

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

A Deep Learning, Model-Predictive Approach to Neighborhood Congestion Prediction and Control

Permalink

<https://escholarship.org/uc/item/6p05k3pt>

Author

Mohanty, Sudatta

Publication Date

2018

Peer reviewed|Thesis/dissertation

**A Deep Learning, Model-Predictive Approach to Neighborhood
Congestion Prediction and Control**

By
Sudatta Mohanty

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Civil and Environmental Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Mike Cassidy (Co-Chair)
Assistant Professor Alexey Pozdnukhov (Co-Chair)
Professor Mark Hansen
Professor Philip Stark

Fall 2018

A Deep Learning, Model-Predictive Approach To Neighborhood Congestion Prediction And Control

Copyright © 2018

by

Sudatta Mohanty

All rights reserved

Abstract

A Deep Learning, Model-Predictive Approach to Neighborhood Congestion Prediction and Control

by

Sudatta Mohanty

Doctor of Philosophy in Engineering - Civil and Environmental Engineering

University of California, Berkeley

Prof. Mike Cassidy (Co-Chair)

Prof. Alexey Pozdnukhov (Co-Chair)

Traffic congestion is a major concern, especially in large cities. To relieve a city of congestion impacts, transportation authorities typically base controls such as toll or congestion prices on expected congestion patterns. However, such strategies can be sub-optimal and may also lead to unintended negative consequences for two main reasons: (i) estimates of congestion state may be inaccurate if the proposed model inputs are too limiting or the assumptions are too restrictive and (ii) predictions may be too late to help. This creates the need for accurate predictions of traffic congestion well ahead of time to avoid delays and gridlocks.

Traffic congestion is analyzed as a network-wide phenomenon. Large-scale spatial correlation and long-term temporal correlation govern traffic congestion propagation across the regional traffic network. Such correlations may be exploited to develop congestion prediction algorithms that are more effective than purely local predictions for the purpose of dynamic controls. Moreover, real-time information that is generated at locations across the network may signal the future congestion state, making it possible to take control measures in advance of worsening traffic situations. However, this approach also presents several new challenges, addressed in this research.

Microscopic models of congestion do not produce realistic representations of network-wide congestion generation and its propagation. This motivates the analysis of congestion at neighborhood scale through macroscopic analysis. Recent literature has shown that Macroscopic Fundamental Diagrams (MFDs) are effective for developing neighborhood-wide congestion controls by controlling inflows from immediately surrounding areas or managing signal timings, in combination with flow conservation laws. To further enhance the use of MFDs for neighborhood-wide congestion management, traffic prediction over much larger geographic scales is incorporated. To that end, a numerically well-behaved score function, called Macroscopic Congestion Level (MCL), is proposed. The score is defined to be the ratio of the neighborhood's vehicle accumulation, to its trip completion rate. Future values of this score are then predicted through a model using network state characteristics over the larger, region-wide network as input. These characteristics are

represented as a vector of Origin-Destination (O-D) demands, link accumulations, link travel times and observed MCL values. The predicted score can be used to describe the likely congestion state in a neighborhood in the near future.

It is challenging to develop congestion prediction algorithms that incorporate both spatial and temporal dependence at a network scale, adapt to sudden changes in demand patterns and forecast accurately over sufficiently long periods to implement controls. Deep learning is used to build sufficiently complex models for this task. Predictions are made using a deep learning model based on Long Short-Term Memory (LSTM) neural network architecture. The ideas are tested using simulation on a simple, hypothetical city-street network. A battery of simulation tests suggests that the model inputs are sensitive to different queue propagation scenarios, within and across days. MCL predictions made by the proposed deep learning model outperform a simple, yet competent, baseline model that assumes congestion on the next day mimics congestion today (referred to as the 1-Nearest Neighbor or 1-NN model).

The prediction accuracy of the deep learning model is compared to that of the baseline 1-NN model in three situations that either aid the baseline model or adversely impact the deep learning model: (i) correlated O-D demands across multiple days, (ii) noisy observations, and (iii) partially observed network state. The limitations to the prediction accuracy in these situations are studied in simulated scenarios. Methods are suggested to improve accuracy in these situations either by modifying certain model hyper-parameters or by understanding the importance of various inputs.

A Neural Attention Model-based framework is developed to extract the importance of various inputs and to better understand the inner workings of the deep learning model. Simulation experiments suggest that such a framework allows identification of major congestion-causing factors that may be targeted during control.

Model predictions and the importance of various inputs are then used to test a dynamic control strategy, namely an app-based dynamic congestion toll at the beginning of a trip. Individuals are charged a toll at the beginning of each trip based on their predicted congestion impact. This dissuades them from traveling during the time and along the routes where they are likely to cause major harm to the performance of the network. An optimization problem is formulated to minimize cumulative MCL across a day in a target neighborhood while maintaining overall travel demand. The conditions for optimal tolls are obtained and approximations for these conditions are proposed through learned deep learning model parameters. Simulations indicate that a deep learning model-based dynamic toll reduces delays and charges lower toll than other tolling strategies that depend on predictions of demand.

Possible improvements to the architecture of the deep learning model are discussed for congestion prediction in large networks. Signals in large networks are assumed to propagate through hypothetical graphs, such as graphs representing the road network or graphs representing the similarities in route-choices made by various individuals. The inputs are transformed to impose a graphical structure on them. A Graph Convolutional Neural Network (Graph-CNN) architecture is implemented for extracting relevant spatial features from this graphical input. An LSTM model makes real-time predictions based on

these extracted features. Simulations of commuter trips on a pared-down freeway/highway network as well as full-scale network representing the San Francisco Bay Area suggest that the model accuracy is superior to that of the 1-NN model, the LSTM-only model, and a Graph-CNN + LSTM model without any road network or route-choice information.

To Mums, Paps, Miks, and Vanya

Contents

1	Introduction	1
1.1	Background	1
1.2	Challenges of Predicting Congestion	2
1.3	Possible Solutions to Challenges	3
1.3.1	Macroscopic Models of Congestion	3
1.3.2	Deep Learning for Congestion Prediction	4
1.3.3	Neural Attention for Model Interpretation	4
1.4	Proposed Research Pipeline	5
2	Literature Review	9
2.1	Brief History of Traffic Congestion Modeling	9
2.2	Brief History of the Macroscopic Fundamental Diagram (MFD)	10
2.3	Traffic Prediction Models	11
2.4	Model Robustness	13
2.5	Interpreting Traffic Prediction Models	14
3	Modeling Framework	16
3.1	Macroscopic Congestion Level	16
3.2	Network State Definition	17
3.3	Deep Learning Framework for MCL prediction	18
4	Viability Conditions	19
4.1	Sensitivity Analysis	19
4.2	Initial Comparisons	31
5	Model Robustness	35
5.1	Correlated O-D Demand Across Days	35
5.2	Noisy Observations	38
5.3	Partially Observed Network State	40
6	Model Application - Dynamic Congestion Control	42
6.1	Neural Attention Model	42
6.2	App-Based Dynamic Congestion Pricing	47
7	Improvements through Graph-based Feature Learning	55
7.1	Graph Transformation of Input	56
7.2	Graph CNN + LSTM Model Formulation	57
7.3	Experimental Setup	58
7.4	Experimental Results	59

8	Conclusions	62
8.1	Key Contributions	62
8.2	Future Work	64
	Appendices	78
	Appendix A Dynamic Origin-Destination Demand Estimation	79
A.1	Problem Overview	79
A.2	Problem Formulation	80
A.2.1	Traditional Approach	80
A.2.2	Proposed Approach	82
A.2.3	Overall Problem Formulation	85
A.2.4	Proposed Algorithm	85
A.2.5	Experimental Results	86
	Appendix B Steps Executed By A Typical Long Short-Term Memory (LSTM) Model	92
	Appendix C Model Robustness - Correlated demand start time and demand slope	93
	Appendix D Model Robustness - Partially observed input variables for large network	96
	Appendix E Model Application - Neural Attention Model performance for large network	97

List of Figures

1.1	Research Pipeline	5
2.1	Macroscopic Fundamental Diagram (MFD)/Network Exit Function (NEF): It represents the correlation between trip completion rate and accumula- tion in a well-defined zone with steady-state traffic. The maximum trip completion rate is μ and the maximum accumulation is η . $\bar{v}$ is a measure of the throughput of trips. Macroscopic Congestion Level is defined as $\frac{1}{\bar{v}}$.	10
4.1	Figures describing Scenario 1 and the effectiveness of MCL and network state for modeling neighborhood congestion.	22
4.2	Figures describing Scenario 2 and the effectiveness of MCL and network state for modeling neighborhood congestion.	23
4.3	Figures describing Scenario 3 and the effectiveness of MCL and network state for modeling neighborhood congestion.	25
4.4	Figures describing Scenario 4 and the effectiveness of MCL and network state for modeling neighborhood congestion.	27
4.5	Figures describing Scenario 5 and the effectiveness of MCL and network state for modeling neighborhood congestion.	29
4.6	Figures describing Scenario 6 and the effectiveness of MCL and network state for modeling neighborhood congestion.	31
5.1	Efficiency bounds for the prediction accuracy of the LSTM model vs the 1-NN model as a function of the amount of correlation in start times of demand and the amount of statistical fluctuation. The points labeled “+” are parameter values for which the LSTM model performed better and the points labeled “o” are parameter values for which 1-NN model performed better. A possible efficiency boundary was constructed using a Decision Tree Classifier.	37
5.2	Efficiency bounds for the prediction accuracy of the LSTM model vs the 1-NN model as a function of the amount of correlation in the slope of demand and the amount of statistical fluctuation in the slope. The points labeled “+” are parameter values for which the LSTM model performed better and the points labeled “o” are parameter values for which 1-NN model performed better. To illustrate the boundary, a Decision Tree (DT) classifier model was fit on the points.	37
5.3	Impact of noise in input data on the prediction accuracy of the LSTM model for different batch sizes	39
6.1	Temporal attention heatmap	45

6.2	Spatial attention heatmap - Variable definitions: Index 0 = Zone a demand, Index 1 = Zone b demand, Index 2 = Zone d demand, Index 3 = First difference of Zone a demand, Index 4 = First difference of Zone b demand, Index 5 = First difference of Zone d demand	46
6.3	Overview of setting for dynamic congestion pricing at source: Agents starting their trips at Zone z_0 at time t_0 are charged congestion toll in proportion to the predicted marginal impact on congestion during their trip. For this example, we consider only the predicted marginal impact on congestion in Zone z_d due to this trip. The agents must display the payment of this toll when they reach Zone z_d at time t_d	47
6.4	Change in the cumulative curve of departing agents under implementation of three toll strategies: (a) Monotonically decreasing toll curve, (b) Triangularly shaped toll curve and (c) Monotonically increasing toll curve. In all cases, it is assumed that the slope of the toll curve is a very small amount ϵ greater than the value of time (\$/min)	51
6.5	(a) No toll: Plot showing demand from the three input zones (i.e., a (yellow), b (green) and d (red), values to be read from the primary y-axis) and the the MCL in Zone c (blue curve, values to be read from the secondary y-axis). Cumulative congestion delay = 59246 min^2 . Average travel time per commuter = 205.93 min . (b) Optimal Toll Using O-D demand: Plot showing demand from the three input zones (i.e., a (yellow), b (green) and d (red), values to be read from the primary y-axis) and the the MCL in Zone c (blue curve, values to be read from the secondary y-axis). Cumulative congestion delay = 23305 min^2 . Average travel time per commuter = 63.74 min	53
6.6	Plot showing variation of demands from Zone a (“.”), Zone b (“-.”) and Zone d (“-”) and resulting macroscopic congestion level in Zone c (solid) as a function of time for - (a) No toll, (b) Average demand-based toll, (c) 1-NN demand-based toll, (d) LSTM model-based toll and (e) Known future demand-based toll	54
7.1	(a) Simplified freeway network and (b) Full-scale network - representing 9 counties of the SF Bay Area. Each region was partitioned into zones. Target zones for MCL prediction are highlighted in yellow.	58
A.1	Figure: Problem Overview For Dynamic Origin-Destination Demand Estimation - The goal is to estimate the number of individuals departing Zone z and traveling towards Zone z' during time interval t , defined as $[D]_{zz'}(t)$	79
A.2	Variation of RMSE vs $\log(\nu)$ for various values of σ (ν as defined in Equation (A.13) and σ as defined in Equation (A.14))	87
A.3	Variation of RMSE vs time of day for $\nu = 1000$ and $\sigma = 0.5$	87
A.4	Variation of RMSE over O-D pairs averaged over entire day for $\nu = 1000$ and $\sigma = 0.5$	89
A.5	Change in the RMSE for the O-D estimate across all zones with incoming information for $\nu = 1000$ and $\sigma = 0.5$ - (a) RMSE at $t = 540$ mins with updates between $t = 540$ mins and $t = 715$ mins every 35 mins, (b) RMSE at $t = 545$ mins with updates between $t = 545$ mins and $t = 720$ mins every 35 mins	90

A.6	Change in the RMSE for the O-D estimate across all zones during peak hours with varying cell phone data coverage	90
A.7	Change in the average RMSE for the O-D estimate across all zones during peak hours with varying noise in link choice probability matrix $P(t', t)$	91
C.1	Efficient bounds for prediction accuracy LSTM model vs the 1-NN model as a function of the amount of correlation in both start time and slope of demand and amount of statistical fluctuation. The points labeled “+” are parameter values for which the LSTM model performed better and the points labeled “o” are parameter values for which 1-NN model performed better. To illustrate the boundary, Decision Tree (DT) classifier model was fit on the points.	95

List of Tables

4.1	A table summarizing the necessary conditions (i.e., physical processes that may affect MCL in the target zone, $\xi^z(t)$) and the components of network state, $X^z(t)$ that is responsive to the occurrence of a particular physical process	31
4.2	A table describing the parameter set for each LSTM model developed and comparison of Root Mean Squared Error (RMSE) for the LSTM model and the 1-NN model	34
5.1	Root Mean Squared For LSTM model for various scenarios with omitted inputs and different prediction times	40
6.1	Cumulative congestion, average travel time and average toll for various toll scenarios	53
7.1	Parameters of Gaussian distributions for the start time and duration of h-w and w-h trips generated for testing the proposed Graph CNN + LSTM model	59
7.2	A table describing the relative performance of 1-NN, Holt-Winters model, LSTM-only, Graph CNN + LSTM (k-NN Euclidean Graph), Graph CNN + LSTM (Shortest Path Graph) and Graph CNN + LSTM (k-NN Trajectory Clustering Graph) for the prediction of $\hat{\xi}^z(t)$ in target zones highlighted in Figure 7.1(b) and demand scenarios described in Table 7.1 . . .	61
D.1	Root Mean Squared error of LSTM predictions in a large network in various scenarios with omitted variables.	96
E.1	Top four O-D demand attentions and the corresponding percentage change in MCL value if the O-D was absent, while predicting MCL in the Zone 2 between 10AM-12PM.	97

Acknowledgment

The journey through my Ph.D. has been a roller coaster ride, with several ups and downs along the way. But at the end of the day, this has been one of the most fulfilling experiences ever. Conceiving an idea, evolving it through literature review, passionate discussions and countless experiments and then finally successfully presenting the idea in front of experts in the field - this is exactly why I wanted to do a Ph.D.!

I would first like to thank my parents and brother for supporting me through every important decision of my life. My dad is my north star, constantly guiding me through difficult times and the first to sparkle with joy at any success that I achieve. My mom has taught me that no feat is impossible if you show sufficient determination and hard work.

I would like to thank my advisers Prof. Mike Cassidy and Alexey Pozdnukhov who have endowed me with the requisite skills to undertake this task. I would also like to thank my committee members, Prof. Mark Hansen and Prof. Philip Stark, whose feedback has been essential towards getting meaningful results. I would also like to thank all the faculty in the TE department for the pleasure of learning from the best people in the field. I would also have not been able to achieve what I have without the help and support from our graduate advisor, Shelly Okimoto, and administrators, Bernedette, Kendra, and Jean. Finally, I would like to thank my dear friends and colleagues - Mogeng, Andrew, Sid, Timothy, Wei, Max, Emin, Danqing, Maddie, Andy, Sreeta, Feras, Alan, Darren, Mustafa, and Mengqiao. Our conversations always kept me motivated and brought so much life into McLaughlin.

I would like to thank Ryan Herring and Christian Sommer for two amazing internships at Apple during my degrees at Berkeley. They have been the most awesome mentors outside college and the best friend-cum-guides that I could have ever desired. I am also thankful to Dr. Venkat Padmanabhan for a lovely time interning at Microsoft Research in Bangalore. I am grateful to Alex Erath at ETH Zurich for developing my interest towards Transportation Engineering during my internship at Future Cities Lab, Singapore and Prof. Geetam Tiwari and Prof. Dinesh Mohan at IIT Delhi who kept the spark in me going throughout my undergrad. I also thank my friends, Anubhav, Akash Rathi, Vibhav, Aditya, Nikunj, Navneet, and Anuj, who ensured that my undergrad was an absolute blast.

Last but definitely not the least, I would like to thank my beautiful, strong and supportive girlfriend Vanya for standing by me at all my difficult times. Without your emotional support and love, I would never be where I am today.

Chapter 1

Introduction

1.1 Background

Congestion can be defined as a negative consequence of an imbalance between traffic demand and capacity which affects users, facilities, and network performance. Congestion costs include *internal costs* such as driver time delay and excess fuel consumption, *external costs* such as infrastructure deterioration and sub-optimal network performance (Litman 2009) as well as *social costs* such as driver stress, which can have a negative impact on the quality of life (Hennessy and Wiesenthal 1999). Congestion in the US has grown rapidly of late. The average delay per commuter per year has risen from 16 hours in 1982 to 38 hours in 2011. This translates to an average cost of \$818 per commuter in 2011, up from \$342 per commuter in 1982. The aggregate direct and indirect cost per year is close to \$300 billion. The total CO_2 emission due to congestion has risen from 10 billion pounds in 1982 to 56 billion pounds in 2011 (Schrank, Eisele, and Lomax 2012). Two classes of strategies have been proposed to mitigate congestion: (i) increasing infrastructure supply and (ii) implementing controls to manage demand. Recent evidence shows that controls have proven far more effective in mitigating congestion effects than infrastructure growth has. The last decade has seen some decline in aggregate numbers for congestion impacts even though the aggregate number of vehicles has increased at a faster rate than infrastructure capacity growth during the same period (Hadley and Tsvetkova 2009). This reduction in congestion impact can be attributed, in large part, to better controls (Schrank, Eisele, and Lomax 2012). The success of such strategies provides hope that there is a tremendous scope for further improvement through smarter controls. One way to facilitate this improvement is to use predictions. Traffic controls can reduce congestion more effectively by incorporating short or medium-term traffic predictions (Van den Berg et al. 2007; Aboudolas, Papageorgiou, and Kosmatopoulos 2009; Aboudolas, Papageorgiou, Kouvelas, et al. 2010). However, most predictive controls focus on link-level or signal-level controls only (Abdulhai, Pringle, and Karakoulas 2003; Prashanth and Bhatnagar 2011). Recent work has shown that neighborhood-wide controls may benefit from predictions of flow in and around the region of interest (Ramezani, Haddad, and Geroliminis 2015, Ni and Cassidy 2018). This research develops a framework for predictive and dynamic controls at even larger geographic scales (for example, a city-wide scale or even a few neighboring cities) and demonstrates its potential through simulations.

1.2 Challenges of Predicting Congestion

Congestion prediction remains challenging even though congestion patterns are generally periodic in nature (Skabardonis, Varaiya, and Petty 2003). The difficulty arises because traffic may deviate from regular patterns due to factors such as inclement weather (Al Hassan and Barker 1999; Smith, Byrne, et al. 2004; Chung et al. 2005; Cools, Moons, and Wets 2010; Xu, He, et al. 2013), accidents (Ceder 1982; Golob and Recker 2003), special events (Wojtowicz and Wallace 2010; Kwoczek, Di Martino, and Nejd1 2014), and statutory holidays (Liu and Sharma 2006). Moreover, in oversaturated networks, the evolution of traffic is highly chaotic with several hidden relationships. For example, route-choice is not straightforward because rational drivers anticipate congestion propagation and change their behavior accordingly (Addison and Heydecker 1993). The situation is further complicated by the fact that driving strategies that maximize individual gain may negatively affect the overall network (Arnott and Small 1994). Even small fluctuations in traffic demand may lead to large fluctuations in the resulting congestion impacts (Daganzo 1996, 1998). This necessitates models that *memorize* important recurring traffic characteristics, yet accommodate the possibility of both structured and unstructured deviations detected from recent data.

Some other challenges in developing traffic prediction models that may be used for dynamic controls are: (i) ensuring responsiveness to sudden changes in traffic demand, (ii) incorporating both spatial and temporal signals for predicting the propagation of congestion, (iii) evaluating the network-wide impact of congestion, and (iv) making predictions sufficiently in advance for meaningful controls to be implemented (Vlahogianni, Karlaftis, and Golias 2014). These requirements often compete with each other. A technique for selecting the appropriate model must not only compare prediction accuracy but also evaluate the properties of the error under various traffic scenarios to ensure that there are few unexpected negative consequences (Chen, Wang, et al. 2012; Vlahogianni and Karlaftis 2013). Due to inadequacies in satisfying all above requirements, traffic prediction models have yet to gain popularity as a tool for real-time congestion mitigation. This leads us to develop a framework that not only predicts congestion accurately, but is also sensitive to the underlying physical processes causing congestion.

Real-world data that is often used as an input to these models, such as data from loop detectors, Global Positioning Systems (GPS) or cell phones, can often become corrupted or unusable at certain locations and/or at certain times (Chen, Kwon, et al. 2003; Tong, Merry, and Coifman 2005; Van Lint and Hoogendoorn 2010; Bar-Gera 2007). Therefore, congestion prediction algorithms that use these data sources must be robust to such dropouts. Some data sources can have measurement-noise. Hence, congestion prediction algorithms must be robust to such noise too (Thiagarajan et al. 2009). Yet another challenge is to build models that are interpretable. This is important for two reasons: (i) it guards against over-fitting the data and (ii) it guides dynamic controls based on the predictions made by the model.

Congestion must be represented in a way that facilitates its prediction over a wide area (one where roads form a large network). Past literature on network-wide congestion representation can be divided into two categories: (i) traffic flow theory-based representation (i.e., microscopic congestion models) and (ii) travel demand model-based representa-

tion (i.e., modeling of aggregate traffic demand and choices made by individuals). Both these representations have limitations. Microscopic models based on traffic flow theory (Lighthill and Whitham 1955; Richards 1956; Newell 1993; Daganzo 1994) represent congestion in a network as an aggregation of link-level queues that propagate over space and time. The major limitations to this representation are: (a) the poor modeling of merging/diverging traffic, (Peeta and Ziliaskopoulos 2001) and (b) not accounting for the heterogeneity in the behavior of different agents in the network. The most popular travel demand models for representing congestion in a network are activity-based travel demand models (ABMs) (Bhat and Koppelman 1999). These models represent congestion as a consequence of individuals traveling to perform activities that leads to traffic demand exceeding the infrastructure capacity at certain locations and times. The key challenge to this representation is the requirement of extensive individual-level data to substantially replicate population behavior. Usually, such models aim to replicate traffic on *typical days* (Tarasov, Kling, and Pozdnoukhov 2013; Coffey and Pozdnoukhov 2013; Wu, Thai, et al. 2015; Yin et al. 2017) and are not sensitive to deviations from recurrent conditions. Generalizing these models to multiple days with varying demand patterns would require a much richer dataset over a very long time period. Such datasets are hard to obtain. In the absence of such rich datasets, correlations in traffic demands over multiple days have to be approximated and the quality of the approximation would significantly affect the model accuracy.

1.3 Possible Solutions to Challenges

1.3.1 Macroscopic Models of Congestion

Macroscopic models of traffic use aggregate characteristics and provide several advantages over microscopic models for network-level congestion representation. These models avoid the requirement of complicated link-level or agent-level information in real-time (Daganzo 2007), allow effective and tractable modeling of merging/diverging traffic (Haddad and Geroliminis 2012; Geroliminis, Haddad, and Ramezani 2013; Hajiahmadi, Haddad, et al. 2015), and may be adjusted to incorporate route-choice heterogeneity (Leclercq and Geroliminis 2013). Tracking aggregate characteristics over time is computationally cheaper than tracking disaggregated information at the same scale. Hence, macroscopic models can analyze congestion over multiple days without requiring rich datasets over very long time periods. These properties are well supported by the Macroscopic Fundamental Diagram (MFD). It evaluates neighborhood-wide congestion state and aids neighborhood-wide controls (Daganzo 2007). When traffic is homogeneous and the demand changes are slow, the congestion performance of a neighborhood can be measured in terms of aggregate characteristics. These correlations exist in real-world networks (Geroliminis and Daganzo 2008). We can also avoid simplifying assumptions about traffic demand across multiple days. Due to these factors, a meaningful representation of network-wide congestion is possible through MFDs. Several hidden spatio-temporal relationships guide congestion propagation. Therefore, a prediction model must be able to discover such relationships. The following section provides the intuition behind a framework that is capable of handling such complexities.

1.3.2 Deep Learning for Congestion Prediction

A deep learning model that uses neural networks for predicting congestion could address the modeling challenges discussed in section 1.2. Neural networks have shown superior predictive power than traditional parametric models while predicting traffic in several complex settings (Miles and Walker 2006; Van Lint and Van Hinsbergen 2012; Vlahogianni, Karlaftis, and Golias 2014). Neural networks also have the ability to extract relevant features for prediction from a high volume and dimensionality of input data. This property is critical for network-wide predictions since the number and size of inputs can be very large. Recurrent Neural Network (RNN) architectures, which are typically used for time-series predictions, can be made robust to noise in the inputs through techniques such as reducing the batch size (Giles, Lawrence, and Tsoi 2001). The layered model structure for neural networks mirrors the spatio-temporal evolution of congestion in a road network. The ability to discover hidden relationships through feature-learning allows building models that are sensitive to factors that are not directly observable. Therefore, it is possible to make models that are sensitive to route-choice behavior and queue spillover. A special type of RNN architecture called Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) can store long-term recurrence patterns that are common in traffic (Vlahogianni, Karlaftis, and Golias 2005). Applications such as visual recognition and image captioning have shown that another special neural network architecture, called Convolutional Neural Network (CNN), can be used as an *encoders* to extract spatial features that an LSTM can use for making dynamic predictions (Vinyals et al. 2015). This kind of a hybrid architecture allows the models to extract complicated and possibly hidden spatial and temporal features simultaneously (Donahue et al. 2015). Although a CNN-LSTM architecture is extremely powerful, it is readily applicable only for grid input data formats (i.e., the data points must occur regularly in the Euclidean space) such as speech (Hinton et al. 2012), images (LeCun et al. 1998; Krizhevsky, Sutskever, and Hinton 2012) or videos (Taylor et al. 2010; Le et al. 2011). Unfortunately, traffic data doesn’t typically display this property. A possible solution to this problem is to represent the traffic data as signals on a graph and then perform analysis through graph signal-processing (Shuman et al. 2013). This technique for generalizing traditional CNNs on arbitrarily structured graphs and manifolds is formally called Graph-CNN (Bruna et al. 2013). The above-mentioned properties of neural networks allow building flexible models for network-level congestion prediction.

1.3.3 Neural Attention for Model Interpretation

As mentioned in section 1.2, a traffic prediction algorithm should not be judged purely by prediction accuracy. A deeper understanding is desirable as to whether the algorithm represents the underlying physical processes involved. In past literature on traffic prediction, this understanding was achieved in two ways. Deterministic models derived by approximating the physical processes were analyzed with the help of closed-form solutions. Probabilistic models that approximately estimate the distributions of certain parameters related to the physical process were analyzed by studying the magnitude, sign and/or behavior of the estimated parameters. However, such approaches are infeasible for interpreting neural networks. It is impractical to analyze all physical processes that govern the outputs from neural network models in complex settings. Moreover, the parameter space of such models might be too large to analyze individually. Neural networks have been criticized for being non-interpretable “black-boxes” in the traffic prediction liter-

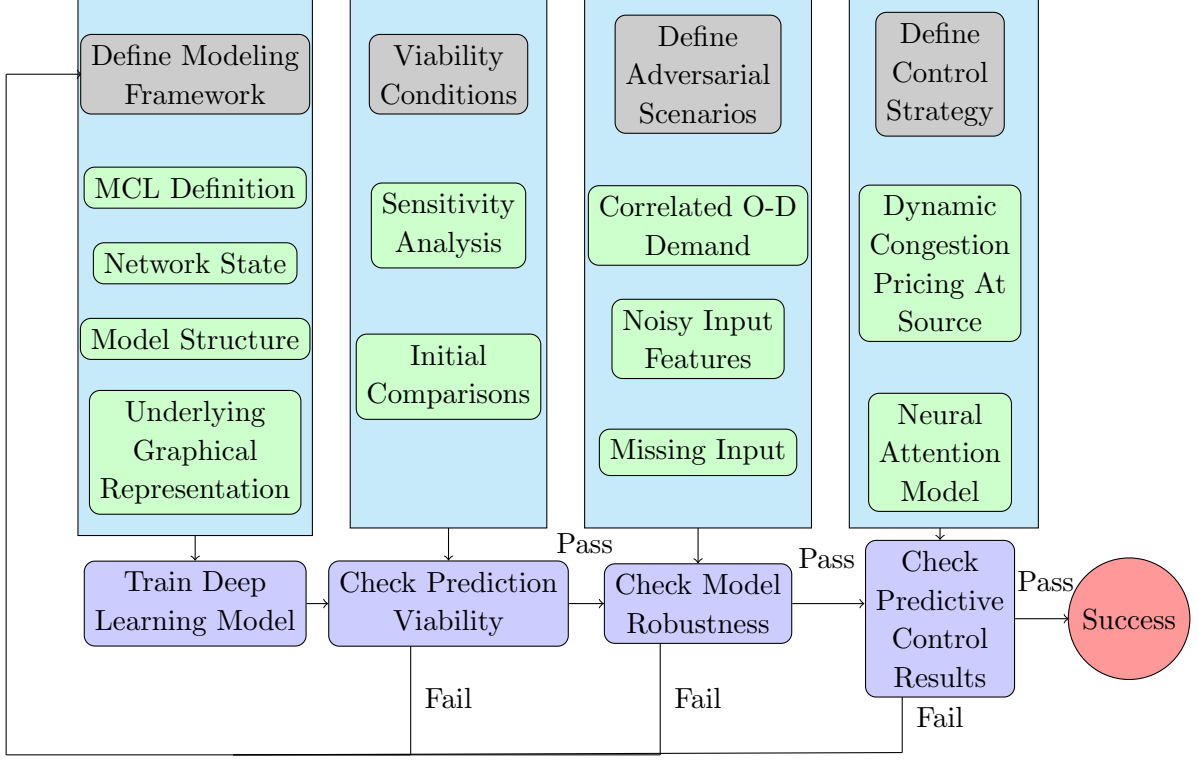


Figure 1.1: Research Pipeline

ature (Smith and Demetsky 1994; Van Lint, Hoogendoorn, and Zuylen 2005; Liu and Sharma 2006; Moayedi and Masnadi-Shirazi 2008). Due to this shortcoming, the direct application of such models to dynamic controls has proven limited. A possible solution is to use *Neural Attention Models* (Fukushima 1987; Tang, Srivastava, and Salakhutdinov 2014; Xu, Ba, et al. 2015; Ramanishka et al. 2017). These models may be employed for discovering the relative spatio-temporal importance of various inputs. Knowledge of the importance of various inputs also provides an intuition behind the cause for congestion. This approach allows developing a better understanding of a neural network model.

1.4 Proposed Research Pipeline

The overview of the proposed research pipeline is described in Figure 1.1. The pipeline is divided into a series of four components, one of which defines the proposed model and three others that check various properties of this model through simulations. Each component is represented through a light blue rectangular box and its outcome is represented through a purple box. A grey box provides an overview of the steps performed. Each green box represents a step in the process. A detailed description of each component is as follows:

(i) Defining Modeling Framework

The modeling framework includes the definition of the inputs to the model, the output predicted by the model and the structure of the model. The inputs are collectively referred to as the network state. The output is defined through a scoring function called Macroscopic Congestion Level (MCL). The proposed model is a deep learning model based on Long Short-Term Memory (LSTM) architecture.

Improvements to the model are proposed by representing the input as signals on a graph. The underlying graphs through which the signals propagate are constructed through the knowledge about the structure of the road network and the routes taken previously by individuals.

(ii) Conditions for a Viable Model

Two conditions are defined to check whether a proposed model is suitable for congestion prediction. These conditions are tested in simulated settings. The first condition checks whether the model inputs are sensitive to various factors that might influence congestion and whether the predicted output is a good representation for congestion in a region. The checks are performed through sensitivity analysis on six hypothetical queue propagation scenarios on an idealized network and five hypothetical scenarios for variation in demand across multiple days. The second condition checks whether the model prediction accuracy is better than that of a simple, yet competent, baseline. The baseline model copies congestion scores observed from the previous day to make predictions and this model is referred to as 1-Nearest Neighbor (1-NN) model. Comparisons are performed in the same set of simulated settings as for the first condition. Results suggest that the proposed deep learning model satisfies both conditions.

(iii) Robustness Against Adversarial Scenarios

Three situations are proposed that either aid the baseline model or adversely impact the deep learning model: (i) correlated demands across multiple days, (ii) noisy observations, and (iii) partially observed network state. The comparisons made in simulated settings are repeated, but with an increasing amount of adversity in each case. The threshold robustness parameter values are determined for which the deep learning model outperforms the baseline 1-NN model. Methods to modify the hyper-parameters for the deep learning model are suggested basing on the simulated results. These methods can help the deep learning model outperform the baseline model when the amount of adversity is high.

(iv) Predictive Control Strategy Performance

The applicability of the proposed model for predictive control is demonstrated through a simulation experiment. A control strategy is defined, namely app-based dynamic congestion pricing at the beginning of each trip. The optimal toll conditions are determined to mitigate congestion while preserving overall demand. The proposed deep learning model for predictions and a Neural Attention Model (which also helps to interpret the deep learning model in a better way) are used to approximate the optimal toll conditions. Simulation results on an idealized scenario indicate a reduction in delays and lower toll rates for the deep learning model-based approximation as compared to other naive tolling strategies that depend on predicting traffic demand.

