COMPUTATIONAL FRAMEWORK FOR MODELING INFRASTRUCTURE NETWORK PERFORMANCE AND VULNERABILITY

A Thesis

Submitted to the Department

of Civil, Architectural, and Environmental Engineering

Drexel University

by

Seyed Hossein Hosseini Nourzad

in partial fulfilment of the

requirements for the degree

of

Doctor of Philosophy

June, 2014

© Copyright 2014
Seyed Hossein Hosseini Nourzad. All Rights Reserved.

DEDICATIONS

I dedicate this thesis to

My beloved **mother**, departed **father**, and dearest **wife**
from whom I learned enthusiasm, perseverance and dedication
I owe to them for all I am

ACKNOWLEDGMENTS

First and foremost, I present my sincerest gratitude to Dr. Anu Pradhan, my research advisor, who greatly supported me throughout my PhD studies with his patience, knowledge and constructive criticism. Besides being a research advisor, he has always been a great friend and brother to me. I am also very grateful to the members of my doctoral dissertation committee: Dr. A. Emin Aktan, Dr. Ivan Bartoli, Dr. Patrick Gurian, Dr. Franklin Moon, Dr. Dario D. Salvucci, and Dr. Yi-Chang Chiu who generously gave their precious time and expertise to read and better my work.

I owe all my accomplishments to my truly exceptional family; to my mother and mother in-law; to my departed father, who is thoughtfully remembered; to my father in-law, who is a grand master, to my grandmothers, who always want to see my success; to my siblings and my closest friends forever, Fatemeh and Ehsan, whose cheer and kindness lighted up my life; and to Amir, Morteza, Saeideh, Alireza, Reihaneh, and Amin. There has not been a single moment in my life without their unconditional support and desire for my success. Words are not able to express my gratitude to them.

The purest thanks go to my lovely wife, Sedigheh, for her endless love, support understanding and for being my better half. For being there when I needed her and believing in me even when I was frustrated. She never gave up on me. Very special thanks also to my awesome sons, Mohammad and Ali, without whom nothing is joyful.

The hardworking and supportive CAEE faculty and staff as well as the staff at the Graduate Office at Drexel University are gratefully recognized for their relentless support. During the last four years at Drexel, I was also fortunate to have Yusuf, Ehsan, Mohammad, Fuad, Long, and Shi as my great friends.

Special thanks go to Mr. Fang Yuan, Mr. William Tsay, and Mr. Michael Becker (from Delaware Valley Regional Planning Commission) for their help and support.

Haghkar family, Dr. Alavi, Mrs. Motemavelian, Mrs. Mohammadi, Mr. Shokrollah, Mohammad, Ali, Hamid, Amir, Hossein, Sajad, Abbas, Majid, Morteza, Hamed, Amin, Mahdi and all families and friends at Mahdieh community are thoughtfully remembered for their kindness, help and support. I would also like to thank my old friends, Mojtaba, Reza, Mohsen, Mohammad, Nasser, Hamid, Alireza, Vahid, Abbas, Mehdi, Sajjad, Hossein, Hamed, Majid, Hassan, Saber, Meysam, Ali, Arash, Ehsan, Najma, Shahab, Farshad and Farid with whom I shared my most memorable moments of life.

I am truly indebted to Dr. Hadi Sabouhi for all the hours he spent training me as an altruistic human being. His persistent help and guidance, immense motivation and dedication, and his generosity has been instrumental in inspiring and encouraging me during these years. I would also like to express my gratitude to my late teacher from high school, Mr. Mahdi Sabouhi. Teachers play a very important role in the development and shaping of our personalities and of what we are, and yet somewhere down the line people tend to forget the importance and greatness of our teachers.

TABLE OF CONTENTS

LIST OF TABLES .... **VIII**

LIST OF FIGURES .... **IX**

ABSTRACT .... **XI**

DEFINITIONS **1**

CHAPTER 1. INTRODUCTION **4**

1.1 Overview of Transportation Network Modeling **5**

1.2 Problem Statement **6**

1.3 Overview of Research Approach **7**

1.4 Scope and High Level Assumptions **10**

1.5 Research Questions **11**

1.5.1 Predicting Macroscopic Measures of Performance based on Multiple Network Structural and Dynamical Attributes without Traffic Simulation (RQ1) *11*

