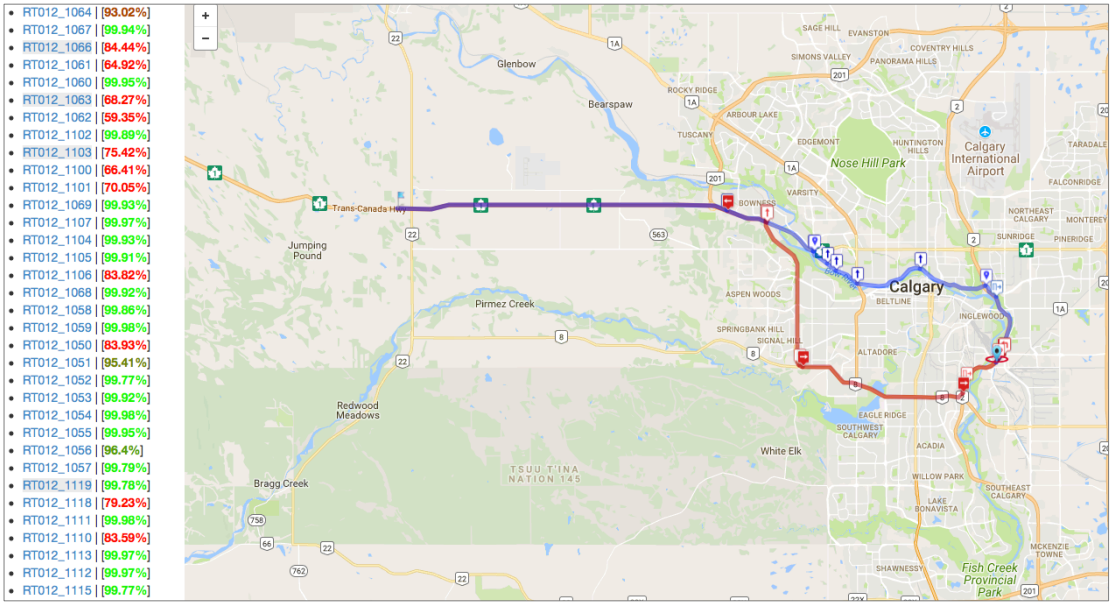

Figure 34 Route similarity index example (two routes with similarity index of 70.05%)

6.4 Google Maps API

The Google Maps Directions API is a service that estimates directions and travel time between locations. The directions between two locations are provided for several modes of transportation including transit, driving, walking, and cycling. The challenge is that the Google cost model is not available to the public to allow comparisons of travel time estimations for the proposed model vs. the Google model. Also, it is not possible to acquire all the link combinations with Google API, as it returns only the most efficient routes in terms of travel time when calculating directions between two locations. For this reason, route similarity is critical to making it possible to compare only the routes that are identical.

6.4.1 General Information

Google API can provide an estimate for current time or future time. There is a limitation of 2,500 free directions requests per day with Google API. There are several parameters that can be changed in the URL, some of them are required while others are optional. Parameters are separated by the "&" character. Two of the important optional parameters for us are "departure_time", and "traffic_model".

6.4.2 Departure Time

Google uses the parameter "departure_time" to identify a specific travel time; the value is the time specified as Unix time, or seconds since midnight, January 1, 1970, UTC.

6.4.3 Travel Time with Traffic

The other optional parameter to use to collect travel time is "traffic_model," which can pull the following values: best _guess, optimistic, and pessimistic. This parameter's values change the "duration_in_traffic" value.

traffic_model parameter values:

1. **Best _guess** (which is the default factor) is based on historical averages.
2. **Optimistic** gives the duration_in_traffic equal to the lowest value in the range Google Maps provides (based on observations for a couple cases).
3. **Pessimistic** gives the duration_in_traffic equals to the highest value in the range Google Maps provides (based on observations for a couple cases).

The best application of the proposed method would be when there is real-time data, and the proposed model could provide a prediction of travel time that could be compared with Google API using the Best_guess and Current_time parameters. However, this data is not available and this approach could not be taken in this study. Nevertheless, it can still be used to find how the model is performing in comparison with Google API. In this way, during the off-peak hours, the estimated travel times could be compared, removing the real-time effects and the base estimation without traffic. The proposed approach could be used in any other studies and most importantly when real-time data is available.

6.5 Waypoint Parameter

There is an optional parameter in Google API through which one can specify the waypoints available for driving, walking and bicycling, but not transit. Using this parameter, however, changes the route from the original in most of the cases, either because of an observed GPS error or based on Google API logic. As such, it would still be necessary to use route similarity. Furthermore, the advantage of using default Google routing instead of a giving waypoint is that the route selections could be compared as well. This is particularly important when there is also a routing suggestion based on an existing cost model. In this way, both the cost model and the routing suggestions could be compared. This parameter in Google API collects the specified array of waypoints (latitude/longitude coordinate) and alters the route by directing it through the specified waypoints. This study tried this feature as well, but it turned out using the route comparison and checking for similarity higher than 0.99 is a better approach.

6.6 Comparison

As mentioned, the comparison of the proposed model and Google travel time estimation were conducted for the off-peak hours and equivalent times (same day, month, time) and 'best_guest' parameters. The proposed model and Google travel time estimations are also compared with the actual travel time observed from the probe data. The steps that need to be taken for the compression between the proposed model, Google model, and the probe data is shown in Figure 35. The indicated steps were

taken, and the results of the comparison for the Baltimore area are demonstrated in the following section.

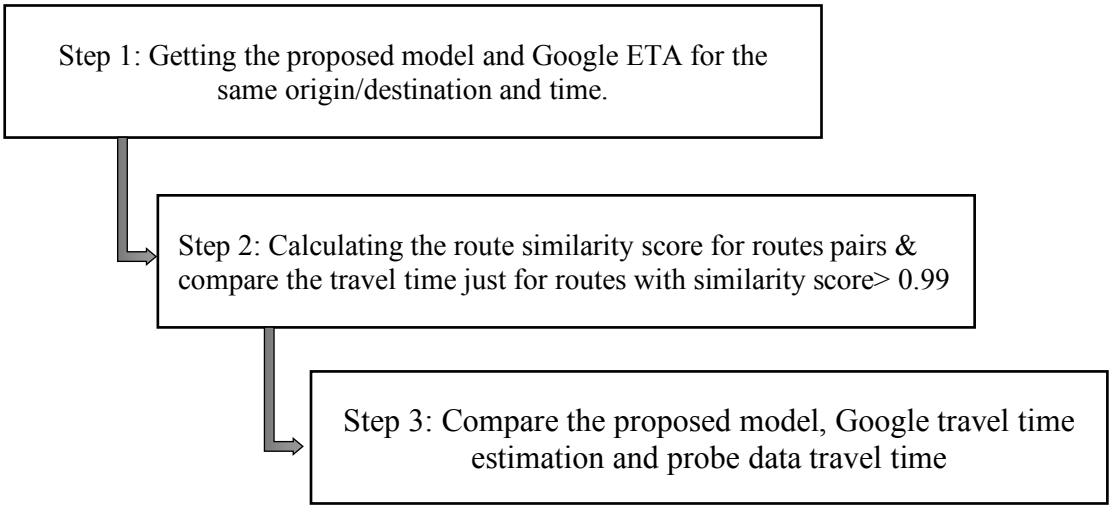

Figure 35 The proposed framework for travel time estimation comparison between models

6.6.1 Case Studies

This section demonstrates how the proposed framework is used for the study area to compare the proposed model in this study to the Google API and the actual observations. As mentioned before, since we do not have access to real-time data, we illustrated the framework using historic data.