The organization of these components into chapters is as follows: Chapter 2 covers the background literature relevant to the topic. Chapter 3 gives the structure of the proposed deep learning model. Chapter 4 discusses the conditions to ensure viability of the model. These conditions are tested in several idealized simulated scenarios. Chapter 5 illustrates the conditions to test the model robustness. Tests in simulated scenarios are continued with varying amount of adversity faced by the deep learning model. Chapter 6 describes a

Neural Attention Model-based framework to better understand the deep learning model. In the same chapter, model predictions and attentions are used to approximate a dynamic congestion toll. Comparisons are performed between the deep learning model-based toll and other naive tolling strategies in an idealized and simulated scenario. In chapter 7, enhancements to the deep learning model are proposed to improve its performance in large networks. Inputs to the deep learning models are treated as signals on hypothetical graphs. A Graph-CNN+LSTM framework is proposed to use these signals and make predictions of congestion in a neighborhood. The prediction accuracy for this model is compared with several baselines in simulated settings that represent commute traffic in the San Francisco Bay Area.

List of Abbreviations Used

ARIMA	Auto-Regressive Integrated Moving Average
ARIMAX	Transfer functions with ARIMA errors
ATIS	Advanced Traveler Information System
CBD	Central Business District
CNN	Convolutional Neural Network
CTM	Cell Transmission Model
DT	Decision Tree
DTW	Dynamic Time Warping
DWT	Discrete Wavelet Transform
FC	Fully Connected
GPS	Global Positioning System
Graph-CNN	Graph Convolutional Neural network
I/O	Input-Output
ITS	Intelligent Transportation Systems
k-NN	k-Nearest Neighbor (where k is a natural number)
LOS	Level Of Service
LSTM	Long Short-Term Memory
MCL	Macroscopic Congestion Level
MDP	Markov Decision Process
MFD	Macroscopic Fundamental Diagram
MLP	Multi-Layered Perceptron
MPC	Model Predictive Control
NEF	Network Exit Function
O-D	Origin-Destination
PDE	Partial Differential Equation
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RMSE	Root Mean Squared Error
Seq2Seq	Sequence-to-Sequence

Chapter 2

Literature Review

2.1 Brief History of Traffic Congestion Modeling

The physical processes governing generation and propagation of traffic congestion have been studied for a long time. Lighthill and Whitham (1955) proposed the theory of kinematic wave propagation on a single-lane freeway with stationary conditions, which led to the widely used Lighthill-Whitham-Richard (LWR) model (Richards 1956). The analysis was simplified by Newell (1993) and it facilitated the extension to multi-lane traffic scenarios (Daganzo 2002). Simplifying assumptions were made to analyze integrated freeway and arterial networks (Mahmassani, Valdes, et al. 1998). Merging/diverging behavior was typically modeled as a solution to a set of Partial Differential Equation (PDEs) and approximated using Godunov's scheme (Godunov 1959; Lebacque 1996) or Hamilton-Jacobi-Bellman equations (Crandall and Lions 1984). However, finding closed-form solutions becomes increasingly challenging with non-linearity and an increasing number of variables (Peeta and Ziliaskopoulos 2001).

Traffic congestion can also be analyzed as a consequence of choices made by individuals regarding the time of departure and the routes to take. Individuals try to maximize their utilities (or minimize their costs) under constraints imposed by the network structure, daily schedules, and other socio-economic variables. The resulting traffic congestion is analyzed at network equilibrium. Wardrop and Whitehead (1952) proposed the existence of a static equilibrium condition, which was validated for a special case of a point-queue model with a single Origin-Destination (O-D) pair and a single bottleneck by Vickrey (1969). Subsequent efforts to generalize the model to address multiple bottlenecks and larger networks made simplifying assumptions regarding route-choice behavior or the network structure to derive closed-form solutions (Smith 1983; Palma et al. 1983; Ben-Akiva, Cyna, and Palma 1984; Mahmassani and Chang 1986; Arnott, Palma, and Lindsey 1990; Arnott, Palma, and Lindsey 1993). The static user equilibrium assumption was relaxed through a *Dynamic User Equilibrium* (DUE) model that relied on a *Dynamic Traffic Assignment* (DTA) algorithm to find approximate solutions for the equilibrium traffic at any given time (Merchant and Nemhauser 1978). However, the DTA framework makes an unreasonable assumption that every user has complete knowledge of the network at every time instant. Moreover, the solution to traditional DTA formulation is computationally expensive and at times non-convergent (Boyce 1989; Peeta and Ziliaskopoulos 2001). Simulation-based modeling frameworks were implemented to replicate realistic traffic scenarios, thereby addressing some of these challenges (Fellendorf 1994; Halati,

Lieu, and Walker 1997; Barceló et al. 1998; Ben-Akiva, Bierlaire, et al. 1998; Bhat and Koppelman 1999; Balmer 2007). While most recent simulation softwares have been shown to replicate real-world conditions well (Zheng, Waraich, et al. 2012), model generation is observed to require a large amount of fine-grained individual-level data and is therefore not suitable for modeling demand across multiple days.

2.2 Brief History of the Macroscopic Fundamental Diagram (MFD)

Macroscopic modeling of network-wide congestion using aggregate characteristics was proposed to alleviate the complexity and the uncertainty in route-choice in microscopic traffic flow models. An approach for macroscopic modeling is through the *Macroscopic Fundamental Diagram* (Daganzo 2007). In a well-defined region with homogeneous traffic conditions and slowly evolving demand, the Macroscopic Fundamental Diagram (MFD) provides a correlation between the network vehicle density and network space-mean flow. A rescaled variation of the MFD is the *Network Exit Function* (NEF), which provides a correlation between more directly observable aggregate characteristics, namely *accumulation* and *trip completion rate*. Accumulation is defined as the number of traveling vehicles in a region and trip completion rate is the frequency of completion of trips in a region (Figure 2.1). According to a typical NEF for a region, trip completion rate increases with accumulation up to a critical point, beyond which there is a strong reduction in the rate. Traffic controls based on MFD/NEF aim to achieve near maximum possible trip completion rate (i.e., “sweet spot” in Figure 2.1) in a target region for the maximum possible time during a day.

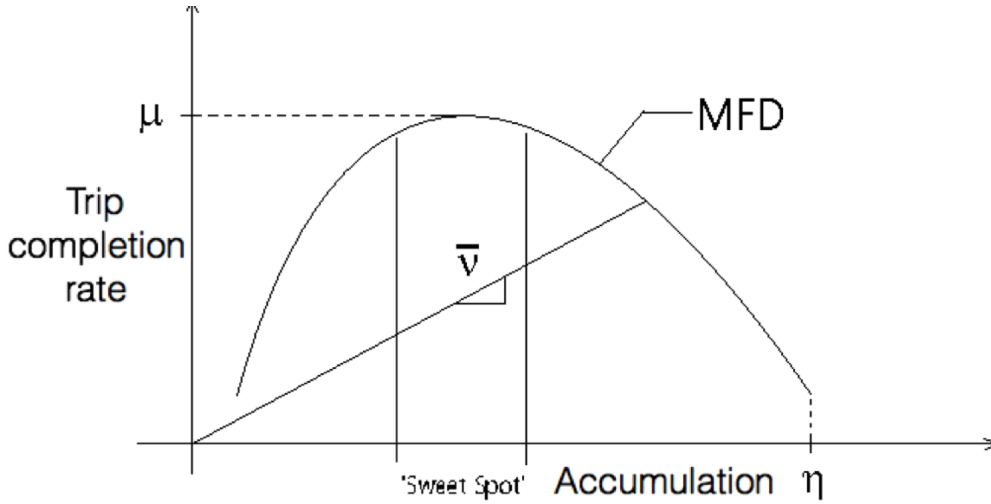


Figure 2.1: Macroscopic Fundamental Diagram (MFD)/Network Exit Function (NEF): It represents the correlation between trip completion rate and accumulation in a well-defined zone with steady-state traffic. The maximum trip completion rate is μ and the maximum accumulation is η . $\bar{v}$ is a measure of the throughput of trips. Macroscopic Congestion Level is defined as $\frac{1}{\bar{v}}$.

The first proposition for the existence of a physical correlation between macroscopic traffic variables was made in by Godfrey (1970). The first model of MFD was introduced by Herman and Prigogine (1979). Mahmassani, Williams, and Herman (1987) showed the

existence of a correlation between vehicle density and network space-mean flow through simulation experiments and Daganzo (2007) extended the correlation to more observable parameters. Geroliminis and Daganzo (2008) showed that such correlations exist on real-world traffic networks also. A key area of research emphasis in the next few years was to define regions in a city where the MFD correlations hold. A solution for partitioning cities with this goal was proposed by Ji and Geroliminis (2012). It was discovered that traffic dynamics may have an impact on the shape of the MFD (Mazlounian, Geroliminis, and Helbing 2010). Doig, Gayah, and Cassidy (2013) also showed that MFDs suffered from a hysteresis effect, i.e., persistence of inhomogeneous conditions that makes the network’s measured traffic states fall beneath its MFD. As a response to the limitations of the traditional MFD, a Generalized MFD (GMFD) framework was proposed by Knoop, Lint, and Hoogendoorn (2015) that is robust against changing traffic dynamics and that also incorporates the hysteresis effects.

MFD/NEF correlations have been used to suggest perimeter controls that aim to maximize trip completion rate for a zone in isolation to its neighbors (Daganzo 2007; Geroliminis and Daganzo 2008). Perimeter control over multiple neighboring zones has been a key research area in the last few years. Geroliminis, Haddad, and Ramezani (2013) solved the optimal perimeter control problem for two urban regions. This work was later extended to the optimal perimeter control over a multi-region heterogeneous network by Aboudolas and Geroliminis (2013). Hajiahmadi, Knoop, et al. (2013) proposed a dynamic control strategy based on MFD for optimizing route guidance through a central controller to keep the number of vehicles entering each region near optimum. More recently, bi-level hierarchical control frameworks have been proposed based on the generalized definition of MFD (Ramezani, Haddad, and Geroliminis 2015; Ni and Cassidy 2018). These frameworks first analyze multiple regions as a single control unit and then propose strategies for specific regions in a second step. However, MFD/NEF-based controls have typically been limited to in and around the target zone. There remains a possibility of further improvement by predicting congestion performance based on traffic conditions across much large geographic scales. To that end, signals from a much larger region-wide network state, such as Origin-Destination (O-D) demand, link accumulations and link travel times could prove extremely useful. O-D demands encode critical high-level information such as peak hour times and peak/non-peak congestion patterns across multiple days. Link accumulations and link travel times are sensitive to several low-level hidden relationships such as drivers’ route preference and spatio-temporal evolution of congestion in the immediate surroundings.

2.3 Traffic Prediction Models

A suitable traffic prediction model is required to make accurate predictions of congestion using signals received from the larger network. Early literature focused on parametric time-series models for traffic prediction (Ahmed and Cook 1979; D’Angelo, Al-Deek, and Wang 1999; Williams 2001; Ishak and Al-Deek 2002). These models were typically limited in their use and were unable to produce the desired level of accuracy in general traffic scenarios (Miles and Walker 2006; Van Lint and Van Hinsbergen 2012; Vlahogianni, Karlaftis, and Golias 2014). Non-parametric methods such as Support Vector Regression (SVR) (Wu, Ho, and Lee 2004), simple feed-forward neural networks (Innamaa 2005),

Recursive Neural Networks (RNNs) (Van Lint, Hoogendoorn, and Van Zuylen 2002), Bayesian models (Fei, Lu, and Liu 2011) and hybrid models (i.e., Bayesian models with neural networks for making prediction) (Van Hinsbergen and Lint 2008) were found to be better suited than parametric models for the task of predicting congestion. These non-parametric models, of course, are unable to store long-term recurrence patterns that are integral to the nature of traffic. Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) architecture provides a solution to this problem. It maintains long-term memory through a gated architecture. Such an architecture has been tried more recently for network-level traffic prediction (Ma, Tao, et al. 2015; Song, Kanasugi, and Shibasaki 2016; Zhao et al. 2017). The general LSTM architecture involves fully connected LSTM (FC-LSTM) networks where a large amount of complex spatial information is transferred while training the neural network. This makes learning more challenging (Xingjian et al. 2015). Special convolutional layers can encode spatial features well and are much easier to train than fully connected layers. Such neural networks are called Convolutional Neural Networks (CNNs). The encoding of spatial features is specially relevant to traffic since queues can spill over from one link to another, causing strong spatial dependencies (Daganzo 1998). Due to this property, CNN-LSTM frameworks have recently been explored in the domain of traffic prediction (Song, Kanasugi, and Shibasaki 2016; Wu and Tan 2016; Ma, Dai, et al. 2017).

A common theme in CNN-LSTM frameworks is the use of images as the input data format. The regular grid structure of pixels in images ensures that there is translation invariance in each direction. However, learning from images can be slow because the information that may appear trivial needs to be first learned by the model to make predictions. This information may include the fact that traffic movement is constrained to the physical road network or the fact that there is a lagged dependence between traffic demand and its resulting congestion. This lag is equal to the travel time along a network path. A possible solution for encoding such information is to represent the input data in a structured format with the help of graphs. This ensures that relative distances between the data at nodes are calculated along paths in this graph rather than calculating Euclidean distances. CNNs extract spatial features from data through two operations, namely convolution (i.e., applying several non-linear functions) and pooling (i.e., aggregating data locally). However, these operations are only defined on data represented as regular grids. Two possible approaches have been proposed to extend these operations to data represented as graphs. The first approach defines special neural network architectures to learn from graphical data (Gori, Monfardini, and Scarselli 2005; Scarselli et al. 2009; Duvenaud et al. 2015; Li, Tarlow, et al. 2015; Jain et al. 2016). These architectures are optimized for specialized tasks and are easy to train. However, the operations do not involve the convolution or pooling operations of traditional CNNs, and thus make these architectures harder to generalize. The second approach is derived from *Spectral Graph Theory* (Chung 1997) and such neural networks are called Graph-CNNs (Bruna et al. 2013; Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2016). The operations involved are described next.

Spectral convolutions are defined as Fourier Transforms (Bracewell 1986) of the graph Laplacian matrix (Chung 1997). This operation generalizes the traditional convolution operation to graphs. Consider a graph, $G = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V}$ is the set of nodes ($|\mathcal{V}| = n$), $\mathcal{E}$ is the set of vertices and $W \in \mathbb{R}^{n \times n}$ is the weighted adjacency matrix

encoding the connection weight between any two vertices. The graph Laplacian for a simple graph is given by $L = D - W$. The normalized graph Laplacian is given by $L = I - D^{-\frac{1}{2}}WD^{\frac{1}{2}}$. In both these definitions, $D \in \mathbb{R}^{n \times n}$ is the diagonal degree matrix such that $D_{ii} = \sum_j W_{ij}$ and I is the identity matrix. The Laplacian can be diagonalized using the Fourier basis, U , as $L = U\Lambda U^T$, where $\Lambda = \text{diag}(\lambda_0, \dots, \lambda_{n-1})$ is a diagonal matrix and its entries correspond to the eigenvalues. Any input signal x may be filtered by a graph $g(\theta)$ to produce the following output:

$$y = g_\theta(L)x = g_\theta(U\Lambda U^T)x = U g_\theta(\Lambda)U^T x, \quad (2.1)$$

where $U^T x$ is the Graph Fourier Transform.

The output of this operation encodes key spatial features for the particular input and can be used to make predictions.

Since operations in the Fourier basis are costly, some approximations for calculating the Fourier transform have been proposed. Defferrard, Bresson, and Vandergheynst (2016) propose a Chebyshev polynomial approximation for the function $g_\theta(\Lambda)$. Kipf and Welling (2016) propose a linear approximation that is even faster to compute than a Chebyshev approximation but may not be as accurate. In this study, the methods proposed by Defferrard, Bresson, and Vandergheynst (2016) are adopted for calculating graph convolutions.

Pooling operations may be performed through agglomerative clustering by defining a suitable neighborhood around a node that is considered as a cluster center. Bruna et al. (2013) mentioned the following possible definition of a neighborhood, $N_\delta(j)$, for node j parameterized by threshold δ :

$$N_\delta(j) = \{i \in \mathcal{E} : W_{ij} > \delta\}, \quad (2.2)$$

where links are represented by $\mathcal{E}$ and weights between (i, j) are represented as $W_{i,j}$.

Such an operation allows coarsening of the graph in the same manner as CNNs reduce the dimensionality of the input data.

A Graph-CNN architecture has recently been used for traffic prediction by Li, Yu, et al. (2017). They proposed a graph structure based on the structure of the road network and assume the traffic flow to be a diffusion process where the transitions may occur due to random walks. They thereby observed a 12-15% improvement over ARIMA baselines. In this research, the structure of the road network and prior knowledge about route-choices are used to define graphs. Signals on these graphs are shown to be sensitive to the physical processes governing congestion propagation. These representations may prove more useful for modeling real-world dynamics than the previous work.

2.4 Model Robustness

In addition to accuracy, model robustness is a highly desirable property. This research focuses on three possible adversities to test for model robustness: (i) statistical fluctuations in demand, (ii) measurement-noise in the input and (iii) missing inputs.

A model is likely to maintain high accuracy in spite of statistical fluctuations in demand if it blends historical data well with real-time data. Several approaches for combining historical and real-time data have been suggested in the past literature. These approaches can be classified as either parametric or non-parametric. Some parametric models are Kalman Filter models (Okutani and Stephanedes 1984; Vythoulkas 1993), random-walk models (Chang and Wu 1994), auto-regressive models (Ashok and Ben-Akiva 1993), and hybrid AR-Kalman Filter models (Ashok and Ben-Akiva 2000; Zhou and Mahmassani 2007). Non-parametric models include clustering of days with similar behavior (Yildirimoglu and Geroliminis 2013) and neural networks (Park and Rilett 1999; Van Lint, Hoogendoorn, and Van Zuylen 2002, 2005; Liu, Zuylen, et al. 2006; Zheng, Lee, and Shi 2006). Non-parametric methods are preferable for traffic prediction since they don't depend on an explicit weighted structure for data fusion (Vlahogianni, Karlaftis, and Golias 2014). The state of the art method for storing a long-term memory while accommodating for statistical fluctuations is a Long Short-Term Memory (LSTM) neural network architecture. It contains a gated structure with a specific gate called the *forget gate* that is responsible for storing only key recurring features over time.

Measurement-noise can corrupt sensors, Global Positioning Systems (GPS), cell phone traces, and other data sources. It may not always be feasible to fix or even detect faulty measurements. Therefore, the denoising of input data is a requirement during modeling. Data aggregation across space and time may lead to some filtering of noise at the expense of high granularity. Statistical techniques such as exponential smoothing (Tikunov and Nishimura 2007; Chan et al. 2012) and Discrete Wavelet Transform (DWT) (Xie, Zhang, and Ye 2007) have also worked well in the past for filtering noise during traffic prediction. Neural network models for time-series predictions can be made robust against noise by reducing the batch size hyper-parameter (Giles, Lawrence, and Tsoi 2001).

Real-world data sources also present the possibility of failure to record data at certain locations/times or losing an input signal altogether. Most prediction methods achieve robustness against the former by imputing missing values in spatio-temporal variables (Van Lint, Hoogendoorn, and Zuylen 2005; Chen, Wang, et al. 2012; Li, Li, and Li 2013). Bayesian techniques for dealing with missing data have also been proposed (Sun, Zhang, and Yu 2006). In the absence of a particular signal, the effect on prediction accuracy depends on its importance. A proxy signal may be used in such a situation if possible.

2.5 Interpreting Traffic Prediction Models

An easily interpretable model is more likely to be adopted for guiding traffic controls. Thus, traffic prediction literature stresses on the importance of developing easily interpretable models (Treiber and Helbing 2002; Sun, Zhang, and Zhang 2005; Vlahogianni, Karlaftis, and Golias 2007; Min and Wynter 2011). A model may be interpreted through the marginal impacts of each input on the prediction at any given time. Deep learning models employ a *Neural Attention Model* for this purpose (Fukushima 1987; Tang, Srivastava, and Salakhutdinov 2014; Xu, Ba, et al. 2015; Ramanishka et al. 2017). This model produces a spatio-temporal saliency map of inputs representing relative variable importance. The theory behind Neural Attention Models is discussed next.

Attention-based models have recently gained popularity for performing tasks such as handwriting synthesis (Graves 2013), machine translation (Bahdanau, Cho, and Bengio 2014), image captioning (Xu, Ba, et al. 2015) and speech recognition (Chorowski et al. 2015; Bahdanau, Chorowski, et al. 2016). These models iteratively process the input by selecting the relevant context at each step. Through this process, they provide a mechanism to extract the most relevant input features and store them, possibly as an external memory (Parisotto and Salakhutdinov 2017). There are two variants of attention models, namely “hard” and “soft” attention. Hard attention models (Mnih, Heess, Graves, et al. 2014) aim to detect the most relevant input features exactly, but may not produce differentiable loss functions. Therefore, training such neural network models can be inefficient. Soft attention models (Xu, Ba, et al. 2015; Yao et al. 2015) aim to detect *regions* that are relevant for prediction. Most implementations of soft attentions may require a change in network architecture and hence, don’t scale well. As an extension to traditional soft attention, Ramanishka et al. (2017) developed a method for discovering spatio-temporal saliency without explicitly changing the network structure. They do this by measuring a representation of loss in predictive power when the input set is constrained. Details of this technique follow.

Consider a classification problem. Given all inputs, the probability of producing a sequence of output, Y_t , at time t , is $p(Y(t))$. Given all inputs except f , the probability of producing the same sequence of output, Y_t , at time t , is $q(Y(t))$. The loss in information at time t due to the constrained input set is given by:

$$\text{Loss}_{f,t} = D_{KL}(p(Y(t)), q(Y(t))), \quad (2.3)$$

where:

$\text{Loss}_{f,t}$ is the loss of information corresponding to feature f at time t ,
 D_{KL} is the KL-divergence between the two probability distributions.

If the prediction problem is a regression problem rather than a classification problem, the KL-divergence loss function can be replaced by an L_2 loss-based function. The normalized values of these losses give the relative importance of the corresponding input that is removed. This method allows derivation of the relative importance of each input at time t .

This study explores model-based predictive controls through a simulated case study. Here, predictions and controls are implemented in separate steps. Future research may aim to jointly derive predictions and controls. A possible technique for doing this is via *Reinforcement Learning* (RL). The congestion state over time can be represented as a Markov Decision Process (MDP) (Bellman 1957). A data-driven optimization jointly forecasts the congestion state and recommends optimal controls to reduce congestion impacts. RL-based adaptive strategies have largely been proposed in the past for signal timing control (Wiering 2000; Bingham 2001; Abdulhai, Pringle, and Karakoulas 2003; Arel et al. 2010; Prashanth and Bhatnagar 2011; El-Tantawy, Abdulhai, and Abdelgawad 2013). More recently, MDP-based dynamic pricing strategies have been developed for link tolls (Rambha and Boyles 2016) and cordon-metering (Ni and Cassidy 2018). However, RL-based dynamic control at city-wide scales are yet to be explored. The framework for congestion prediction that is described in this study enables such a possibility.

Chapter 3

Modeling Framework

This chapter describes the modeling framework for zone/neighborhood-wide congestion prediction. A scoring function is defined to represent the congestion state in a target neighborhood. Input signals from a larger region-wide network are proposed to forecast this score. A function maps the recently observed input signals to the future congestion scores. A deep learning model based on LSTM architecture approximates this function.

The setup is as follows. A graph, $G = (\mathcal{V}, \mathcal{A})$, represents the road network for the entire region of interest. Here, $\mathcal{A}$ represents all directed road links approximated by straight line segments and $\mathcal{V}$ represents the endpoints of all road links. The network is partitioned into Z neighborhoods through a network partitioning procedure proposed by Ji and Geroliminis (2012). This partitioning method ensures that:

- traffic is homogeneously loaded across each individual neighborhood, z ,
- traffic evolves slowly over time within each neighborhood ensuring nearly steady-state conditions at any given time, and
- any street link, i , lies exactly within a single neighborhood.

These conditions ensure that MFD/NEF correlations hold within each partitioned neighborhood. The set of links within a neighborhood, z , is A^z .

3.1 Macroscopic Congestion Level

A congestion score for a neighborhood is proposed based on its MFD. The score depends on two neighborhood variables, namely *accumulation* and *trip completion rate*. Both the variables can be derived from the cumulative number of vehicles entering and exiting each link in the neighborhood. Let $A_i(t)$ and $L_i(t)$ represent the cumulative number of vehicles that have arrived and left link i by time t . Here, $n_i(t) = A_i(t) - L_i(t)$ represents the vehicle accumulation on link i respectively by time t .

The aggregate accumulation, $n^z(t)$, in z can be derived as a function of $n_i(t)$ as follows:

$$n^z(t) = \sum_{i \in A^z} n_i(t). \quad (3.1)$$

Let $E_i(t)$ be the endogenous portion of $D_i(t)$; i.e., the cumulative number of trips that ended on link i . The total trip completion rate, $T^z(t)$, is defined as:¹

$$T^z(t) = \sum_{i \in A^z} \frac{\partial E_i(t)}{\partial t}. \quad (3.2)$$

A congestion score for z at time t is defined as a function of $n^z(t)$ and $T^z(t)$. This score is called **Macroscopic Congestion Level (MCL)**. Its definition is as follows:

$$\xi^z(t) = \begin{cases} 0, & T^z(t) < \tau^z, \quad n^z(t) < \alpha^z \\ n^z(t)/T^z(t), & \text{otherwise,} \end{cases} \quad (3.3)$$

where τ^z is a threshold trip completion rate and α^z is a threshold accumulation for Zone z .

When the values of $n^z(t)$ and $T^z(t)$ are below their respective thresholds, their ratio might be unstable. These are situations where traffic demand is very low and hence, the trip completion rate observed is also low. MCL doesn't model congestion very accurately for such scenarios. But these scenarios are deemed less important since there are no massive delays. The value of $\xi^z(t)$ is thus set to zero. Further smoothening may be applied to the value of $\xi^z(t)$ using LOWESS (Locally Weighted Scatterplot Smoothening) (Cleveland 1981) or nearest-neighbors to remove noise.

Though exact measurements of $\xi^z(t)$ would require arrival and departure counts on all links in z , the assumption of homogeneous loading in each partitioned neighborhood indicates that estimates can be made by sampling counts from a partition of z 's links. This makes it possible to track MCL values even when data from a neighborhood is incomplete.

3.2 Network State Definition

The time-dependent state of the regional network, $X(t)$, is defined as the following vector of vectors:

$$X(t) = \begin{bmatrix} D(t) \\ C(t) \\ TT(t) \\ \xi(t) \end{bmatrix}, \quad (3.4)$$

where:

$D(t)$ is the vector of O-D demands between all Z neighborhoods in the network²,

$C(t)$ is the vector of link accumulations,

$TT(t)$ is the vector of link travel times,

$\xi(t)$ is the vector of $|Z|$ MCL values estimated using Equation (3.3).

¹This definition assumes that street links in z are of uniform length. The trip completion rate in neighborhoods with unequal link lengths can be calculated as per Edie's generalized definitions (Edie 1965).

²Unlike other components of vector $X(t)$, real-time O-D demands, $D(t)$, are not directly observable quantities. For a technique to dynamically estimate and update O-D demands using modern data sources, please see Appendix A

Dimensionality of $X(t)$ may be very high. $X^z(t)$ is defined as a vector of a subset of elements of $X(t)$ that are expected to affect the congestion state in z significantly. Several heuristics can help reduce the number of feature dimensions that need to be explored to build an accurate prediction algorithm. For example, O-D demands used to populate $X^z(t)$ might be restricted to those with their destination in z . This would reduce the number of O-D inputs from $\mathcal{O}(|Z|^2)$ to $\mathcal{O}(|Z|)$. Similar heuristics can be applied to reduce the dimensionality of vectors $C(t)$, $TT(t)$ and $\xi(t)$ in $X^z(t)$. One might, for example, filter out input signals from links or neighborhoods that reside greater than some specified distance from the centroid of z . A possible data-driven approach for dimensionality reduction is through graph sampling (Chepuri et al. 2017) after transforming the network state input to impose a graphical structure (see Chapter 7).

The hypothesis is that MCL values can be forecasted into the future for each z^{th} neighborhood using the values $x^z(t)$ of vector $X^z(t)$.

3.3 Deep Learning Framework for MCL prediction

A deep learning model is used to forecast the MCL values for each Zone z , ξ^z , using the values x^z of the relevant network state vector, X^z . The model has the form:

$$[\xi^z(t + h^z + p^z), \xi^z(t + h^z + p^z - 1), \dots, \xi^z(t + h^z)] = f(x^z(t), \dots, x^z(t - p^z)), \quad (3.5)$$

where:

f represents a function approximator (in this case, a Seq2Seq LSTM model³), h^z is defined as the *minimum dependency lag* which describes the shortest possible time required for a measured input signal, $x^z(t)$, to travel to the periphery of neighborhood z , p^z is defined as the *maximum dependency persistence* which describes the greatest duration over which the signal can directly impact congestion within Zone z after reaching its periphery.

Equation (3.5) logically indicates that at time t , the latest input signal observed will have a likely impact on z 's congestion scores in the interval $[t + h^z, t + h^z + p^z]$. Therefore, the forecast horizon is chosen as such. The other observed signals that may impact congestion scores in the same time horizon lie over the period $[t - p^z, t]$. Hence, these signals are chosen as the model inputs.

The hyper-parameters h_z and p_z vary over space and time but were treated as constants to limit the number of iterations needed to train the LSTM model. h^z was conservatively chosen to be the signal's minimum free-flow travel time across all possible locations or origins within the network. Similarly, p^z is conservatively taken to be the maximum of all signal durations over space and time. Both estimates were obtained via cross-validation.

³For details about the steps executed by the LSTM model, please refer to Appendix B.

Chapter 4

Viability Conditions

The model is tested for sensitivity to the physical processes affecting congestion generation and propagation. A battery of simulation experiments suggests that neighborhood traffic conditions are nicely described by MCL. They also demonstrate that the network state signals, that are considered as model inputs, are useful in conveying how traffic features emerging elsewhere within a regional network are carried to neighborhoods, and locally impact the scores there. The model accuracy is compared with an intuitive and competent baseline model in simulated settings. Outcomes suggest that a deep learning-based function responds better to the changes in the network structure and variations in traffic demand within and across days.

4.1 Sensitivity Analysis

This section aims to test two requirements:

- MCL must be able to mimic major trends in time-varying congestion patterns in a neighborhood.
- There must be at least one input signal that helps forecast the major trend in MCL over time.

A baseline network was constructed as a directed graph G containing 7 links (i.e., links 1-7) representing a simple city street network containing only 4 zones of interest (i.e., a, b, c, and d) (Figure 4.1a). A battery of tests was performed using this simple (hypothetical and idealized) network. The network link parameters are coarse abstractions of reality. Illustrations try to approximately mimic commute traffic, heading into a Central Business District (CBD) from residential neighborhoods via arterial roads. This is a common scenario that leads to congestion in large cities. Trends in congestion patterns, which are a function of the network structure, the demand scenarios, and individual choices, are approximations of reality too. These simplifications allow analysis and prediction of neighborhood congestion using deterministic models based on traffic flow theory. The predictions can then act as *ground truth* for evaluating the performance of any other prediction model.

Scenario 1: Time-Varying O-Ds

The first test examines a time-varying O-D demand pattern on the baseline network shown in Figure 4.1a. All trips originating in the centers of neighborhoods a, b, and d are bound for the center of neighborhood c. Trips from the neighborhood a are served by three directed links labeled 1, 5, and 4. Similarly, trips from neighborhood b are served by three directed links labeled 3, 7, 4, and trips from neighborhood d are served by directed links labeled 2, 5, 4. The physical lengths of each link are shown in Figure 4.1a.

Note first that links 1, 2, 3, and 4 reside solely within neighborhoods a, d, b, and c, respectively. Each neighborhood's entire street network is thus represented by a single directed link. This abstraction dispenses with the concerns regarding whether or not neighborhoods are homogeneously loaded with traffic since inhomogeneous loading can render MFDs unusable; see Daganzo (2007) and Daganzo and Geroliminis (2008). And MFDs, with this simplified representation, reduce to ordinary Fundamental Diagrams (FDs).

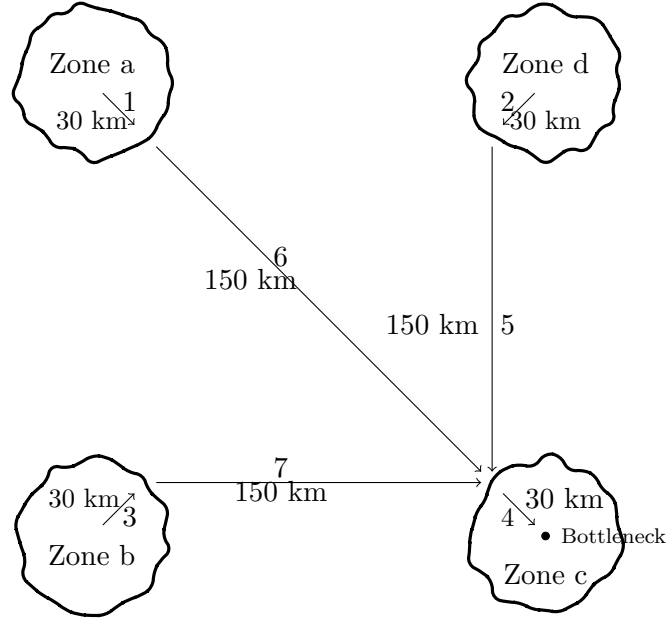
The FDs used for the baseline network are shown in Figure 4.1b. These were selected to distinguish relative differences between traffic features in a single city-street lane (links 1, 2, 3, and 4) and those in a single lane of a signalized arterial (links 5, 6, and 7). Importantly, a bottleneck with a capacity of 5 vehs/min is placed at the downstream end of link 4 so as to congest neighborhood c.

The time-varying demands from neighborhoods a, b, and d to c were chosen to serve the same purpose and to approximate typical commute demand scenarios. The demand is shown by the dotted curves in Figure 4.1c. The resulting congestion pattern in c is shown through cumulative vehicle count curves in Figure 4.1d. They have been constructed in such a way that their vertical displacements are the vehicle accumulations in Zone c; see Daganzo (2001).