1.5.2 Modeling Network Vulnerability due to Critical Disruptions (RQ2) *17*

1.5.3 Modeling Network Vulnerability due to Driver Distraction (RQ3) *19*

1.6 Dissertation Organization **20**

1.6.1 Predicting Macroscopic Measures of Performance (Chapter 2) *21*

1.6.2 Modeling Network Vulnerability due to Critical Disruptions (Chapter 3) *21*

1.6.3 Modeling Network Vulnerability due to Driver Distraction (Chapter 4) *22*

CHAPTER 2. PREDICTING MACROSCOPIC MEASURES OF PERFORMANCE **23**

2.1 Introduction **23**

2.2 Background Research **25**

2.2.1 Transportation Networks *25*

2.2.2 Structure of Transportation Networks *28*

2.2.3 Dynamics of Transportation Networks *32*

2.2.4 Performance of Transportation Networks *35*

2.3 Proposed Approach **37**

2.3.1 Framework *37*

2.4 Results **44**

2.5 Validation **52**

CHAPTER 3. MODELING NETWORK VULNERABILITY DUE TO CRITICAL DISRUPTIONS **57**

3.1 Introduction **57**

3.2 Background Research **58**

3.2.1 Background Research on Road Disruption Analysis *58*

3.2.2 Background Research on Network Science *61*

3.3 Proposed Approach **63**

3.3.1 Objective *63*

3.3.2 Framework *63*

3.4 Results **75**

3.5 Validation **79**

3.5.1 Simulation-based Validation *80*

3.5.2 Observation-based Validation *85*

CHAPTER 4. MODELING NETWORK VULNERABILITY DUE TO DRIVER DISTRACTION **92**

4.1 Introduction **92**

4.2 Background Research **93**

4.2.1 Background Research on Driver Distraction *93*

4.2.2 Background Research on Distracted Drivers and Surrounding Traffic *97*

4.2.3 Limitations of Existing Studies *98*

4.3 Proposed Approach **99**

4.3.1 Framework *100*

4.4 Validation **104**

4.4.1 Validation: case I *105*

4.4.2 Validation: case II *108*

4.4.3 Naturalistic Verification *111*

4.5 Results **112**

4.5.1 Local Case Studies *112*

4.5.2 Preliminary Results on Large-Scale Case Studies *114*

CHAPTER 5. CONCLUSIONS **119**

5.1 Research Contributions **119**

5.1.1 Performance Prediction Model *119*

5.1.2 Critical Area Identification *121*

5.1.3 Driver Distraction Analysis *122*

5.2 Practical Implications **123**

5.2.1 Performance Prediction Model *123*

5.2.2 Critical Area Identification *123*

5.2.3 Driver Distraction Analysis *124*

5.3 Limitations and Future Research Directions **124**

5.3.1 Performance Prediction Model *124*

5.3.2 Critical Area Identification *125*

5.3.3 Driver Distraction Analysis *125*

5.3.4 The Integrated Framework *125*

REFERENCES **127**

APPENDIX A: LIST OF PYTHON CODES **142**

Build a graph in i-Graph (RQ1 & RQ2) *142*

Calculate the structural attributes of a road network (RQ1) *142*

Calculate the criticality of links and cluster them (RQ2) *143*

Replicate Stavrinos's model for LOS A in VISSIM (RQ3) *148*

Replicate Stavrinos's model for LOS C in VISSIM (RQ3) *152*

Replicate Stavrinos's model for LOS E in VISSIM (RQ3) *153*

Replicate Salvucci's model for Standard Scenario in VISSIM (RQ3) *154*

Replicate Salvucci's model for Circular Scenario in VISSIM (RQ3) *158*

Developed interface for VISSIM – Real scenarios (RQ3) *159*

APPENDIX B: SAMPLES OF OBSERVATION-BASED VALIDATION **163**

VITA **166**

LIST OF TABLES

Table 2.1. The Network-wide Structural Attributes of Road Networks 28

Table 2.2. The Local Node-based Structural Attributes of Road Networks 30

Table 2.3. The Local Link-based Structural Attributes of Road Networks 31

Table 2.4. The Local Cell-based Structural Attributes of Road Networks 31

Table 2.5. Different Scenarios for Changing Links' Capacities 46

Table 2.6. Proposed Structural Attributes 47

Table 2.7. Comparison of Simulated and Predicted Results for Real Scenarios 54

Table 3.1. Computational Complexity (i.e., running time) of three Different Approaches 70

Table 3.2. The results of the two-sample Kolmogorov-Smirnov and t-tests 84

Table 4.1. KS Test Results for Speed Deviation 108

LIST OF FIGURES

Figure 1.1. Problem Definition......6

Figure 1.2. Overview of Three Research Questions......9

Figure 1.3. Schematic View of the Proposed Approach in RQ1......11

Figure 1.4. The Results of Three Different Simulations for Epidemic Spreading......14

Figure 1.5. The Results of Three Different Simulations for Congestion Propagation......16

Figure 1.6. Schematic View of RQ2......17

Figure 1.7. Impacts of Disruptions on Guam Road Network......18

Figure 1.8. Schematic View of RQ3......19

Figure 1.9. Speed Fluctuations for Different Distraction Scenarios......20

Figure 2.1. The Observed Distribution of Daily Traffic Presented in [29]......34

Figure 2.2. An Example of MFD presented in [45]......36

Figure 2.3. Flowchart of the Proposed Approach for Research Question 1......38

Figure 2.4. Process of Performance Prediction......41

Figure 2.5. The Study Area (The Greater Philadelphia Region)......45

Figure 2.6. Relationships between Canonical Variates......50

Figure 2.7. Simulated Results for Real Scenarios (i.e., test) Plotted on the Captured Canonical Variates ...55

Figure 3.1. Overview of the Proposed Framework......64

Figure 3.2. Change in Capacities of a Small Weighted Graph......66

Figure 3.3. Flowchart for the Exhaustive Approach......67

Figure 3.4. Flowcharts for the Two Proposed Approaches......68

Figure 3.5. Three Methods for Finding Critical Link of Guam Island Network......69

Figure 3.6. Defining Criticality of Links......71

Figure 3.7. The Results for 60% Severity of Disruption at Different Times of a Day......76

Figure 3.8. The Results for Different Severity of Disruptions during PM Peak Period......78

Figure 3.9. Three Selected Clusters: High-Critical (Red), Medium-Critical (Orange), and Low-Critical (Green)......81

Figure 3.10. Simulated Results for Validation 81

Figure 3.11. The Aggregated Buffer Region around the Three Clusters 83

Figure 3.12. Changes of V/C after Disruption Compared to before Disruption (Red: Significant Increase, Orange: Medium Increase, Green: Small Increase, Gray: No Increase) 85

Figure 3.13. Locations of Observed Real Incidents 86

Figure 3.14. Sample Observations within a High-Critical Cluster during MD Period 88

Figure 3.15. Sample Camera Records within a High-Critical Cluster during MD Period 89

Figure 3.16. Sample Observations within a Medium-Critical Cluster during PM Period 90

Figure 3.17. Sample Observations within a Low-Critical Cluster during MD Period 91

Figure 4.1. Levels of Abstraction in the Proposed Framework 99

Figure 4.2. Flowchart of the Proposed Approach 100

Figure 4.3. Sample 30-second-long Snapshots of Three Distraction Time Profiles 102

Figure 4.4. Snapshots of Three Different Models 106

Figure 4.5. Comparison of the Results for Speed Deviation 107

Figure 4.6. Configurations of Two Experiments Proposed by Salvucci (2013) 108

Figure 4.7. Standard Car-Following Results (Changes Compared to No Distraction). There Are Ten Graph Lines, each of Which Represents a Group of Model Drivers Simulated Across Three Conditions, Namely with 0, 1, or 3 Distracted Drivers. 109

Figure 4.8. Circular Car-Following Results (Changes Compared to No Distraction). There Are Ten Graph Lines, each of Which Represents a Group of Model Drivers Simulated Across Three Conditions, Namely with 0, 1, or 3 Distracted Drivers. 110

Figure 4.9. Speed Deviation for Each Individual Driver 111

Figure 4.10. Snapshots of Two Intersections in: Germany (Left, (a) and (c)) and China (Right, (b) and (d)) 113

Figure 4.11. Results for Germany (left) and China (right) Including Change in: Speed c.o.v. (<u>SV</u>), Headway Distance c.o.v. (<u>HV</u>), Speed Average (<u>SA</u>), Headway Distance Average (<u>HA</u>), and Lane Change (<u>LC</u>) , for Different Levels of Distraction (5%, 10%, and 30% of drivers). Positive Values Show Increase and Negative Values Show Decrease in the Measure. 114

Figure 4.12. A Real-World Large-Scale Case Study from Philadelphia Region 116

Figure 4.13. Difference between Distracted and Not-Distracted Drivers (The Values on Vertical Axis Are Relative to 0% Distracted Drivers) 116