To do this, all the historic data for all the training datasets were used as the input for the model to be trained. Once the model was trained, the model predictions of travel times for the test day again for 12 AM to 6 PM time, and with the same weather conditions are obtained. As such, for all the trips that occurred on the test date, we have the origin/destination, GPS track, and the proposed model's predicted travel time. For

all those trips, we acquired the Google estimates as well. In Google API, we entered the same day, time of day, day of week and month and same weather condition as the test day, and used the "best_guess" parameter as the API input for the future data since only current and future times are available through Google, however, If we had the current data, we could simply use the current Google API estimation. In this way, we obtained the origin-destination, GPS track, and the estimated travel time of Google as well.

The next step was to find those trips with a shape similarity index between the actual observations and the Google proposed path. For trips that the observation track and Google proposed route are the same, the average of travel times and errors for proposed model, the Google, and actual observation is demonstrated in Table 12. The details of travel times are shown for a sample of these trips in Table 13. What follows demonstrate the results of this comparison between the Google API, proposed model, and the observations.

6.6.2 Results

6.6.2.1 Off-peak Travel Time Comparison

Figure 36 demonstrates an example of the comparison between the Google API and proposed model estimated travel time and observed travel time. The given trip is on Tuesday, in winter, and with fair weather conditions and non-peak hour traffic (February 10th at 00:57 AM).

The example trip route has a similar route based on a Google API proposed route, which means that the shape similarity index between the two routes is higher than 0.99 and we can compare the travel times.

The observed travel time is 500 seconds; whereas, the Google API estimate is 480 seconds; finally, the proposed model estimation is 508 seconds. In this case, there is a 4% difference between the observed travel time and Google API estimation, and a 1% difference between the observed travel time and proposed model estimation. Also, the difference between Google and the proposed model estimation is 0.3 %. As mentioned, this is just one example to demonstrate the comparison process. As such, the comparison is done for the test dataset and results are demonstrated and discussed as follows.

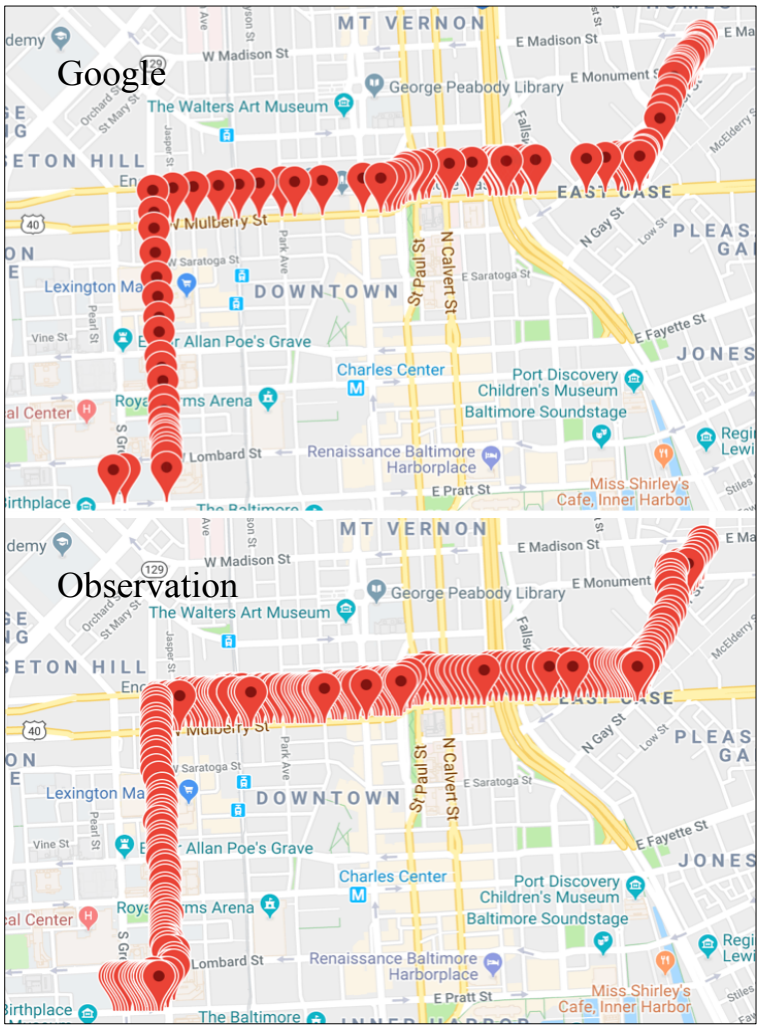

Figure 36 Proposed Model, Google API, and observed travel time comparison (off-peak)

The selected test case is based on trips on a Tuesday in winter under normal weather conditions, for midnight to 5 am in the morning, for a travel length of more than five minutes. Around 190 cases were randomly selected from which around 40% of them had the same route as the Google API route with a shape similarity index of 0.99 and above.

The sample 90 tested cases have an average trip of 420 seconds, with a maximum of 608 seconds. From this sample, 10 trips were doubled-checked manually (both probe trajectory and timestamps) and all appeared normal.

Figure 37 Google and proposed model Percentage of error trend demonstrates the travel difference from the ground truth trend (they are not necessarily equivalent points) for the proposed model and Google. As it is shown, the proposed model has the tendency to estimate a higher travel time than the ground truth, and Google's estimates are lower than ground truth 50% of the time.

As is shown in Table 12, the average estimation error for the proposed model is 20%; whereas, the average estimation error for the Google API is 22%. This analysis demonstrates the proposed process and the result of the analysis for 90 cases. The analysis shows that the proposed model estimation does not yield results that stay far from Google API, even though historical data is used.

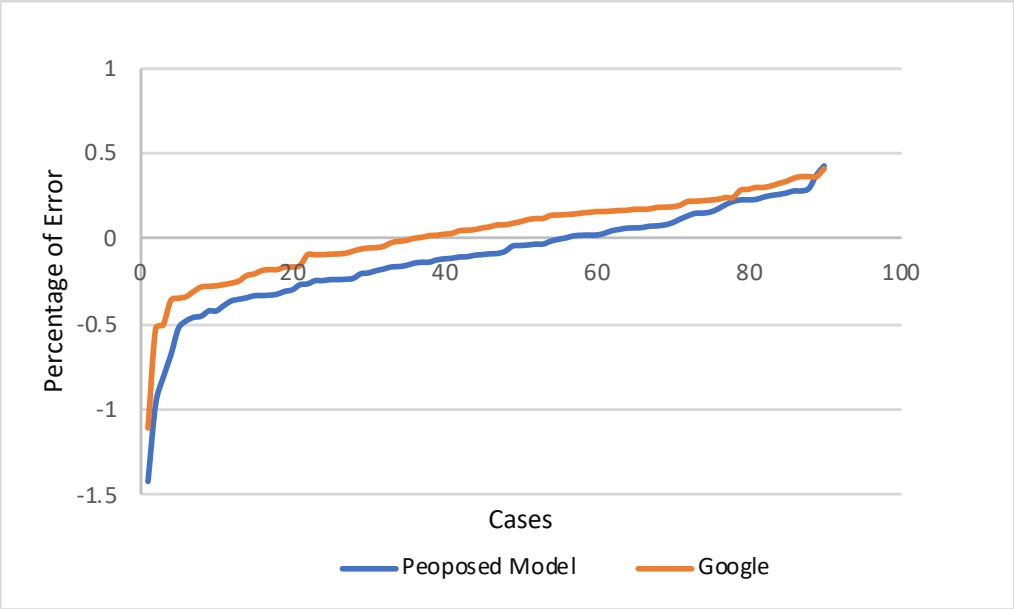

Figure 37 Google and proposed model Percentage of error trend (off-peak)