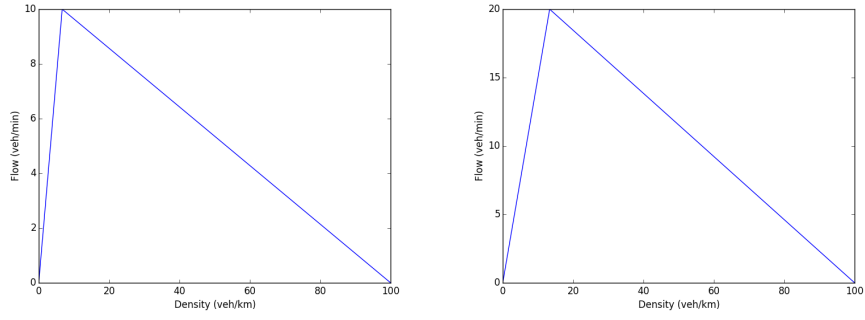
It may be noted how the MCL nicely captures the pattern displayed in Figure 4.1d. The reader can see this by referring back to Figure 4.1c. Its solid curve is the time-series of Zone c's MCL. Note first how the MCL's sinusoidal pattern aligns with the growth and dissipation of accumulation shown by the count curves in Figure 4.1d. The similarity between the MCL's pattern and that of the demand may also be noted from Figure 4.1c. The similarity highlights how O-D pattern influences MCL. It may further be noted how the congestion score is shifted in time relative to the demand. The shift is a function of free flow travel times on all links except link 4. The free flow travel times need not be provided as a separate input to the model as they can be inferred through observations across multiple such days.

Scenario 2: Coalescing O-Ds

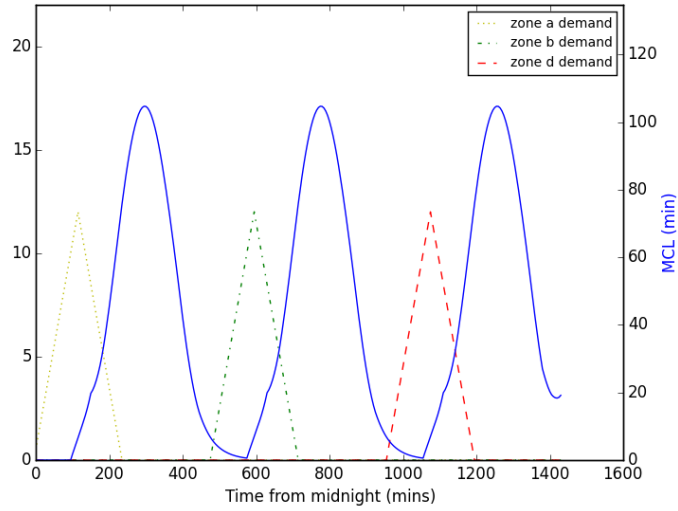
Similar support come for MCL and its input signals, when demands from multiple zones coalesce. Traffic on all links is described by the FDs in Figure 4.1b and the baseline network remains unchanged. The demand patterns originating in a and b are shown with dotted and dashed curves respectively in Figure 4.2a. These demands emanate closely in time, which causes flows from both neighborhoods to coalesce in c. The resulting traffic pattern in c is well described by the time-series of congestion scores; see the solid curve



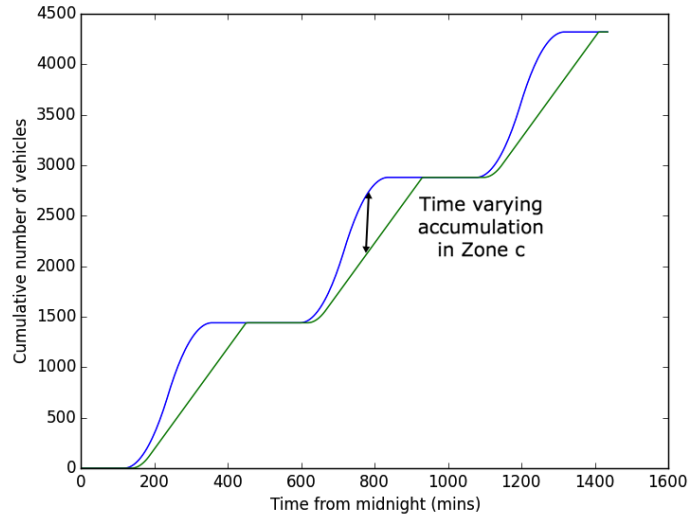
(a) Baseline Network



(b) FD for links 1, 2, 3 & 4 (left) and links 5, 6 & 7 (& 8) (right)



(c) Scenario 1: Plot showing demand from the three input zones over time (values to be read from the primary y-axis) and the MCL in the Zone c over time (plotted as a solid line, values to be read from the secondary y-axis). The threshold output was 1 veh/min and nearest neighbor smoothening was applied with 10 nearest neighbors.

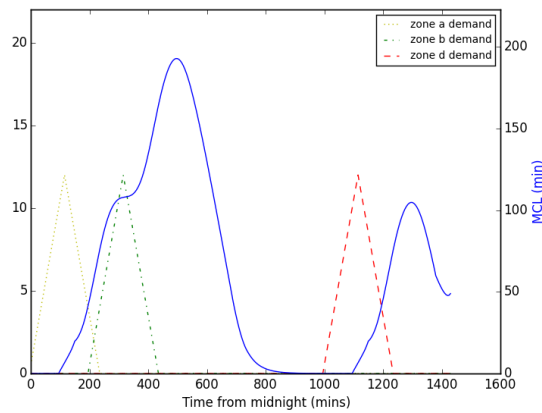


(d) Cumulative demand and departure curve in Zone c

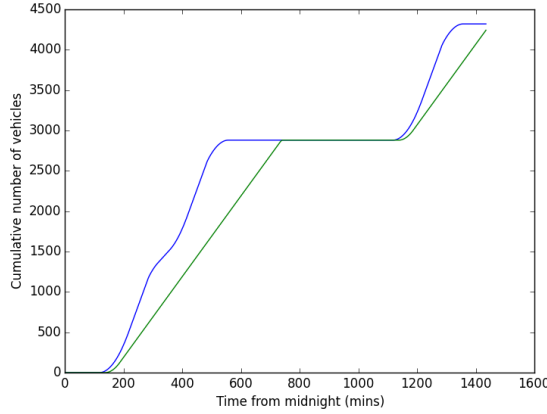
Figure 4.1: Figures describing Scenario 1 and the effectiveness of MCL and network state for modeling neighborhood congestion.

in Figure 4.2a. Such a scenario may arise, for example, when the desired arrival times for individuals residing in multiple neighborhoods are close to each other.

Note the irregular wavy pattern displayed by the MCL in the early going, when flows from a and b begin to coalesce. Note too how the additional input from b produces both, a heightened MCL value and a longer period of congestion relative to the baseline outcomes previously shown by the congestion scores in Figure 4.1c. A similar pattern can be observed with respect to time-varying accumulation in Zone c, see Figure 4.2b. This highlights the use of MCL. The effect of coalescing O-Ds on the congestion score highlights the importance of tracking input signals over a sufficiently long time horizon to accommodate temporal correlations.



(a) Scenario 2: Plot showing demand from the three input zones over time (values to be read from the primary y-axis) and the MCL in the Zone c over time (plotted as a solid line, values to be read from the secondary y-axis).



(b) Scenario 2: Plot showing the cumulative input curve (blue) and cumulative output curve (green) for link 4.

Figure 4.2: Figures describing Scenario 2 and the effectiveness of MCL and network state for modeling neighborhood congestion.

Scenario 3: Effect of Bottlenecks on Preceding Links

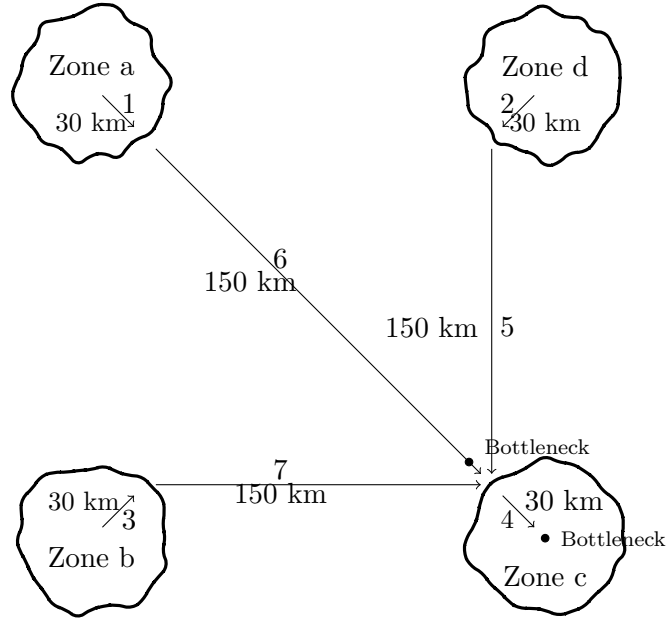
The next experiment features the baseline network and the original demand pattern with a bottleneck of a capacity of 5 vehicles/min at the downstream end of link 6; see Figure 4.3a. The new bottleneck generates queueing on link 6, which dissipates when the demand subsided. Telltale signs of this queueing are evident in link 6's travel times (i.e., $TT(t)$). A time-series curve for this signal is presented in Figure 4.3d. Note how link 6's queueing episode is clearly evident in the figure.

The newly placed bottleneck causes Zone c's accumulation to level-off because the vehicles now enter the zone at a rate equal to the capacity of the bottleneck on link 6 (5 veh/min). This is evident in Zone c's MCL shown with solid lines in Figure 4.3b. The score plateaus due to a constant rate of vehicles entering Zone c. The duration of the level-off exactly matches with the duration of time for which link 6's travel times exhibit a sinusoidal pattern.

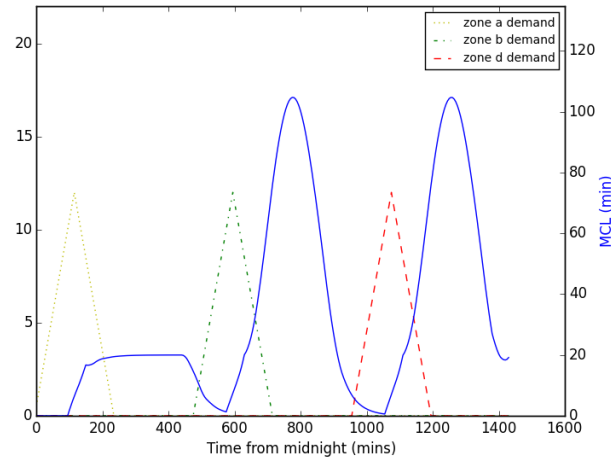
In sum, the second scenario reconfirms the descriptive power of MCL. It also highlights the role of link travel times in forecasting that score. Link accumulations and travel times can together play the same role in a more complex setting, as shown below.

Scenario 4: Effect of Route Choice

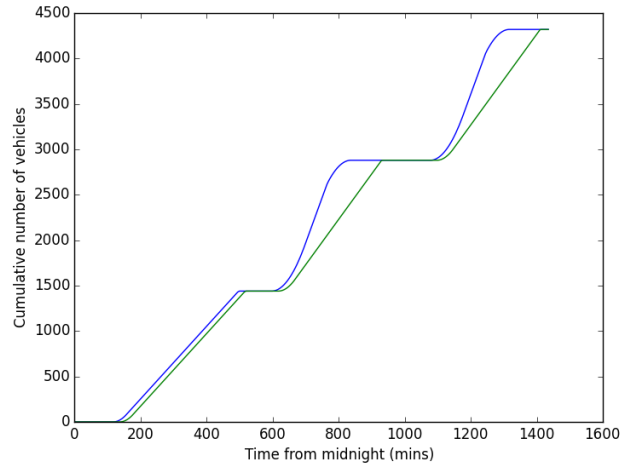
The relevance of the scoring function and input signals for modeling neighborhood congestion in the presence of route-choice is shown next. The network structure is described in Figure 4.4a. Note the addition of link 8 as compared to Figure 4.3a. The link FD for link 8 is as described in Figure 4.1b. Similar to the situation described in Scenario 3 (see Figure 4.3), the presence of an arterial bottleneck on link 6 causes delays on link 6. However, as a consequence, some individuals now have the option to switch to link 8 and subsequently move along link 7. The route-choice is assumed to occur in accordance to Dynamic User Equilibrium (DUE) (i.e., no user can lower his/her travel time by changing



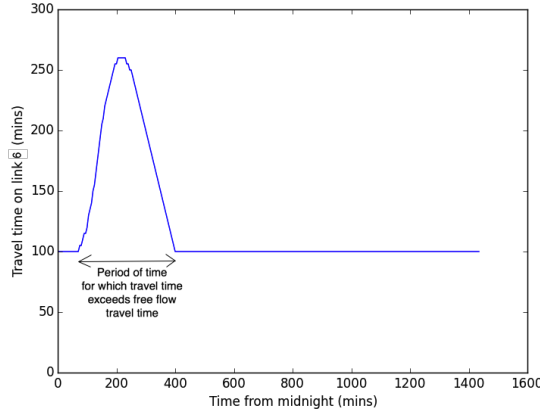
(a) Baseline Network With Arterial Bottleneck



(b) Scenario 3: Plot showing demand from the three input zones over time (values to be read from the primary y-axis) and the MCL in the Zone c over time (plotted as a solid line, values to be read from the secondary y-axis).



(c) Cumulative demand and departure curve in Zone c



(d) Travel time on link 6 over time

Figure 4.3: Figures describing Scenario 3 and the effectiveness of MCL and network state for modeling neighborhood congestion.

routes). This condition assumes that each commuter has perfect knowledge of the network state. Real-world scenarios may not follow DUE. However, this doesn't affect the viability of the proposed model as no such assumption is made during model formulation.

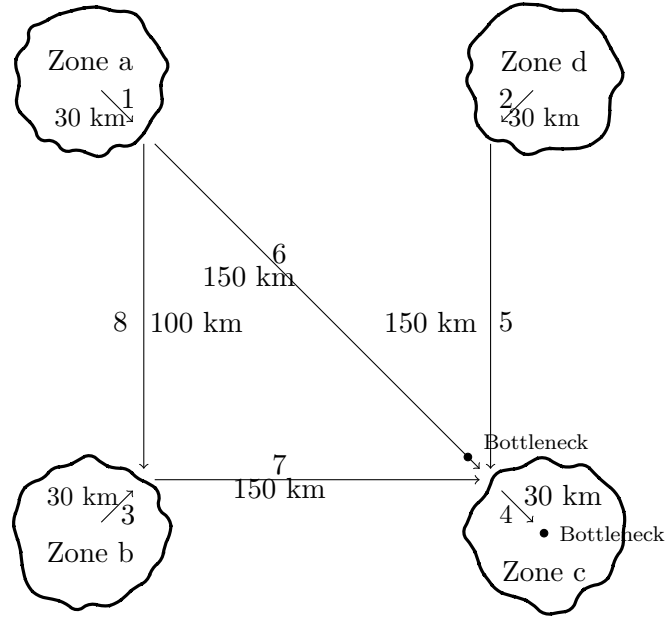
Note how MCL values have a sudden jump at around $t=400$ mins in Figure 4.4b. This corresponds to the time at which individuals who chose the route along (link 1 $\rightarrow$ link 8 $\rightarrow$ link 7) start arriving at Zone c. Also, note that the time-varying accumulation has a similar jump in values at the same time (see Figure 4.4c). This confirms the hypothesis that MCL is a good representation of the congestion performance in Zone c.

The key signals that can aid prediction of congestion performance are link travel times, $TT(t)$, and link accumulations, $C(t)$, corresponding to link 6. Figures 4.4d and 4.4e compare the travel times and accumulations on link 6 in Scenarios 3 and 4 respectively. The comparison shows that an indicator of the occurrence of the 'jump' in MCL values (Figure 4.4b) is that there is a 'flat' section after an increase in link travel time and link accumulations. The period for which the travel times and counts stay flat corresponds to the period when route-switching takes place.

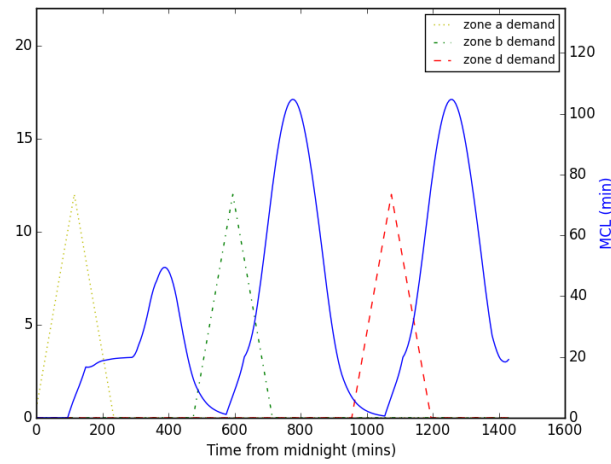
Scenario 5: Effect of Queue Spillover

The next scenario tested for the relevance of the scoring function and input signals is that of queues spilling over to neighboring zones. It features the baseline network with the lengths of link 4 and link 6 shortened to 5 kms each (see Figure 4.5a). Due to the shortened link lengths, the queue created due to the bottleneck at the downstream end of link 4 spills over to link 6 and then eventually to link 1. The zone analyzed is now Zone a. The MCL in the Zone c (i.e., $\xi^c(t)$) provides an initial indication of the possibility of queue spillover. Subsequently, additional information from the travel times and accumulations on link 6 (i.e., $TT(t)$ and $C(t)$) confirms queue spilling over to the Zone a as well as the time it takes for that to happen.

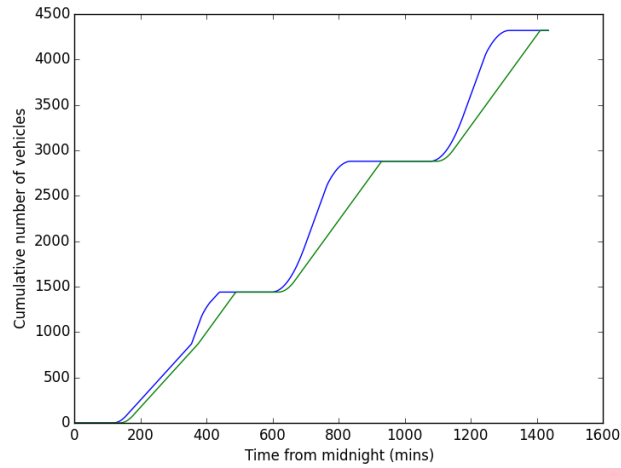
In figures 4.5b and 4.5d, the congestion score in the Zone a is plotted with dashed curves.



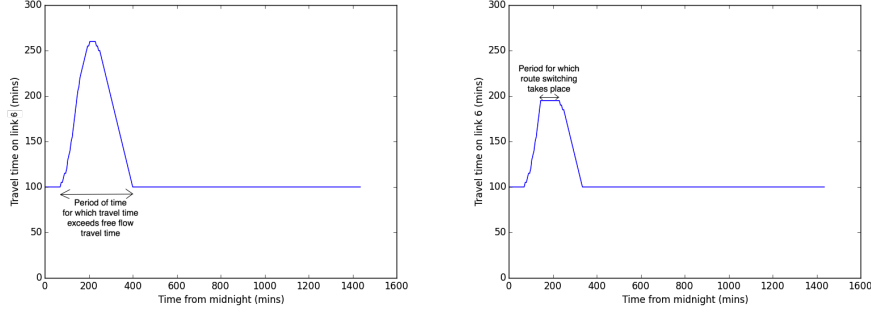
(a) Baseline Network With Arterial Bottleneck And Route-Choice



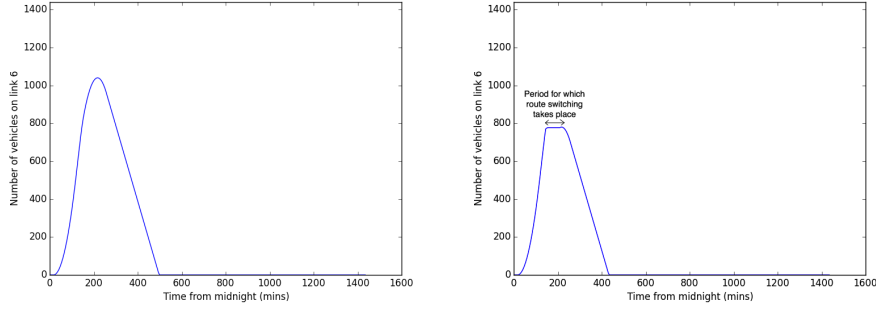
(b) Scenario 4: Plot showing demand from the three input zones over time (values to be read from the primary y-axis) and the MCL in the Zone c over time (plotted as a solid line, values to be read from the secondary y-axis).



(c) Cumulative demand and departure curve in Zone c



(d) Travel time on link 6 over time for Scenario 3 (left) and Scenario 4 (right)



(e) Number of vehicles on link 6 over time for Scenario 3 (left) and Scenario 4 (right)

Figure 4.4: Figures describing Scenario 4 and the effectiveness of MCL and network state for modeling neighborhood congestion.

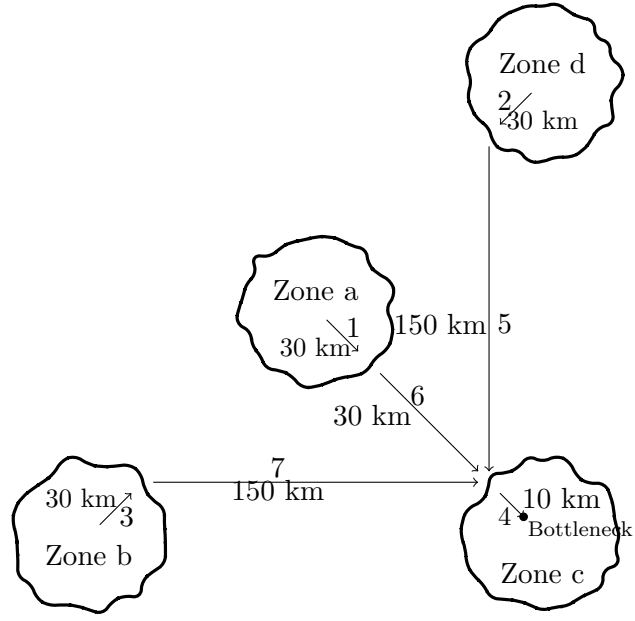
The occurrence of queue spillover is nicely captured by the sudden spike in the MCL values at around $t = 150$ mins. From Figure 4.5b, we can notice a plateau in MCL values in Zone c occurring a few minutes before the onset of congestion spillover. This provides an indication of the possibility of queue spillover to its surrounding zones. Figure 4.5d provides confirmation of the queue spillover and also the time at which it occurs. Congestion spills over to Zone a right at the time of the plateauing of both the link travel times and link accumulation curves for link 6.

To summarize, MCL values in neighboring zones (i.e., $\xi(t)$) provide an early indication of a possible spillover. Together with link travel times and accumulations in preceding links (i.e., $TT(t)$ and $C(t)$), they allow prediction of MCL in the target zone.

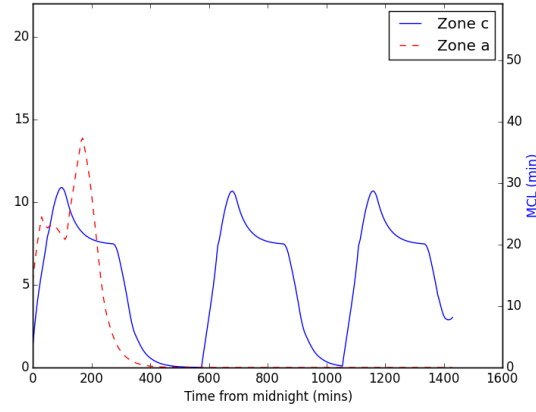
Scenario 6: Effect of Incidents

Several kinds of incidents may affect the structure and properties of a traffic network. Incidents may (i) impact the way people choose routes, (ii) change travel times along certain links, or (iii) force changes in the demand pattern. Changes in link travel times or O-D patterns are directly observable through the change in values of the inputs to the model (Equation 3.4). A scenario is evaluated where the network structure changes have an impact on the route-choice. Input parameters values that are responsive to this occurrence are discovered.

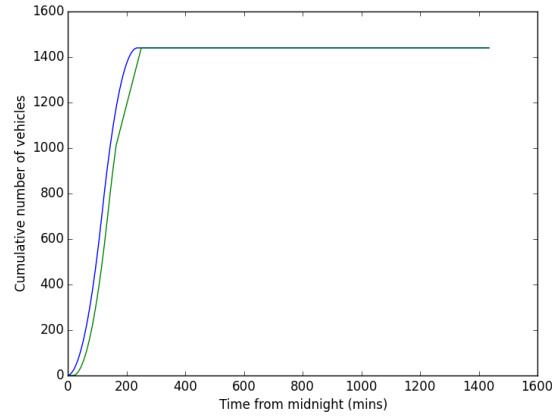
In this scenario, the network structure is the same as that in Scenario 4 (Figure 4.4a). However, unlike Scenario 4, due to the occurrence of an incident at $t = 180$ mins, link 8



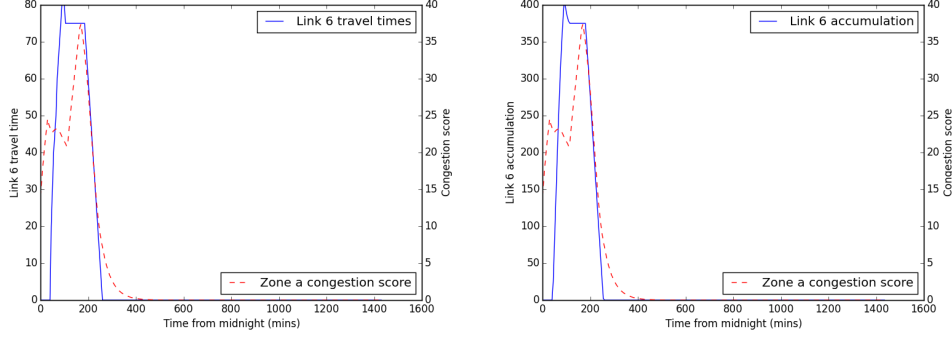
(a) Baseline Network With Short Links



(b) The plot of congestion score in Zone a (dotted line, secondary y-axis) and congestion score in Zone c (solid line, secondary y-axis) vs time in Scenario 5



(c) Cumulative demand and departure curve for the Zone a in Scenario 5



(d) Travel times and accumulation on link 2 in Scenario 4 plotted with the congestion score (dashed, secondary axis) in neighborhood A.

Figure 4.5: Figures describing Scenario 5 and the effectiveness of MCL and network state for modeling neighborhood congestion.

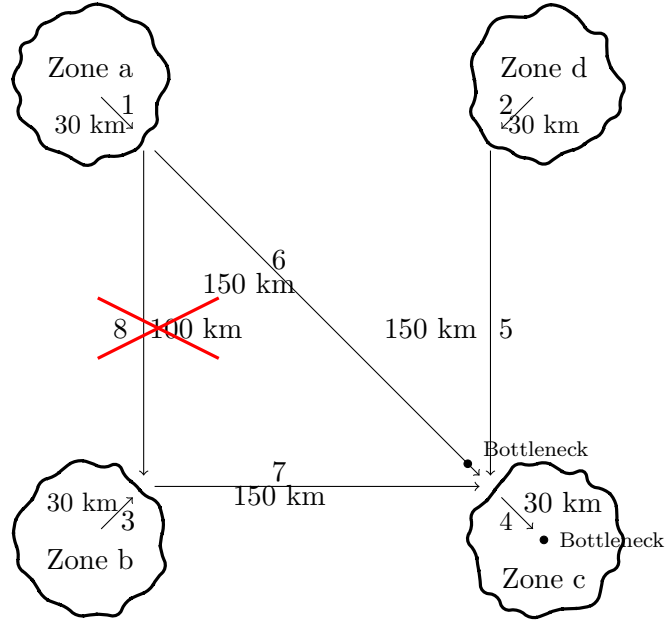
gets closed, making route-choice infeasible.

Figure 4.6b shows the comparison of MCL in the Zone c for Scenario 4 (route-choice without an incident) and Scenario 6 (route-choice with an incident). It may be noted that there is a jump in MCL at the time at which individuals choosing the alternate route (link 8 $\rightarrow$ link 7 $\rightarrow$ link 4) start arriving at Zone c. However, the jump is significantly lower for Scenario 6 as compared to Scenario 4 since route-choice is affected by the occurrence of an incident. It may also be noted that in Figure 4.6c, the time-varying accumulation in Zone c shows a similar pattern to the one described above (i.e., there is an increase in the accumulation for both scenarios, but the increase was much lower in Scenario 6). As a result, MCL proves to be a valid representation of the congestion pattern in Zone c.

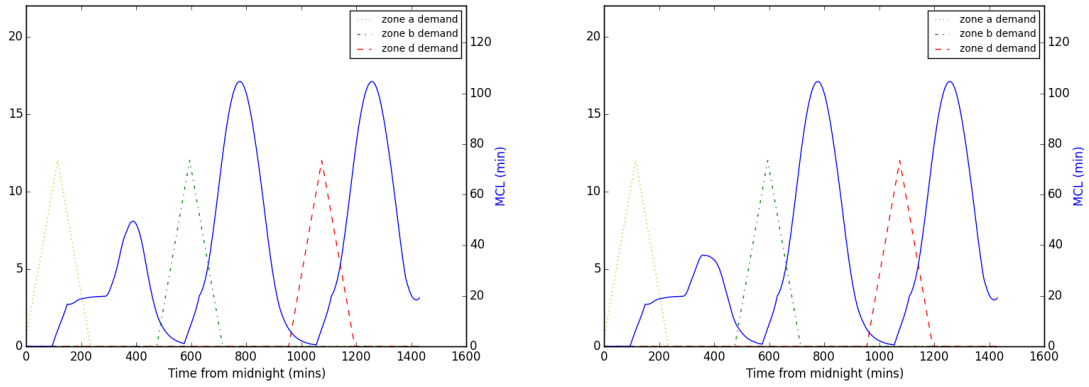
As seen before in Scenario 4, the key signal that helps in forecasting the congestion score is link accumulations, $C(t)$, in link 6, see Figure 4.6d. The presence of a ‘flat’ trend in link accumulations for link 6 indicates the occurrence of route-switching. However, in Scenario 6, that period is much smaller than that in Scenario 4, indicating that route-switching occurs for a much shorter period. Thus, one can expect a jump in MCL values similar to that in Scenario 4, but the magnitude of the jump is lower.

Table 4.1 summarizes the various physical processes that may impact MCL in the target zone, $\xi^z(t)$, and the respective components of network state, $X^z(t)$, that are sensitive to the occurrence of each physical process.

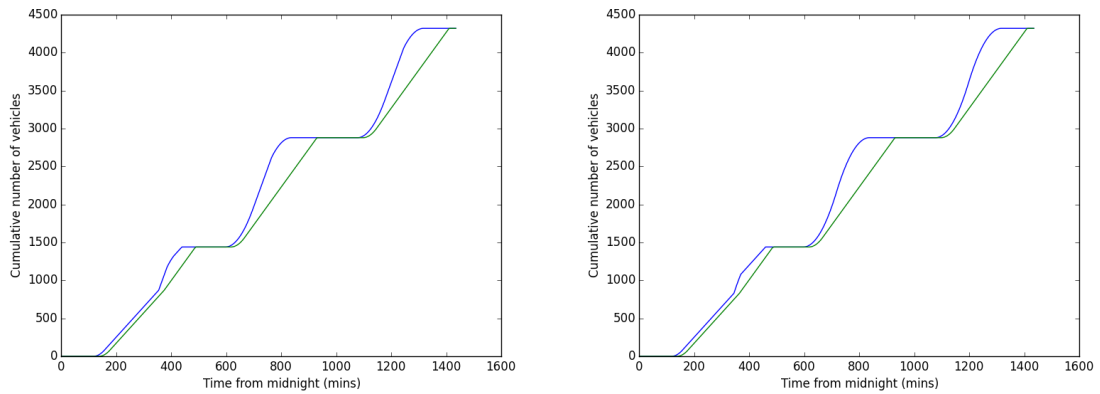
The analysis of MCL in varying traffic/queue propagation scenarios and the response of various input state variables (summarized in Table 4.1) suggests that the proposed model (Equation 3.5) with network state input as defined in Equation 3.4 is **sensitive to the various physical processes affecting congestion**.



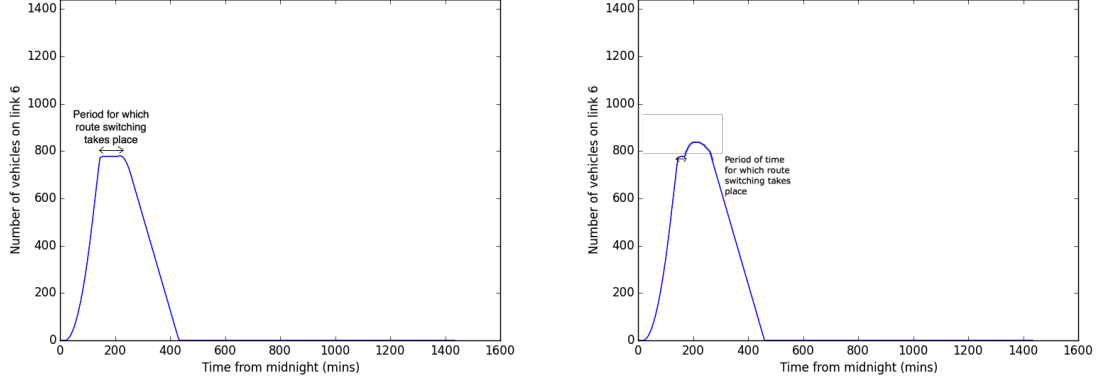
(a) Baseline Network With Arterial Bottleneck And Route-Choice And Link Closure



(b) The plot of demand (dotted line, primary y-axis) and congestion score (solid line, secondary y-axis) vs time for Scenario 4 (left) and Scenario 6 (right)



(c) Cumulative demand and departure curve in Zone c for Scenario 4 (left) and Scenario 6 (right)



(d) Number of vehicles on link 6 over time for Scenario 3 (left) and Scenario 6 (right)

Figure 4.6: Figures describing Scenario 6 and the effectiveness of MCL and network state for modeling neighborhood congestion.