ABSTRACT

COMPUTATIONAL FRAMEWORK FOR MODELING INFRASTRUCTURE NETWORK PERFORMANCE AND VULNERABILITY

Seyed Hossein Hosseini Nourzad

Networked infrastructures serve as essential backbones of our society. Examples of such critical infrastructures whose destruction severely impacts the defense or economic security of our society include transportation, telecommunications, power grids, and water supply networks. Among them, road transportation networks have a principal role in people’s everyday lives since they facilitate physical connectivity. The performance of a road transportation network is governed by the three principal components: (a) structure, (b) dynamics, and (c) external causes. The structure defines the topology of a network including links and nodes. The dynamics (i.e., traffic flow) defines what processes are happening on the network. The external causes (e.g., disasters and driver distraction) are the phenomena that impact either structure or dynamics. These principal components do tend to influence each other. For example, the collapse of a bridge (i.e., external cause) could render certain nodes and links (i.e., structure) ineffective thereby affecting traffic flow (i.e., dynamics). A distracted driver (i.e., external cause) on a road can also cause accidents that can negatively impact traffic flow. Thus, to model the performance and vulnerability of a network, it is necessary to consider such interactions among these principal components. The main objective of this research is to formalize and develop a computational framework that can: (a) predict the macroscopic performance of a transportation network based on its multiple structural and dynamical attributes (Chapter 2), (b) analyze its vulnerability as a result of man-made/natural disruption that minimizes

network connectivity (Chapter 3), and (c) evaluate network vulnerability due to driver distraction (Chapter 4). An integrated framework to address these challenges—which have largely been investigated as separate research topics, such as distracted driving, infrastructure vulnerability assessment and traffic demand modeling—needs to simultaneously consider all three principal components (i.e., structure, dynamics, and external causes) of a network. In this research, the integrated framework is built upon recent developments (theories and methods) in interdisciplinary domains, such as network science, cognitive science and transportation engineering. This is the novelty of the proposed framework compared to existing approaches. Finally, the framework were validated using real-world data, existing studies and traffic simulated results.

DEFINITIONS

Serviceability: The serviceability of a road network describes the possibility to use the road network during a given time period [1].

Performance: The performance of a road network refers to level of service of the network. In other words, it is a measure of network serviceability [1, 2].

Vulnerability: The vulnerability of a road network is the susceptibility to events that may result in considerable reductions in road network serviceability. The events may be voluntarily or involuntarily, caused by man or nature. [1]

Structure: The structure defines the topology and geometry of a network including links and nodes [3].

Dynamics: The dynamics defines what processes are happening on the network [4]. In the case of road networks, dynamics refers to the traffic flow dynamics, such as congestion.

External causes: The external causes are phenomena that impact either structure or dynamics. In the case of road networks, examples include disruption and driver distraction [5, 6].

Area-covering disruption: The area-covering disruptions impact multiple links within an area of the network [6]. The examples include flood, and heavy snow.

Driver distraction: The driver distractions are the issues that impact the traffic dynamics by changing drivers' behavior [7]. Examples include talking on cellphones and text-messaging.

Structural attribute: The structural attributes are the measures that quantify characteristics related to the structure of the network. Examples of structural attributes include betweenness and degree distribution [3].

Dynamical attribute: The dynamical attributes are the measures that quantify characteristics related to the dynamics of the network [3]. Examples of dynamical attributes include traffic volume between origin-destination pairs [8].

Network attributes: The network attributes are the set of all structural and dynamical attributes that measure the network characteristics.

Measures of effectiveness (MOE): The MOEs are the measures that quantify the network performance [2, 9, 10]. In the case of road networks, macroscopic MOEs include average speed, delay, and traffic volume of the road segments.

Network science: The area of science concerning the study of networks is called network science [11].

Civil infrastructures: Many of civil infrastructures consist of a set of objects (called nodes) that are connected together with links. Therefore, we can refer to these infrastructures as networks. For example, in road transportation networks, the nodes and links represent the intersections and road segments.

Weighted and un-weighted networks: A network can be either weighted or un-weighted [12]. A weighted network is a network where the links among nodes have weights assigned to them. In contrast, in an un-weighted network, the links does not have any weights assigned to them.

Adjacency matrix: An adjacency matrix is used to represent the structure of a network. The adjacency matrix captures which nodes of a network are adjacent to which other nodes. Adjacency matrix of a weighted network G (with N nodes) is an N×N matrix where the entry a_{ij} is the weight of the link from node i to node j. For an un-weighted network, the adjacency matrix is a matrix of ones and zeroes where a one indicates the presence of a connection and a zero indicates the absence of a connection [11].

Centrality: In network science, centrality of a node quantifies its relative importance within a network [13]. There are several measures that quantify network centrality. One of the well-known measures is the *degree* of a node which is the number of connections it has with other nodes [11]. In addition, in a weighted network, the *weighted degree* (aka strength) of a node is the total weights of its adjacent links [12]. Although node's degree is a measure of centrality, it just looks at individual nodes at a time. So, it is a localized measure. The other measure of centrality is the *betweenness* centrality of a node which is equal to the number of shortest paths from all nodes to all others that pass through that node. The betweenness centrality is also defined for each individual node, and does not represent the connectivity of the entire network [11].

Connectivity: A network is connected if there are paths connecting every pair of nodes. There are several measures that quantify connectivity at a network-level [14-16]. For example, the *gamma index* is the ratio of the number of links (of a network) to maximal number of links (of a connected network with the same number of nodes) [16]. The gamma ranges between 0 and 1, where a value of 1 indicates a connected network. However, Prakash et al. (2013) showed that the best single measure of connectivity is the *largest eigenvalue of the adjacency matrix* of a network, which represents the connectivity of a network as a whole rather than individual links [14, 15]. For network-wide analysis, it is better to use the largest eigenvalue instead of the localized measures, such as degree and betweenness.

CHAPTER 1. INTRODUCTION

Networked infrastructures serve as important backbones of our society [17]. In President's Commission on Critical Infrastructure Protection (PCCIP), an infrastructure was defined as: "the framework of interdependent networks and systems comprising of identifiable industries, institutions (including people and procedures), and distribution capabilities that provide a reliable flow of products and services essential to the defense and economic security of the United States, the smooth functioning of government at all levels, and society as a whole" [18]. In addition, the PCCIP highlighted the essential role of certain infrastructures that their destruction impacts the defense or economic security of the United States. Examples of such critical infrastructures include transportation, telecommunications, power grids, gas and oil storage/transportation, banking and finance, water supply networks, and emergency services (e.g., medical, police, fire, and rescue). Among them, transportation networks have a principal role in people's everyday lives (e.g., citizens' personal, communication, and economic activities) since they facilitate physical connectivity. Researchers have studied the performance of different types of transportation networks, such as road, railway, and air traveling networks [19-29].

Any disruption in transportation networks (e.g., due to accidents or natural disasters) negatively influences network vulnerability. The changes may originate from either external sources (e.g., heavy snow, storm, or earthquake) or internal issues (e.g., accidents) [6]. Therefore, it is important to analyze the impacts of disruptions and mitigate the risks of disruptions. In addition, it is also important to analyze the impacts of driver distraction that can worsen network safety and vulnerability [30]. Hence, this research focuses on modeling performance and vulnerability of road networks, as an example of transportation networks, under normal, disrupted and distracted conditions.