Table 12 Google and proposed travel time prediction summary (off-peak)

Variable	Value
Number of Observations	90
Average Trip Time (s)	420
Average Proposed Model Average Error (s)	75
Average Google Average Error (s)	84
Proposed Model Average Error %	20
Google Average Error %	22

Table 13 Sample of Google, proposed model travel time prediction and observations travel time comparison (off-peak)

Trip	Observation (s)	Proposed model (s)	Google (s)	Google-Obs (s)	Proposed-Obs (s)
1	507	500	480	-20	7
2	455	322	429	107	132
3	547	507	444	-63	40
4	377	362	525	163	15
5	405	353	380	27	52
6	420	374	339	-35	46
7	418	301	415	114	117
8	429	354	296	-58	75
9	372	443	315	-128	-71
10	495	340	468	128	155
11	606	527	539	12	79
12	479	580	452	-128	-101
13	476	450	407	-43	26
14	436	533	392	-141	-97
15	377	484	286	-198	-107
16	465	445	472	27	20
17	303	360	288	-72	-57
18	353	414	344	-70	-61
19	491	540	333	-207	-49
20	528	435	349	-86	93
21	328	360	280	-80	-32
22	491	540	333	-207	-49
23	390	502	350	-152	-112
24	390	410	350	-60	-20
25	318	430	277	-153	-112
26	461	302	409	107	159
27	412	344	384	40	68
28	608	522	465	-57	86
29	440	372	295	-77	68
30	600	360	509	149	240

6.6.2.2 PM peak Travel Time Comparison

As mentioned before, to remove the effects of real-time traffic variability, the comparison was done during off-peak hours. However, what follows is the demonstration of comparison between the proposed model, Google API, and the actual travel-time for a sample of observations on a Monday during PM peak hours. The sample cases are randomly selected from the test date. Around 30 trips are selected from which 11 cases met the comparison criteria. The average percentage of error for the proposed model is 10.8 % whereas the Google average percentage of error is 24.4%. Figure 38 demonstrates an example of the comparison between the Google API and proposed model estimated travel time and observed travel time. The given trip is on Monday, in winter, and with fair weather conditions and peak hour traffic. The example trip has a similar route as the one proposed by the Google API, meaning that the shape similarity index between the two routes is higher than 0.99 and consequently the travel times can be compared.

The observed travel time is 422 seconds; whereas, the Google API estimate is 477 seconds; and the proposed model estimation is 391 seconds. In this case, there is a 13% difference between the observed travel time and Google API estimation, and a 7.4 % difference between the observed travel time and proposed model estimation. Table 14 shows all the 11 cases which are considered in this analysis.

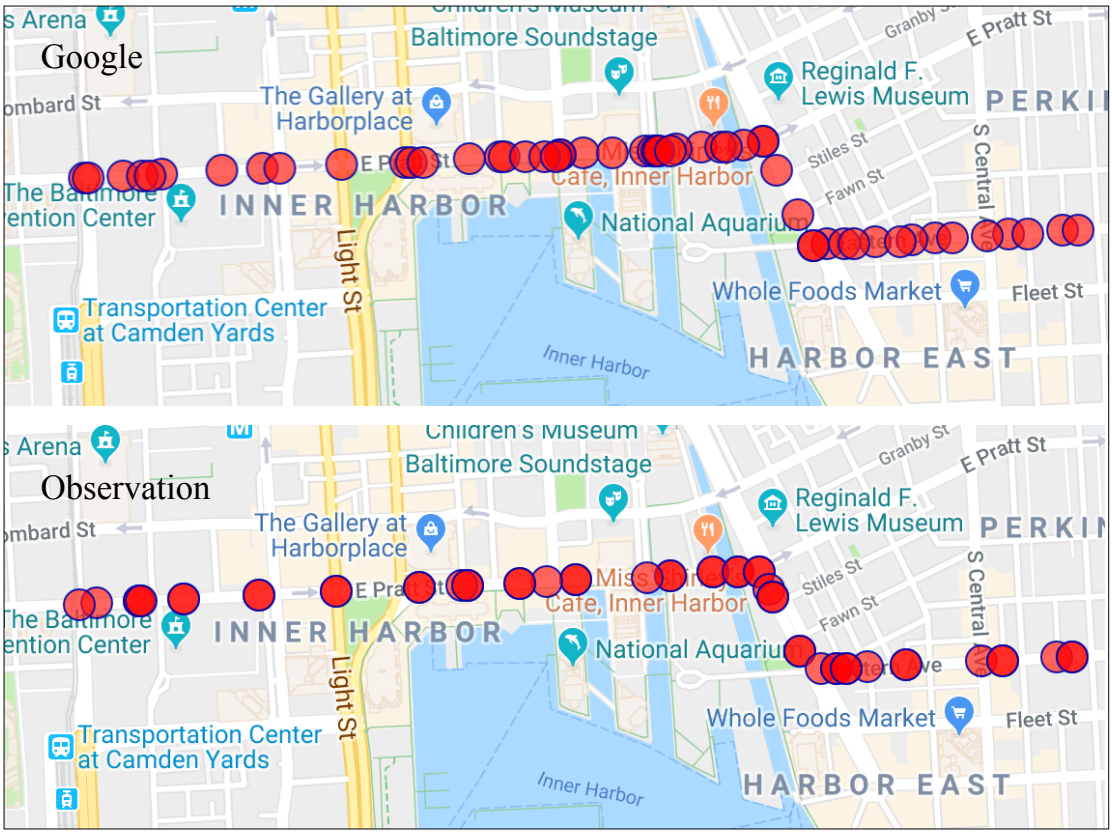

Figure 38 Proposed Model, Google API, and observed travel time comparison (PM peak)

Table 14 Google, proposed model travel time prediction and observations travel time comparison (PM peak)

Trip	Observation (s)	Proposed model (s)	Google (s)	Google-Obs (s)	Proposed-Obs (s)
1	238	220	429	191	-18
2	414	365	455	41	-49
3	540	492	598	58	-48
4	382	421	647	265	39
5	308	355	474	166	47
6	422	391	477	55	-31
7	378	411	620	242	33
8	630	490	618	-12	-140
9	233	246	192	-41	14
10	540	492	598	58	-48
11	847	779	772	-75	-68

6.7 Sensitivity Analysis

Sensitivity analysis is the study of how the output of a mathematical model can be sensitive to the inputs and assumptions. In this process, the outcome is recalculated under alternative assumptions/inputs to determine the impact of each assumption/input. There are several benefits to conducting sensitivity analysis such as understanding the potential relationships between input and output variables in the model, evaluating the robustness of the results of the model when uncertainty exists, finding any possible errors in the model, and developing a possible enhancement in the model by providing better input and so on. This study is an empirical study and the model is a data-driven model; as such, there are not many input factors/assumptions in the model. In this section, the input factors are explained, and the results of sensitivity analysis are demonstrated.