Physical process affecting MCL	Responsive network state input
Time-varying O-Ds	O-D demand values
Coalescing O-Ds	O-D demand times
Congestion in preceding links	Link travel times in preceding links
Route-choice	Link travel time and accumulations in preceding links
Queue Spillover	Congestion score in neighboring zones
Incidents	Link accumulations in the links neighboring the incident

Table 4.1: A table summarizing the necessary conditions (i.e., physical processes that may affect MCL in the target zone, $\xi^z(t)$) and the components of network state, $X^z(t)$ that is responsive to the occurrence of a particular physical process

4.2 Initial Comparisons

A viable model must predict congestion more accurately than a competent baseline for varying demand scenarios and traffic conditions. Due to the absence of any existing baselines for predicting congestion in a neighborhood, the following baseline model is proposed:

- *1-Nearest Neighbor (1-NN) Baseline Model*

MCL at time t was predicted as the observed MCL at the same time on the previous day. It's equation is as follows:

$$\hat{\xi}^z(t + \Delta) = \xi^z(t), \quad (4.1)$$

where $\Delta = 1440 \text{ mins}$.

Even though the function governing the baseline model is extremely trivial to evaluate, it still provides a competitive prediction accuracy because of the typical daily cycles inherent to traffic patterns.

In addition to the various traffic/queue propagation scenarios within a day, comparisons made across multiple days also depend on the correlations between demands across days. The following demand scenarios across multiple days were proposed for the simulation experiments described in section 4.1:

(i) *Constant demand* (a):

Both the start time and magnitude of O-D demand from each of the Zones a, b, and d to Zone c were assumed to remain constant across all days

$$\begin{aligned} st_{t+1,z} &= \mu_{st,z}, \\ sl_{t+1} &= \mu_{sl,z}. \end{aligned}$$

(ii) *Varying magnitude* (b)

The peak O-D demand from each input zone to the target zone across multiple days was drawn from an i.i.d. Gaussian distribution with a mean value of 60 veh/min and a standard deviation of 6 veh/min (i.e., 10% of mean)

$$\begin{aligned} st_{t+1,z} &= \mu_{st,z}, \\ sl_{t+1,z} &= \mu_{sl,z} + \epsilon_{sl,z}, \quad \epsilon_{sl,z} \sim \mathcal{N}(0, \sigma_{sl,z}). \end{aligned}$$

(iii) *Varying start time* (c)

The start time for the O-D demand from each input zone to each target zone across multiple days was drawn from i.i.d Gaussian distributions. The mean start time was chosen as per the scenarios in section 4.1. The standard deviation was chosen as 30 mins.

$$\begin{aligned} st_{t+1,z} &= \mu_{st,z} + \epsilon_{st,z}, \quad \epsilon_{st,z} \sim \mathcal{N}(0, \sigma_{st,z}), \\ sl_{t+1,z} &= \mu_{sl,z}. \end{aligned}$$

(iv) *Varying magnitude and varying start time* (d)

The peak O-D demand across multiple days was drawn from an i.i.d. Gaussian distribution with a value of 60 veh/min and a standard deviation of 6 veh/min. The start time for each O-D was drawn from an i.i.d Gaussian distribution with the mean value as described in the scenarios in section 4.1 and a standard deviation of 30 mins.

$$\begin{aligned} st_{t+1,z} &= \mu_{st,z} + \epsilon_{st,z}, \quad \epsilon_{st,z} \sim \mathcal{N}(0, \sigma_{st,z}), \\ sl_{t+1,z} &= \mu_{sl,z} + \epsilon_{sl,z}, \quad \epsilon_{sl} \sim \mathcal{N}(0, \sigma_{sl}). \end{aligned}$$

(v) *Correlated demand* (e)

The start times for O-D demand were picked from i.i.d. Gaussian distribution. The mean start times for O-D demand from Zones a, b, and d were fixed at $t = 360$ mins, $t = 420$ mins and $t = 480$ mins respectively while the standard deviation was set at 30 mins. This was done to ensure that O-Ds coalesce. The peak O-D demand across multiple days was drawn from an i.i.d. Gaussian distribution with a mean value of 60 veh/min and a standard deviation of 6 veh/min.

$$\begin{aligned} st_{t+1,z} &= \mu_{st,z} + \epsilon_{st,z}, \quad \epsilon_{st,z} \sim \mathcal{N}(0, \sigma_{st,z}), \\ sl_{t+1,z} &= \mu_{sl,z} + \epsilon_{sl,z}, \quad \epsilon_{sl} \sim \mathcal{N}(0, \sigma_{sl}), \\ \mu_{st,z_3} &= k_{2,3} + \mu_{st,z_2} = k_{1,3} + \mu_{st,z_1}. \end{aligned}$$

In all the above equations:

$st_{t+1,z}$ is the start time of demand on day $(t + 1)$ for Zone z ,

$sl_{t+1,z}$ is the slope of demand on day $(t + 1)$ for Zone z ,

$\epsilon_{st,z}$ is the random fluctuation in the start time of demand at time $(t + 1)$ for Zone z (assumed Gaussian distribution with a mean zero and a standard deviation $\sigma_{st,z}$), $\epsilon_{sl,z}$ is the random fluctuation in the slope of demand at time $(t + 1)$ for Zone z (assumed Gaussian distribution with mean zero and standard deviation $\sigma_{sl,z}$), $k_{2,3}$ and $k_{1,3}$ are known constants.

For traffic scenarios 1-4 in section 4.1, the target variable for prediction was the MCL in the Zone c whereas, in traffic scenario 5, the target variable for prediction was the MCL in the Zone a. For Scenario 6, only one demand scenario (i.e., constant demand across all days) was considered. An incident closing link 8 could occur on any day with a probability of 0.5. The incident time was picked from a uniform distribution throughout the day.

The prediction accuracy of the deep learning model (Equation 3.5) was compared to that of the baseline 1-NN model. The hyper-parameters h and p (in Equation 3.5) were trained as follows:

- h was set as the free flow travel time from Zones a, b, and d to Zone c (i.e., 2 hrs).
- p was trained using cross-validation.

Each model was trained on a sufficient amount of data until convergence was achieved.

A sequence of actual MCL values was considered as *ground truth* and the Root Mean Squared Error (RMSE) of this vector was calculated from the vector of predicted MCL values at the same times. The following equations describe how the accuracy metric was calculated:

$$\begin{aligned}\xi^z &= [\xi^z(t+h), \xi^z(t+h+1), \dots, \xi^z(t+h+p)], \\ \hat{\xi}^z &= [\hat{\xi}^z(t+h), \hat{\xi}^z(t+h+1), \dots, \hat{\xi}^z(t+h+p)], \\ RMSE &= \|\xi^z - \hat{\xi}^z\|_2,\end{aligned}\tag{4.2}$$

where $RMSE$ is the Root Mean Squared Error.

The simulation results for the analysis of the performance of the proposed LSTM model against the baseline 1-NN model are summarized in Table 4.2.

The results show that the accuracy of the LSTM model was superior to that of the 1-NN in all scenarios where demand varied across multiple days. The only case where the accuracy was not superior was for the case where demand was constant (i.e., Scenario a). This is intuitive since the 1-NN model assumes no variation in demand across consecutive days. These simulation results suggest that **the LSTM model is able to respond to the physical processes and the demand fluctuations better than the baseline 1-NN model**¹.

¹An open source implementation of the algorithm on the sample scenarios was made available at: <https://github.com/sudatta0993/Dynamic-Congestion-Prediction>

T.S.	D.S.	n_{in}	p (mins)	n_{hid}	n_{lay}	α	n_{iter}	Batch	LSTM RMSE	1-NN RMSE
1	a	6	1440	1	1	10	100	1	0.165	0.0
1	b	6	1440	1	1	10	100	1	4.059	5.080
1	c	6	1440	1	1	10	200	1	7.229	18.703
1	d	6	1440	1	1	10	400	10	5.783	19.154
1	e	12	240	100	1	0.1	25000	10	4.898	9.026
2	a	6	1440	1	1	10	100	1	0.250	0.0
2	b	6	1440	1	1	10	100	1	5.263	5.408
2	c	6	1440	1	1	10	200	10	7.271	18.703
2	d	6	1440	1	1	10	400	10	5.487	19.154
2	e	12	240	100	1	0.01	25000	10	4.737	9.026
3	a	6	1440	1	1	10	100	1	0.152	0.0
3	b	6	1440	1	1	10	100	1	4.017	5.069
3	c	6	1440	1	1	10	200	10	7.173	18.703
3	d	6	1440	1	1	10	400	10	4.876	19.037
3	e	24	1440	100	10	0.1	200	10	8.322	13.957
4	a	6	1440	1	1	10	100	1	0.134	0.0
4	b	6	1440	1	1	10	200	1	4.357	4.737
4	c	6	1440	1	1	10	200	10	9.634	15.954
4	d	6	1440	1	1	10	200	10	10.204	14.815
4	e	28	1440	100	10	0.01	700	10	7.324	9.982
5	a	2	1440	1	1	10	100	1	0.045	0.0
5	b	6	1440	1	1	10	400	1	1.746	1.903
5	c	2	1440	1	1	10	300	1	1.407	2.182
5	d	6	1440	1	1	10	500	1	1.652	2.502
5	e	6	1440	100	10	0.01	200	1	3.703	4.179
6	a	6	1440	1	1	0.01	400	10	10.342	15.874

T.S.: Traffic Scenario Index (Section 4.1)

D.S.: Demand Scenario Index (Section 4.2)

n_{in} : Number of input features for LSTM model

n_{hid} : Number of hidden layers for LSTM model

n_{lay} : Number of stacked LSTM layers for LSTM model

α : Learning rate for LSTM model

n_{iter} : Number of iterations for LSTM model

Table 4.2: A table describing the parameter set for each LSTM model developed and comparison of Root Mean Squared Error (RMSE) for the LSTM model and the 1-NN model

Chapter 5

Model Robustness

The deep learning model's accuracy is compared with that of the baseline model under adverse situations:

- correlated O-D demands across multiple days,
- noisy input observations, and
- partially observed network state.

They either aid the baseline model or adversely impact the deep learning model. Methods are suggested to help the deep learning model with increasing adversity.

All experiments in this chapter are conducted for:

- the baseline network defined in Figure 4.1a, and
- the link fundamental diagrams described in Figure 4.1b.

5.1 Correlated O-D Demand Across Days

Experimental Setup

In section 4.2, a triangular demand function was assumed to model O-D demand variation from a single zone on a given day. The demand pattern was defined by two parameters, namely the start time of demand ($st_{t,z}$) and the slope of the demand curve ($sl_{t,z}$). For each case, no correlation was assumed between the start times or slopes on days t and $(t + 1)$. In the past literature, models for O-D demand estimation have assumed autoregressive structures relating demands on consecutive days (Ashok and Ben-Akiva 1993 Ashok and Ben-Akiva 2000 Zhou and Mahmassani 2007). Following these examples, a simple AR-1 process is defined to introduce correlation between O-D demand parameters. Equations governing the parameters are as follows:

$$st_{t+1,z} = \rho_{st,z} * st_{t,z} + (1 - \rho_{st,z}) * \mu_{st,z} + \epsilon_{st,z}, \quad \epsilon_{st,z} \sim \mathcal{N}(0, \sigma_{st,z}), \quad (5.1)$$

$$sl_{t+1,z} = \rho_{sl,z} * sl_{t,z} + (1 - \rho_{sl,z}) * \mu_{sl,z} + \epsilon_{sl,z}, \quad \epsilon_{sl,z} \sim \mathcal{N}(0, \sigma_{sl,z}), \quad (5.2)$$

where ρ_{st} and ρ_{sl} are auto-correlation parameters for start time and slope of demand respectively

A higher value of ρ is hypothesized to favor the baseline 1-NN model. The parameters ρ_{st} , ρ_{sl} , σ_{st} , and σ_{sl} were varied over multiple experiments and the model accuracy of the deep learning model was compared with that of the baseline 1-NN model for each experiment. Threshold values were determined for the set of parameters for which the LSTM model performs better.

Experimental Results

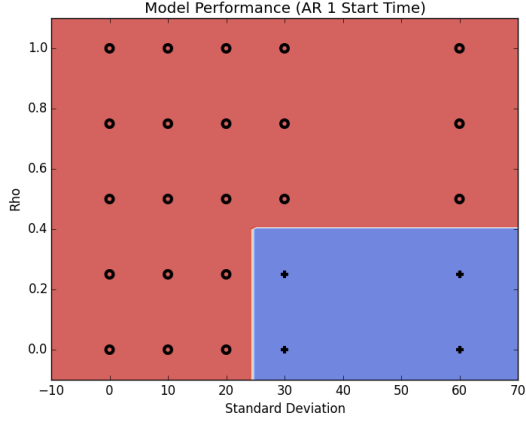
First, the impact of correlated start times was studied with the slope remaining constant. This was done by varying the parameters $\rho_{st,z}$, and $\sigma_{st,z}$ for demand from Zones a, b, and d to Zone c (Equation 5.1). For simplicity, these parameters were assumed to be equal for Zones a, b, and d. Higher variation in demand across zones favors the deep learning model more than the baseline model. Therefore, the assumption of equal demand across zones leads to a lower bound on the relative superiority of the deep learning model.

In Figure 5.1, the parameters governing the demand-generating process were plotted on the x and y axes and an indicator of the better performing model was plotted using symbols “+” and “o” (“+” signifies that LSTM model outperformed 1-NN model and “o” signifies the opposite). A possible efficiency boundary for the LSTM model was also constructed using a Decision Tree (DT) classifier. The area labeled blue represents cases for which the deep learning model performs better. In Figure 5.1(a), the parameters h and p for the LSTM model (Equation 3.5) were set as 2 hours and 24 hours respectively, implying that the model tries to predict MCL in the Zone c up to 26 hours in the future. As seen from the figure, the LSTM model performed poorly for several combinations of $\rho_{st,z}$ and $\sigma_{st,z}$. In Figure 5.1(b), the LSTM model predicted only up to 4 hours in the future. We can see that by changing the hyper-parameter p , the LSTM model performed better over a wider range of simulation scenarios. This suggests that the **hyper-parameter p can be tuned to improve model robustness when O-D demands are correlated across days.**

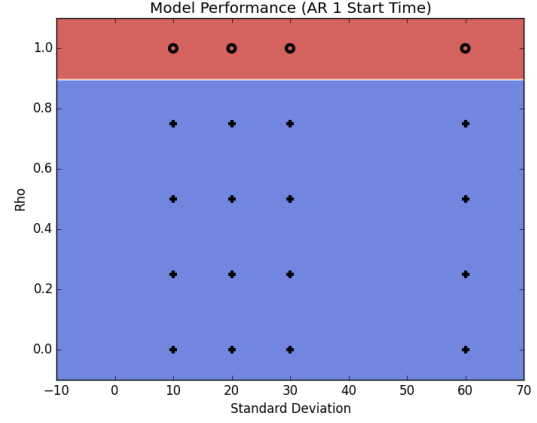
Next, the impact of correlated slopes was studied with the start time remaining constant. This was done by varying the parameters $\rho_{sl,z}$ and $\sigma_{sl,z}$ (Equation 5.2). For simplicity, these parameters were assumed to be equal for Zones a, b, and d which lead to a lower bound on the relative superiority of the deep learning model.

As in the experiments above, the indicators for relative performance of the deep learning model versus the baseline 1-NN model (i.e., symbols “+” and “o”) were plotted for two scenarios, i.e., (a) predictions up to 26 hours in the future and (b) predictions up to 4 hours in the future (Figure 5.2). In this case, we don’t see any drastic improvement by changing the prediction horizons as in the case of varying start times. This suggests that **MCL values are more sensitive to variation in the start time than variation in the slope.** This is intuitive since start times greatly govern the fact as to whether there is overlap in the arrival times for traffic originating from different zones, and thus, potentially influence the MCL pattern to a greater extent than slopes¹.

¹For the analysis and plots when both start time and slope were varied, please refer to Appendix C.

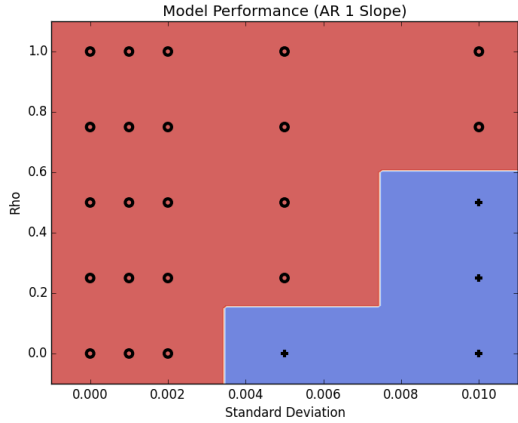


(a) $h = 2 \text{ hrs}$, $p = 24 \text{ hrs}$

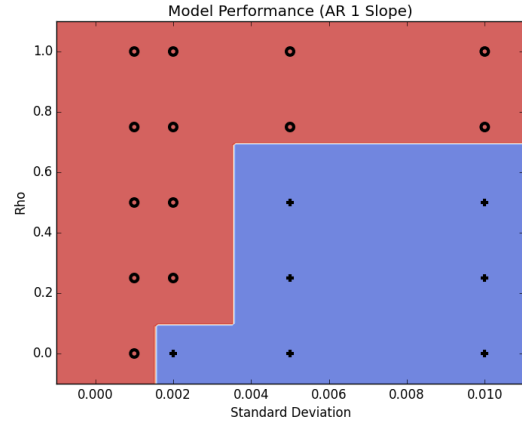


(b) $h = 2 \text{ hrs}$, $p = 2 \text{ hrs}$

Figure 5.1: Efficiency bounds for the prediction accuracy of the LSTM model vs the 1-NN model as a function of the amount of correlation in start times of demand and the amount of statistical fluctuation. The points labeled “+” are parameter values for which the LSTM model performed better and the points labeled “o” are parameter values for which 1-NN model performed better. A possible efficiency boundary was constructed using a Decision Tree Classifier.



(a) $h = 2 \text{ hrs}$, $p = 24 \text{ hrs}$



(b) $h = 2 \text{ hrs}$, $p = 2 \text{ hrs}$

Figure 5.2: Efficiency bounds for the prediction accuracy of the LSTM model vs the 1-NN model as a function of the amount of correlation in the slope of demand and the amount of statistical fluctuation in the slope. The points labeled “+” are parameter values for which the LSTM model performed better and the points labeled “o” are parameter values for which 1-NN model performed better. To illustrate the boundary, a Decision Tree (DT) classifier model was fit on the points.

5.2 Noisy Observations

Experimental Setup

This experiment is aimed to study the impact of measurement-noise on the accuracy of the predictions. The observed input data was assumed to be stochastic to simulate the effect of noise. The following distribution was assumed:

$$O^z(t) = X^z(t) + X^z(t) \odot \epsilon^z, \quad \epsilon^z \sim \mathcal{N}(\mathbf{0}, \Sigma^z), \quad (5.3)$$

where:

ϵ^z is a vector of random noise corrupting the input data (assumed to be drawn from a Multivariate Gaussian distribution with mean $\mathbf{0}$ and covariance matrix Σ^z),

$\odot$ represents element-wise multiplication of vectors.

For simplicity, no cross-correlation was assumed among the noisy variables and the relative noise in each dimension was assumed to be equal. Therefore, the expression for the covariance of the noise term can be simplified to:

$$\Sigma^z = \sigma^z * I, \quad (5.4)$$

where:

σ^z is the noise parameter equal to (standard deviation/mean) of input data in each dimension.

The assumption of equal noise in each dimension leads to a lower-bound on the maximum noise tolerance of the proposed model. The amount of noise impacting input data can be varied by changing the value of σ^z . The break-even σ^z value can be evaluated where the prediction accuracy of the deep learning model equals that of the 1-NN baseline model.

Giles, Lawrence, and Tsoi (2001) suggest two methods to increase the noise tolerance of neural networks in the context of time-series prediction, namely:

- Remove noisy inputs
(*Stdev/Mean*) ratio gives a proxy for the amount of noise in a particular input. A threshold (*Stdev/Mean*) ratio can be set above which the input may be discarded.
- Reduce batch size
A key step during training of neural networks is calculations of gradients for optimizing the weights. Typically, the gradients are calculated from several batches of data. Reduction in the batch size limits the amount of noise propagating to the gradients from each batch of data. This technique assumes that noise from several independent batches may cancel each other and thus lead to accurate gradients.

The effect of removing inputs may have a negative effect on the accuracy if the input removed is important. This idea is discussed further in Section 5.3 and Section 6.1. In this experiment, the effect of reduction in the batch size on the noise tolerance of the deep learning model is tested.

Experimental Results

The experiments assumed that:

- the O-D demand within a day varies as per Figure 4.1c and
- the O-D demand across multiple days varies as per Scenario d in Section 4.2

The simulation results are summarized in Figure 5.3. On the x-axis, σ^c was plotted on a log scale and on the y-axis the RMSE value for LSTM model (blue) and 1-NN model (red) were plotted. The dotted-cum-dashed blue curve represents a batch size of 10, the dashed blue curve represents a batch size of 100 and the dotted blue curve represents a batch size of 1.

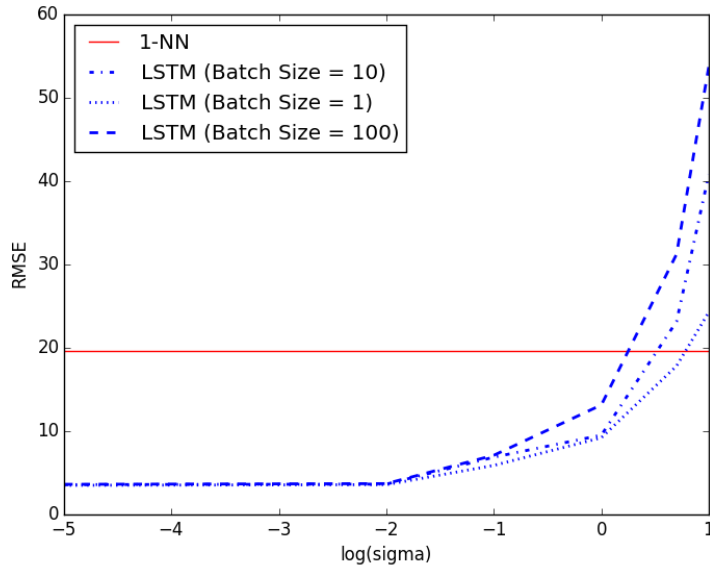


Figure 5.3: Impact of noise in input data on the prediction accuracy of the LSTM model for different batch sizes

The plot shows that reducing the batch size has a greater effect on the model accuracy when the amount of noise is high. The approximate break-even value of σ^c (i.e., the value of σ^c where the red curve intersects each blue curve) for various batch sizes are:

- 1.8 for batch size = 100,
- 4.1 for batch size = 10 and
- 8.2 for batch size = 1.

A higher σ^c value indicates more robustness against measurement noise. This suggests that **reduction in the batch size increases the noise tolerance of the deep learning model.**

5.3 Partially Observed Network State

Experimental Setup

Equation 3.4 defined the network state vector, $X(t)$, that is used as an input to the deep learning model. For prediction of $\xi^z(t)$, a subset of relevant inputs (i.e., $X^z(t)$) may be chosen. In a real-world setting, all relevant inputs may not be observable. Let the observable inputs be $O^z(t) \subseteq X^z(t)$. The prediction accuracy is a function of the set of elements in $O^z(t)$. This experiment aimed to determine the critical elements for accurate predictions. This was done by testing various combinations of the elements present in $O^z(t)$ and comparing the prediction accuracy in each scenario to the case when $O^z(t) = X^z(t)$. The critical inputs determined through this process were then compared to those known from intuition.

Experimental Results

The experiments assumed that:

- the O-D demand within a day varies as per Figure 4.1c and
- the O-D demand across multiple days varies as per Scenario d in Section 4.2

The following combinations of inputs in $O^c(t)$ were tested:

$$(i) O^c(t) = \begin{bmatrix} D_{a,c} \\ D_{b,c} \\ D_{d,c} \end{bmatrix}, (ii) O^c(t) = \begin{bmatrix} D_{b,c} \\ D_{d,c} \end{bmatrix}, (iii) O^c(t) = \begin{bmatrix} D_{a,c} \\ D_{d,c} \end{bmatrix} \text{ and } (iv) O^c(t) = \begin{bmatrix} D_{a,c} \\ D_{b,c} \end{bmatrix}$$

Prediction accuracy was evaluated for the time intervals of 6-8 AM, 2-4 PM, and 8-10 PM. Table 5.1 displays the RMSE value between the ground truth and predicted MCL values in each scenario.

Prediction Inputs Omitted \ Time	6-8 AM	2-4 PM	8-10 PM
None	6.274	3.359	5.281
Zone a demand	45.283	3.360	5.570
Zone b demand	6.723	42.457	5.345
Zone d demand	6.186	3.381	52.843

Table 5.1: Root Mean Squared For LSTM model for various scenarios with omitted inputs and different prediction times

Intuition suggests that the demand in Zone a is critical for predicting MCL in Zone c between 6-8 AM, the demand in Zone b is critical for predicting MCL in Zone c between 2-4 PM and the demand in Zone d is critical for predicting MCL in Zone c between 8-10 PM. The aim of this experiment was to corroborate the fact that these inputs are critical for accurate prediction of MCL for the deep learning model as well. In case the deep learning model predicts MCL accurately even in the absence of critical inputs, we can conclude that the model has not learned the required features for prediction of congestion. However, Table 5.1 does not indicate that. The table suggests that **the absence**

of information regarding a critical input can be catastrophic to the accuracy in prediction. In more complex settings, we may not be able to identify critical inputs intuitively. This motivates a deeper understanding of the importance of various inputs at a given space and time. A technique for doing this is described in section 6.1 ².

²In scenarios where the effect of one input is not dominating, the impact of the removal of any one input may not be as significant. For details, please refer to Appendix D

Chapter 6

Model Application - Dynamic Congestion Control

Experiments performed in section 5.3 indicate that it is important to discover the critical input parameters required for accurate MCL prediction. Since we may want to implement controls that provide maximum “bang-for-the-buck”, information about the relative importance of various inputs plays a crucial role in defining dynamic congestion controls. Yet another benefit of deriving the relative importance of inputs is that it allows us to understand the inner workings of neural network models in a better way rather than treating it as a “black box”. In this chapter, a procedure based on *Neural Attention Models* is discussed to infer the importance of inputs. Thereafter, a case study is analyzed where model predictions and the importance of inputs are used to frame a dynamic control strategy. The proposed strategy is as follows.

An app-based dynamic congestion toll is calculated at the beginning of each trip based on the marginal impact on the predicted congestion. This toll is aimed at discouraging travel into a target neighborhood at times when the rate of inflow exceeds capacity. Thus, this strategy achieves a similar effect to a cordon metering scheme (Daganzo 2007) but relieves the commuter of the uncertainty of being denied entry into the destination neighborhood midway during a trip. The reduction in cumulative congestion impact during the course of a day is shown as a consequence of the toll through simulation experiments.

All experiments in this chapter are conducted for:

- the baseline network defined in Figure 4.1a,
- the link fundamental diagrams described in Figure 4.1b, and
- O-D demand variation within a day as per Figure 4.1c

6.1 Neural Attention Model

Mathematical Formulation

Attentions are weights assigned to each input in addition to the standard neural network weights. These attentions serve as proxies for the contributions made by the inputs while predicting the output. For the purpose of modeling attentions, the sequence of predictions

produced by the trained model is considered the *baseline prediction*. These predictions may be derived from Equation 3.5. The output is:

$$\hat{\xi}^z = [\hat{\xi}^z(t+h+p), \dots, \hat{\xi}^z(t+h)]. \quad (6.1)$$

The attention weights can be derived through a procedure described by Ramanishka et al. (2017). The procedure involves constraining the input set, making predictions based on the constrained input set and comparing these predictions to the baseline predictions. When the input set is constrained, the unobserved input values are estimated through prior distributions based on previously observed data. Let the prediction vector evaluated based on the constrained input be $\tilde{\xi}^z$. The value of the loss for a constrained input set is calculated as the deviation (in L_2 norm) between $\tilde{\xi}^z$ and $\hat{\xi}^z$. The detailed procedure is described below.

Equation B.2 (see Appendix B) shows how the hidden state is calculated for an LSTM model. The hidden state value calculated at time t when all inputs are present is hid_t . When the input set is constrained such that only information about input f is available at lag l from the time of prediction, the hidden state value is recalculated. As a result of this modification, all hidden state values for lags 1 to $(l-1)$ also get affected. These modified hidden states are represented as $\widehat{hid}_t$. The hidden state value is recalculated again for the case when no information for any input is available at lag l . Once again, all hidden state values for lags 1 to $(l-1)$ get affected. These modified hidden state values at time t are represented as $\widetilde{hid}_t$. The values of hid_t , $\widetilde{hid}_t$ and $\widehat{hid}_t$ are used to calculate the *temporal attention* (i.e., the importance of a given lag, l) and the *spatial attention* (i.e., the importance of a given input f at lag l).

Temporal Attention

The loss due to the absence of all inputs at lag l is evaluated to determine the temporal attention value at lag l . This value is calculated as the difference in the values of the loss for two constrained input sets:

- (i) a constrained input set where hidden states from lag p to lag $(l-1)$ are absent and
- (ii) a constrained input set where hidden states from lag p to lag l are absent

The following equations summarize the process:

$$\begin{aligned} \hat{\xi}_{1,l}^z &= \sum_{i=1}^{l-1} \{W_{\text{LSTM}_i} * \widehat{hid}_{t-h-i} + B_{\text{LSTM}_i}\} + \sum_{i=l}^p \{W_{\text{LSTM}_i} * hid_{t-h-i} + B_{\text{LSTM}_i}\}, \\ \hat{\xi}_{2,l}^z &= \sum_{i=1}^l \{W_{\text{LSTM}_i} * \widehat{hid}_{t-h-i} + B_{\text{LSTM}_i}\} + \sum_{i=l+1}^p \{W_{\text{LSTM}_i} * hid_{t-h-i} + B_{\text{LSTM}_i}\}, \\ \text{Loss}_{1,l} &= \|\hat{\xi}^z - \hat{\xi}_{1,l}^z\|_2, \\ \text{Loss}_{2,l} &= \|\hat{\xi}^z - \hat{\xi}_{2,l}^z\|_2, \\ \text{Loss}_l &= \text{Loss}_{1,l} - \text{Loss}_{2,l}, \end{aligned} \quad (6.2)$$

where:

$\widehat{hid}_t$ represents the estimated value of hid_t through a prior distribution based on all observed values of hid_t .

The attention value at lag l is calculated as the normalized loss over all lags as follows:

$$att_l = \begin{cases} \frac{Loss_l}{\sum_{i=1}^p Loss_i}, & \sum_{i=1}^p Loss_i \neq 0 \\ \frac{1}{p}, & \sum_{i=1}^p Loss_i = 0. \end{cases} \quad (6.3)$$

The value of att_l gives a proxy for the importance of the input at a time lag of l from the time of making the predictions.

Spatial Attention

The hidden state values at lag l are calculated when the input set is constrained to only input f . This is done by modifying Equation B.2 (see Appendix B) as follows:

$$\begin{aligned} \widetilde{fg}_{t-l} &= \sigma(W_{f,t-l}^T[h_{t-l-1}, \hat{x}_{t-l}] + b_{f,t-l}), \\ \widetilde{i}_{t-l} &= \sigma(W_{i,t-l}^T[h_{t-l-1}, \hat{x}_{t-l}] + b_{i,t-l}), \\ \widetilde{o}_{t-l} &= \sigma(W_{o,t-l}^T[h_{t-l-1}, \hat{x}_{t-l}] + b_{o,t-l}), \\ \widetilde{c}_{t-l} &= fg_{t-l} \odot \widetilde{c}_{t-l-1} + \widetilde{i}_{t-l} \odot \tanh(W_{c,t-l}^T[h_{t-l-1}, \hat{x}_{t-l}] + b_{c,t-l}), \\ \widetilde{hid}_{t-l} &= \widetilde{o}_{t-l} \odot \tanh(\widetilde{c}_{t-l}), \end{aligned} \quad (6.4)$$

where:

$$\hat{x}_t = f_t * e_f + \sum_{f' \neq f} \hat{f}'_t * e_{f'},$$

f_t is the value of data corresponding to input f at time t ,

$\hat{f}_t$ is the estimated value of data for input f at time t based on the prior distribution of observed values, and

e_f is a unit vector in input space in the direction of input f .

All hidden states between lags 1 and $(l-1)$ are recalculated by using Equation 6.4 recursively for each lag. Then, the steps described in section 6.1 for calculating the value of the loss at lag l may be repeated. The following equations summarize the process:

$$\begin{aligned} \hat{\xi}_{1,f,l}^z &= \sum_{i=1}^{l-1} \{W_{\text{LSTM}_i} * \widehat{hid}_{t-h-i} + B_{\text{LSTM}_i}\} + \sum_{i=l}^p \{W_{\text{LSTM}_i} * hid_{t-h-i} + B_{\text{LSTM}_i}\}, \\ \hat{\xi}_{2,f,l}^z &= \sum_{i=1}^l \{W_{\text{LSTM}_i} * \widetilde{hid}_{t-h-i} + B_{\text{LSTM}_i}\} + \sum_{i=l+1}^p \{W_{\text{LSTM}_i} * hid_{t-h-i} + B_{\text{LSTM}_i}\}, \\ \text{Loss}_{1,f,l} &= \|\hat{\xi}^z - \hat{\xi}_{1,f,l}^z\|_2, \\ \text{Loss}_{2,f,l} &= \|\hat{\xi}^z - \hat{\xi}_{2,f,l}^z\|_2, \\ \text{Loss}_{f,l} &= \text{Loss}_{1,f,l} - \text{Loss}_{2,f,l}. \end{aligned} \quad (6.5)$$

The spatial attention value may be calculated as:

$$att_{f,l} = \begin{cases} \frac{\text{Loss}_{f,l}}{\sum_{k \in F} \sum_{i=1}^p \text{Loss}_{k,i}}, & \sum_{k \in F} \sum_{i=1}^p \text{Loss}_{k,i} \neq 0 \\ \frac{1}{p*N}, & \sum_{k \in F} \sum_{i=1}^p \text{Loss}_{k,i} = 0, \end{cases} \quad (6.6)$$

where:

F is the set of all inputs,

N is the dimensionality of the input set.