1.1 Overview of Transportation Network Modeling

The performance and vulnerability of transportation networks have been studied from two different viewpoints: (a) complex networks (i.e., networks that have many elements with nonlinear interactions) [3, 4, 11, 12, 16, 31-37], and (b) transportation engineering [2, 38-48]. Furthermore, the impacts of interrelation between structure and dynamics on the network performance was highlighted in many research studies [4, 49-51]. Examples include the study of onset of traffic congestion [4, 51], propagation of congestion [50], and cascading failure [49].

In addition, researchers studied vulnerability of infrastructures to large-scale collapse in modern societies [6, 31, 48, 49, 52-56]. In particular, any disruption in a road network may degrade network performance [56]. For instance, the researchers investigated the impacts of single-link blockages on total delay over the entire road network [31]. Also, the researchers investigated the impacts of multiple-links blockage on the network performance [6, 56].

Finally, driver distraction is the major cause of vehicle crashes in the United States [57]. During the last few years, different methods have been developed to analyze driver distraction due to performing secondary tasks. Examples include physical driving simulation, naturalistic experiment, statistical analysis, and computational modeling [7, 58-76]. Among these methods, physical driving simulation is the common method to analyze the impacts of driver distraction [65]. However, this method is time-consuming and expensive [30]. Hence, the computational modeling (e.g., cognitive models) has been proposed to evaluate distraction without physical simulators [30, 77-79]. Recently, the researchers investigated simple scenarios of multiple-drivers distraction using computational modeling [61].

1.2 Problem Statement

The performance of a transportation network is governed by the three principal components: (a) structure, (b) dynamics, and (c) external causes. The structure defines the topology of a network including links and nodes. The dynamics defines what processes are happening on the network. The external causes (e.g., driver distraction) are the phenomena that impact either structure or dynamics. These principal components do tend to influence each other. For example, the collapse of a bridge (i.e., external cause) could render certain nodes and links (i.e., structure) ineffective thereby affecting traffic flow (i.e., dynamics). A distracted driver (i.e., external cause) on a road can also cause accidents that can impact traffic flow. Thus, to model the performance and vulnerability of a network, it is necessary to consider such interactions among these principal components (Figure 1.1).

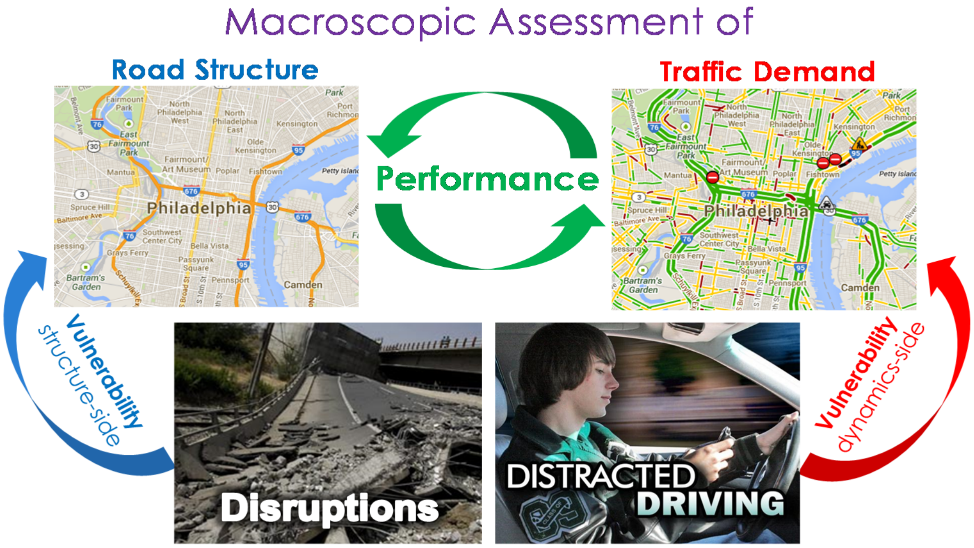


Figure 1.1. Problem Definition

The main objective of this research is to formalize and develop a computational framework that can: (a) predict the macroscopic performance of a transportation network based on its multiple structural and dynamical attributes, (b) analyze its vulnerability as a result of man-made/natural disruption that minimizes network connectivity, and (c) evaluate network vulnerability in response to driver distraction. An integrated framework to address these challenges—which have largely been investigated as separate research topics, such as distracted driving, infrastructure vulnerability assessment and traffic demand modeling—needs to consider all three principal components (i.e., structure, dynamics, and external causes) of a transportation network (Figure 1.1).

1.3 Overview of Research Approach

This research attempts to address the three major limitations associated with current approaches of evaluating performance of road transportation networks (Figure 1.2). First, the prevalent approaches based on **M**acroscopic **F**undamental **D**iagrams (MFD) and phase transition models do not model multiple network **m**easures **o**f **e**ffectiveness (MOE) based on various structural and dynamical attributes [19, 45, 46]. For example, the MFDs capture the relationship between two network MOEs, such as density and flow. Phase transition models evaluate the impacts of a single network attribute (e.g., largest eigenvalue of the adjacency matrix) on a single measure of performance (e.g., congestion propagation). However, in addition to density and flow, other MOEs, such as average speed and delay of road networks, are also necessary to quantify the performance of a network. Second, the existing approaches to analyze the vulnerability of a transportation network focus mainly on the failure of either a single node/link or multiple links or simple scenarios of disruptions [6, 56, 80]. What is missing is the consideration of the impact of disruptions (e.g., heavy snow, flood or earthquake) that can interrupt a number of large areas on a network. In this research, these disruptions are referred to as area-covering disruptions. Third, recent approaches to assess the impacts of driver distraction on road have focused on simple scenarios in which vehicles follow each other on a straight single-lane highway [60, 61, 81]. However, these approaches do not evaluate the impacts of multiple distracted drivers in terms of overall network performance and safety. For instance, how will the traffic flow be impacted if 5% of drivers on road are involved in text-messaging at a given time?

To address these challenges, the framework is built upon recent developments (theories and methods) in interdisciplinary domains, such as network science, cognitive science and transportation engineering. This is the novelty of the proposed framework compared to existing frameworks and approaches. Network science seeks to understand the underlying principles that govern the structure, dynamics and co-evolution of complex networks (e.g., social network, gene regulatory network and computer network). Thus, network science theories could help us to better understand the structure and dynamics of a transportation network. Similarly, cognitive science, which focuses on how information is represented, processed and transformed within (human or animals) nervous systems, could help to investigate driver distraction. The proposed framework would enable expert modelers to: (a) evaluate new design scenarios of transportation network and their potential macroscopic impact on performance, (b) analyze the vulnerability of a transportation network under various disruptions, and (c) understand the impact of distractive devices on network performance and safety.