6.7.1 Number of States

As mentioned before, here we have a few assumptions on which we can conduct the sensitivity analysis, one of which is the fact that we considered only two states of traffic congestion. The reason to choose two states of congestion is that in some of the cases, two distributions with two different means can be detected in the histogram of the links raw travel time data. However, there is no robust empirical study showing that that is the case. On the other hand, the more states we have, the more data we need to do the analysis, so even if there were more traffic states, we did not think with the available amount of data in this study more than two states can be considered. To demonstrate

whether or not the higher number of states could result in a different measure of model performance, and in order to explain the observation with accuracy, we conducted sensitivity analysis on the number of states in the model.

In this part, the three states and four states are considered for the model. The result for the three states is shown in Table 15. As shown in Table 15 and Figure 39, the performance of the model utilizing a higher number of states decreased. To ensure that the result is not an accident, three different days of week and months are randomly selected and the performance of the model with two states and three states are compared. Figure 39 also demonstrates this trend of diminishing performance by incorporating a higher number of states for several test days. As mentioned before, this could be attributed to the small sample size. For example, utilizing three states means that we need to have three distributions for each link, and we need to have enough observations in all the cases to have sound data.

It should be noted that the increasing the number of states results in much higher computational complexity for the training part. The initial test on the 4 states case showed that training time was very high which made it inapplicable in the real-world situation and therefore it is not considered here. Thus, the two-state option seems to be a good choice, at least for this amount of data. With a larger amount of data, the number of states could be considered to see if it can improve model performance.

Table 15 Sensitivity analysis on the number of states

Test Day Season	Wednesday, Winter	Tuesday Summer	Friday, Fall
Average test trip	546	529	537
Model with 2 states error (s)	99	95	108
Model with 3 states error (s)	118	112	132
Model with 2 states error %	18.5	18.5	20.2
Model with 3 states error %	21.3	21.0	23.7

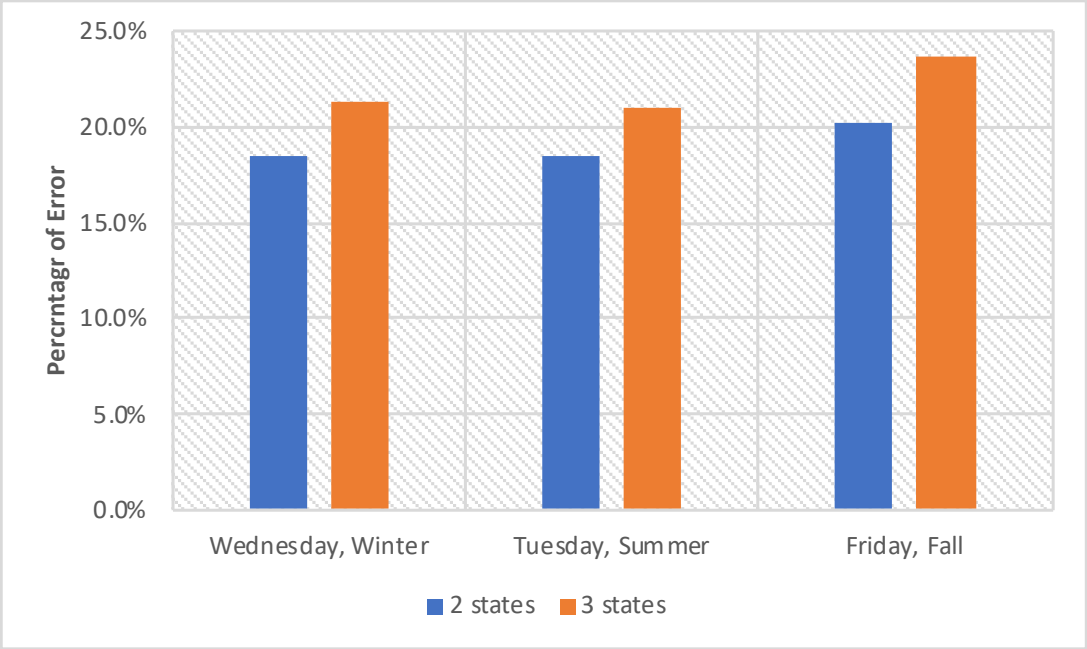

Figure 39 Sensitivity analysis on the number of states

6.7.2 Number of Observations

In most of the data-driven models, the model is assumed to be very sensitive to the amount of data. In this study, all the available data is used for the model. However, to show how sensitive the model is to the amount of data, the sensitivity analysis is conducted on the amount of data. For this purpose, since we do not have more data, less data is used to see how the model performance could potentially change based on the amount of available data.

In this part, 80% and 50% of the whole data is used for the modeling. 80% (50%) of the data is selected from all of the available days to have a better representative data (not just using 50% of the days). The test day is randomly selected for this analysis. The test day is Friday, October 2nd which is in the fall season and the trained model is tested on this day.

As noted, the model is trained with 80% of the data and 50% of the data and the results are demonstrated in the Table 16. As it is shown in Table 16, for the case of the trained model with 80% of the data the decrease in the percentage of the error is not significant and we can say the performance of the model with 100% of the data and 80% of the data, at least on this test day, is the almost the same. However, there is an evident improvement in the performance of model comparing the model with 50% of the data with models with 80% and 100% of the data. Thus in general, we can say the amount of data could have an impact on the model performance. The lower the number of observations we have, the lower the performance would be, and the closer the proposed

model performance would be to the base model with real-time average and regression model.

Table 16 Sensitivity analysis on the amount of data

Variable	Value
Average test trip	537
Model with 100 % of data error (s)	108
Model with 80 % of data error (s)	109
Model with 50 % of data errors (s)	126
Base model errors (s)	152
Model with 100 % of data error %	20.2
Model with 80% of data error %	20.3
Model with 50 % of data errors %	23.7
Base model errors %	28.1

This trend is also shown in Figure 40, which illustrates the improvement in the performance of the proposed graphical model and its correspondence with the increase in the number of observations.