The value of $att_{f,l}$ represents the importance of input f at lag l from the time of making predictions.

Experimental Results

The analysis of the impact of partially observed input in Section 5.3 confirmed the intuition that the demand in Zone a is critical for making predictions between 6-8 AM, the demand in Zone b is critical for making predictions between 2-4 PM and the demand in Zone d is critical for making predictions between 8-10 PM (Table 5.1). This experiment aims to draw the same conclusion through the Neural Attention Model-based proxies of input variable importance.

The prediction horizon for each prediction interval is 2 hrs. Figure 6.1 shows a heatmap of the temporal attention values, att_l , for 24 lags of duration 5 mins each. Figure 6.2 shows a heatmap of the spatial attention values, $att_{f,l}$ for six inputs, namely the three O-D demands and their corresponding first differences, for the same lags.

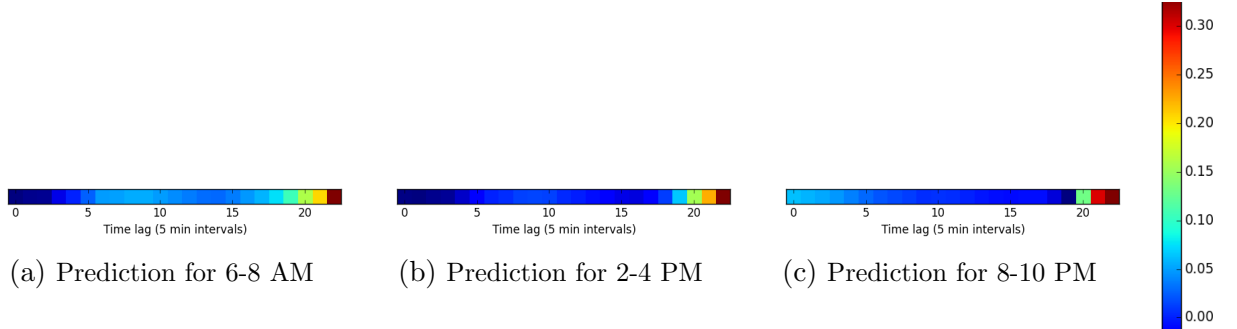


Figure 6.1: Temporal attention heatmap

We can observe that in all the temporal attention heatmaps, the most important time lag is at lag 24 (i.e., lag time = 2 hrs). This is intuitive since the free flow travel time from each demand zone to Zone c is 2 hrs and the demands have been chosen in such a way that there is no congestion delay. Due to this correlation, any trip beginning at time t affects the congestion score in Zone c at $(t + 2 \text{ hrs})$.

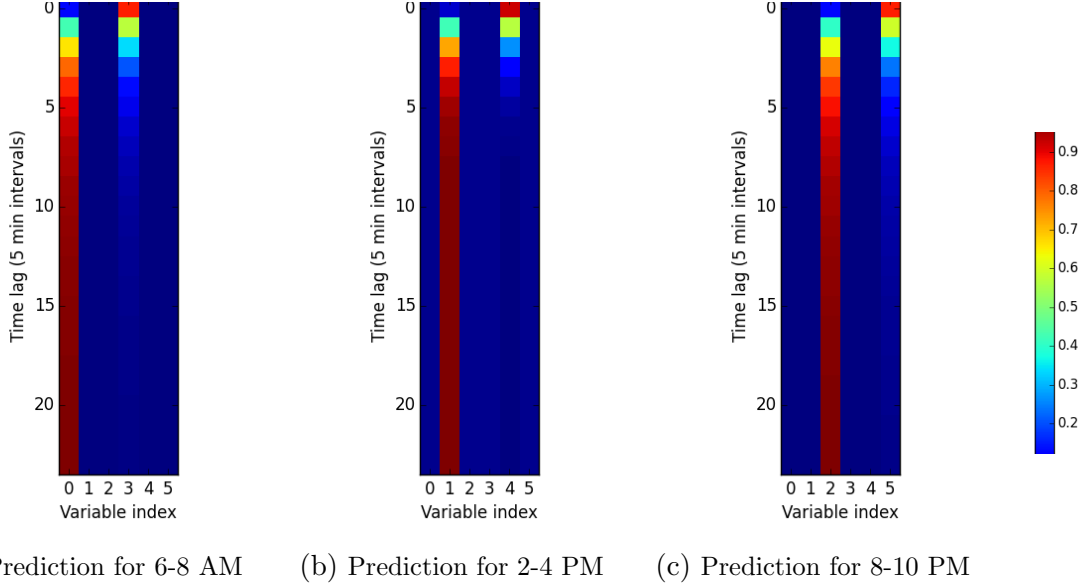


Figure 6.2: Spatial attention heatmap - Variable definitions: Index 0 = Zone a demand, Index 1 = Zone b demand, Index 2 = Zone d demand, Index 3 = First difference of Zone a demand, Index 4 = First difference of Zone b demand, Index 5 = First difference of Zone d demand

As the spatial attention heatmaps show, the only input variables that impact the congestion scores during 6-8 AM were variables corresponding to indices 0 and 3 (i.e., the demand in Zone a and its first difference respectively). The relative importance for higher lag values is higher for variable index 0, whereas the relative importance for higher lag values is lower for variable index 3. This is because the congestion curve is concave in nature (Figure 4.1c) which reverses the sign of the second derivative from that of the first derivative. The simulation results for the predictions made at 6-8 AM can be extended to the predictions made at 2-4 PM and 8-10 PM with Zone b and Zone d acting as the key input zones respectively (Figures 6.1 and 6.2). Thus, the simulation experiment suggests that the **Neural Attention Model approximates the spatio-temporal importance of variables well**¹. In the next section, the derived attentions are used for the purpose of proposing a predictive dynamic congestion control.

¹For analysis of the Neural Attention Model-based framework on a larger, less-idealized network, please refer to Appendix E.

6.2 App-Based Dynamic Congestion Pricing

Modeling Framework

Consider a situation where we would like to charge agents as they are departing for a trip based on their predicted impact on congestion at the target zone (Figure 6.3). Assume that any agent must enter his/her destination on an app before leaving. Based on this entry, the app displays a toll amount. This toll amount may be paid only at the time of entry to the destination zone (i.e., if you change your mind during the trip, you will have to re-enter your destination and the toll amount may be modified accordingly). But unless the agent displays the payment of this toll, he/she is not allowed entry into the destination zone. The toll is aimed at deterring agents arriving at the target zone during peak hours. The toll amount charged in this example is only a function of predicted impact on one particular target zone (i.e., Zone z). The analysis may be extended to multiple target zones in future studies. The overall objective is defined and the system-optimal conditions are obtained by solving an optimization problem. A mechanism is provided to approximate those conditions either based on a deep learning model or through O-D demand prediction. The approximation methods are compared through a simulation experiment.

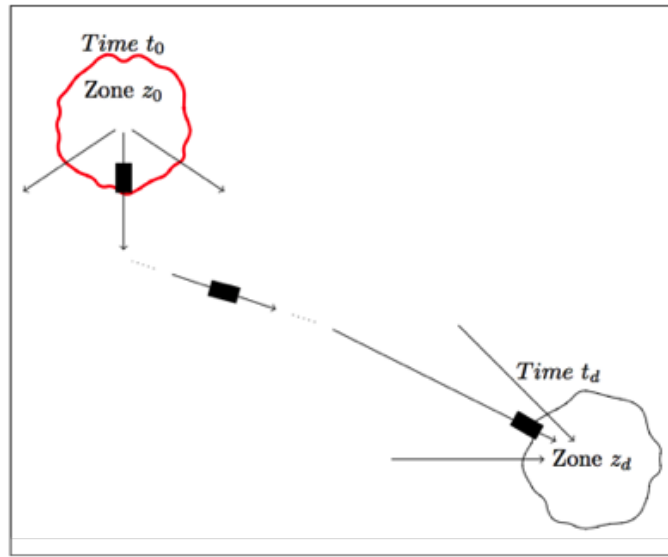


Figure 6.3: Overview of setting for dynamic congestion pricing at source: Agents starting their trips at Zone z_0 at time t_0 are charged congestion toll in proportion to the predicted marginal impact on congestion during their trip. For this example, we consider only the predicted marginal impact on congestion in Zone z_d due to this trip. The agents must display the payment of this toll when they reach Zone z_d at time t_d .

Conditions for Optimal Toll

The objective of the congestion toll is to minimize cumulative MCL while maintaining overall demand. A trip departing Zone z_i at time t_i is assumed to arrive at the periphery of Zone z at time t . The following optimization problem is formulated with optimal values giving us the target O-D demands at any time:

$$\begin{aligned}
& \underset{D^{z_i,z}(t)}{\text{Minimize}} \int_{t_{start}}^{t_{end}} \xi^z(t) dt \\
& \text{s.t.} \quad \sum_{i \in \{1,2,\dots,Z\}} \int_{t_{start}}^{t_{end}} D^{z_i,z}(t) dt = \text{Constant},
\end{aligned} \tag{6.7}$$

where:

$\xi^z(t)$ represents the MCL value at time t , which is a function of the O-D demands from all Zones z_i at times t_i ,

$D^{z_i,z}(t)$ represents the O-D demand from z_i to z at time t , and

t_{start} , t_{end} are the start and end time intervals of analysis respectively.

The optimization problem in Equation 6.7 is non-convex due to the non-convex nature of the objective and constraint functions. We may achieve a lower bound on the optimal objective by solving the Lagrangian dual version of the problem. From simulations, it is later shown that the lower bound is nearly sharp. The Lagrangian is formed as follows:

$$\begin{aligned}
\mathcal{L} &= \int_{t_{start}}^{t_{end}} \xi^z(t) dt + \nu \left(\sum_{i \in \{1,2,\dots,Z\}} \int_{t_{start}}^{t_{end}} D^{z_i,z}(t) dt - \text{Constant} \right) \\
&= \int_{t_{start}}^{t_{end}} \{ \xi^z(t) + \nu \left(\sum_{i \in \{1,2,\dots,Z\}} D^{z_i,z}(t) - \text{Constant} \right) \} dt.
\end{aligned} \tag{6.8}$$

The first order condition for minimizing the Lagrangian function gives the following condition:

$$\begin{aligned}
& \frac{\partial \mathcal{L}}{\partial D^{z_i,z}(t)} = \int_{t_{start}}^{t_{end}} \left\{ \frac{\partial \xi^z(t)}{\partial D^{z_i,z}(t)} + \nu \right\} dt = 0 \\
& \Rightarrow \int_{t_{start}}^{t_{end}} \frac{\partial \xi^z(t)}{\partial D^{z_i,z}(t)} dt = \text{Constant}, \quad \forall i \in \{1, 2, \dots, Z\},
\end{aligned} \tag{6.9}$$

where:

$\frac{\partial \xi^z(t)}{\partial D^{z_i,z}(t)}$ represents the marginal impact of O-D demand from z_i to z on MCL at time t .

The physical interpretation of the optimal condition is that the impact of all O-D demands on MCL over the analysis period should be equal. One must ensure that **the impact of all O-D demands on MCL is constant over space and time** to make the condition hold over any analysis period.

An expression must be derived to estimate $\frac{\partial \xi^z(t)}{\partial D^{z_i,z}(t)}$. A few terms are defined to facilitate this derivation, namely:

(i) $\beta^{imp,z}(t)$:

This term represents the marginal impact or the rate of change in MCL in z per unit arrival at time t . An expression is derived to evaluate $\beta^{imp,z}(t)$.

The cumulative number of arrivals in Zone z at time t is $\sum_{i \in A^z} A_i(t)$ (see Section 3.1). $\beta^{imp,z}(t)$ can be represented as:

$$\begin{aligned}\beta^{imp,z}(t) &= \frac{\partial \xi^z(t)}{\partial \sum_{i \in A^z} (A_i(t))} \\ &= \frac{\partial \xi^z(t) / \partial t}{\partial \sum_{i \in A^z} (A_i(t)) / \partial t} \\ &= \frac{\beta_{\xi,z}(t)}{\partial \sum_{i \in A^z} (A_i(t)) / \partial t},\end{aligned}\tag{6.10}$$

where:

$\beta_{\xi,z}(t)$ is the rate of change in MCL for Zone z at time t .

The denominator can be further simplified following the assumption that the number of exogenous trips to z are either nearly constant over time or are very small compared to the number of endogenous trips. The cumulative rate of arrivals can be represented in terms of the trip completion and the rate of change of accumulation as $(n_i(t))$ and $L_i(t)$ are as defined in Section 3.1):

$$\begin{aligned}\frac{\partial \sum_{i \in A^z} (A_i(t))}{\partial t} &= \frac{\partial \sum_{i \in A^z} n_i(t)}{\partial t} + \frac{\partial (\sum_{i \in A^z} L_i(t))}{\partial t} \\ &\approx \frac{\partial \sum_{i \in A^z} n_i(t)}{\partial t} + \frac{\partial (\sum_{i \in A^z} E_i(t))}{\partial t} \\ &= \frac{\partial \sum_{i \in A^z} n_i(t)}{\partial t} + T^z(t) \\ &= \beta^{n,z}(t) + T^z(t),\end{aligned}\tag{6.11}$$

where $\beta^{n,z}(t)$ is the rate of change in accumulation in Zone z at time t .

This leads to the following expression of $\beta^{imp,z}(t)$:

$$\beta^{imp,z}(t) = \frac{\beta_{\xi,z}(t)}{\beta_{n,z}(t) + T^z(t)}.\tag{6.12}$$

In practice, the values of $\beta_{\xi,z}(t)$ and $\beta_{n,z}(t)$ are unknown when the toll is charged. Therefore, the value of $\beta^{imp,z}(t)$ must be forecasted into the future.

(ii) $\underline{p_{z_i,z}(t_i, t)}$:

This represents the probability that a trip departing from z_i at time t_i arrives at z at time t . In practice, $p_{z_i,z}(t_i, t)$ is unknown at the time when the toll is charged. Therefore, the value must be forecasted into the future.

Both of the derived terms are independent and are assumed to affect $\frac{\partial \xi^z(t)}{\partial D^{z_i,z}(t)}$ linearly. Therefore, the derived expression is as follows:

$$\frac{\partial \xi^z(t)}{\partial D^{z_i,z}(t)} = p_{z_i,z}(t_i, t) * \beta^{imp,z}(t).\tag{6.13}$$

A toll must be charged in proportion to $\frac{\partial \xi^z(t)}{\partial D^{z_i,z}(t)}$ to ensure that the optimal condition in Equation 6.9 holds. One possible toll rate is as follows:

$$Toll_{z_i,t_i} = k_{z_i,t}(t_i, t) \frac{\partial \xi^z(D^{z_i,z}(t))}{\partial D^{z_i,z}(t)} = k_{z_i,t}(t_i, t) * p_{z_i,z}(t_i, t) * \beta^{imp,z}(t), \quad (6.14)$$

where:

$k_{z_i,z}(t_i, t)$ is a constant of proportionality for converting the expression to a dollar amount. This constant might vary over time, but was assumed constant during the simulation experiment to reduce the number of iterations required for training.

Assumption Regarding Departure-Behavior Under Toll

Assumptions are made regarding the departure-behavior of agents under a congestion toll. The assumptions are based on the work by Arnott, De Palma, and Lindsey (1993) and the existence of network-wide equilibrium described by Vickrey (1969). Figure 6.4 describes the behavior under three possible toll scenarios:

- a) a monotonically decreasing toll curve,
- b) a triangularly shaped toll curve, and
- c) a monotonically increasing toll curve

The value of time of all individuals is assumed to be constant (= \$30/hr). The toll curve has a slope that is a small amount, ϵ , greater than this value of time. The curves in the figures are justified next.

Let us analyze case (b) in further detail. The “blue” curve represents the cumulative departure curve before applying toll. In the presence of toll, the first departing person can unilaterally improve his/her situation by departing earlier. He/She continues to experience a gain until the toll vanishes. Therefore, it is optimal for this person to depart at the beginning of the toll period. Similarly, the last departing person can unilaterally improve his/her situation by departing later. He/she continues to gain until the toll vanishes. Therefore, it is optimal for this person to depart at the end of the toll period. It is assumed that the toll rate is flat for a very small duration, δ , at the time at which the rate is at its peak. Therefore, it is optimal for the agent departing at this time not to change his/her departure time. This justifies the “green” curve as the new equilibrium departure curve under toll. This explanation can be extended to the other two figures with the same argument.

Optimal Toll Using Deep Learning Model

The parameters for estimating the optimal toll are approximated through a deep learning model as follows:

- $\beta^{imp,z}(t)$:

This value can be forecasted through a deep learning model, much like the model described in Equation 3.5. The input signals are the same as those for predicting MCL. The expression for predicting $\beta^{imp,z}(t)$ through an LSTM model is as follows:

$$[\beta^{imp,z}(t+h+p), \dots, \beta^{imp,z}(t+h)] \approx W_{LSTM}^T [hid_{t-h}, \dots, hid_{t-h-p}] + B_{LSTM}. \quad (6.15)$$

- $p_{z_i,z}(t_i, t)$:

This value may be approximated by the attention values corresponding to the O-D demand for Zone z_i at any given time as defined in Equation 6.6. The expression is as follows:

$$p_{z_i,z}(t_i, t) \approx att_{f^*,l} = \begin{cases} \frac{\text{Loss}_{f^*,l}}{\sum_{k \in F} \sum_{i=1}^p \text{Loss}_{k,i}}, & \sum_{k \in F} \sum_{i=1}^p \text{Loss}_{k,i} \neq 0 \\ \frac{1}{p*N}, & \sum_{k \in F} \sum_{i=1}^p \text{Loss}_{k,i} = 0, \end{cases} \quad (6.16)$$

where:

f^* is the input representing O-D demand from Zone z_i to Zone z .

- $k_{z_i,z}(t_i, t)$:

This value is a hyper-parameter which may be trained using cross-validation. It is assumed that the value of this parameter is constant over time to reduce the parameter search space.

These approximations can be substituted back into Equation 6.14 to derive an approximately optimal toll.

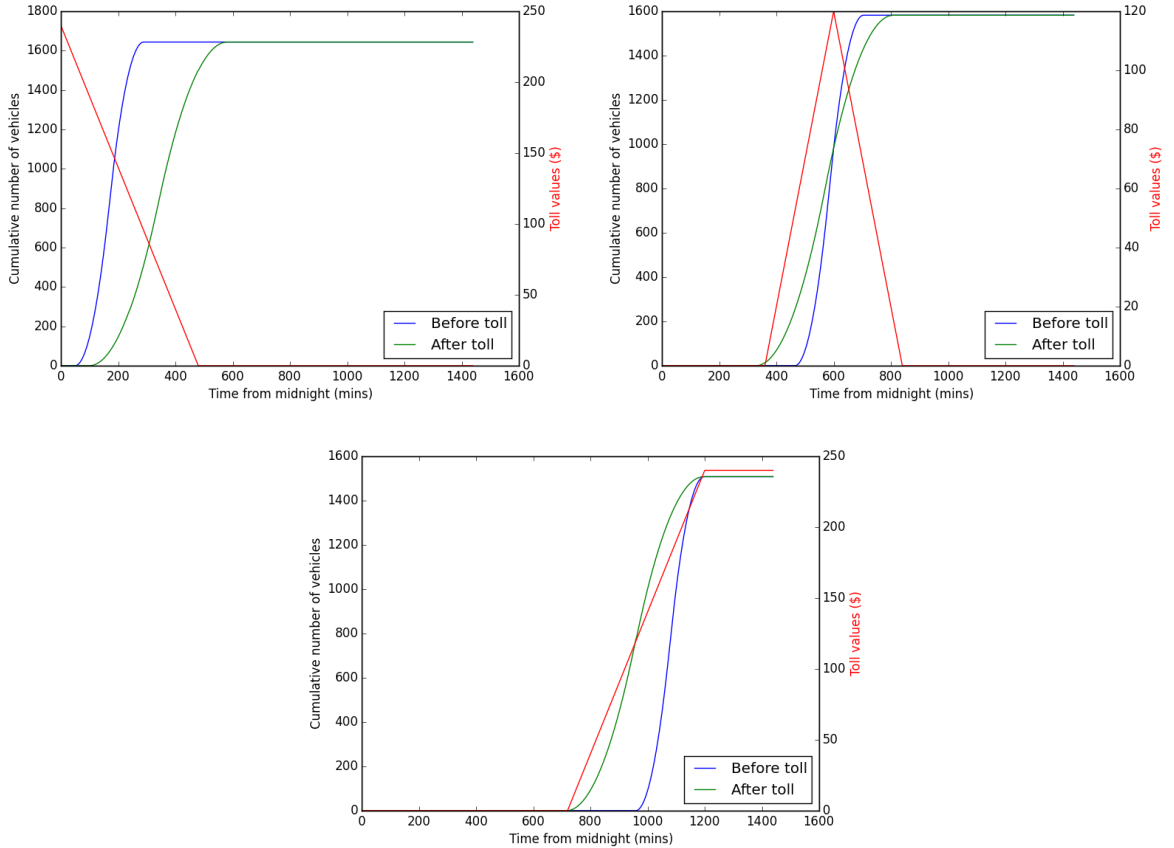


Figure 6.4: Change in the cumulative curve of departing agents under implementation of three toll strategies: (a) Monotonically decreasing toll curve, (b) Triangularly shaped toll curve and (c) Monotonically increasing toll curve. In all cases, it is assumed that the slope of the toll curve is a very small amount ϵ greater than the value of time (\$/min)

Optimal Toll Using O-D Demand Prediction

An alternative is suggested to the deep learning model-based approximations for the optimal toll. Future O-D demands may be predicted and tolls may be proposed based on those predictions to minimize cumulative MCL through the condition described in Equation 6.9. This section provides a heuristic for the same for a special case when all zones are equidistant from the target zone and demands from multiple zones are well separated in time.

The physical interpretation of the optimal toll condition suggests that the impact of O-D demands on MCL must be constant over space and time. This suggests that a **toll must be charged to ensure that departures from a zone are spread uniformly over time while ensuring that demands from multiple zones don't coalesce**. For the special case when all zones are equidistant to the target zone and demands from multiple zones are well separated in time, the optimal toll follows the pattern described below:

- (a) a monotonically increasing curve for the zone where departures begin first,
- (b) a monotonically decreasing curve for the zone where departures begin last, and
- (c) a triangularly shaped (with the peak at roughly the middle of the demand period) for all other zones.

Figure 6.5 shows the benefit of implementing such a strategy. The experiment assumes the baseline network (Figure 4.1a) with O-D demands well separated in time (Figure 4.1c). We can notice the reduction in MCL values by observing the “blue” curves in the two plots. The cumulative MCL through the day reduces from 59,246 min^2 to 23,305 min^2 with the application of toll. The average delay per commuter reduces from 205.93 mins to 63.74 mins when the toll is applied.

The key challenge for this strategy is the prediction of O-D demand for the day. Two possible heuristics for predicting O-D demand are:

- (i) approximating demand by the average of observed demands across all days and
- (ii) approximating demand by the observed demand on the previous day.

These heuristics assume that O-D demands on previous days were either observed completely or estimated accurately. Errors in estimation may further impact the effect of the tolling strategy. In the next section, the effect of the deep learning approximation for optimal toll is compared with the effect of O-D demand prediction-based toll.

Experimental Results

Simulation experiments were run to evaluate the performance of the LSTM model-based toll (in terms of cumulative congestion delay, average toll and average travel time in Zone c) against four possible tolling strategies:

- (i) no toll,

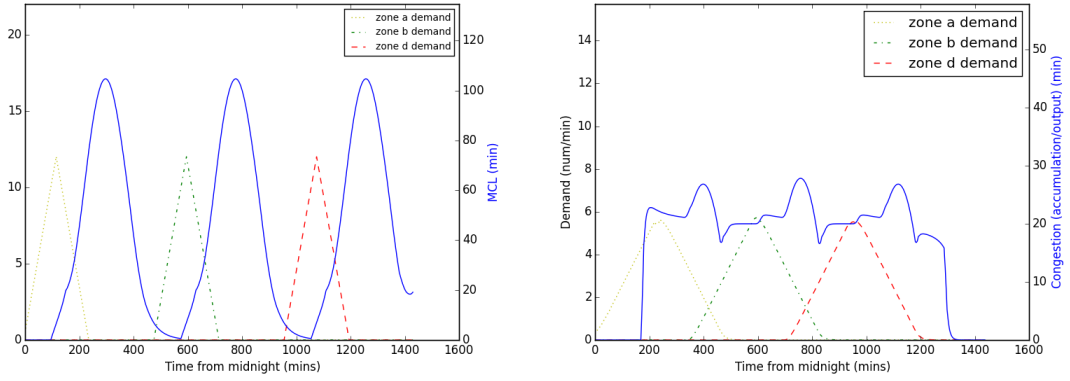


Figure 6.5: (a) No toll: Plot showing demand from the three input zones (i.e., a (yellow), b (green) and d (red), values to be read from the primary y-axis) and the the MCL in Zone c (blue curve, values to be read from the secondary y-axis). Cumulative congestion delay = 59246 min^2 . Average travel time per commuter = 205.93 min . (b) Optimal Toll Using O-D demand: Plot showing demand from the three input zones (i.e., a (yellow), b (green) and d (red), values to be read from the primary y-axis) and the the MCL in Zone c (blue curve, values to be read from the secondary y-axis). Cumulative congestion delay = 23305 min^2 . Average travel time per commuter = 63.74 min .

- (ii) toll based on approximating O-D demand by the average of observed demands across all days (i.e., Average demand-based toll),
- (iii) toll based on approximating O-D demand by the previous day's demand (i.e., 1-NN demand-based toll), and
- (iv) a hypothetical idealized scenario where future O-D demands are known (i.e., Known future demand-based toll).

The results from the simulation are summarized in Table 6.1 and the corresponding demand-congestion plots are shown in Figure 6.6.

Scenario	Cumulative congestion (min^2)	Average travel time (mins)	Average Toll (\$)
No Toll	77374.18	90.59	0.0
Average demand-based	65106.44	75.60	99.84
1-NN demand-based toll	56889.50	64.30	102.20
LSTM model-based toll	49643.22	60.93	3.01
Known future demands-based toll	41095.43	45.86	83.06

Table 6.1: Cumulative congestion, average travel time and average toll for various toll scenarios

The table suggests that approximating optimal toll with the help of LSTM model predictions and attentions lead to the lowest cumulative MCL among all strategies where future demands are unknown. Moreover, it is the only scenario with non-zero toll where the average toll amount charged from road users is reasonable (i.e., \$3.01/user). The average travel time for the LSTM-based toll is 60.93 mins , which saves 29.66 mins , 14.67 mins , and 3.37 mins over no toll, average demand-based toll, and 1-NN demand-based

toll respectively. The simulation results suggest that an **LSTM model-based toll is successful in reducing overall congestion impact while resulting in low discomfort for travelers**. A possible method to improve the LSTM model predictions for large networks is discussed in the next chapter.

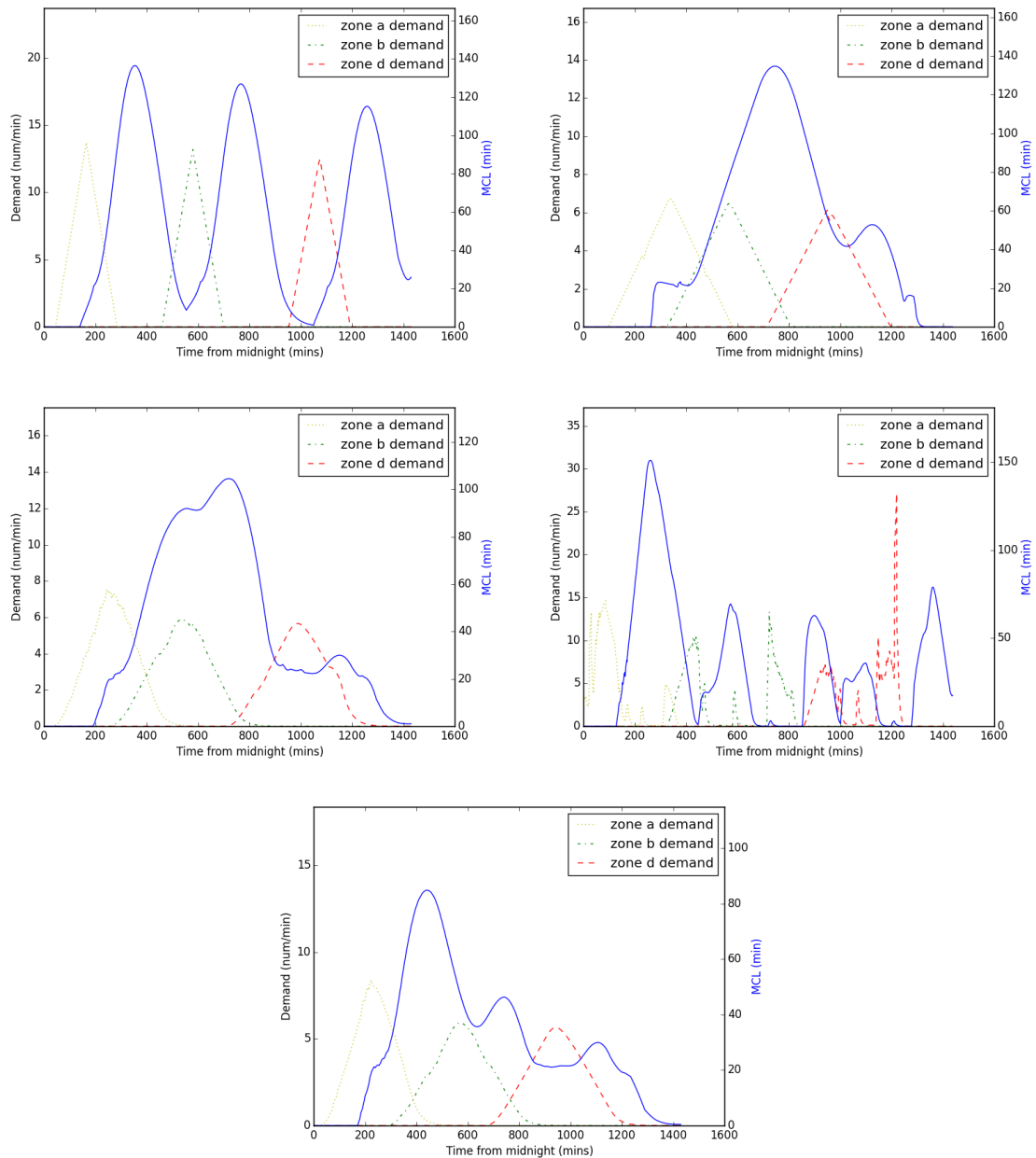


Figure 6.6: Plot showing variation of demands from Zone a (“.”), Zone b (“-.”) and Zone d (“-”) and resulting macroscopic congestion level in Zone c (solid) as a function of time for - (a) No toll, (b) Average demand-based toll, (c) 1-NN demand-based toll, (d) LSTM model-based toll and (e) Known future demand-based toll

Chapter 7

Improvements through Graph-based Feature Learning

The deep learning models dealt in all previous chapters have been tested with use of an LSTM architecture only. Past studies have shown that the addition of convolutional layers to an LSTM model allows better learning of spatial features and thus improves prediction accuracy (Xingjian et al. 2015; Song, Kanasugi, and Shibasaki 2016; Wu and Tan 2016; Ma, Dai, et al. 2017; Li, Yu, et al. 2017; Ning et al. 2017). This chapter aims to improve the prediction accuracy for the deep learning model proposed in Equation 3.5 for predicting MCL through better learning of spatial features by using Convolutional Neural Networks (CNNs).

Large traffic networks produce signals that may have a large-scale spatial impact and a long-term temporal impact. These signals can be assumed to propagate through hypothetical graphs. These graphs may represent the road network or the similarity in route-choices made by various individuals. These graphs also provide some structure to the input signals and thus allow learning of features with a low amount of training data.

The input signals are transformed to incorporate the effect of a graphical structure. A CNN architecture, namely Graph-CNN, is employed that is geared towards extracting spatial features from graphical inputs. Simulations of commute trips on:

- (a) a simplified representation of the freeway/highway network and
- (b) a full-scale representation of the network

in the San Francisco Bay Area suggest that a Graph-CNN+LSTM architecture predicts MCL more accurately than the following baselines:

1. a 1-NN model,
2. a Holt-Winters model,
3. an LSTM-only model.

7.1 Graph Transformation of Input

A weighted undirected graph, $\tilde{G} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, W)$, is constructed using the road network graph, $G = (\mathcal{V}, \mathcal{A})$, and the zones, denoted by Z . The zones are partitioned through the procedure defined by Ji and Geroliminis (2012). An input signal, x^z , from the observed network state is transformed into an input signal, $\tilde{x}^z$, incorporating the structure of the graph, $\tilde{G}$. Each transformed signal emanates at a node in $\tilde{G}$. Two possible procedures are discussed for the derivation of $\tilde{G}$ and the transformation of signals as described below:

1. Shortest-path-based Weights

The key idea here is to assign weights based on the shortest path between zones along the edges of G . The details of the procedure are as follows:

- A node, $\tilde{v} \in \tilde{\mathcal{V}}$, is constructed at the centroid of each Zone $z \in Z$.
- The weight between any two nodes, $\tilde{v}_\alpha, \tilde{v}_\beta \in \tilde{\mathcal{V}}$, representing the Zones $z_\alpha, z_\beta \in Z$, is determined as the shortest directed path length between the nodes along the edges of G . This weight is represented as

$$W_{\alpha,\beta} = D_{SP,G}(v_\alpha, v_\beta), \quad (7.1)$$

where $D_{SP,G}$ represents the length of the shortest path on graph G .

- The transformed input signal, $\tilde{x}_\alpha^z$, emanating from node v_α is estimated as follows:
 - Half of each O-D demand originating at z_α is assigned to $\tilde{x}_\alpha^z$.
 - The probability that an individual observed on a given link, i , originates or terminates its trip in Zone z_α is $p(\alpha|i)$. Half of the link accumulation on link i weighted by this probability is assigned to $\tilde{x}_\alpha^z$.
 - Half of the link travel time on link i weighted by $p(\alpha|i)$ is assigned to $\tilde{x}_\alpha^z$.