Figure 1.2 depicts the relationships among the three principal components of a road (transportation) network: (1) **Network Structure**, (2) **Traffic Flow Dynamics** and (3) **External Causes**.

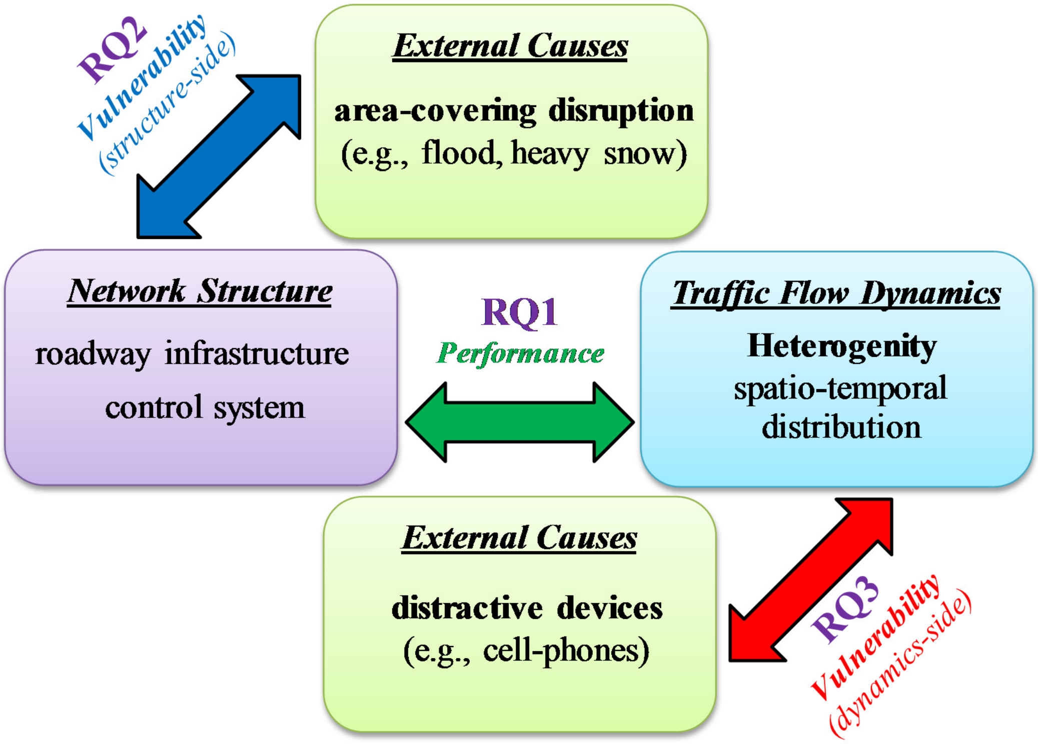


Figure 1.2. Overview of Three Research Questions

Figure 1.2 depicts the three research questions which investigate and formalize: (a) the network macroscopic performance based on multiple network structural and dynamical attributes without traffic simulation (RQ1), (b) the network vulnerability due to critical area-covering disruptions (RQ2), and (c) the network vulnerability due to driver distraction (RQ3). Section 1.5 describes in detail these three research questions.

1.4 Scope and High Level Assumptions

This section lists the scope and high-level assumptions of my PhD research:

- I limited the scope of this research to (road) transportation networks. Other infrastructure networks (e.g., water supply networks, power grids) and different transportation networks (e.g., air traveling and railway) are beyond the scope of this research.
- The performance prediction model in RQ1 returns macroscopic measures of performance (i.e., MOEs). While it does not replace the simulation models, it is a useful model for pre-screening process of numerous design alternatives. The pre-screening process is performed to select a few number of design alternatives out of an initial long list of alternatives. The selected alternatives will then be analyzed using simulation models, leading to a significant saving in time and computational resources.
- The proposed research employed the road network of the Greater Philadelphia region. I assumed that the acquired road network is a representative real-world network.
- While a driver can get distracted due to many reasons (e.g., texting, dialing, talking, eating, etc.), this research focused on driver distraction due to cell-phone dialing, conversation, and text messaging. I measured the effects of distraction via changes in average and deviation of speed and headway distance. However, the possibility of distraction-related crashes and its impacts on traffic conditions are beyond the scope of the current research.

1.5 Research Questions

This section discussed the three major research questions associated with computational modeling of performance and vulnerability of (road) transportation networks.

1.5.1 Predicting Macroscopic Measures of Performance based on Multiple Network Structural and Dynamical Attributes without Traffic Simulation (RQ1)

As shown in Figure 1.2, the RQ1 focuses on modeling the network performance based on the network structure and traffic flow dynamics. The objective is to formalize and develop an approach that can predict the macroscopic (also referred to as network-wide) measures of performance for new networks without undertaking traffic simulations. This is especially useful for pre-screening of a long list of numerous design alternatives. While the proposed model does not replace simulation models, it helps us to save a significant amount of time and computational resources during the pre-screening process. As shown in Figure 1.3, the inputs of the approach are multiple structural and dynamical attributes of a new network, and the outputs are multiple network-wide measures of performance (i.e., MOEs).