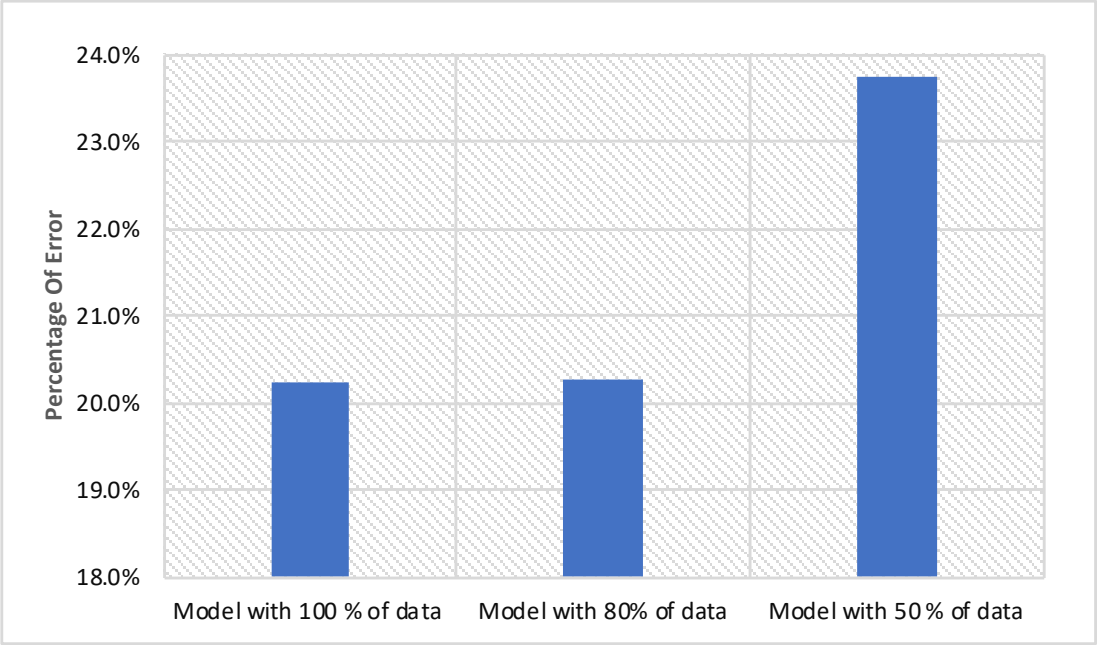

Figure 40 Sensitivity analysis on the amount of data

The analysis outcome shows that the model performance could potentially improve with more data and the model could be very promising when we have enough samples of data. In the future, this model could be applied with more data in order to see how performance could be improved. The availability of data is critical not only because the analysis shows that the higher amount of data could itself result in higher model performance, but also because it makes it possible for other proposed variables such as intersections and movements and more weather conditions to be included in the model and potentially increase the performance of the model across a variety of conditions.

6.8 Summary

The proposed model to compare this study model and Google API has three main steps. The first step involves having a number of trips for the test and changing the Google API parameters to match the test trip. The Google API parameters include origin, destination, departure time, traffic model, and waypoint, all of which are changed according to each trip. Then, the route proposed by Google API and the observation are compared to ensure that we are comparing the travel times on the same routes. To achieve this, this study proposes the use of a well-known shape similarity algorithm, Hausdorff Distance, and incorporates only those trips for which the Google routes and the observation routes are identical. Finally, this study compares the travel times provided by the Google API, the proposed model, and the observation.

The analysis for the sample of trips during off-peak hours shows that, in general, Google API and the proposed model have a comparable difference from the ground truth (observation) and, in general, Google API overestimates the time in 50% of cases and underestimates travel time in 50% of cases. The proposed model, however, has a tendency toward overestimation.

Finally, sensitivity analyses were conducted for two parameters of the model. One was the number of congestion states, which was considered two in this study. To find out how the model is sensitive to the states of congestion, three and four congestion states were considered and tested. The analysis results showed that, in instances where there were a higher number of congestion states, the model had lower performance. This could be because of the low number of observations or because two congestion states

are the best in representing the reality. The other parameter that was analyzed was the number of available observations. This analysis showed that, with a lower number of observations, we would have lower performance. This implies that, with larger amount of data, the proposed model's performance can improve.

Chapter 7: Summary and Conclusion

In this Chapter, we provide a complete summary of the research. Then we discuss the findings of this study. Finally, in conclusion, some of the limitations of this study and concrete suggestions for future research are proposed and discussed.

7.1 Summary

Travel time prediction is very important to everyday life. Reliable travel time prediction helps both road users and system controllers to be informed about the future conditions on roadways and enables them to make their best decisions based on that information. In a world where we are facing a rapid increase of traffic congestion, accurate Traveler Information Systems are increasingly crucial. Accurate and reliable travel time information could help to reach control and could alleviate this congestion by providing a more reliable network. The first step toward this goal is the creation of accurate real-time traffic monitoring systems. This is especially important in arterial networks since they are major city streets that provide for travel within and between cities. Yet, there is not much work that is being done to improve arterial real-time traffic monitoring.

In general, travel time estimation and prediction are very complex and challenging tasks. This is due to unknown variables such as varying driver behaviors, uncertainty in demand, weather, incidents, roadway conditions, etc. This is even more challenging for arterials since there are other factors playing a role in this uncertainty, such as traffic signals and the relation between the links and their effects on each other.

Many of these fluctuations could be captured in a travel time predictions model. In fact, the goal is to capture these fluctuations and to be able to predict the fluctuations as much as possible. Nevertheless, there is still some uncertainty associated with travel time that poses challenges for capturing this information. Still, knowing about these uncertainties and the range of uncertainties is very important.

To address the need for travel time predictions, many studies have proposed different methodologies. Most of these studies are for freeway travel time prediction, and fewer are for arterial travel time prediction.

An overview of these methods for travel time prediction in general and arterial travel time prediction was done in Chapter 2. The approaches to this problem can be categorized based on the scope, output, and methodology used. There are two primary different outcomes: large-scale navigation systems (mostly deterministic) and probability distributions of travel times, mostly for the smaller scale. There are two main scopes that are distinguishable in the literature, microscopic (includes a small number of intersections) or macroscopic. Finally, in terms of the used methods, the approaches can be categorized into three main approaches: parametric, non-parametric, and hybrid modeling. The parametric methods, such as Feedforward Neural Network, Hidden Markov Model, and Autoregressive Models, have strong model assumptions. Comparatively, the non-parametric methods, such as the k-nearest neighbor, require fewer assumptions and they grow with the amount of data. Some of the proposed probabilistic methods used specifically for arterial travel time prediction, such as STARMA, Dynamic Bayesian network using EM algorithm, and Spatial Moving Average (SMA), were discussed in more detail.

This study proposed a probabilistic model using raw probe data. Using the raw probe data for the modeling purpose presents a lot of challenges and requires steps to be taken prior to the modeling. However, as of the time of this writing, the author has no knowledge of a study that features a framework for each of these steps. Chapter 3 of this research provided a framework for each preliminary step needed to be taken before the modeling. The steps include the data cleaning framework, map matching of the data to the network, and creation of the network. This study proposed a framework to work with raw and dirty probe data, including two main steps, trip splitter and trip filtering. The three main splitters are Time Difference Splitter, Waypoint Sequence Splitter, and Location Difference Splitter. The trip filters are Same Time Filter, Same Location Filter, Speed Boundary Filter, Idle Filter, Location Displacement Filter, Trip Length Filter, Shaking Probe Filter, Large Error Size Filter, and Shaking Probe Filter.

Next, the map matching and existing methods for map matching were discussed. This study used Hidden Markov Model (HMM) map matching methods to find the match that maximizes the multiplication of the likelihoods of the points on the map for each latitude/longitude, as well as the maximum likelihood of the sequence between two consecutive latitude/longitude pairs.