The resulting transformed input signal at node v_α representing Zone z_α is

$$\tilde{x}_\alpha^z(t) = \left[\begin{array}{c} D^{z_\alpha, z}(t)/2 \\ p(\alpha|i)C_i(t)/2 \\ p(\alpha|i)TT_i(t)/2 \end{array} \right]. \quad (7.2)$$

2. Trajectory-Clustering-based Weight

The key idea here is to determine a measure of distance between O-Ds based on the similarities in the route-choices between those O-Ds.

- A node, $\tilde{v} \in \tilde{\mathcal{V}}$, is constructed representing each pair of Zones, $(z_\alpha, z_\beta) \in Z$.
- The weight between any two nodes, $v_a, v_b \in \tilde{\mathcal{V}}$ representing the pairs of Zones $(z_\alpha, z_\beta), (z_\gamma, z_\delta) \in Z^2$ is determined based on the similarity of routes chosen by individuals when traveling from z_α to z_β and from z_γ to z_δ . This measure of similarity is calculated through the following steps:

(a) Trajectory clustering

All observed trips are clustered based on the similarities in executed trajectories. Some trajectory clustering algorithms from past literature are based on a partition and group approach (Lee, Han, and Whang 2007), a mixture of regression models (Gaffney and Smyth 1999) and *Dynamic Time Warping* (DTW) (Sankararaman et al. 2013).

- (b) Calculating the probability distribution over clusters for each O-D
A discrete probability distribution, $\Delta^{z_\alpha, z_\beta}$, over derived clusters is determined for every O-D, (z_α, z_β) . The probability assigned to a cluster is the proportion of trips belonging to that cluster.
- (c) Calculating the relative entropy between probability distributions
The relative entropy between two probability distributions derived in the previous step, $\Delta^{z_\alpha, z_\beta}$ and $\Delta^{z_\gamma, z_\delta}$, is calculated as

$$W_{ab} = D_{KL}(\Delta^{z_\alpha, z_\beta}, \Delta^{z_\gamma, z_\delta}), \quad (7.3)$$

where D_{KL} represents the K-L divergence between two distributions.
 W_{ab} represents the weight of the edge between nodes v_a and v_b in $\tilde{G}$.

- The transformed input signal, $\tilde{x}_a^z$, emanating from node v_a is estimated as follows:
 - The O-D demand D^{z_α, z_β} is assigned to $\tilde{x}_a^z$.
 - The probability that an individual observed on link i originates its trip at z_α and terminates its trip at z_β is $p(\alpha, \beta|i)$. The link accumulation on link i weighted by $p(\alpha, \beta|i)$ is assigned to $\tilde{x}_a^z$.
 - The link travel time on link i weighted by $p(\alpha, \beta|i)$ is assigned to $\tilde{x}_a^z$.

The transformed input signal at node v_a representing the pair of Zones (z_α, z_β) is

$$\tilde{x}_a^z(t) = \begin{bmatrix} D^{z_\alpha, z_\beta}(t) \\ p(\alpha, \beta|i)C_i(t) \\ p(\alpha, \beta|i)TT_i(t) \end{bmatrix}. \quad (7.4)$$

7.2 Graph CNN + LSTM Model Formulation

A deep-learning model is proposed to predict the MCL in Zone z , $\xi^z(t)$, as a function of the input signals $\tilde{x}^z$ on an instance of graph $\tilde{G}$ parameterized by θ (i.e., g_θ). The model has the following form:

$$\begin{aligned} [\hat{\xi}^z(t+h+p), \dots, \hat{\xi}^z(t+h)] &= f(\tilde{g}_\theta(L)\tilde{x}^z(t), \dots, \tilde{g}_\theta(L)\tilde{x}^z(t-p)) \\ &= \mathcal{U}f(\tilde{g}_\theta(\Lambda)\tilde{x}^z(t), \dots, \tilde{g}_\theta(\Lambda)\tilde{x}^z(t-p))\mathcal{U}^T, \end{aligned} \quad (7.5)$$

where:

f represents a function approximator (in this case, a Graph CNN-LSTM model)

h is the *minimum dependency lag* (same interpretation as in Equation 3.5)

p is the *maximum dependency persistence* (same interpretation as in Equation 3.5)

$\mathcal{U} = [U, U, \dots, U]$ is a tensor where each element in a column is a matrix of eigenvectors for Laplacian L for instance of graph $\tilde{G}$ parameterized by θ

$\Lambda = \text{diag}(\lambda_0, \dots, \lambda_{n-1})$ is a diagonal matrix with entries as the eigenvalues arranged in the same order as the corresponding eigenvectors in U

7.3 Experimental Setup

The prediction accuracy for the proposed model (see Equation (7.5)) was tested on two networks described in Figure 7.1. The first network (Figure 7.1(a)) comprises 39 nodes and 54 links representing a simplified freeway/highway network for nine counties of the San Francisco Bay Area. The entire region was categorized into 54 zones such that a zone's entire street network was represented by a single link. The second network was a detailed road network representing the nine counties of the San Francisco Bay Area, consisting of 352,012 nodes and 564,368 links. Here, the area was partitioned into 1454 zones as per the Metropolitan Transportation Commission's (MTC's) Travel Model One¹.

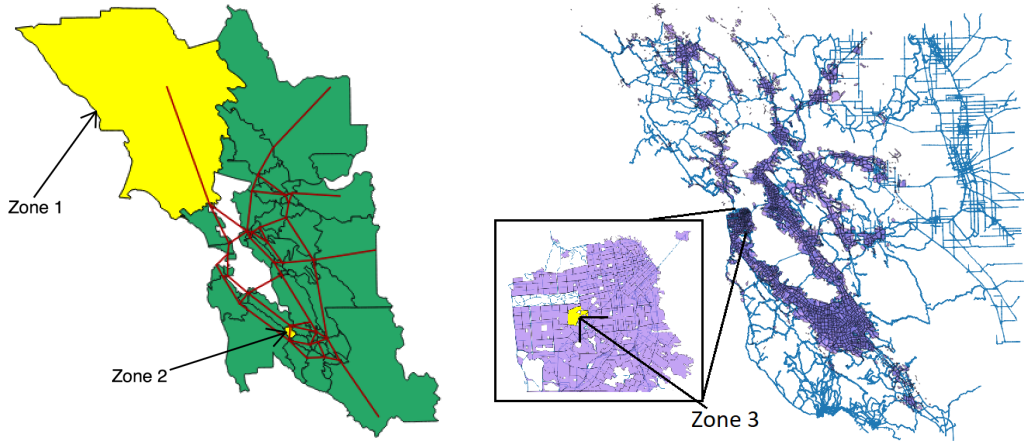


Figure 7.1: (a) Simplified freeway network and (b) Full-scale network - representing 9 counties of the SF Bay Area. Each region was partitioned into zones. Target zones for MCL prediction are highlighted in yellow.

The congestion patterns for commute trips were generated using an agent-based activity demand model through a well-known open-source traffic simulation software called MATSim (Illenberger, Flötteröd, and Nagel 2007). The desired *activity chains* (i.e., a sequence of activities performed in a typical day) of individuals was derived and a network equilibrium was determined by maximizing individual utilities. Individuals gain utility by performing an activity at their desired times and lose utility due to increased travel times. The process results in a set of executed trajectories during simulation. The home and work locations were estimated from Census Transportation Planning Products (CTPP) data for years 2006-2010². Variability in demand across multiple days was ensured by sampling the start times and commute duration from Gaussian probability distributions. Four scenarios were generated based on the mean and the standard deviation of the probability distribution, as described in Table 7.1. Data was generated for 5,000 days in each scenario with 100,000 individuals commuting on each day.

¹See <https://github.com/BayAreaMetro/modeling-website/wiki/TravelModel> and <https://mtc.maps.arcgis.com/home/item.html?id=b85ba4d43f9843128d3542260d9a2f1f>

²See <http://ctpp.transportation.org/Pages/5-Year-Data.aspx>

Scenario	μ_{st} h-w	μ_{st} w-h	σ_{st}	μ_d	σ_d
1	8.5	17.5	1	1	0
2	8.5	17.5	2	1	0
3	8.5	17.5	1	1	1
4	8.5	17.5	2	1	1

μ_{st} : Mean start time

μ_d : Mean duration

σ_{st} : Standard deviation of start time

σ_d : Standard deviation of duration

h-w: Home-to-work trip

w-h: Work-to-home trip

Table 7.1: Parameters of Gaussian distributions for the start time and duration of h-w and w-h trips generated for testing the proposed Graph CNN + LSTM model

Two implementations of the Graph CNN + LSTM model were tested, namely the shortest-path-based graph and the trajectory clustering-based graph. The construction of these graphs is described next.

For the shortest-path-based model, the centroid of each zone in Figure 7.1 represented the node $\tilde{v}$. Shortest paths were calculated between each pair of centroids using Dijkstra’s algorithm. In case the centroid did not lie on an existing link, straight line paths (in Euclidean space) were assumed from the centroid to its nearest link. A complete graph, $\tilde{G}$ was derived. Shortest paths were calculated between each pair of centroids the freeway network (Figure 7.1(a)). Only the k ($= 2$) nearest neighbors were connected by shortest paths in the detailed network (Figure 7.1(b)) to conserve memory.

A sample of 565,000 trips was analyzed for the trajectory clustering-based model. Pair-wise distances were calculated between the trips using DTW distance. Clustering was performed using standard *K-Means* clustering with $\kappa = 100$ clusters. The number of clusters was motivated from a finding by Wu, Thai, et al. (2015) that most real-world trips get clustered around a few routes. The O-D adjacency matrix was calculated from these clusters using the procedure described in Section 7.1. A threshold of 50 trips per O-D was chosen in the freeway network and a threshold of 10 trips per O-D was chosen for in detailed network to limit memory consumption. This left 673 O-Ds in the freeway network and 2173 O-Ds in the detailed network. k-NN graphs with $k = 5$ for the freeway network and $k = 2$ for the detailed network were constructed, further limiting memory consumption.

7.4 Experimental Results

The proposed Graph CNN + LSTM model was implemented to predict congestion in the highlighted zones in Figure 7.1. The performance of the proposed framework was tested against three baseline models:

- i *1-NN model* (See Equation 4.2)
- ii *LSTM-only model* (See Equation 3.5)

iii *k-NN Euclidean Graph-CNN + LSTM model*:

A weighted graph $\tilde{G} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, W)$ was constructed assuming no knowledge of the underlying road network. It was hypothesized that the features that are in the neighborhood of each other have similar observed values. The nodes $\tilde{v}$ represent the input set F and the weights W are determined as the mean Euclidean distance between the observed values along two input dimensions. The following equation describes the calculation of weights between nodes:

$$W_{i,j} = \sum_t (X_i(t) - X_j(t))^2,$$

where X_i and X_j represent the observed network state inputs along dimensions i and j respectively

The graph was made sparse by constructing edges only along the k ($= 5$) nearest neighbors to each node.

iv *Holt-Winters model* (Holt 1957; Winters 1960):

A purely statistical exponential smoothening approach for time-series prediction was implemented for predicting MCL values. This model incorporates seasonality through a weighted linear structure.

The performance of each model for prediction of MCL in the Zones 1, 2, and 3 (Figure 7.1) for the scenarios mentioned in Table 7.1 was tested by calculating RMSE values as per equation 4.2. The simulation results are displayed in Table 7.2³.

The table suggests that the **Graph CNN + LSTM models are more accurate than the LSTM-only model, which in turn is more accurate than the two implicit models (i.e., the 1-NN model and the Holt-Winters model)**. This is because implicit models don't learn any useful features from the network state. Among the two implicit models, the 1-NN model performs better when the amount of variation across days is low and Holt-Winters model performs better when the variation is high. The difference in performance between deep learning models and the implicit models generally grows with higher uncertainty in the start time of demand and the travel duration. We may attribute the gain in performance of the LSTM model over the implicit models to a better learning of the temporal difference between various input signals and the output. We may attribute the gain in performance of the Graph-CNN+LSTM models over the LSTM-only model to better learning of spatial features by exploiting the graphical nature of the input data.

The different graphs for the Graph-CNN + LSTM models are compared next. The table suggests that the trajectory clustering-based graph is slightly more accurate than the shortest-path-based graph, which in turn is more accurate than the Euclidean distance-based graph. The higher prediction accuracy for the shortest-path-based graph over the Euclidean distance-based graph is intuitive since the structure of the road network is used in the former but not the latter. The higher prediction accuracy for the trajectory clustering-based graph over the shortest-path-based graph may be simply attributed to

³An open source implementation of the algorithm on the sample scenarios is made available at: <https://github.com/sudatta0993/Dynamic-Congestion-Prediction>

more road network information (i.e., executed routes of agents) being encoded into the generated graph.

Sc.	T.Z.	1-NN	H.W.	LSTM	GCN+LSTM (k-NN Eu)	GCN+LSTM (SP)	GCN+LSTM (k-NN TC)
1	1	0.503	1.104	0.356	0.222	0.194	0.187
1	2	1.300	3.014	0.801	0.637	0.601	0.596
1	3	0.932	0.988	0.752	0.671	0.551	0.530
2	1	1.077	1.121	0.875	0.715	0.462	0.358
2	2	1.831	3.011	1.107	0.998	0.612	0.558
2	3	1.202	1.070	1.050	0.965	0.738	0.537
3	1	1.010	1.510	0.871	0.660	0.205	0.102
3	2	2.256	2.015	1.228	1.070	0.789	0.435
3	3	1.866	1.803	1.657	1.474	1.401	1.386
4	1	0.924	1.082	0.609	0.605	0.377	0.194
4	2	3.155	2.700	2.124	2.024	1.694	0.858
4	3	1.144	0.554	0.467	0.433	0.343	0.313

Sc: Scenario (Table 7.1)

T.Z.: Target Zone (Figure 7.1(b))

H.W.: Holt-Winters Model

GCN+LSTM (k-NN Eu): Graph CNN + LSTM, k-NN graph, Euclidean distance weights

GCN+LSTM (SP): Graph CNN + LSTM, complete graph, shortest path weights

GCN+LSTM (k-NN TC): Graph CNN + LSTM, k-NN graph, trajectory clustering weights

Table 7.2: A table describing the relative performance of 1-NN, Holt-Winters model, LSTM-only, Graph CNN + LSTM (k-NN Euclidean Graph), Graph CNN + LSTM (Shortest Path Graph) and Graph CNN + LSTM (k-NN Trajectory Clustering Graph) for the prediction of $\hat{\xi}^z(t)$ in target zones highlighted in Figure 7.1(b) and demand scenarios described in Table 7.1

Chapter 8

Conclusions

8.1 Key Contributions

The key contributions of this research may be summarized as follows:

- The congestion state in a neighborhood with homogeneous and slowly evolving traffic conditions is approximated through a scoring function, called Macroscopic Congestion Level (MCL). This score is derived from two aggregate characteristics, namely the total vehicle accumulation, and the total trip completion rate in the region. A battery of simulation experiments on a hypothetical, idealized city street network suggests that MCL represents the true congestion state in the neighborhood quite well.
- The network state in a larger region-wide network, defined as a vector of O-D demands, link accumulations and link travel times and MCL values, may signal the future MCL in a particular neighborhood. Simulation experiments on an idealized network suggest that the network state is responsive to various physical processes that may affect traffic congestion.
- A deep learning model based on Long Short-Term Memory (LSTM) architecture is formulated to predict the MCL in a neighborhood. The network state in a larger network is used as the model input. Simulation experiments suggest that the prediction of MCL with the deep learning model is more accurate than that using a model based on copying the previous day's MCL (i.e., 1-NN model).
- The deep-learning model is tested for robustness under three scenarios, namely
 - correlated O-D demands across multiple days,
 - noisy observations of the model inputs, and
 - partially observed model inputs.

Techniques to improve model robustness are suggested for each scenario by changing the hyper-parameters of the deep learning model or by understanding the importance of various inputs.

- A Neural Attention Model-based framework is developed to derive the importance of various inputs to the deep learning model. Simulation experiments conducted on a simple, idealized network and a hypothetical O-D demand suggest that the

Neural Attention model-based framework approximates the importance of inputs quite well.

- The deep learning model predictions and the importance of various inputs are used to define a model-based predictive controls. A novel app-based dynamic congestion price is suggested to control the departure times of individuals. The optimal toll, which depends on future O-D demand, is approximated by using the deep learning model. The effect of deep learning model-based predictive toll is compared to three tolling strategies, namely:
 - no toll,
 - a toll that approximates future O-D demand by the average O-D demand observed on all previous days, and
 - a toll that approximates future O-D demand by the O-D demand on the previous day.

Simulation experiments on an idealized network and a hypothetical O-D demand suggest that the deep learning model-based toll not only reduces delay more than the other tolling strategies but also charges a low toll to travelers.

- Improvements to the prediction accuracy of the LSTM model are suggested by representing the input signals in the form of a graph and extracting useful spatial features through a Graph-CNN architecture. Two possible graphs are proposed as follows:
 - (i) a graph that represents the centroid of each neighborhood through a node and derives the weights of the edges between nodes as the length of the shortest path between the centroids in the road network (i.e., shortest path graph), and
 - (ii) a graph that represents each pair of neighborhoods as a node and derives the weights between nodes based on the similarity in the routes chosen by individuals traveling between the pair of neighborhoods (i.e., trajectory clustering graph).

Simulation experiments on commute trips on (a) a simplified representation of the freeway/highway network and (b) a full-scale representation of the network representing the San Francisco Bay Area suggest that a Graph CNN + LSTM model based on the shortest path graph and the trajectory clustering graph shows better prediction accuracy than three baselines:

- (i) a 1-NN model,
- (ii) an LSTM-only model, and
- (iii) a Graph CNN + LSTM model based on a graph that assumes no prior information such as the structure of the underlying road network or the history of route-choices of individuals.

8.2 Future Work

The mechanism discussed for making predictions of the Macroscopic Congestion Level (MCL) in a target neighborhood as well as deriving the importance of various inputs lends itself well to model-predictive dynamic control for mitigating congestion. Several possible future directions of research emerge from the current study. A few such directions are as follows:

- The framework for predicting congestion in a single target neighborhood can be generalized to predict congestion across multiple neighborhoods. This may allow one to predict MCL at a large network-wide scale and extracting important inputs for the same. Controls may be developed to reduce delays based on congestion impacts in the entire region-wide network.
- The case study analyzed for dynamic controls assumed that the behavior of individuals during toll was known. Techniques may be discussed to modify the control in case the behavior is not known prior to the control.
- Predictions and optimal control can be derived jointly. A *Markov Decision Process* (MDP) (Bellman 1957) can be formulated representing the evolution of MCL over time. Model-based *Reinforcement Learning* (RL) may be employed to predict MCL as well as test the impact of various dynamic controls at the same time.

Bibliography

- [1] Baher Abdulhai, Rob Pringle, and Grigoris J Karakoulas. “Reinforcement learning for true adaptive traffic signal control”. In: *Journal of Transportation Engineering* 129.3 (2003), pp. 278–285.
- [2] KMAE Aboudolas, M Papageorgiou, A Kouvelas, et al. “A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks”. In: *Transportation Research Part C: Emerging Technologies* 18.5 (2010), pp. 680–694.
- [3] Konstantinos Aboudolas and Nikolas Geroliminis. “Perimeter and boundary flow control in multi-reservoir heterogeneous networks”. In: *Transportation Research Part B: Methodological* 55 (2013), pp. 265–281.
- [4] Konstantinos Aboudolas, Markos Papageorgiou, and E Kosmatopoulos. “Store-and-forward based methods for the signal control problem in large-scale congested urban road networks”. In: *Transportation Research Part C: Emerging Technologies* 17.2 (2009), pp. 163–174.
- [5] JD Addison and Benjamin G Heydecker. “A Mathematical Model For Dynamic Traffic Assignment.” In: *Transportation and Traffic Theory* (1993).
- [6] Mohammed S Ahmed and Allen R Cook. *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. 722. 1979.
- [7] Y Al Hassan and Derek J Barker. “The impact of unseasonable or extreme weather on traffic activity within Lothian region, Scotland”. In: *Journal of Transport Geography* 7.3 (1999), pp. 209–213.
- [8] Itamar Arel et al. “Reinforcement learning-based multi-agent system for network traffic signal control”. In: *IET Intelligent Transport Systems* 4.2 (2010), pp. 128–135.
- [9] R. Arnott, A. de Palma, and R. Lindsey. “Departure time and route choice for the morning commute”. In: *Transportation Research Part B: Methodological* 24.3 (1990), pp. 209–228. ISSN: 0191-2615. DOI: [http://dx.doi.org/10.1016/0191-2615\(90\)90018-T](http://dx.doi.org/10.1016/0191-2615(90)90018-T). URL: <http://www.sciencedirect.com/science/article/pii/019126159090018T>.
- [10] Richard Arnott, Andre De Palma, and Robin Lindsey. “A structural model of peak-period congestion: A traffic bottleneck with elastic demand”. In: *The American Economic Review* (1993), pp. 161–179.
- [11] Richard Arnott, André de Palma, and Robin Lindsey. “A Structural Model of Peak-Period Congestion: A Traffic Bottleneck with Elastic Demand”. English. In: *The American Economic Review* 83.1 (1993), pp. 161–179. ISSN: 00028282. URL: <http://www.jstor.org/stable/2117502>.

- [12] Richard Arnott and Kenneth Small. “The economics of traffic congestion”. In: *American scientist* (1994), pp. 446–455.
- [13] Kalidas Ashok and Moshe E Ben-Akiva. “Dynamic origin-destination matrix estimation and prediction for real-time traffic management systems”. In: *International Symposium on the Theory of Traffic Flow and Transportation (12th: 1993: Berkeley, Calif.)*. *Transportation and traffic theory*. 1993.
- [14] Kalidas Ashok and Moshe E Ben-Akiva. “Alternative approaches for real-time estimation and prediction of time-dependent origin–destination flows”. In: *Transportation Science* 34.1 (2000), pp. 21–36.
- [15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).
- [16] Dzmitry Bahdanau, Jan Chorowski, et al. “End-to-end attention-based large vocabulary speech recognition”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE. 2016, pp. 4945–4949.
- [17] Michael Balmer. “Travel demand modeling for multi-agent transport simulations: Algorithms and systems”. PhD thesis. ETH Zurich, 2007.
- [18] J Barceló et al. “Microscopic traffic simulation for att systems analysis. a parallel computing version”. In: *Contribution to the 25th Anniversary of CRT* (1998).
- [19] Hillel Bar-Gera. “Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: A case study from Israel”. In: *Transportation Research Part C: Emerging Technologies* 15.6 (2007), pp. 380–391.
- [20] Richard Bellman. “A Markovian decision process”. In: *Journal of Mathematics and Mechanics* (1957), pp. 679–684.
- [21] Moshe Ben-Akiva, Michel Bierlaire, et al. “DynaMIT: a simulation-based system for traffic prediction”. In: *DACCORS short term forecasting workshop, The Netherlands*. Citeseer. 1998.
- [22] Moshe Ben-Akiva, Michèle Cyna, and André de Palma. “Dynamic model of peak period congestion”. In: *Transportation Research Part B: Methodological* 18.4–5 (1984), pp. 339–355. ISSN: 0191-2615. DOI: [http://dx.doi.org/10.1016/0191-2615\(84\)90016-X](http://dx.doi.org/10.1016/0191-2615(84)90016-X). URL: <http://www.sciencedirect.com/science/article/pii/019126158490016X>.
- [23] Chandra R Bhat and Frank S Koppelman. “Activity-based modeling of travel demand”. In: *Handbook of transportation Science*. Springer, 1999, pp. 35–61.
- [24] Ella Bingham. “Reinforcement learning in neurofuzzy traffic signal control”. In: *European Journal of Operational Research* 131.2 (2001), pp. 232–241.
- [25] David E. Boyce. “Contributions of Transportation Network Modeling to the Development of a Real-Time Route Guidance System”. In: *Transportation for the future* 3 (1989), pp. 161–177.
- [26] Stephen Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge university press, 2004.
- [27] Ronald Newbold Bracewell. *The Fourier transform and its applications*. Vol. 31999. McGraw-Hill New York, 1986.

- [28] Joan Bruna et al. “Spectral networks and locally connected networks on graphs”. In: *arXiv preprint arXiv:1312.6203* (2013).
- [29] Avishai Ceder. “Relationships between road accidents and hourly traffic flow—II: Probabilistic approach”. In: *Accident Analysis & Prevention* 14.1 (1982), pp. 35–44.
- [30] Kit Yan Chan et al. “Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm”. In: *IEEE Transactions on Intelligent Transportation Systems* 13.2 (2012), pp. 644–654.
- [31] Gang-Len Chang and Jifeng Wu. “Recursive estimation of time-varying origin-destination flows from traffic counts in freeway corridors”. In: *Transportation Research Part B: Methodological* 28.2 (1994), pp. 141–160.
- [32] Chao Chen, Jaimyoung Kwon, et al. “Detecting errors and imputing missing data for single-loop surveillance systems”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1855 (2003), pp. 160–167.
- [33] Chenyi Chen, Yin Wang, et al. “The retrieval of intra-day trend and its influence on traffic prediction”. In: *Transportation research part C: emerging technologies* 22 (2012), pp. 103–118.
- [34] Sundeep Prabhakar Chepuri et al. “Learning sparse graphs under smoothness prior”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE. 2017, pp. 6508–6512.
- [35] Jan K Chorowski et al. “Attention-based models for speech recognition”. In: *Advances in Neural Information Processing Systems*. 2015, pp. 577–585.
- [36] Edward Chung et al. “Effect of rain on travel demand and traffic accidents”. In: *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*. IEEE. 2005, pp. 1080–1083.
- [37] Fan RK Chung. *Spectral graph theory*. 92. American Mathematical Soc., 1997.
- [38] William S Cleveland. “LOWESS: A program for smoothing scatterplots by robust locally weighted regression”. In: *The American Statistician* 35.1 (1981), pp. 54–54.
- [39] Cathal Coffey and Alexei Pozdnoukhov. “Temporal decomposition and semantic enrichment of mobility flows”. In: *Proceedings of the 6th ACM SIGSPATIAL International Workshop on Location-Based Social Networks*. ACM. 2013, pp. 34–43.
- [40] Mario Cools, Elke Moons, and Geert Wets. “Assessing the impact of weather on traffic intensity”. In: *Weather, Climate, and Society* 2.1 (2010), pp. 60–68.
- [41] Michael G Crandall and P-L Lions. “Two approximations of solutions of Hamilton-Jacobi equations”. In: *Mathematics of Computation* 43.167 (1984), pp. 1–19.
- [42] Carlos F Daganzo. “The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory”. In: *Transportation Research Part B: Methodological* 28.4 (1994), pp. 269–287.
- [43] Carlos F Daganzo. “The nature of freeway gridlock and how to prevent it”. In: *International symposium on transportation and traffic theory*. 1996, pp. 629–646.

- [44] Carlos F Daganzo. “Queue spillovers in transportation networks with a route choice”. In: *Transportation Science* 32.1 (1998), pp. 3–11.
- [45] Carlos F Daganzo. “A simple traffic analysis procedure”. In: *Networks and Spatial Economics* 1.1-2 (2001), pp. 77–101.
- [46] Carlos F Daganzo. “A behavioral theory of multi-lane traffic flow. Part I: Long homogeneous freeway sections”. In: *Transportation Research Part B: Methodological* 36.2 (2002), pp. 131–158.
- [47] Carlos F Daganzo. “Urban gridlock: Macroscopic modeling and mitigation approaches”. In: *Transportation Research Part B: Methodological* 41.1 (2007), pp. 49–62.
- [48] Carlos F Daganzo and Nikolas Geroliminis. “An analytical approximation for the macroscopic fundamental diagram of urban traffic”. In: *Transportation Research Part B: Methodological* 42.9 (2008), pp. 771–781.
- [49] Matthew D’Angelo, Haitham Al-Deek, and Morgan Wang. “Travel-time prediction for freeway corridors”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1676 (1999), pp. 184–191.
- [50] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. “Convolutional neural networks on graphs with fast localized spectral filtering”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 3844–3852.
- [51] Jean Doig, Vikash Gayah, and Michael Cassidy. “Inhomogeneous Flow Patterns in Undersaturated Road Networks: Implications for Macroscopic Fundamental Diagram”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2390 (2013), pp. 68–75.
- [52] Jeffrey Donahue et al. “Long-term recurrent convolutional networks for visual recognition and description”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2625–2634.
- [53] David K Duvenaud et al. “Convolutional networks on graphs for learning molecular fingerprints”. In: *Advances in neural information processing systems*. 2015, pp. 2224–2232.
- [54] LC Edie. “Discussion of Traffic Stream Measurement and Description”. In: *Proc. Second Intern. Symp. on the Theory of Road Traffic Flow, J. Almond, Ed., OECD, Paris*. 1965.
- [55] Xiang Fei, Chung-Cheng Lu, and Ke Liu. “A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction”. In: *Transportation Research Part C: Emerging Technologies* 19.6 (2011), pp. 1306–1318.
- [56] Martin Fellendorf. “VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority”. In: *64th Institute of Transportation Engineers Annual Meeting*. Springer. 1994, pp. 1–9.
- [57] Kunihiro Fukushima. “Neural network model for selective attention in visual pattern recognition and associative recall”. In: *Applied Optics* 26.23 (1987), pp. 4985–4992.
- [58] Scott Gaffney and Padhraic Smyth. “Trajectory clustering with mixtures of regression models”. In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 1999, pp. 63–72.

- [59] Nikolas Geroliminis and Carlos F Daganzo. “Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings”. In: *Transportation Research Part B: Methodological* 42.9 (2008), pp. 759–770.
- [60] Nikolas Geroliminis, Jack Haddad, and Mohsen Ramezani. “Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach”. In: *IEEE Transactions on Intelligent Transportation Systems* 14.1 (2013), pp. 348–359.
- [61] C Lee Giles, Steve Lawrence, and Ah Chung Tsoi. “Noisy time series prediction using recurrent neural networks and grammatical inference”. In: *Machine learning* 44.1 (2001), pp. 161–183.
- [62] JW Godfrey. “The mechanism of a road network”. In: *Traffic Engineering & Control* 8.8 (1970).
- [63] Sergei Konstantinovich Godunov. “A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics”. In: *Matematicheskii Sbornik* 89.3 (1959), pp. 271–306.
- [64] Thomas F Golob and Wilfred W Recker. “Relationships among urban freeway accidents, traffic flow, weather, and lighting conditions”. In: *Journal of transportation engineering* 129.4 (2003), pp. 342–353.
- [65] Marco Gori, Gabriele Monfardini, and Franco Scarselli. “A new model for learning in graph domains”. In: *Neural Networks, 2005. IJCNN’05. Proceedings. 2005 IEEE International Joint Conference on*. Vol. 2. IEEE. 2005, pp. 729–734.
- [66] Alex Graves. “Generating sequences with recurrent neural networks”. In: *arXiv preprint arXiv:1308.0850* (2013).
- [67] Jack Haddad and Nikolas Geroliminis. “On the stability of traffic perimeter control in two-region urban cities”. In: *Transportation Research Part B: Methodological* 46.9 (2012), pp. 1159–1176.
- [68] Stanton W Hadley and Alexandra A Tsvetkova. “Potential impacts of plug-in hybrid electric vehicles on regional power generation”. In: *The Electricity Journal* 22.10 (2009), pp. 56–68.
- [69] Mohammad Hajiahmadi, Jack Haddad, et al. “Optimal hybrid perimeter and switching plans control for urban traffic networks”. In: *IEEE Transactions on Control Systems Technology* 23.2 (2015), pp. 464–478.
- [70] Mohammad Hajiahmadi, Victor L Knoop, et al. “Optimal dynamic route guidance: A model predictive approach using the macroscopic fundamental diagram”. In: *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE. 2013, pp. 1022–1028.
- [71] Abolhassan Halati, Henry Lieu, and Susan Walker. “CORSIM-corridor traffic simulation model”. In: *Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and Opportunities*. 1997.
- [72] Dwight A Hennessy and David L Wiesenhal. “Traffic congestion, driver stress, and driver aggression”. In: *Aggressive behavior* 25.6 (1999), pp. 409–423.
- [73] Robert Herman and Ilya Prigogine. “A two-fluid approach to town traffic”. In: *Science* 204.4389 (1979), pp. 148–151.

- [74] Geoffrey Hinton et al. “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97.
- [75] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.
- [76] CC Holt. *Forecasting trends and seasonals by exponentially weighted averages*. Carnegie Institute of Technology. Tech. rep. Pittsburgh ONR memorandum, 1957.
- [77] Andreas Horni et al. “Location choice modeling for shopping and leisure activities with MATSim: combining microsimulation and time geography”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2135 (2009), pp. 87–95.
- [78] Johannes Illenberger, Gunnar Flötteröd, and Kai Nagel. “Enhancing matsim with capabilities of within-day re-planning”. In: *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*. IEEE. 2007, pp. 94–99.
- [79] Satu Innamaa. “Short-term prediction of travel time using neural networks on an interurban highway”. In: *Transportation* 32.6 (2005), pp. 649–669.
- [80] Sherif Ishak and Haitham Al-Deek. “Performance evaluation of short-term time-series traffic prediction model”. In: *Journal of Transportation Engineering* 128.6 (2002), pp. 490–498.
- [81] Ashesh Jain et al. “Structural-RNN: Deep learning on spatio-temporal graphs”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5308–5317.
- [82] Bruce N Janson and Frank Southworth. “Estimating departure times from traffic counts using dynamic assignment”. In: *Transportation Research Part B: Methodological* 26.1 (1992), pp. 3–16.
- [83] Yuxuan Ji and Nikolas Geroliminis. “On the spatial partitioning of urban transportation networks”. In: *Transportation Research Part B: Methodological* 46.10 (2012), pp. 1639–1656.
- [84] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).
- [85] Victor L Knoop, Hans van Lint, and Serge P Hoogendoorn. “Traffic dynamics: Its impact on the macroscopic fundamental diagram”. In: *Physica A: Statistical Mechanics and its Applications* 438 (2015), pp. 236–250.
- [86] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.
- [87] Simon Kwocek, Sergio Di Martino, and Wolfgang Nejdl. “Predicting and visualizing traffic congestion in the presence of planned special events”. In: *Journal of Visual Languages & Computing* 25.6 (2014), pp. 973–980.
- [88] Quoc V Le et al. “Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis”. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 3361–3368.