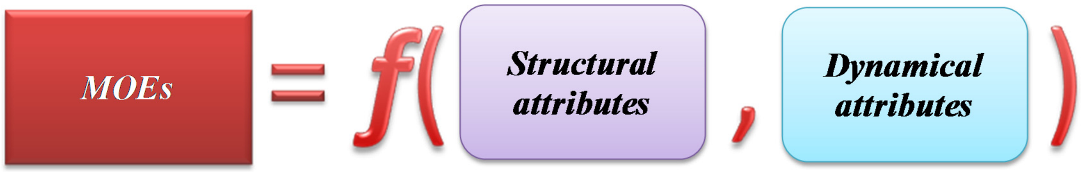


Figure 1.3. Schematic View of the Proposed Approach in RQ1

Motivation of RQ1

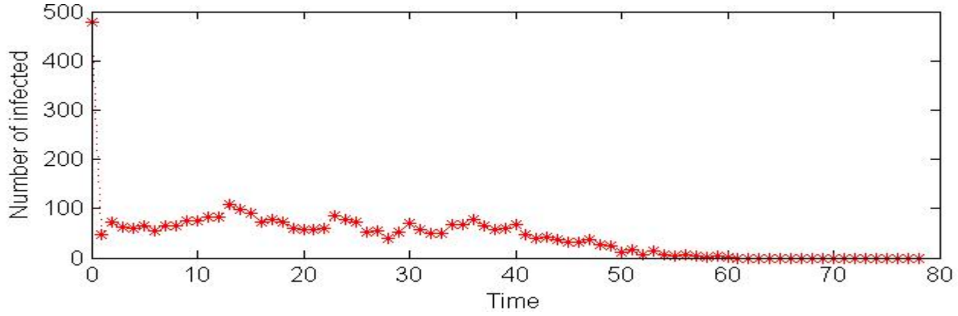
In our previous study, we presented two simple case studies to investigate the relationship between network performance and network attributes. The first case was epidemic spreading in social networks (i.e., network of individuals linked through some interactions), in which individuals and their relationships are considered as network nodes and links [50, 82, 83]. The second case was congestion propagation in road networks, in which intersections and road segments are considered as network nodes and links. I chose these two case studies since there are some similarities between them: the epidemic spreads among individuals through their relationships and the congestion propagates among the intersections through the roads. I investigated the impacts of network structure (i.e., the largest eigenvalue of adjacency matrix) on epidemic spreading and congestion propagation. Therefore, these two simple case studies can illustrate the impacts of structure and dynamics on the performance of different networks.

The epidemic spreading is a critical problem in social networks. The aim of epidemic spreading modeling is to reproduce actual dynamics of epidemic and to understand effects of network topology on epidemic spreading [3, 82]. At each time, each node could be in only one of the two states: (a) **S**usceptible (i.e., those who can catch the infection), or (b) **I**nfected (i.e., those who have caught the infection and can transfer the infection). In this research, I focused on the **S**usceptible–**I**nfected–**S**usceptible models, which are related to the diseases that do not confer immunity to their survivors (e.g., tuberculosis and gonorrhea). In the SIS models, the probability of epidemic spreading from an infected node to a susceptible node in its neighborhood is referred to as the spreading rate (equal to δ). In addition, once a node gets infected, the probability of recovering from the disease and returning to the susceptible state is referred to as the recovery rate (equal to μ) [82].

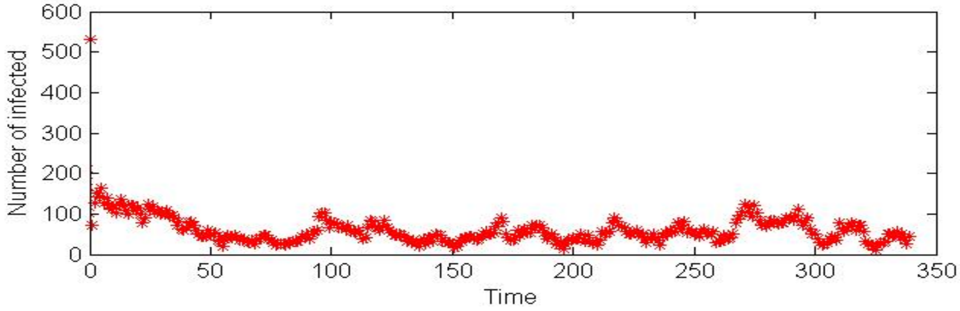
In the SIS models, epidemic starts from one or a few nodes within the network. The epidemic will then spread from the infected nodes to their neighboring nodes (with the rate δ). At the same time, some of the infected nodes will recover (with the rate μ). At each time step, next state of each node is a function of its current state and of states of its neighbors on the network. Therefore, the network dynamics depends on both the spreading and recovery rates and specifically on their ratio (i.e., $\sigma = \delta / \mu$). In fact, there is an epidemic transition (i.e., a critical value of the ratio, $\sigma = \sigma_c$). In the SIS models, for $\sigma > \sigma_c$ the epidemic persists in the network, while for $\sigma < \sigma_c$ it does not. Wang et al. (2003) presented a generic solution that the epidemic threshold of any given network is $\sigma_c = 1/\lambda_{1,A}$ in which $\lambda_{1,A}$ is the largest eigenvalue of its adjacency matrix [83].

I ran computer simulations to verify the mathematical solution of the proposed epidemic threshold. To do so, I generated a network using the tools presented in [84]. The number of nodes was equal to 10000. The recovery rate was constant (μ=0.5) and the spreading rate changed from δ=0.00525 to δ=0.2. The largest eigenvalue of the adjacency matrix ($\lambda_{1,A}$) was equal to 90.90 leading to the epidemic threshold $\sigma_c = 1/\lambda_{1,A} = 0.011$.

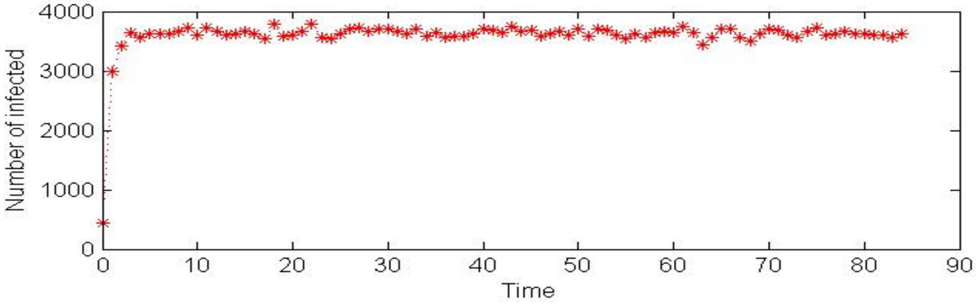
Figure 1.4 depicts the number of infected nodes in each time step for three simulations. Figure 1.4 (a) shows when σ was less than epidemic threshold, the epidemic died after 60 time step. Figure 1.4 (b) shows when σ was equal to the epidemic threshold, there was a fluctuation in the number of infected nodes, i.e., the network was in the transition phase. Figure 1.4 (c) shows when σ was more than the epidemic threshold, the number of infected nodes reached a nonzero number, i.e., the epidemic persisted in the network.



(a) δ=0.00525, σ = 0.0105



(b) δ=0.0055, σ = 0.011



(c) δ=0.1, σ = 0.2

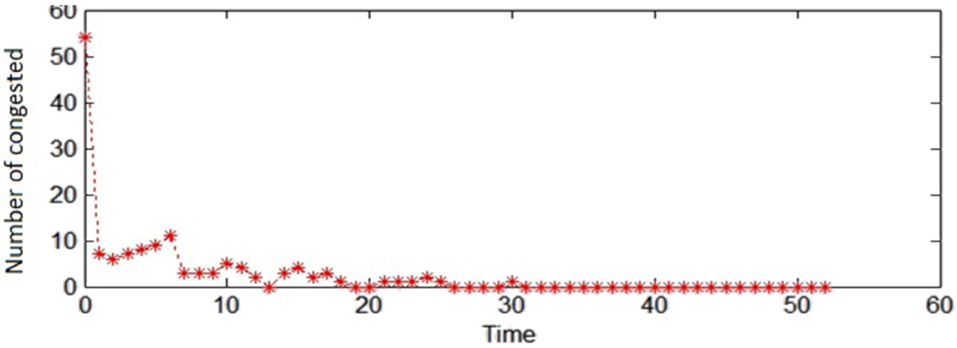
Figure 1.4. The Results of Three Different Simulations for Epidemic Spreading

Therefore, I observed that in social networks there is an interrelation between network structure (i.e., largest eigenvalue of the adjacency matrix) and dynamics (i.e., the epidemic spreading). The knowledge of such interrelation is necessary for epidemiologist to assess spreading of different epidemics and to mitigate the risks of the epidemics.