Finally, the initial structure of the model was explained. The proposed model in this study is an enhanced graphical model. In the graphical model, it is assumed that there is a conditional dependency between random variables, which are demonstrated by a graph.

As such, this chapter explained how a network or the base Dynamic Acyclic Graph (DAG) of the roads is created by creating the graph of the street map. In the base DAG,

a variable is assigned to each direction of a link and for time step t and the time step of t+1. The state of each link (variable) is conditioned on the neighboring links' (variables) state at the previous time step. Consequently, the most important part of creating the network is creating this adjacency matrix of the connection between links. The enhanced graphical model was explained in Chapter 4. The enhanced graphical model has a global variable for time of the day, the day of the week, season, and weather conditions. All of the variations in either demand or travel behavior imposed by each of these variables are included in the model, and the model is able to estimate/predict travel time on each of these scenarios. The chapter continued with an explanation of Streaming Variational Bayes (SVB) algorithm. The SVB is an approximate inference algorithm that is used instead of exact Bayesian inference – the latter is very time consuming, and it is often inapplicable for complex models. Streaming Variational Bayes deploys the Bayes rule to convert a conventional Variational Bayes method into an incremental approach.

The validation of the models and analysis of the variables influence was demonstrated in Chapter 5. The results of the enhanced graphical model were demonstrated for different scenarios, including different seasons and different days of the week. In most of the scenarios, the percentage of error of the model for short-term travel time prediction is less than 20%. The graphical models were compared to each other's with incrementally added variables in order to allow the evaluation of the effects of inclusion of each variable in the model. The performance of the model was also compared to a base model, which is a real-time average, as well as a combination of real-time average and a regression model. The proposed model outperformed these two models in all of

the scenarios. The weather conditions variable was separately evaluated and discussed in this chapter, and the accuracy of the model in bad weather conditions was demonstrated again for different scenarios.

The chapter concluded with detailed suggestions on extending the model by bringing the intersections and driver behavior into the modeling process. The extended model for inclusion of intersections, for example, includes a movement variable in the graph for each of the possible movements in the intersection. However, this model could not be tested in this research due to a lack of data regarding all the movements. In this study, there were enough data to test the original proposed model without the extensions. We provided comprehensive details for developing the extended model and testing it in future studies pending availability of data.

Even though the model is evaluated based on the test data, in Chapter 6, this research proposed a new method to compare the proposed model (or any real-time prediction model) with any publicly available routing API such as Google. The proposed framework has three main steps. The first step involves having information from a sample trip (including GPS tracks) for the test date, and matching the Google API (or any available API) parameters according to each of the trips; it also involves obtaining the Google APT prediction/estimation for the same trip. The next step involves comparing the observed trip GPS tracks for the proposed model, and the Google API routes using Hausdorff Distance to make sure the travel times for the same route are being compared. The Hausdorff Distance is a Shape-Matching algorithm and is a measure that shows how similar are two subsets which in our case are the GPS tracks. Lastly, there is a comparison of the estimations/predictions for those routes that are

identical. The results showed that the proposed model and Google API have a comparable difference with the ground truth.

Sensitivity analysis was done on two factors of the study. First, we evaluated the impact of the higher number of traffic states (three, and four were considered). The results of the analysis showed that, with this amount of data, the performance of the model decreases when the number of states is increased. Then we analyzed the impact of the amount of data available for use in the modeling process. The analysis showed that using a smaller amount of data (since we did not have more) could result in lower performance of the model, or even make it difficult or impossible to capture the influence of a variable at all.

7.2 Conclusions

What follows is the list of this study findings:

- The first step in developing a good model is gaining access to the cleanest data available. The cleaned data could result in a better estimation/prediction model. This study proposed a framework to clean the data has proven to be effective.
- Using the Streaming Variational Bayes could make the model usable for large-scale networks with more variables in the model.
- There is a time pattern in the travel time that can be captured. This variable could be incorporated in the model.
- There is a seasonal pattern in travel time that can be captured by including this variable in the model.

- The weather condition does affect either driving behavior or demand, and it influences travel time and its uncertainty. The analysis confirms that the travel time distribution in bad weather conditions has a higher mean and wider standard deviation.
- There is an average of 2% difference for the Google travel time estimation (with the available parameters) and our proposed model travel time estimation. Also, there is an average of 20% difference for the Google travel time estimation and actual travel time observations.
- Less amount of data could result in poorer estimation/prediction and lower model performance. This means that more data could potentially result in better modeling and higher performance.
- The model has the highest performance when it considers two states for each link's travel time distributions. Considering a higher number of states results in lower performance. This could be attributed to the fact that two states can explain the observation the best, or it could be that, with the existing sample size, adding more variables into the model could decrease model performance.

7.3 For Future Research

This research provided contributions to the existing literature in the area of real-time arterial travel time prediction. There are some model enhancements that could not be accomplished in this study due to a lack of data. The most important enhancement would be incorporating intersections in the model. This means that we would decompose the trip travel time to link travel times and delays at the intersection. For

this purpose, this research suggested incorporating a movement-based variable in the model. By doing so, the most important variation in the intersection based on the type of movement (left, right, and straight) could be captured in the modeling. Many commercial models still encounter this challenge in producing an accurate estimation/prediction waiting time for the traffic signals, specifically when it comes to left turns. If and when more data are available as it is becoming increasingly the case, this extended model can be developed and tested by using more data.

This research assumed the links travel time to be independent for the sake of data availability, in the future and with the availability of more data, the interdependency between the links travel time could be captured by learning the joint travel time distributions for neighboring links.

The other variable that could be considered in the graphical model is the driver behavior variable, which could be captured by individual vehicle trip speed. The effects of bringing this variable into the graphical model or other models could be evaluated in future studies.

This study also included weather condition as an explanatory variable in the model. Two weather conditions were used in this study, however, with greater availability of data, more weather condition scenarios (such as heavy rain, heavy snow, storm etc.) could be used to expand this variable. This can lead to a more accurate travel time estimation/prediction in real-world conditions. Another interesting consideration about weather conditions would be the possibility of bringing weather predictions into the travel time modeling, as well as the uncertainties associated with it.

There are other events that can influence travel time significantly, including incidents, work zones, special holidays, etc.. However, enough vehicle probe data are needed for each of these events in order for them to be incorporated into the model. With rapidly improving technology development, increasing amounts of data are becoming available, which makes it possible to include these events in the modeling. Including these events could yield more accurate estimation/prediction.

Bibliography

Ahmed, M. S., and Cook, A. R. (1979) "Analysis of freeway traffic time-series data by using Box-Jenkins techniques," *Transportation Research Record*.

Alt, H., and Godau., M. (1995) "Computing the Fr´echet distance between two polygonal curves," Int. J. Comput. Geom. Appl., 5:75–91.

Alt, H., and Guibas, L. (1999) "Discrete geometric shapes: Matching, interpolation, and approximation - a survey," In J.-R. Sack and J. Urrutia, editors, Handbook of Computational Geometry. Elsevier, 1999.

Ban, X. Herring, R., Hao, P., and Bayen, A. (2009) "In Proceedings of the 88th Annual Meeting of the Transportation Research Board, Washington, D.C.