- [89] Jean-Patrick Lebacque. “The Godunov scheme and what it means for first order traffic flow models”. In: *International symposium on transportation and traffic theory*. 1996, pp. 647–677.
- [90] Ludovic Leclercq and Nikolas Geroliminis. “Estimating MFDs in simple networks with route choice”. In: *Procedia-Social and Behavioral Sciences* 80 (2013), pp. 99–118.
- [91] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.
- [92] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. “Trajectory clustering: a partition-and-group framework”. In: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. ACM. 2007, pp. 593–604.
- [93] Li Li, Yuebiao Li, and Zhiheng Li. “Efficient missing data imputing for traffic flow by considering temporal and spatial dependence”. In: *Transportation research part C: emerging technologies* 34 (2013), pp. 108–120.
- [94] Yaguang Li, Rose Yu, et al. “Graph Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting”. In: *arXiv preprint arXiv:1707.01926* (2017).
- [95] Yujia Li, Daniel Tarlow, et al. “Gated graph sequence neural networks”. In: *arXiv preprint arXiv:1511.05493* (2015).
- [96] Michael J Lighthill and Gerald Beresford Whitham. “On kinematic waves. II. A theory of traffic flow on long crowded roads”. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 229. 1178. The Royal Society. 1955, pp. 317–345.
- [97] Todd Litman. “Transportation cost and benefit analysis”. In: *Victoria Transport Policy Institute* 31 (2009).
- [98] Hao Liu, Henk van Zuylen, et al. “Predicting urban arterial travel time with state-space neural networks and Kalman filters”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1968 (2006), pp. 99–108.
- [99] Zhaobin Liu and Satish Sharma. “Statistical investigations of statutory holiday effects on traffic volumes”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1945 (2006), pp. 40–48.
- [100] Xiaolei Ma, Zhuang Dai, et al. “Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction”. In: *Sensors* 17.4 (2017), p. 818.
- [101] Xiaolei Ma, Zhimin Tao, et al. “Long short-term memory neural network for traffic speed prediction using remote microwave sensor data”. In: *Transportation Research Part C: Emerging Technologies* 54 (2015), pp. 187–197.
- [102] H Mahmassani, James C Williams, and R Herman. “Performance of urban traffic networks”. In: *Transportation and Traffic Theory (Proceedings of the Tenth International on Transportation and Traffic Theory Symposium, Cambridge, Massachusetts), NH Gartner, NHM Wilson, editors, Elsevier*. 1987, pp. 1–18.

- [103] Hani S. Mahmassani and Gang-Len Chang. “Experiments with departure time choice dynamics of urban commuters”. In: *Transportation Research Part B: Methodological* 20.4 (1986), pp. 297–320. ISSN: 0191-2615. DOI: [http://dx.doi.org/10.1016/0191-2615\(86\)90045-7](http://dx.doi.org/10.1016/0191-2615(86)90045-7). URL: <http://www.sciencedirect.com/science/article/pii/0191261586900457>.
- [104] Hani S Mahmassani, Didier M Valdes, et al. “Integrated Arterial and Freeway Operation Control Strategies for IVHS Advanced Traffic Management Systems: Research Report”. In: *Work* 1468 (1998), p. 1.
- [105] Amin Mazlounian, Nikolas Geroliminis, and Dirk Helbing. “The spatial variability of vehicle densities as determinant of urban network capacity”. In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 368.1928 (2010), pp. 4627–4647.
- [106] Deepak K Merchant and George L Nemhauser. “A model and an algorithm for the dynamic traffic assignment problems”. In: *Transportation science* 12.3 (1978), pp. 183–199.
- [107] JC Miles and AJ Walker. “The potential application of artificial intelligence in transport”. In: *IEE Proceedings-Intelligent Transport Systems*. Vol. 153. 3. IET. 2006, pp. 183–198.
- [108] Wanli Min and Laura Wynter. “Real-time road traffic prediction with spatio-temporal correlations”. In: *Transportation Research Part C: Emerging Technologies* 19.4 (2011), pp. 606–616.
- [109] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. “Recurrent models of visual attention”. In: *Advances in neural information processing systems*. 2014, pp. 2204–2212.
- [110] H Zare Moayedi and MA Masnadi-Shirazi. “Arima model for network traffic prediction and anomaly detection”. In: *Information Technology, 2008. ITSIM 2008. International Symposium on*. Vol. 4. IEEE. 2008, pp. 1–6.
- [111] Gordon F Newell. “A simplified theory of kinematic waves in highway traffic, part I: General theory”. In: *Transportation Research Part B: Methodological* 27.4 (1993), pp. 281–287.
- [112] Wei Ni and Michael Cassidy. “City-wide traffic control: modeling impacts of cordon queues”. In: 2018.
- [113] Guanghai Ning et al. “Spatially supervised recurrent convolutional neural networks for visual object tracking”. In: *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*. IEEE. 2017, pp. 1–4.
- [114] Iwao Okutani and Yorgos J Stephanedes. “Dynamic prediction of traffic volume through Kalman filtering theory”. In: *Transportation Research Part B: Methodological* 18.1 (1984), pp. 1–11.
- [115] Andre de Palma et al. “Stochastic equilibrium model of peak period traffic congestion”. In: *Transportation Science* 17.4 (1983), pp. 430–453.
- [116] Emilio Parisotto and Ruslan Salakhutdinov. “Neural map: Structured memory for deep reinforcement learning”. In: *arXiv preprint arXiv:1702.08360* (2017).

- [117] Dongjoo Park and Laurence R Rilett. “Forecasting freeway link travel times with a multilayer feedforward neural network”. In: *Computer-Aided Civil and Infrastructure Engineering* 14.5 (1999), pp. 357–367.
- [118] Srinivas Peeta and Athanasios K Ziliaskopoulos. “Foundations of dynamic traffic assignment: The past, the present and the future”. In: *Networks and Spatial Economics* 1.3-4 (2001), pp. 233–265.
- [119] Victor Pillac et al. “A review of dynamic vehicle routing problems”. In: *European Journal of Operational Research* 225.1 (2013), pp. 1–11.
- [120] LA Prashanth and Shalabh Bhatnagar. “Reinforcement learning with function approximation for traffic signal control”. In: *IEEE Transactions on Intelligent Transportation Systems* 12.2 (2011), pp. 412–421.
- [121] Vasili Ramanishka et al. “Top-down Visual Saliency Guided by Captions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.
- [122] Tarun Rambha and Stephen D Boyles. “Dynamic pricing in discrete time stochastic day-to-day route choice models”. In: *Transportation Research Part B: Methodological* 92 (2016), pp. 104–118.
- [123] Mohsen Ramezani, Jack Haddad, and Nikolas Geroliminis. “Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control”. In: *Transportation Research Part B: Methodological* 74 (2015), pp. 1–19.
- [124] Paul I Richards. “Shock waves on the highway”. In: *Operations research* 4.1 (1956), pp. 42–51.
- [125] Swaminathan Sankararaman et al. “Model-driven matching and segmentation of trajectories”. In: *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM. 2013, pp. 234–243.
- [126] Franco Scarselli et al. “The graph neural network model”. In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80.
- [127] David Schrank, Bill Eisele, and Tim Lomax. “TTI’s 2012 urban mobility report”. In: *Texas A&M Transportation Institute. The Texas A&M University System* (2012), p. 4.
- [128] David I Shuman et al. “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains”. In: *IEEE Signal Processing Magazine* 30.3 (2013), pp. 83–98.
- [129] Alexander Skabardonis, Pravin Varaiya, and Karl Petty. “Measuring recurrent and nonrecurrent traffic congestion”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1856 (2003), pp. 118–124.
- [130] Brian L Smith, Kristi G Byrne, et al. “An investigation into the impact of rainfall on freeway traffic flow”. In: *83rd annual meeting of the Transportation Research Board, Washington DC*. 2004.
- [131] Brian L Smith and Michael J Demetsky. “Short-term traffic flow prediction models—a comparison of neural network and nonparametric regression approaches”. In: *Systems, Man, and Cybernetics, 1994. Humans, Information and Technology., 1994 IEEE International Conference on*. Vol. 2. IEEE. 1994, pp. 1706–1709.

- [132] M.J. Smith. “The existence and calculation of traffic equilibria”. In: *Transportation Research Part B: Methodological* 17.4 (1983), pp. 291–303. ISSN: 0191-2615. DOI: [http://dx.doi.org/10.1016/0191-2615\(83\)90047-4](http://dx.doi.org/10.1016/0191-2615(83)90047-4). URL: <http://www.sciencedirect.com/science/article/pii/0191261583900474>.
- [133] Xuan Song, Hiroshi Kanasugi, and Ryosuke Shibasaki. “DeepTransport: Prediction and Simulation of Human Mobility and Transportation Mode at a Citywide Level.” In: *IJCAI*. 2016, pp. 2618–2624.
- [134] Shiliang Sun, Changshui Zhang, and Guoqiang Yu. “A Bayesian network approach to traffic flow forecasting”. In: *IEEE Transactions on intelligent transportation systems* 7.1 (2006), pp. 124–132.
- [135] Shiliang Sun, Changshui Zhang, and Yi Zhang. “Traffic flow forecasting using a spatio-temporal bayesian network predictor”. In: *International Conference on Artificial Neural Networks*. Springer. 2005, pp. 273–278.
- [136] Yichuan Tang, Nitish Srivastava, and Ruslan R Salakhutdinov. “Learning generative models with visual attention”. In: *Advances in Neural Information Processing Systems*. 2014, pp. 1808–1816.
- [137] Samah El-Tantawy, Baher Abdulhai, and Hossam Abdelgawad. “Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto”. In: *IEEE Transactions on Intelligent Transportation Systems* 14.3 (2013), pp. 1140–1150.
- [138] Alexey Tarasov, Felix Kling, and Alexei Pozdnoukhov. “Prediction of user location using the radiation model and social check-ins”. In: *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*. ACM. 2013, p. 8.
- [139] Graham W Taylor et al. “Convolutional learning of spatio-temporal features”. In: *European conference on computer vision*. Springer. 2010, pp. 140–153.
- [140] Arvind Thiagarajan et al. “VTrack: accurate, energy-aware road traffic delay estimation using mobile phones”. In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. ACM. 2009, pp. 85–98.
- [141] Denis Tikunov and Toshikazu Nishimura. “Traffic prediction for mobile network using Holt-Winter’s exponential smoothing”. In: *Software, Telecommunications and Computer Networks, 2007. SoftCOM 2007. 15th International Conference on*. IEEE. 2007, pp. 1–5.
- [142] Daoqin Tong, Carolyn J Merry, and Benjamin Coifman. “Traffic information deriving using GPS probe vehicle data integrated with GIS”. In: *Center for Urban and Regional Analysis and Department of Geography. The Ohio State University. Ohio, USA.[Links]* (2005).
- [143] Martin Treiber and Dirk Helbing. “Reconstructing the spatio-temporal traffic dynamics from stationary detector data”. In: *Cooperative Transportation Dynamics* 1.3 (2002), pp. 3–1.
- [144] Monique Van den Berg et al. “Integrated traffic control for mixed urban and freeway networks: A model predictive control approach”. In: *European journal of transport and infrastructure research EJTIR*, 7 (3) (2007).

- [145] CPIJ Van Hinsbergen and J van Lint. “Bayesian combination of travel time prediction models”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2064 (2008), pp. 73–80.
- [146] J Van Lint, S Hoogendoorn, and H Van Zuylen. “Freeway travel time prediction with state-space neural networks: modeling state-space dynamics with recurrent neural networks”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1811 (2002), pp. 30–39.
- [147] JWC Van Lint and Serge P Hoogendoorn. “A robust and efficient method for fusing heterogeneous data from traffic sensors on freeways”. In: *Computer-Aided Civil and Infrastructure Engineering* 25.8 (2010), pp. 596–612.
- [148] JWC Van Lint, SP Hoogendoorn, and Henk J van Zuylen. “Accurate freeway travel time prediction with state-space neural networks under missing data”. In: *Transportation Research Part C: Emerging Technologies* 13.5 (2005), pp. 347–369.
- [149] JWC Van Lint and CPIJ Van Hinsbergen. “Short-term traffic and travel time prediction models”. In: *Artificial Intelligence Applications to Critical Transportation Issues* 22 (2012), pp. 22–41.
- [150] William S. Vickrey. “Congestion Theory and Transport Investment”. English. In: *The American Economic Review* 59.2 (1969), pp. 251–260. ISSN: 00028282. URL: <http://www.jstor.org/stable/1823678>.
- [151] Oriol Vinyals et al. “Show and tell: A neural image caption generator”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3156–3164.
- [152] Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. “Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach”. In: *Transportation Research Part C: Emerging Technologies* 13.3 (2005), pp. 211–234.
- [153] Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. “Spatio-Temporal Short-Term Urban Traffic Volume Forecasting Using Genetically Optimized Modular Networks”. In: *Computer-Aided Civil and Infrastructure Engineering* 22.5 (2007), pp. 317–325.
- [154] Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. “Short-term traffic forecasting: Where we are and where we’re going”. In: *Transportation Research Part C: Emerging Technologies* 43 (2014), pp. 3–19.
- [155] Eleni Vlahogianni and Matthew Karlaftis. “Testing and comparing neural network and statistical approaches for predicting transportation time series”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2399 (2013), pp. 9–22.
- [156] PC Vythoulkas. “Alternative Approaches to Short-Term Traffic Forecasting For Use In Driver Information Systems”. In: *International Symposium on the Theory of Traffic Flow and Transportation (12th: 1993: Berkeley, Calif.)*. *Transportation and traffic theory*. 1993.
- [157] John Glen Wardrop and JI Whitehead. “Correspondence. Some Theoretical Aspects of Road Traffic Research.” In: *Proceedings of the Institution of Civil Engineers* 1.5 (1952), pp. 767–768.

- [158] MA Wiering. “Multi-agent reinforcement learning for traffic light control”. In: *Machine Learning: Proceedings of the Seventeenth International Conference (ICML’2000)*. 2000, pp. 1151–1158.
- [159] Billy Williams. “Multivariate vehicular traffic flow prediction: evaluation of ARIMA modeling”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1776 (2001), pp. 194–200.
- [160] Peter R Winters. “Forecasting sales by exponentially weighted moving averages”. In: *Management science* 6.3 (1960), pp. 324–342.
- [161] Jeffrey Wojtowicz and William A Wallace. “Traffic management for planned special events using traffic microsimulation modeling and tabletop exercises”. In: *Journal of Transportation Safety & Security* 2.2 (2010), pp. 102–121.
- [162] Cathy Wu, Jérôme Thai, et al. “Cellpath: fusion of cellular and traffic sensor data for route flow estimation via convex optimization”. In: *Transportation Research Procedia* 7 (2015), pp. 212–232.
- [163] Chun-Hsin Wu, Jan-Ming Ho, and Der-Tsai Lee. “Travel-time prediction with support vector regression”. In: *IEEE transactions on intelligent transportation systems* 5.4 (2004), pp. 276–281.
- [164] Yuankai Wu and Huachun Tan. “Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework”. In: *arXiv preprint arXiv:1612.01022* (2016).
- [165] Yuanchang Xie, Yunlong Zhang, and Zhirui Ye. “Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition”. In: *Computer-Aided Civil and Infrastructure Engineering* 22.5 (2007), pp. 326–334.
- [166] SHI Xingjian et al. “Convolutional LSTM network: A machine learning approach for precipitation nowcasting”. In: *Advances in neural information processing systems*. 2015, pp. 802–810.
- [167] Feifei Xu, Zhaocheng He, et al. “Assessing the impact of rainfall on traffic operation of urban road network”. In: *Procedia-Social and Behavioral Sciences* 96 (2013), pp. 82–89.
- [168] Kelvin Xu, Jimmy Ba, et al. “Show, attend and tell: Neural image caption generation with visual attention”. In: *International Conference on Machine Learning*. 2015, pp. 2048–2057.
- [169] Li Yao et al. “Describing videos by exploiting temporal structure”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4507–4515.
- [170] Mehmet Yildirimoglu and Nikolas Geroliminis. “Experienced travel time prediction for congested freeways”. In: *Transportation Research Part B: Methodological* 53 (2013), pp. 45–63.
- [171] Mogeng Yin et al. “A Generative Model of Urban Activities from Cellular Data”. In: *IEEE Trans on Intelligent Transportation Systems (to appear)*. 2017.
- [172] Zheng Zhao et al. “LSTM network: a deep learning approach for short-term traffic forecast”. In: *IET Intelligent Transport Systems* 11.2 (2017), pp. 68–75.
- [173] Nan Zheng, Rashid A Waraich, et al. “A dynamic cordon pricing scheme combining the Macroscopic Fundamental Diagram and an agent-based traffic model”. In: *Transportation Research Part A: Policy and Practice* 46.8 (2012), pp. 1291–1303.

- [174] Weizhong Zheng, Der-Horng Lee, and Qixin Shi. “Short-term freeway traffic flow prediction: Bayesian combined neural network approach”. In: *Journal of transportation engineering* 132.2 (2006), pp. 114–121.
- [175] Xuesong Zhou and Hani S Mahmassani. “A structural state space model for real-time traffic origin–destination demand estimation and prediction in a day-to-day learning framework”. In: *Transportation Research Part B: Methodological* 41.8 (2007), pp. 823–840.
- [176] Henk J. Van Zuylen and Luis G. Willumsen. “The most likely trip matrix estimated from traffic counts”. In: *Transportation Research Part B: Methodological* 14.3 (1980), pp. 281–293. ISSN: 0191-2615. DOI: [http://dx.doi.org/10.1016/0191-2615\(80\)90008-9](http://dx.doi.org/10.1016/0191-2615(80)90008-9). URL: <http://www.sciencedirect.com/science/article/pii/0191261580900089>.

Appendices

Appendix A

Dynamic Origin-Destination Demand Estimation

All entries of the network state vector, $X(t)$ in Equation 3.4 are directly observable quantities except Origin-Destination demand, $D(t)$. Therefore, it must be estimated. Estimation of Origin-Destination demand has been a well studied problem since early 1980's (Zuylen and Willumsen 1980). Recent developments in data-driven optimization techniques allow the use of real-time data to accurately estimate the values of $D(t)$, as well as to update these estimates with more data becoming available with time. One such technique is discussed in this chapter. An *Entropy-Maximization* framework developed by Janson and Southworth (1992) is implemented with an additional convex optimization sub-routine for determining route flows developed by Wu, Thai, et al.(2015).

A.1 Problem Overview

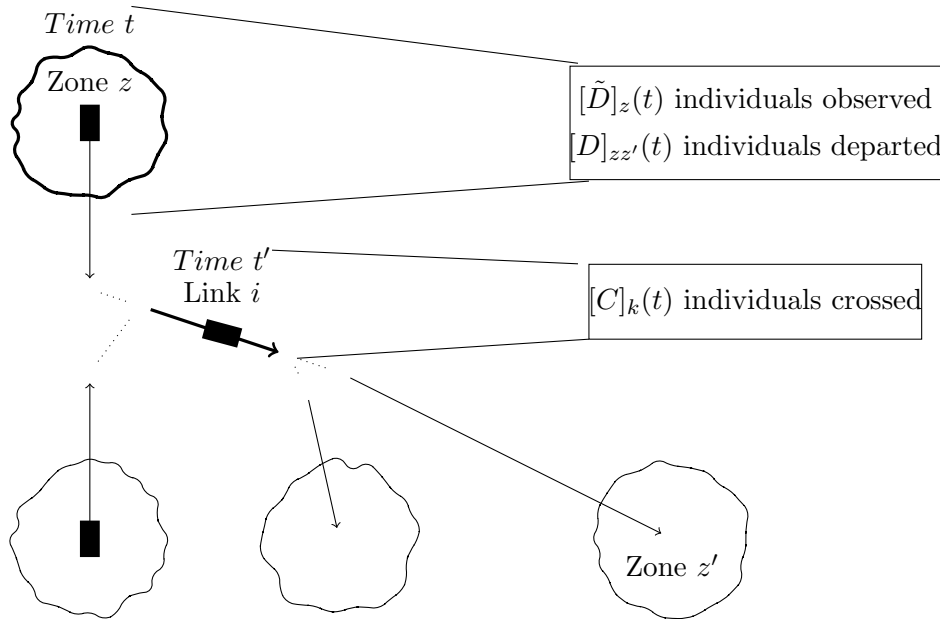


Figure A.1: Figure: Problem Overview For Dynamic Origin-Destination Demand Estimation - The goal is to estimate the number of individuals departing Zone z and traveling towards Zone z' during time interval t , defined as $[D]_{zz'}(t)$.

A road network graph $G = (\mathcal{V}, \mathcal{A})$ is divided into Z zones such that each i^{th} street link lies exactly within its (single) Zone z . The objective here is to estimate the Origin-Destination (O-D) demand matrix in real-time. The O-D demand values during a time interval t are represented as entries in a $|Z|^2$ dimensional vector $D(t)$ where the element labeled $[D]_{zz'}(t)$ represents the number of individuals leaving Zone z and traveling towards Zone z' during time interval t .

The following data sources are assumed to be available:

- (i) At the link-level, complete information on the total number of individuals crossing link i during each time interval t' is assumed. These values are represented through the elements of the vector $[C]_i(t)$. This is an aggregation of individuals whose trips over several O-Ds. The dimensionality of such a vector is $|\mathcal{A}|$. Typically, this information is available from sensors installed inside the roads which detect the passage of any vehicle.
- (ii) Only some of the individuals in Zone z are observed at the end of each time interval t . These values are represented as elements of a $|Z|$ dimensional vector $[\tilde{D}]_z(t)$. This is a lower bound of the total number of individuals present in Zone z at the end of the time interval t . Typically, this information is available from aggregated cell phone *Call Detail Records* (CDRs) up to the granularity of the nearest cell phone tower.
- (iii) Perfect knowledge is assumed regarding link travel times, $TT(t)$, across the whole network at any instance. The dimensionality of such a vector is $|\mathcal{A}|$. Typically, this information can be extracted from travel time APIs for navigation apps.

A.2 Problem Formulation

A.2.1 Traditional Approach

Janson and Southworth (1992) proposed an *Entropy-Maximization* approach for estimating the O-D matrix from link accumulations. The general hypothesis behind this approach is that in the absence of any constraints regarding desired activity schedules, the trip departure choice of individuals is likely to be dispersed over space and time, leading to lower traffic congestion. Janson and Southworth (1992) claim that this hypothesis holds even in the presence of constraints imposed by desired activity schedules with a minor modification. In presence of such constraints, individuals are likely to choose departure times that are as dispersed as possible while ensuring that the desired activity schedules of all individuals are met. The dispersion in departure times may be represented as the entropy of the distribution of the number of departures over space and time. As a result, the problem of estimating the number of individuals departing from any Zone z towards any other Zone z' can be expressed as the problem of maximizing this entropy in the presence of constraints imposed by desired activity schedules. However, since activity schedules of individuals are not directly observable, they propose the use of a proxy that captures such behavior. The proxy chosen here is that of link accumulations on certain links in the network. These link accumulations combined with the structure of the road network provide a rough idea about the times of the day and locations with high activity concentrations. The overall problem formulated by Janson and Southworth (1992) is as

follows:

$$\begin{aligned} \min_{\{D(t)\}_{t \in \mathcal{T}}} \quad & \sum_{t \in \mathcal{T}} -H(D(t)) \\ \text{s.t.} \quad & \sum_{t \in \mathcal{T}} P(t', t) D(t) = C(t), \quad \forall t' \in \mathcal{T}, \end{aligned} \quad (\text{A.1})$$

where:

H represents the Entropy function defined by:

$$H(x) = \sum_{i=1}^n \{x_i * \log(x_i) - x_i\}, \quad n = \text{card}(x). \quad (\text{A.2})$$

$[D]_{zz'}(t)$ represents the number of individuals departing from Zone z to Zone z' during time interval t

$[P]_{i,zz'}(t', t)$ represents the probability that a trip departing Zone z to Zone z' during time interval t uses link i during time interval t'

To solve the optimization problem stated above, the entries of matrix $P(t', t)$ must be estimated first. Janson and Southworth (1992) suggested for solving *Dynamic Traffic Assignment* (DTA) (Merchant and Nemhauser 1978) as a sub-routine for estimating the link-choice probability matrix $P(t', t)$. This involves solving the non-convex optimization problem, as described below, approximately to estimate the number of individuals departing Zone z during time interval t , and heading towards Zone z' and arriving at link i at time interval t' . This number can be represented as entries of the matrix $[\mathcal{D}]_{i,zz'}(t', t)$. The algorithm proposed for solving DTA aims to find an equilibrium condition wherein no individual can change his/her path to reduce the time that he/she takes to reach the desired destination. The problem is formulated as:

$$\begin{aligned} \min_Q \quad & \sum_{(t', t) \in \mathcal{T} \times \mathcal{T}} \sum_{j \in Z \times Z} \phi(A\mathcal{D}_j(t', t)) \\ \text{s.t.} \quad & \sum_{t \in \mathcal{T}} \sum_{j \in Z \times Z} A\mathcal{D}_j(t', t) = C(t), \quad \forall t' \in \mathcal{T}, \end{aligned} \quad (\text{A.3})$$

where:

$$[A]_{i,zz'} = \begin{cases} 1 & \text{if link } i \text{ lies along shortest path (or least disutility path) b/w } z \text{ and } z', \\ 0 & \text{otherwise,} \end{cases}$$

$\mathcal{D}_j(t', t)$ represents the j^{th} column of $\mathcal{D}(t', t)$

ϕ is a non-convex function dependent on the travel utility of individuals

After estimating the entries of the matrix $\mathcal{D}(t', t)$, the entries of the matrix $P(t', t)$ may be estimated by calculating frequency ratios as follows:

$$[P]_{i,zz'}(t', t) = \frac{[\mathcal{D}(t', t)]_{i,zz'}}{\sum_{i \in \mathcal{A}} [\mathcal{D}(t', t)]_{i,zz'}}. \quad (\text{A.4})$$

However, the traditional approach suffers from the following shortcomings:

- (i) Since the DTA problem is non-convex, local minima are not guaranteed to be global minima. There is no approximation guarantee on the values of $[\mathcal{D}]_{i,zz'}(t', t)$ estimated by the DTA solver as compared to the corresponding optimal values $[\mathcal{D}^*]_{i,zz'}(t', t)$. As a result, the estimated values of $[P]_{i,zz'}(t', t)$ may not be accurate.
- (ii) There is also no approximation guarantee on the distribution of the estimated values of $[\mathcal{D}]_{i,zz'}(t', t)$ over space and time. This may significantly affect the quality of the estimated matrix $P(t', t)$ and in turn, may significantly affect the estimated O-D values $D(t)$.
- (iii) This method assumes that individuals maximize their utility based on complete information about the network. But it has since been shown that this condition doesn't hold in practice (Vlahogianni, Karlaftis, and Golias 2014).
- (iv) The decisions taken by individuals cannot be modified once their trip has begun. However, in practice, this is not true. Due to the use of Advanced Travel Information Systems (ATIS), decisions about route-choices are made dynamically (Pillac et al. 2013). The choice of the location of the destination for secondary trips such as shopping might also be dynamic (Horni et al. 2009).

A.2.2 Proposed Approach

The traditional formulation of the problem for dynamic O-D demand estimation is extended with the help of real-time data sources, namely Call Detail Records (CDRs) and travel time APIs. The following extensions are proposed:

1. Adding constraints based on observing a part of all departures:

The total number of trips departing from Zone z in time interval t is greater than or equal to the difference of the number of observed individuals in Zone z at the end of the time interval t and the number of observed individuals at the end of the time interval $(t - 1)$. This data is obtained from Call Data Records (CDRs). The constraint is represented as:

$$\sum_{z' \in Z} [D]_{zz'}(t) \geq \max\{[\tilde{D}]_z(t) - [\tilde{D}]_z(t - 1), 0\}. \quad (\text{A.5})$$

2. Estimation $P(t', t)$ via data-driven approach:

A data-driven estimation of the value of $P(t', t)$ is proposed via the formulation of Quadratic Program (QP) and using network-wide link travel times:

- QP Formulation:

First, it involves formulating a QP problem based on Wu, Thai, et al. (2015). This optimization problem aims to estimate the number of individuals traveling along any given route r in the network during a specified time interval t . This may be represented as elements of a vector $\Gamma(t)$. Typically, the dimensionality of this matrix might be very large. However, Wu, Thai, et al. (2015) showed through experiments in downtown Los Angeles that nearly 95% of all trips were covered if only the top 50 routes between each O-D pair were considered. Such a distribution of route-choices between each O-D pair is hypothesized to be true in most of the large cities. This allows us to consider

only a constant M number of trips between each O-D pair. The dimensionality of the vector $\Gamma(t)$ reduces to $M|Z|^2$ through this transformation. The proposed approach assumes the availability of individual-level data in a form of a cellpath, c , defined as a time-stamped sequence of discrete regions within which a user can be located during a trip. It is a common format of mobility data available from cellular network carriers and IT service providers. For this approach, we assume that the area covered by the road network is divided into cell paths C . The problem formulation is as follows:

$$\begin{aligned} \min_{\Gamma(t)} \quad & \frac{1}{2} \|\tilde{A}\Gamma(t) - C(t)\|_2^2 \\ \text{s.t.} \quad & U\Gamma(t) = F(t) \\ & \Gamma(t) \succeq 0, \end{aligned} \tag{A.6}$$

where:

$$[\tilde{A}]_{i,r} = \begin{cases} 1 & \text{if link } i \text{ lies along the route } r, \\ 0 & \text{otherwise.} \end{cases}$$

$$[U]_{c,r} = \begin{cases} 1 & \text{if cellpath } c \text{ is covered by route } r, \\ 0 & \text{otherwise.} \end{cases}$$

$[F]_c(t)$ = total number of observed individuals along cellpath c during t

- Reduction To Least Squares:

The QP formulation can be further reduced to a least squares formulation. The constraint in equation (A.6) can be re-written as:

$$\begin{aligned} U\Gamma(t) &= F(t) \\ \iff \sum_{\hat{r}^* \in \mathcal{R}^c} [\Gamma(t)]_{\hat{r}^*} &= [F]_c(t), \quad \forall c \in C, \end{aligned} \tag{A.7}$$

where:

$\mathcal{R}^c$ = subset of routes along cellpath c

Here, note that $\mathcal{R}^c$ is a disjoint set since each route has at most 1 cellpath associated with it. Suppose route $\hat{r}^*$ has associated cellpath $\hat{c}$. Then, the following change of variables can be applied:

$$[\tilde{\Gamma}]_{\hat{c}}(t) = [F]_{\hat{c}}(t).$$

The problem reduces to least squares as follows:

$$\begin{aligned} \min_{\tilde{\Gamma}(t)} \quad & \frac{1}{2} \|\tilde{A}(t)\tilde{\Gamma}(t) - C(t)\|_2^2 \\ \text{s.t.} \quad & \mathbf{1}^T \tilde{\Gamma}(t) = 1 \\ & \tilde{\Gamma}(t) \succeq 0. \end{aligned} \tag{A.8}$$

$$[\tilde{A}]_{i,c}(t) = \begin{cases} [F]_c(t) & \text{if link } i \text{ lies along any route } r \text{ which covers cell path } c, \\ 0 & \text{otherwise.} \end{cases}$$

Wu, Thai, et al. (2015) then propose for solving the least squares problem by eliminating equality constraints by a technique discussed in Boyd and Vandenberghe (2004), Section 4.2.4. They then apply accelerated gradient descent to find the optimal value, $\Gamma^*(t)$.

- Estimating route-choice probability matrix $\tilde{P}(t)$:

It may be noted that each route r has at the most one associated departing Zone z and one associated arrival Zone z' . Therefore, the solution to the least squares problem can be used to calculate the route-choice probability matrix (i.e., the probability of taking route r for an individual departing Zone z and traveling to Zone z') as follows:

$$[\tilde{P}]_{zz',r} = \frac{[\Gamma^*(t)]_r}{\sum_{\tilde{r}' \in \mathcal{R}^r} [\Gamma^*(t)]_{\tilde{r}'}} \quad (\text{A.9})$$

where:

$\mathcal{R}^r$ = subset of routes departing Zone z and arriving at Zone z'

- Estimating link departure time incidence matrix $\hat{A}(t', t)$:

Real-time information about network-wide link travel time can be used to derive a matrix storing the probability that a trip departing along the route r during time interval t reaches link i at time interval t' . Elements of this matrix are represented as:

$$[\hat{A}]_{r,i}(t', t) = \begin{cases} 1, & \text{if link } i \text{ lies along the route } r \text{ and travel time to link } i \\ & \text{lies in the interval } (t' - t) \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.10})$$

- Representing $P(t', t)$ as a product of two estimated matrices:

The probability matrix $P(t', t)$ can be estimated as a product of the two matrices estimated in the previous two steps:

$$\begin{aligned} [P]_{i,zz'}(t', t) &= \sum_{r \in \mathcal{R}} [\tilde{P}]_{zz',r}(t) * [\hat{A}]_{r,i}(t', t) \\ \Rightarrow P(t', t) &= \tilde{P}(t) * \hat{A}(t', t), \end{aligned} \quad (\text{A.11})$$

where:

$\mathcal{R}$ = set of all routes that are analyzed in the network

3. Addition of prior OD estimates from Wu, Thai, et al. (2015):

Wu, Thai, et al. (2015) assume a quasi-static setting where flows are constant along each route and propose a convex optimization framework that makes use of cell phone CDRs to estimate route flow. While the quasi-static assumption may not always hold true for time-varying networks, it does provide a prior estimate for ODs which may be used as part of the Entropy Maximization framework. A KL-divergence term between the prior and estimated O-Ds is introduced in the objective function to penalize heavy discrepancy from prior estimates. A hyper-parameter term ν is introduced to determine the relative weight given to the KL-divergence term. This term depends on the accuracy of the prior O-D estimates which in turn depends on the validity of the quasi-static assumption. The value of this term for a given scenario is determined through cross-validation. The new objective function is as shown below:

$$\min_{\{D(t)\}_{t \in \mathcal{T}}} \sum_{t \in \mathcal{T}} \nu D_{KL}(D(t), \hat{D}(t)) - H(D(t)), \quad (\text{A.12})$$

where:

$D_{KL}(a, b)$ = KL-divergence between the distributions of vectors a and b

$\hat{D}(t) = \sum_{\tilde{r}' \in \mathcal{R}^r} [\Gamma^*(t)]_{\tilde{r}'}$ (where $\Gamma^*(t)$ and $\mathcal{R}^r$ are as described in equation A.9)

ν = hyper-parameter for relative weight between the KL-divergence term and Entropy term

A.2.3 Overall Problem Formulation

The overall optimization problem is formulated as follows:

$$\begin{aligned} \min_{\{D(t)\}_{t \in \mathcal{T}}} \quad & \sum_{t \in \mathcal{T}} \nu D_{KL}(D(t), \hat{D}(t)) - H(D(t)) \\ \text{s.t.} \quad & \sum_{t \in \mathcal{T}} P(t', t) D(t) = C(t'), \quad \forall t' \in \mathcal{T} \\ & \sum_{z' \in \mathcal{Z}} [D]_{zz'}(t) \geq \max\{[\tilde{D}]_z(t) - [\tilde{D}]_z(t-1), 0\}, \end{aligned} \quad (\text{A.13})$$

where:

ν , $\hat{D}(t)$ and $P(t', t)$ are estimated using the data-driven approach described above

A.2.4 Proposed Algorithm

The overall algorithm proposed for dynamic O-D estimation is as follows:

Algorithm For Dynamic O-D Estimation

Require:

- Set of all O-D pairs zz'
- Set of all cellpaths c
- Set of all routes r
- Road network graph G
- link accumulations $C(t)$

At each time step t :

For each route r , find corresponding cellpath c

For each route r , find set of links i along the route and populate matrix $\tilde{A}(t)$

Solve LS problem A.8 and apply a change of variables to obtain $\Gamma(t)$

Estimate the departure time link incidence matrix $\hat{A}(t', t)$ from equation A.10

For each O-D pair zz' :

Estimate route-choice probability $[\tilde{P}]_{zz', r}(t)$ using equation A.9

Estimate link use probability $[P]_{i, zz'}(t', t)$ using equation A.11

Observe a part of the departures from CDR data $\sum_{z' \in \mathcal{Z}} [D]_{zz'}(t)$

Solve convex optimization problem A.13 to obtain O-D estimate $D(t)$

Repeat for $\hat{t} = t - h$ and update $D(\hat{t})$; if no significant change, then break

A.2.5 Experimental Results

Simulation experiments were conducted for the test network described in Figure (7.1) with simulated data generated for commute trips. The equilibrium demand patterns were determined by running simulations on agent-based traffic simulation software *MAT-Sim*. The ground truth O-D patterns were recorded and compared with O-D estimation using the proposed algorithm (see section A.2.4) with updates made at every 5 minutes. The accuracy of estimates was determined through the RMSE values between the ground truth O-Ds and the estimated O-Ds. To simulate the departure constraints based on observing a part of the departures, partial observations were assumed to follow a uniform distribution in the range of $[0, \frac{\text{CDR coverage}}{100} * \text{True departure count}]$.