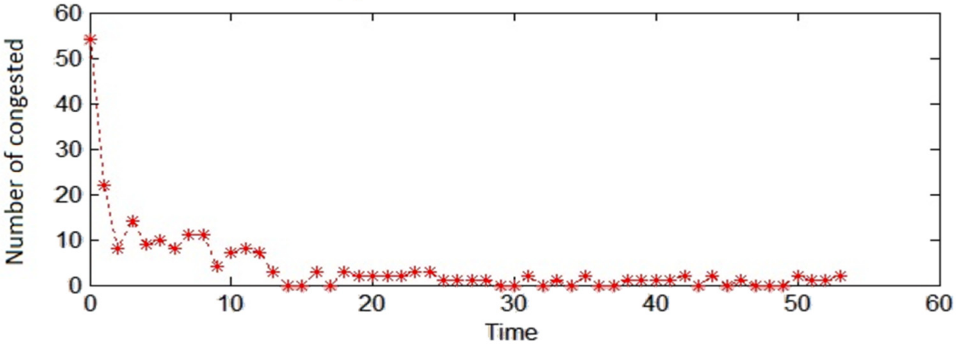
In road networks, I conjectured that the structure (e.g., largest eigenvalue of the adjacency matrix) influences the dynamics (e.g., congestion propagation). To test this conjecture, I simulated congestion propagation in road networks. For a given road network, when a node is congested, the congestion may spillback to upstream nodes with the average propagation rate (δ). At the same time, some of the congested nodes may become decongested with the average relief rate (μ). In other words, each node can be in only one of the two states: (a) uncongested, or (b) congested. Therefore, I could simulate congestion propagation using a SIS model. The quantity of interest is the critical threshold associated to the phase transition from free-flow to congested phase [50].

I simulated the congestion propagation on the road network of the Island of Guam, which had 539 nodes and 1183 links. The largest eigenvalue of the adjacency matrix ($\lambda_{1,A}$) was 2.819 and therefore, the transition threshold (σ_c) was 0.355. The simulation period was 4 hours (i.e., 48 five-minute time steps).

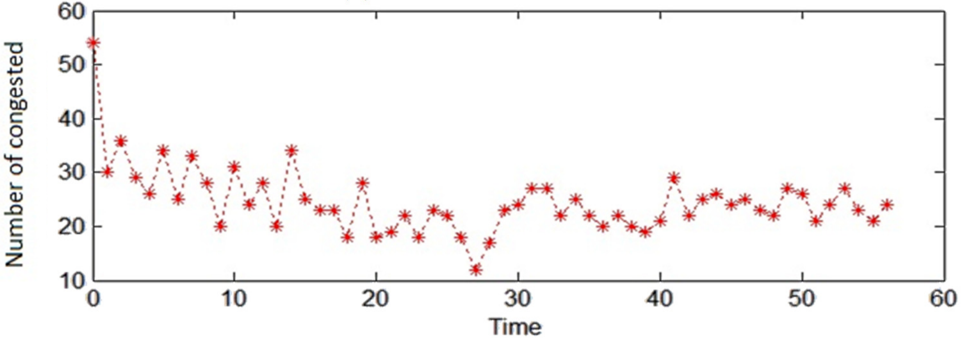
Figure 1.5 depicts the number of nodes that were congested at each time step for three different simulations. Figure 1.5 (a) shows when $\sigma < \sigma_c$, the network became decongested after 150 minutes. Figure 1.5 (b) shows when $\sigma \approx \sigma_c$, there was a fluctuation in the number of congested nodes, i.e., the network was in the transition phase. Figure 1.5 (c) shows when $\sigma > \sigma_c$, the number of congested nodes reached a nonzero number, i.e., the congestion persisted in the network.



(a) δ=0.044, σ = 0.177



(b) δ=0.089, σ = 0.355



(c) δ=0.177, σ = 0.709

Figure 1.5. The Results of Three Different Simulations for Congestion Propagation

Therefore, the results show that: (i) There was a relationship between largest eigenvalue, $\lambda_{1,A}$, and congestion dynamics, σ; and (ii) more importantly, this relationship impacted the congestion propagation. These simulations only considered one structural attribute and one dynamical attribute. I hypothesize that understanding the relationship among multiple structural (e.g., betweenness and degree) and dynamical (e.g., speed, delay) attributes would enable us to perform a macroscopic analysis of performance as few attributes often fail to capture all the characteristics of a network. In general, different domain–specific network scientists in biology and social science have leveraged a number of attributes to capture the characteristics of their networks, such as gene regulatory network and social graphs [2, 10]. Different network measures, such as centrality, assortativity, modularity, and betweenness, are used to find critical nodes based on its connectivity with respect to other nodes, and to find clusters of similar nodes, etc. Details about different network measures can be found in [11].

1.5.2 Modeling Network Vulnerability due to Critical Disruptions (RQ2)

As shown in Figure 1.2 shows, any external cause (e.g., disruption) that changes the structure will also impact the performance of a network. Different types of disruptions may impact a network, ranging from the events (e.g., car accident or bridge collapse) that impact a single link to the events (e.g., heavy snow or storm) that impact a large area which includes several links. As shown in Figure 1.6, to analyze network vulnerability, the RQ2 investigates large-scale impacts of disruptions thereby focusing on area-covering disruptions.