Brakatsoulas, S., et al. (2005) "On Map Matching Vehicle Tracking Data," *in 31st International Conference on Very Large Databases*: Trondheim, Norway p. 853-864.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., and Jordan, M. I. (2013) "Streaming Variational Bayes," *In Advances in Neural Information Processing Systems* 26, 2013.

Budge, S., Ingolfsson, A., and Zerom, D. (2010) "Empirical analysis of ambulance travel times: The case of Calgary emergency medical services," *Management Science*, 56:716–723.

Butler, S., Ringwood, J. and Fay, D. (2007) "Use of Weather Inputs in Traffic Volume Forecasting," Proceeding of Irish Signals and Systems Conference, Derry, Ireland.

Cetin, M., and Comert, G. (2006) "Short-term traffic flow prediction with regime switching models," Transportation Research Record: *Journal of the Transportation Research Board*, vol. 1965, pp. 23-31.

Chung, E., Sarvi, M., Murakami, Y., Horiguchi, R., Kuwahara, M. (2003) Cleansing of probe car data to determine trip OD, in: Proceedings of 21st ARRB and 11th REAAA Conf

erence, Cairns, Australia.

Cui, Z. Ruimin, K. Yinhai W. "Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction", access on January 2018 https://arxiv.org/abs/1801.02143.

Davis, G. A., Nihan, N. L., Hamed, M. M., and Jacobson, L. N. (1990) "Adaptive forecasting of freeway traffic congestion," *Transportation Research Record*.

Farokhi Sadabadi, K., Hamedi, M., and Haghani, A. (2010) "Evaluating moving average techniques in short-term travel time prediction using an AVI data set," *in Transportation Research Board 89th Annual Meeting*.

Ghosh, B., Basu, B., and O'Mahony, M. (2009) "Multivariate Short-Term Traffic Flow Forecasting Using Time-Series Analysis," *Intelligent Transportation Systems*, IEEE Transactions on, vol. 10, pp. 246-254.

Google API, https://developers.google.com/maps/documentation/directions/intro.

GraphHopper, https://github.com/graphhopper/map-matching, accessed December 2016.

Greenfeld, J.S., (2002) “Matching GPS Observations to Locations on a Digital Map,” *in 81st Annual Meeting of the Transportation Research Board*: Washington, DC, USA.

Haghani, A., Hamedi, M., Sadabadi, K. F., Young, S., and Tarnoff, P. (2010) "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.

Hamed, M., Al-Masaeid, H., and Said, Z. (1995) "Short-Term Prediction of Traffic Volume in Urban Arterials," *Journal of Transportation Engineering*, volume. 121, pp. 249-254.

Hamner, B. (2010) "Predicting travel times with context-dependent random forests by modeling local and aggregate traffic flow," in Data Mining Workshops (ICDMW), 2010 IEEE International Conference, pp. 1357-1359.

Hao, P., Boriboonsomsin, K., Wu, G., Barth, M.J. (2017)” Modal Activity-Based Stochastic Model for Estimating Vehicle Trajectories from Sparse Mobile Sensor Data”. IEEE Trans. *Intelligent Transportation System*.

Herring, R. (2010) “Real-time traffic modeling and estimation with streaming probe data using machine learning,” Ph.D. dissertation, *Univ. California, Berkeley*, CA.

Herring, R., Hofleitner, A., Abbeel, P., and Bayen, A. (2010) “Estimating arterial traffic conditions using sparse probe data,” In Proceedings of the 13th IEEE International Conference on Intelligent Transportation Systems. IEEE, 923–929.

Hofleitner, A., Herring, R., and Bayen, A. (2012 a) "Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning," *Transportation Research Part B*, 46:1097–1122.

Hofleitner, A., Herring, R., Abbeel, P., and Bayen, A. (2012 b) "Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network," *IEEE Transactions on Intelligent Transportation Systems*, 13: 1679–1693.

Hranac, R., E. Sterzin, D. Krechmer, H. Rakha, and M. Farzaneh. (2006) "Empirical Studies on Traffic Flow in Inclement Weather" Report FHWAHOP-07-073.

http://planet.openstreetmap.org/, accessed March 2016.

https://onlinecourses.science.psu.edu/stat501/node/358, accessed December 2016.

https://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based datasets/quality-controlled-local-climatological-data-qclcd, accessed for year 2015.

https://www.ncdc.noaa.gov/oa/climate/ghcn-daily/, accessed for the year 2015.

https://www.ncdc.noaa.gov/stormevents/, accessed for the year 2015.

https://www.openstreetmap.org, accessed March 2016.

https://wiki.openstreetmap.org/wiki/, accessed March 2016.

Huang, S., and Ran, B. (2003) "An Application of Neural Network on Traffic Speed Prediction Under Adverse Weather Condition," Presented at 82nd Annual Meeting of the Transportation Research Board, Washington, D.C.

Huang, W.H., Song G.J., Hong H.K, and Xie K.Q. (2014) “Deep Architecture for Traffic Flow Prediction: Deep Belief Networks with Multitask Learning.” IEEE Transactions on Intelligent Transportation Systems. 15(5):2191–201.

Hunter, T., Herring, R., Abbeel, P., Bayen, A. (2009). “Path and travel time inference from GPS probe vehicle data,” *Neural Information Processing Systems Foundation (NIPS) Conference*, Vancouver, Canada.

Hunter, T., Hofleitner, A., Reilly, J., Krichene, W., Thai, J., Kouvelas, A., Abbeel, P., and Bayen, A. (2013) “Arriving on time: Estimating travel time distributions on large-scale road networks,” *Technical Report arXiv*: 1302.6617, arXiv.

Jenelius, E., and Koutsopoulos, H. N. (2013) “Travel Time Estimation for Urban Road Networks Using Low Frequency Probe Vehicle Data,” *Transportation Research Part B*: Methodological 53: 64–81.

Jiang, X., and Adeli, H. (2005) "Dynamic wavelet neural network model for traffic flow forecasting," *Journal of Transportation Engineering*, vol. 131, pp. 771-779.

Kamarianakis, Y., and Prastacos, P. (2005) "Space–time modeling of traffic flow," *Computers & Geosciences*, vol. 31, pp. 119-133.

Karlaftis, M. G., and Vlahogianni, E. I. (2009) "Memory properties and fractional integration in transportation time-series," *Transportation Research Part C: Emerging Technologies*, vol. 17, pp. 444-453.

Khoei, A. M., Bhaskar, A., and Chung, E. (2013) “Travel time prediction on signalized urban arterials by applying SARIMA modelling on Bluetooth data,” In 36th Australasian Transport Research Forum (ATRF), Brisbane, Australia.

Kim, S., and Kim, J.-H. (2001) “Adaptive Fuzzy-Network-Based C-Measure Map Matching Algorithm for Car Navigation System,” *IEEE Transactions on Industrial Electronics*. 48(2): p. 432-441.

Koller, D., and Friedman, N. (2009). Probabilistic Graphical Models. Massachusetts: *MIT Press*. ISBN 0-262-01319-3.

Krumm, J., Letchner, J., and Horvitz, E. (2007) “Map Matching with Travel Time Constraints,” *in Society of Automotive Engineers (SAE) 2007 World Congress*: Detroit, Michigan, USA.