The following section describes the effect of variation in parameter ν , the effect of quality of priors, the effect of observing a part of all departures through cell phone CDRs and the effect of updating predictions over time on the accuracy of O-D estimates (Equation (A.13)).

Tuning of Hyper-parameter ν Based on Quality of Priors

The first step was to tune hyper-parameter ν in Equation (A.13) based on the quality of priors, $\hat{D}(t)$.

For this analysis, the CDR coverage was assumed to be 100 percent. Next, ν was tuned based on average RMSE values of demand across all O-Ds during peak demand hours. Finally, the amount of potential noise in the prior estimate, $\hat{D}(t)$ was varied by assuming a Gaussian noise term similar to that introduced in equations (5.3) and (5.4). The value of a prior noise may be represented as:

$$\hat{D}_N(t) = (\hat{D}(t) + \hat{D} \odot \epsilon)_+, \epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma), \Sigma = \sigma * I, \quad (\text{A.14})$$

where $\odot$ represents element-wise multiplication of vectors

Here, we also impose a constraint that the prior O-D estimates cannot be negative. Now, the value of σ was varied and the corresponding optimal ν was determined by minimizing the RMSE. The simulation results are summarized in Figure A.2.

As per intuition, with noisier prior estimates (i.e., higher value of σ), the optimal value of ν for the best estimates decreases since we have lower confidence on the priors. For $\sigma = 0$ and $\sigma = 0.25$, the RMSE was a decreasing function of ν . This indicates that we don't require any further correction in prior estimates through link accumulations and observing a part of the departures. However, for $\sigma = 0.5$ and $\sigma = 0.75$, the optimal value for $\log(\nu)$ was 3, which implies that the optimal value of ν was 1000.

In real-world scenarios, the quality of prior estimates may be first evaluated by determining the percentage of accuracy based on previously observed demand during peak hours. In case the accuracy is high (error ≤ 25 percent), there is no need to re-evaluate O-Ds. In case the accuracy is low, the optimal value of parameter ν may be determined using plots similar to those in Figure A.2.

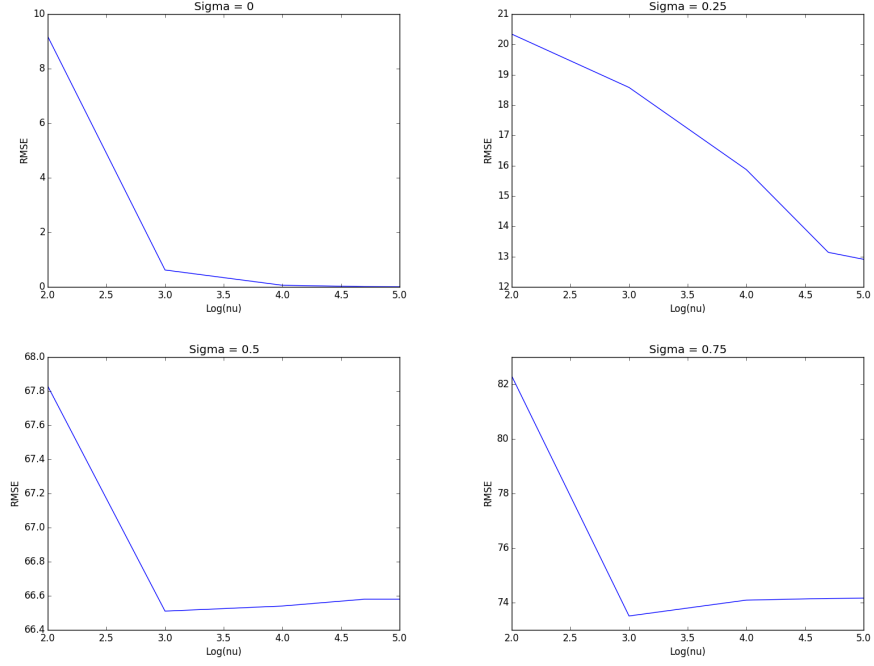


Figure A.2: Variation of RMSE vs $\log(\nu)$ for various values of σ (ν as defined in Equation (A.13) and σ as defined in Equation (A.14))

Temporal Distribution of Errors

The temporal distribution of RMSE across a day was plotted in Figure A.3 for $\nu = 1000$ and $\sigma = 0.5$ for two scenarios:

- (i) with constraints based on observing a part of the departures and
- (ii) without any observed departures.

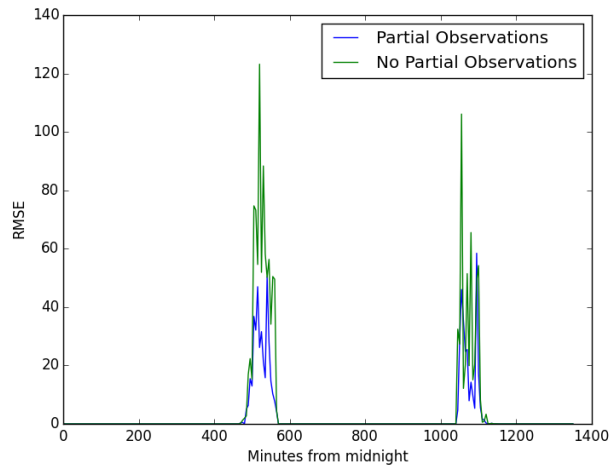


Figure A.3: Variation of RMSE vs time of day for $\nu = 1000$ and $\sigma = 0.5$

The plot shows that the addition of constraints based on observing a part of the departures helps in reducing the overall RMSE across all O-D pairs during peak hours. Note that

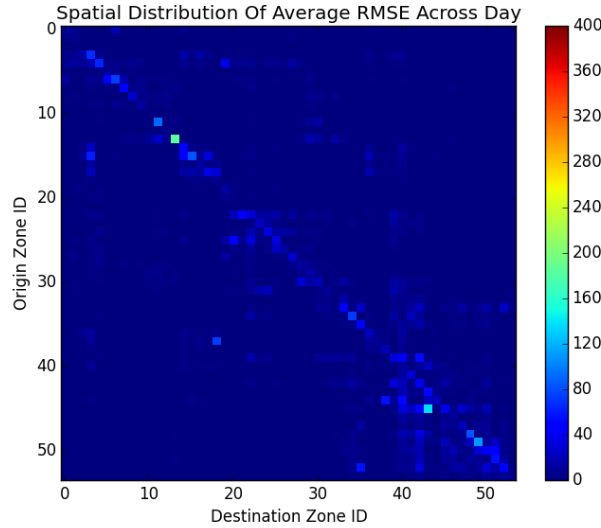
the feasible region with constraints is a subset of the feasible region without constraints. Therefore, the optimal objective value with these constraints was at least as large as that without these constraints. However, the optimal value was closer to the ground truth. This was because ground truth observations were influenced by daily activity plans that may make the observed O-D counts deviate from the entropy maximization estimate as well as the prior estimate.

Spatial Distribution of Errors

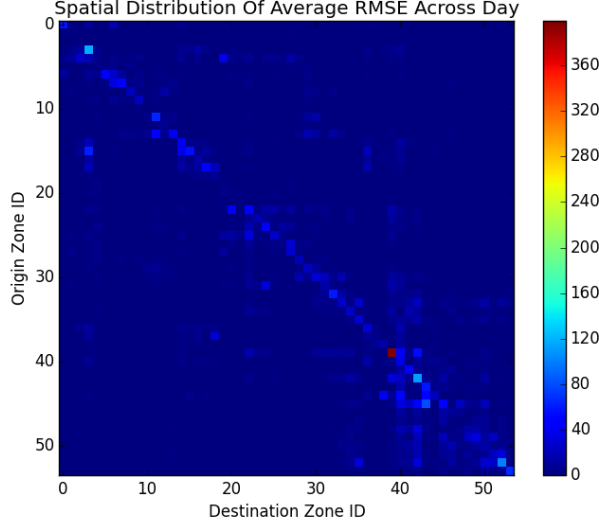
The spatial distribution of RMSE across all O-D pairs averaged over the course of the day was plotted in Figure A.4 for $\nu = 1000$ and $\sigma = 0.5$ for two scenarios:

- (i) with constraints based on observing a part of the departures and
- (ii) without any observed departures

From both figures, it can be noted that since most errors lie along the diagonal, trips originating and terminating in the same zone are not predicted accurately in either framework. This might be because the starting position of all trips was assumed to be at the centroid of each zone. Therefore, the probability matrix $P(t', t)$ does not incorporate the time required to reach the centroid of a particular zone as well as the possibility of trips never traveling through the centroid of the origin. These errors start to dissipate in trips involving well separated origins and destinations since most of the travel time is spent on the freeway part of the trip and the inaccuracies during the trip start/end don't play a crucial role. This problem may be rectified by updating the map to include a more detailed network.



(a) With constraints based on observing a part of the departures



(b) Without any observed departures

Figure A.4: Variation of RMSE over O-D pairs averaged over entire day for $\nu = 1000$ and $\sigma = 0.5$

It may also be observed that both frameworks tend to have higher errors when estimating O-Ds for trips with origins and destinations in the Zones 38-44. These zones represent Santa Clara county where trip lengths and commuting times are typically shorter than average in the Bay Area because most individuals residing here work in the nearby Silicon Valley (see <http://www.vitalsigns.mtc.ca.gov/>). As a result, it is harder to judge the exact origin/destination of vehicles observed on each link.

Finally, there are more “light blue” regions in the heatmap for RMSE with a part of the departures observed. This indicates that errors were more evenly spread out across various zones when constraints based on a part of all departures were added. Without these constraints, the error was heavily concentrated on trips with origin and destination Zone 39. This effect suggests that the addition of these constraints prevents extremely high errors, but may lead to small errors across multiple zones. This is a desirable property since congestion patterns typically depend on whether aggregate demand is greater than aggregate capacity in a particular zone. Therefore, large errors in even a few demands may lead to higher variation in the estimates of congestion as compared to small errors in a larger number of demands.

Effect of Updating Estimates Over Time

With more information getting revealed over time, the change in average RMSE across all O-D estimates over time was studied to measure the effect of updating estimates of O-D. The value of ν and σ were set as 1000 and 0.5 respectively and constraints based on observing a part of all departures were also included in the optimization framework. In Figure A.5, the change in the RMSE for the O-D estimate across all zones at $t = 540$ mins, and $t = 545$ mins (start of morning peak) was studied between $t = 540$ mins and $t = 720$ mins. Updates for estimates were made every 35 mins. As per intuition, the RMSE decreases with incoming information. The effect was more pronounced when the RMSE values were higher (as in the case of O-D estimates at $t = 545$ mins).

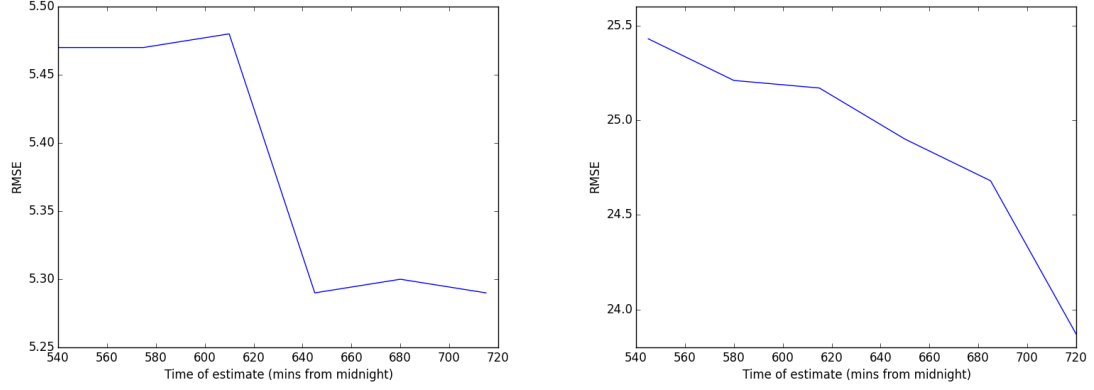


Figure A.5: Change in the RMSE for the O-D estimate across all zones with incoming information for $\nu = 1000$ and $\sigma = 0.5$ - (a) RMSE at $t = 540$ mins with updates between $t = 540$ mins and $t = 715$ mins every 35 mins, (b) RMSE at $t = 545$ mins with updates between $t = 545$ mins and $t = 720$ mins every 35 mins

Effect of Cell Phone Data Coverage

The next experiment was to test the effect of cell phone data coverage on the quality of O-D estimates. To mimic the effect of various amounts of cell phone coverage, it was assumed that partial observations follow a uniform distribution in the range of $[0, \frac{\text{CDR coverage}}{100} * \text{True departure count}]$. The value of ν and σ were set as 1000 and 0.5 respectively. The RMSE for O-D estimates across all O-Ds was calculated for peak hours only. The variation of RMSE with cell phone coverage is displayed in Figure A.6. As per intuition, the addition of more accurate constraints significantly reduces overall RMSE.

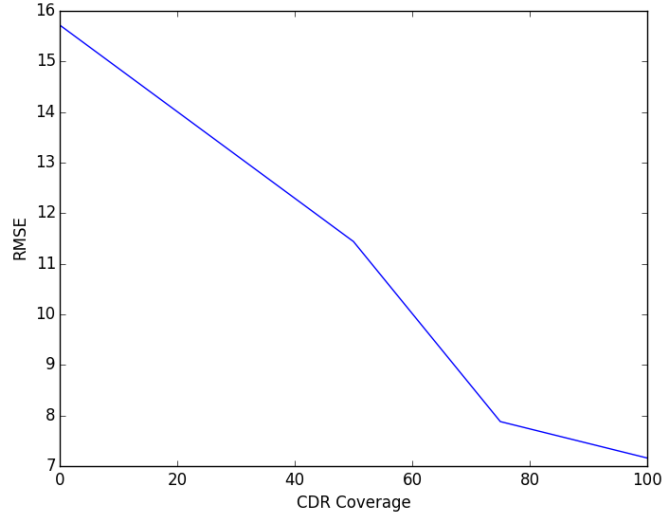


Figure A.6: Change in the RMSE for the O-D estimate across all zones during peak hours with varying cell phone data coverage

Effect of Noise in the Link Choice Probability Matrix

The final experiment was designed to test the impact of noise in the link choice probability matrix which may arise because of inaccurate predictions from travel time APIs, the

invalidity of the quasi-static setting assumption made by Wu, Thai, et al. (2015) or insufficient sample of trips used for estimating $P(t', t)$. To approximate the impact of noise in the data, $P(t', t)$ matrix was corrupted by a Gaussian noise term. The noisy link choice probability matrix may be represented as:

$$P(t', t)_N(t) = (P(t', t) + P(t', t) \otimes \epsilon)_+, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma), \quad \Sigma = \sigma * I, \quad (\text{A.15})$$

where $\otimes$ represents Hadamard product of matrices

Figure A.7 displays the change in average RMSE across all O-D estimates during peak hours with the increase in σ . We can notice that the RMSE was quite sensitive to noise in the link choice probability matrix. However, if the quality of the data provided through travel time APIs improves, we can expect to see large improvements in the quality of the O-D estimates.

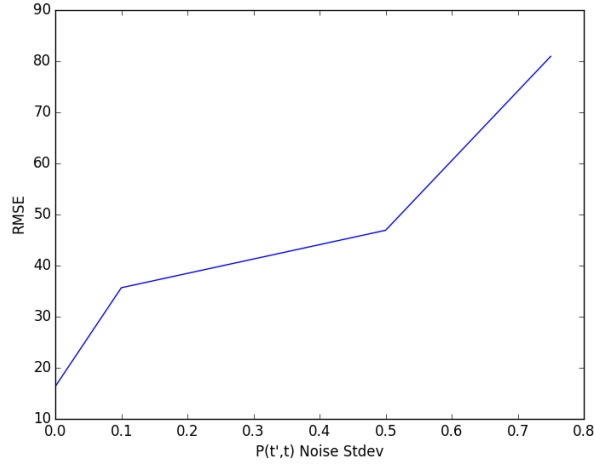


Figure A.7: Change in the average RMSE for the O-D estimate across all zones during peak hours with varying noise in link choice probability matrix $P(t', t)$

Appendix B

Steps Executed By A Typical Long Short-Term Memory (LSTM) Model

A typical LSTM model (Hochreiter and Schmidhuber 1997) undertakes the following operations at time step t :

$$\begin{aligned}
 fg_t &= \sigma(W_{f,t}^T[h_{t-1}, x_t] + b_{f,t}) \\
 i_t &= \sigma(W_{i,t}^T[h_{t-1}, x_t] + b_{i,t}) \\
 o_t &= \sigma(W_{o,t}^T[h_{t-1}, x_t] + b_{o,t}) \\
 c_t &= fg_t \odot c_{t-1} + i_t \odot \tanh(W_{c,t}^T[h_{t-1}, x_t] + b_{c,t}) \\
 hid_t &= o_t \odot \tanh(c_t),
 \end{aligned} \tag{B.1}$$

where:

fg_t , i_t , o_t , c_t and hid_t are the forget, input, output, cell-state, and hidden state values at time t

$W_{f,t}$, $W_{i,t}$, $W_{o,t}$ and $W_{c,t}$ are the weight vectors for the forget, input, output, and cell-state gates respectively for the hidden state at time $(t - 1)$ and observed state at time t

$b_{f,t}$, $b_{i,t}$, $b_{o,t}$ and $b_{c,t}$ are the bias vectors for the forget, input, output, and cell-state gates respectively at time t

σ and $\tanh$ represent the Sigmoid and Hyperbolic Tangent Activation Functions respectively

$\odot$ represents the Hadamard product between two vectors

If the model implements stacked LSTM layers, the last hidden layer is calculated iteratively from the previous hidden layers with usual weights, biases and activation functions.

For the model proposed in equation 3.5, the final output can be expressed in terms of the input as:

$$\begin{aligned}
 [\hat{\xi}^z(t + h + p), \hat{\xi}^z(t + h + p - 1), \dots, \hat{\xi}^z(t + h)] &= W_{\text{LSTM}}^T[hid_{t-h}, \dots, hid_{t-h-p}] + B_{\text{LSTM}} \\
 &= \sum_{i=0}^p \{W_{\text{LSTM}_i} * hid_{t-h-i} + B_{\text{LSTM}_i}\},
 \end{aligned} \tag{B.2}$$

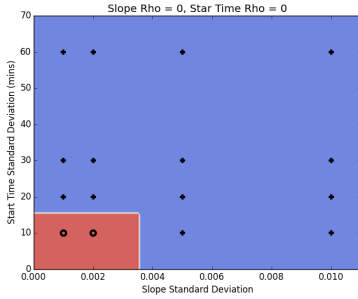
where:

W_{LSTM} and B_{LSTM} are weight and bias vectors for the final prediction.

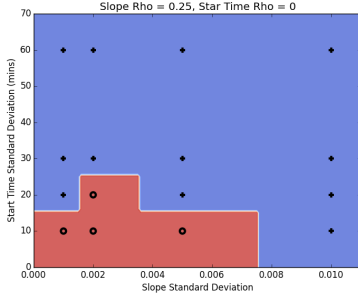
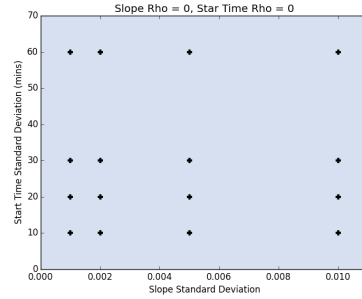
Appendix C

Model Robustness - Correlated demand start time and demand slope

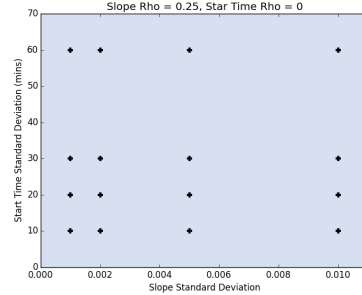
To study model sensitivity with both start times and slopes produced by AR 1 processes (Equations (5.1) and (5.2)), the relative performance of the LSTM model versus the 1-NN model was plotted. We have four possible parameter values, namely $\rho_{st,z}$, $\rho_{sl,z}$, $\sigma_{st,z}$, and $\sigma_{sl,z}$, that govern the demand pattern. For each combination of $(\rho_{st,z}, \rho_{sl,z})$, the efficiency boundary was plotted (Figure C.1) using DT classifier. We can notice that in each case, the figures on the right have greater region shaded “blue”. This shows that, as seen in section 5.1, the LSTM model outperforms the 1-NN model for a larger combination of parameter values when the prediction horizon is shorter.

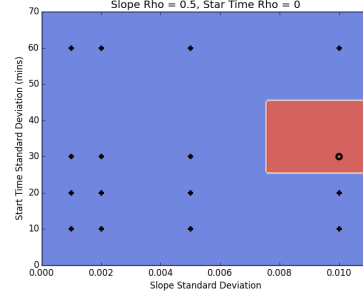
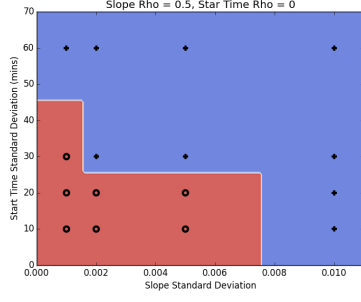


(a) $\rho_{st,z} = 0$, $\rho_{sl,z} = 0$, $h = 2 \text{ hrs}$, $p = 24 \text{ hrs}$ (b) $\rho_{st,z} = 0$, $\rho_{sl,z} = 0$, $h = 2 \text{ hrs}$, $p = 2 \text{ hrs}$

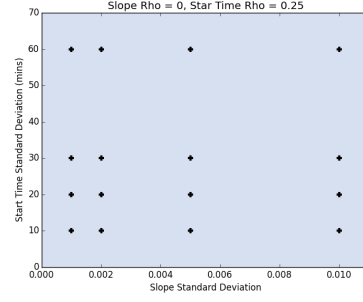
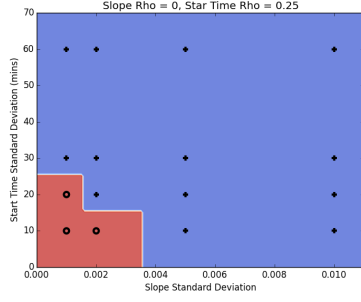


(c) $\rho_{st,z} = 0$, $\rho_{sl,z} = 0.25$, $h = 2 \text{ hrs}$, $p = 24 \text{ hrs}$ (d) $\rho_{st,z} = 0$, $\rho_{sl,z} = 0.25$, $h = 2 \text{ hrs}$, $p = 2 \text{ hrs}$

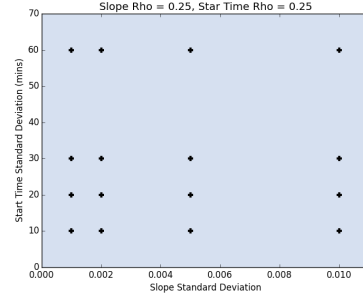
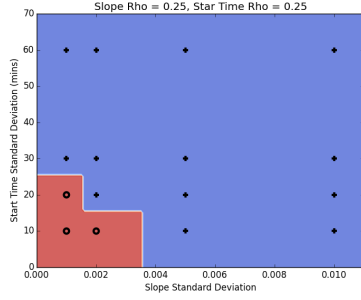




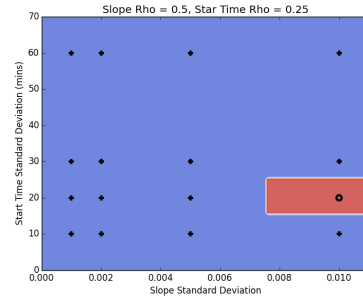
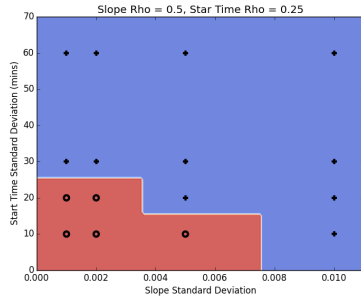
(e) $\rho_{st,z} = 0, \rho_{sl,z} = 0.5, h = 2 \text{ hrs}, p = 24 \text{ hrs}$ (f) $\rho_{st,z} = 0, \rho_{sl,z} = 0.5, h = 2 \text{ hrs}, p = 2 \text{ hrs}$



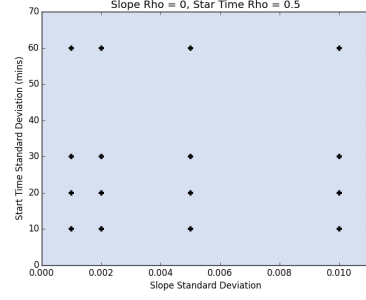
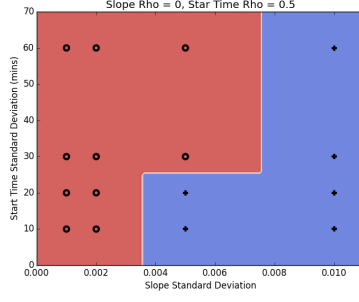
(g) $\rho_{st,z} = 0.25, \rho_{sl,z} = 0, h = 2 \text{ hrs}, p = 24 \text{ hrs}$ (h) $\rho_{st,z} = 0.25, \rho_{sl,z} = 0, h = 2 \text{ hrs}, p = 2 \text{ hrs}$



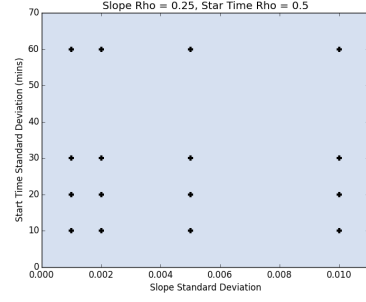
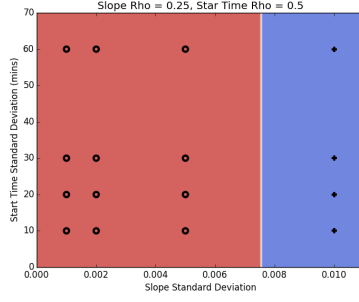
(i) $\rho_{st,z} = 0.25, \rho_{sl,z} = 0.25, h = 2 \text{ hrs}, p = 24 \text{ hrs}$ (j) $\rho_{st,z} = 0.25, \rho_{sl,z} = 0.25, h = 2 \text{ hrs}, p = 2 \text{ hrs}$



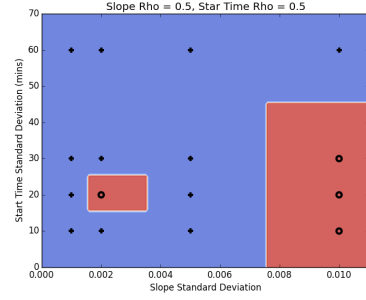
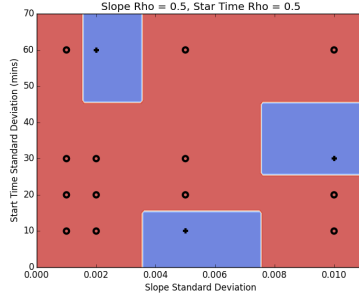
(k) $\rho_{st,z} = 0.25, \rho_{sl,z} = 0.5, h = 2 \text{ hrs}, p = 24 \text{ hrs}$ (l) $\rho_{st,z} = 0.25, \rho_{sl,z} = 0.5, h = 2 \text{ hrs}, p = 2 \text{ hrs}$



(m) $\rho_{st,z} = 0.5, \rho_{sl,z} = 0, h = 2 \text{ hrs}, p = 24 \text{ hrs}$ (n) $\rho_{st,z} = 0.5, \rho_{sl,z} = 0, h = 2 \text{ hrs}, p = 2 \text{ hrs}$



(o) $\rho_{st,z} = 0.5, \rho_{sl,z} = 0.25, h = 2 \text{ hrs}, p = 24 \text{ hrs}$ (p) $\rho_{st,z} = 0.5, \rho_{sl,z} = 0.25, h = 2 \text{ hrs}, p = 2 \text{ hrs}$



(q) $\rho_{st,z} = 0.5, \rho_{sl,z} = 0.5, h = 2 \text{ hrs}, p = 24 \text{ hrs}$ (r) $\rho_{st,z} = 0.5, \rho_{sl,z} = 0.5, h = 2 \text{ hrs}, p = 2 \text{ hrs}$

Figure C.1: Efficient bounds for prediction accuracy LSTM model vs the 1-NN model as a function of the amount of correlation in both start time and slope of demand and amount of statistical fluctuation. The points labeled “+” are parameter values for which the LSTM model performed better and the points labeled “o” are parameter values for which 1-NN model performed better. To illustrate the boundary, Decision Tree (DT) classifier model was fit on the points.

Appendix D

Model Robustness - Partially observed input variables for large network

The effect of partially observed input variables on prediction accuracy was analyzed for the network described in Figure 7.1 and demand Scenario 1 described in table 7.1. For this experiment, MCL was only predicted for Zone 1 as per Figure 7.1(b). Three possible scenarios defined by vector $O^z(t)$ in section 5.3 are analyzed:

$$O^z(t) = \begin{bmatrix} D_{1,z} \\ D_{2,z} \\ \cdot \\ \cdot \\ \cdot \\ D_{54,c} \end{bmatrix}, O^z(t) = \begin{bmatrix} D_{1,z} \\ D_{2,z} \\ \cdot \\ \cdot \\ \cdot \\ D_{27,z} \end{bmatrix} \text{ and } O^z(t) = \begin{bmatrix} D_{28,z} \\ D_{29,z} \\ \cdot \\ \cdot \\ \cdot \\ D_{54,z} \end{bmatrix}$$

For each scenario, the prediction accuracy was evaluated for 6-8 AM, 2-4 PM, and 8-10 PM. The simulation results are described in Table D.1.

In this scenario, no single variable was responsible for the accuracy of the prediction. Therefore, the impact of the removal of variables was not significant. This was because the LSTM model aims to find an approximate function in the absence of the omitted variables and tries to appropriately weigh the given inputs to make up for the missing input demands.

Prediction Time Variables Omitted	6-8 AM	2-4 PM	8-10 PM
None	0.344	0.172	0.302
Zone 28-54 demands	0.320	0.412	0.278
Zone 1-27 demands	0.345	0.316	0.376

Table D.1: Root Mean Squared error of LSTM predictions in a large network in various scenarios with omitted variables.

Appendix E

Model Application - Neural Attention Model performance for large network

The Neural Attention Model-based framework (see equation (6.6)) was tested for predicting congestion in Zone 2 in the freeway network described in Figure 7.1(a) and demand Scenario 1 described in Table 7.1. The analysis was restricted to only the evaluation of the relative importance of all O-D demands between 10AM-12PM with the destination zone as Zone 2. While the ground truth contribution of each O-D demand towards MCL was hard to evaluate, a proxy for the same was the percentage change in MCL in case that particular O-D demand was absent. We expect O-D demands from zones with high relative importance to have the maximum impact on MCL.

Table E.1 displays the top four contributing O-D demands (out of the possible 54) towards MCL prediction as per the Neural Attention framework. The percentage change in MCL in case of absence was also highest for these four O-Ds. We can note that these O-D demands contribute 78.1% of the total Neural Attention weights. The predicted order of importance for these O-D demands was in accordance with the order of percentage change in MCL values if the O-D demand was absent. The zone that contributes the maximum weight was Zone 2 itself. This was due to several short trips within the zone during this time. One possible reason was the presence of several tech companies in the region and the preference of employees to live close to the work place¹.

att_{O-D}	Relative Weight	MCL % change if O-D absent
	58.7 %	37.3 %
	9.1 %	13.4 %
	6.1 %	5.8 %
	4.2 %	2.5 %

Table E.1: Top four O-D demand attentions and the corresponding percentage change in MCL value if the O-D was absent, while predicting MCL in the Zone 2 between 10AM-12PM.

¹See <http://www.vitalsigns.mtc.ca.gov/>