Kumar, S. V., Vanajaksh L. (2015) “Short-term traffic flow prediction using seasonal ARIMA model with limited input data,” *European Transport Research Review*, 7(21):1-9.

Lecture on Hausdorff and Frechet distance, accessed July 2017, https://www2.cs.duke.edu/courses/spring07/cps296.2/scribe_notes/lecture23.pdf.

Levin, M. and Tsao, Y.D. (1980) "On Forecasting Freeway Occupancies and Volumes (Abridgment)," *Transportation Research Record.*

Leshem, G., and Ritov, Y. a. (2007) "Traffic Flow Prediction using Adaboost Algorithm with Random Forests as a Weak Learner," International Journal of Intelligent Technology, vol. 2.

Markovic, N., Sekuła, P., Laan, Z. V., Andrienko, G., and Andrienko, N. (2017) “Applications of Trajectory Data in Transportation: Literature Review and Maryland Case Study”. *IEEE Transactions On Intelligent Transportation Systems,*

Min, W., and Wynter, L. (2011) "Real-time road traffic prediction with spatiotemporal correlations," *Transportation Research Part C*: Emerging Technologies, vol. 19, pp. 606-616.

Murphy, K.P. (2012) “Machine learning: a probabilistic perspective.” The MIT Press.

Myung, J., Kim, D. K., Kho, S. Y., and Park, C. H. (2011) "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.

Newson, P. and Krumm, J., (2009) “Hidden Markov Map Matching Through Noise and Sparseness,” Proceedings of the 17th ACM SIGSPATIAL *International Conference on Advances in Geographic Information Systems* (GIS).

Park, T., and Lee, S. (2004) “A Bayesian approach for estimating link travel time on urban arterial road network,” *Lecture Notes in Computer Science*, p. 1017, – 1025.

Peng, Y. Lei, M. Li, J. B., and Peng, X. Y. (2014) “A novel hybridization of echo state networks and multiplicative seasonal ARIMA model for mobile communication traffic series forecasting,” *Neural Computing and Applications*, vol. 24, no. 3-4, pp. 883–890.

Perry, R. W., & Greene, M. R. (1982). “The role of ethnicity in the emergency decision making process,” *National Emergency Training Center*.

Qiao, W., Haghani, A., Hamedi, M. (2012) “Short-Term Travel Time Prediction Considering the Effects of Weather,” *Transportation Research Record: Journal of the Transportation Research Board* 2308 (1), 61–72.

RITIS. (2017) The regional integrated transportation information system (RITIS). http://www.cattlab.umd.edu/?portfolio=ritis. Accessed: 2017.

Robinson, S., and Polak, J. (2005) "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.

Russell, S., and Norvig, P. (2009) “Artificial Intelligence: A Modern Approach,” *Pearson*; 3rd edition, p. 496.

Sharifi, E., Ernest, S. Y., Eshragh, S., Hamedi, M, Reuben ,M. J., and Kaushik,K. (2017) “Outsourced probe data effectiveness on signalized arterials,” *Journal of Intelligent Transportation Systems*, 21:6, 478-491

Smith, B., and Demetsky, M. (1997) "Traffic Flow Forecasting: Comparison of Modeling Approaches," *Journal of Transportation Engineering*, vol. 123, pp. 261-266.

Stathopoulos A., and Karlaftis, M. G. (2003)"A multivariate state space approach for urban traffic flow modeling and prediction," *Transportation Research Part C: Emerging Technologies*, vol. 11, pp. 121-135.

Sun, S., Zhang, C. and Yu, G. (2006) "A Bayesian network approach to traffic flow forecasting," *Intelligent Transportation Systems*, IEEE Transactions on, vol. 7, pp. 124-132.

Tang, L., Kan, Z. Zhang, X., Yang, X., Huang, F., and Li, Q. (2016) “Travel time estimation at intersections based on low-frequency spatial-temporal GPS trajectory big data”, *Cartography and Geographic Information Science*, 43:5, 417-426.

Telenav, http://www.telenav.com/products, accessed June 2017.

Tiesyte, D., and Jensen, C. S. (2008) “Similarity-based prediction of travel times for vehicles traveling on known routes,” *In Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM.

Tsirigotis, L., Vlahogianni, E. I., and Karlaftis, M. G. (2011) “Does Information on Weather Affect the Performance of Short-term Traffic Forecasting Models?” Presented at 90th Annual Meeting of the Transportation Research Board, Washington, D.C.

Veltkamp, R.C., (2001) “Shape matching: similarity measures and algorithms. in Shape Modeling and Applications”, SMI 2001 International Conference.

Vlahogianni, E. I., Golias, J. C., Karlaftis, M. G. (2004) "Short- term traffic forecasting: Overview of objectives and methods," *Transport Reviews*, vol. 24, pp. 533-557.

Westgate, B. S., Woodard, D. B., Matteson, D. S., and Henderson, S. G. (2016) “Large-network travel time distribution estimation, with application to ambulance fleet management,” *European Journal of Operational Research*.

Westgate, B. S., Woodard, D. B., Matteson, D. S., and Henderson, S. G. (2013) “Travel time estimation for ambulances using Bayesian data augmentation,” *Annals of Applied Statistics*, 7:1139–1161.

Williams, B. M., Durvasula, P. K., and Brown, D. E. (1998) "Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transportation Research 158 Record: Journal of the Transportation Research Board*, vol. 1644, pp. 132-141.

Williams, B.M. and Hoel, L.A. (2003) "Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results," *Journal of Transportation Engineering*, 129(6), 664–672.

Woodard, D. B., Nogin, G., Koch, P., Racz, D., Goldszmidt, M, and Horvitz, E. (2017). "Predicting travel time reliability using mobile phone GPS data," *Transportation Research Part C*, 75: 30-44. Pdf

Yang, Q., Wu, G., Boriboonsomsin, K., & Barth, M. (2017). "A novel arterial travel time distribution estimation model and its application to energy/emissions estimation," *Journal of Intelligent Transportation Systems*, 1–13.

Yin, H., Wong, S., Xu, J., and Wong, C. (2002) "Urban traffic flow prediction using a fuzzy-neural approach," *Transportation Research Part C: Emerging Technologies*, vol. 10, pp. 85-98.

Yu, B. Yang, Z. Z. and Chen, K. (2010) "Hybrid model for prediction of bus arrival times at next station," *Journal of Advanced Transportation*, vol. 44, pp. 193-204.

Yu, R., Li, Y., Shahabi, C., Demiryurek, U., and Liu, Y. (2017) "Deep learning: A generic approach for extreme condition traffic forecasting". In Proceedings of the

SIAM International Conference on Data Mining (SDM), Houston, TE, USA, 27–29.

Zeng, X., and Zhang, Y. (2013) "Development of Recurrent Neural Network Considering Temporal-Spatial Input Dynamics for Freeway Travel Time Modeling," *Computer-Aided Civil and Infrastructure Engineering*.

Zhang, Y. (2015) “Uncertainty Associated with Travel Time Prediction: Advanced Volatility Approaches and Ensemble Methods,” PhD thesis, University of Maryland, 789 East Eisenhower Parkway.