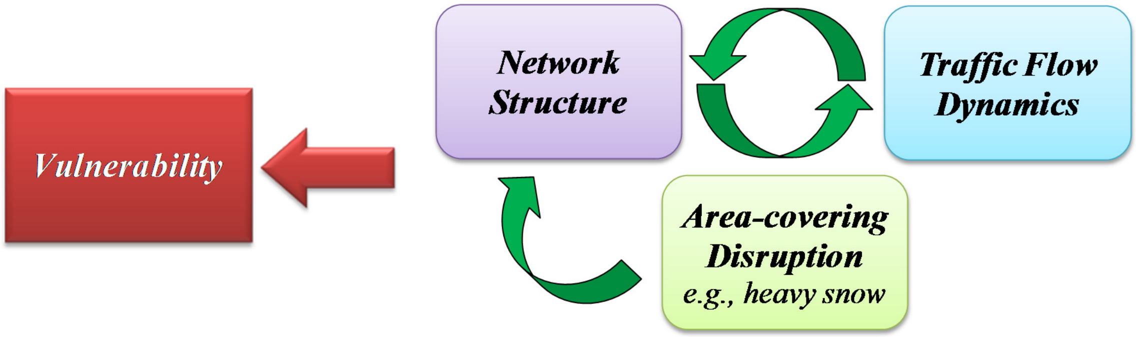


Figure 1.6. Schematic View of RQ2

Motivation of RQ2

In our preliminary study, I investigated how the capacity reduction of different nodes (in a road network) due to disruption impacts its connectivity (Figure 1.7). The network connectivity was represented by the eigenvalue of its adjacency matrix. In Figure 1.7, the *x-axis* represents the proportion of disrupted nodes, and the *y-axis* represents the drop in the largest eigenvalue. For the analysis, we selected 15% nodes of the network (i.e., Guam network) using different selection methods that include random and targeted selections. The targeted selection methods are based on nodes with higher degree, betweenness, weighted betweenness and the largest eigenvalue. I observed that targeted method based on the largest eigenvalue was the most effective method to minimize network connectivity. This observation supports the research finding of Prakash et al. (2013) where they concluded that the Exhaustive method was effective in minimizing network connectivity.

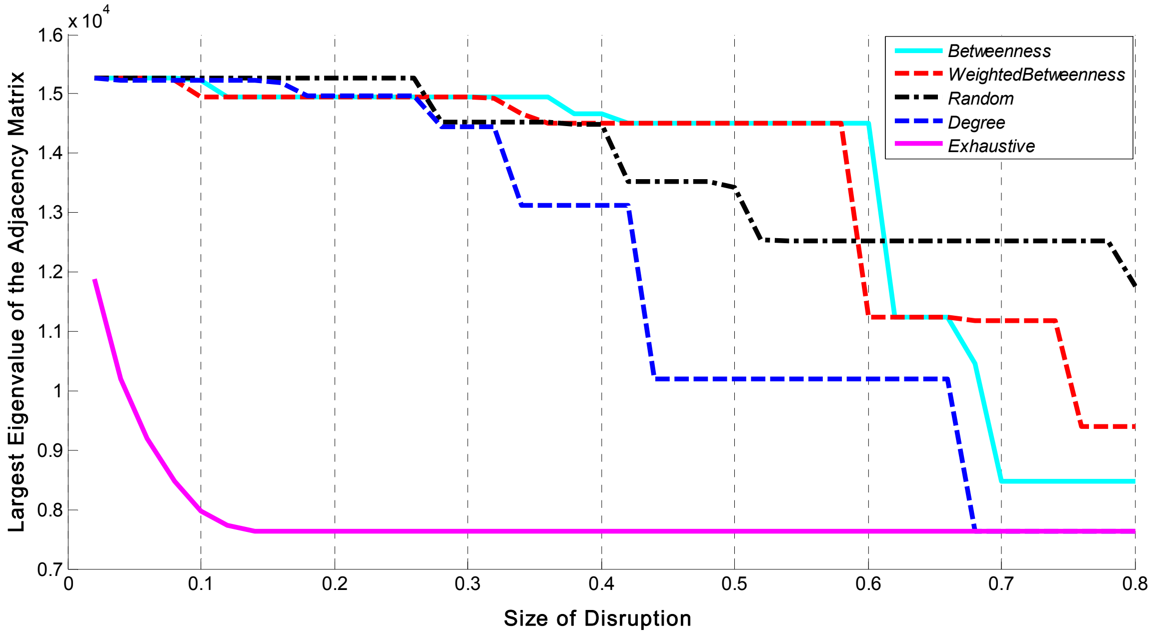


Figure 1.7. Impacts of Disruptions on Guam Road Network

1.5.3 Modeling Network Vulnerability due to Driver Distraction (RQ3)

Finally, any external cause (e.g., driver distraction) that changes traffic flow dynamics will also impact the performance of a network. For example, distracted drivers negatively impact the traffic conditions and safety. As shown in Figure 1.8, to analyze the network vulnerability, the RQ3 considers network-wide impacts of distraction on traffic flow dynamics.

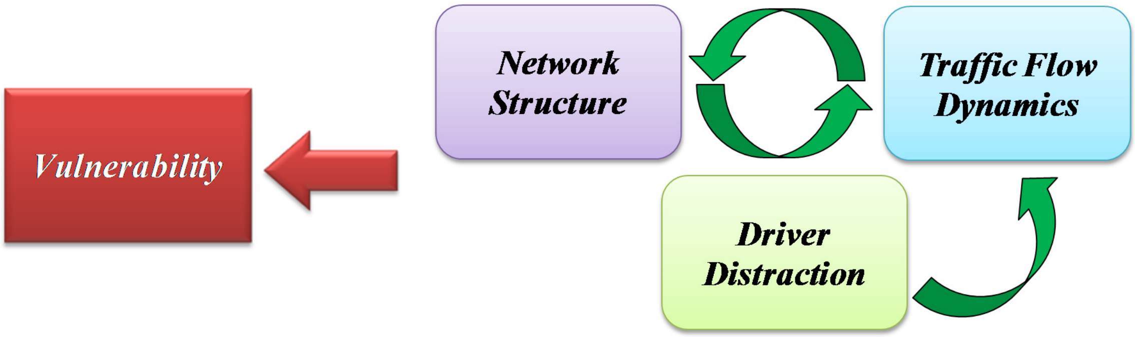


Figure 1.8. Schematic View of RQ3

Motivation of RQ3

On July 17, 2009, 21-year-old Casey Feldman of Philadelphia was struck and killed by a distracted driver as she crossed the street in Ocean City, New Jersey. The distracted driver had taken his eyes off the road for just a few seconds. According to NHTSA'S National Center for Statistics and Analysis, Casey was just one of the 15,254 people killed in distraction-affected fatal crashes across the United States between 2009 and 2012 [85-87]. Moreover, on average, around 20 percent of distracted drivers were distracted by the use of cell phones [87]. At any given daylight time across the United Stated, approximately 660,000 drivers are using cell-phones or other electronic devices while driving, a number that has held steady since 2010 [88]. All these facts and statistics highlight the importance of large-scale distraction simulation to quantitatively assess the impacts of distraction on traffic condition and safety.

In our previous research, we started formalizing a framework to simulate large-scale impacts of driver distraction. We replicated an existing experiment, in which sixteen drivers drove in single lane straight and curved roadway [61]. We tested the three different experiments, where 0, 1, and 3 distracted drivers were involved. The preliminary investigation demonstrated that the proposed framework could account for statistically significant changes in speed fluctuation in the presence of a significant number of distracted drivers (Figure 1.9).

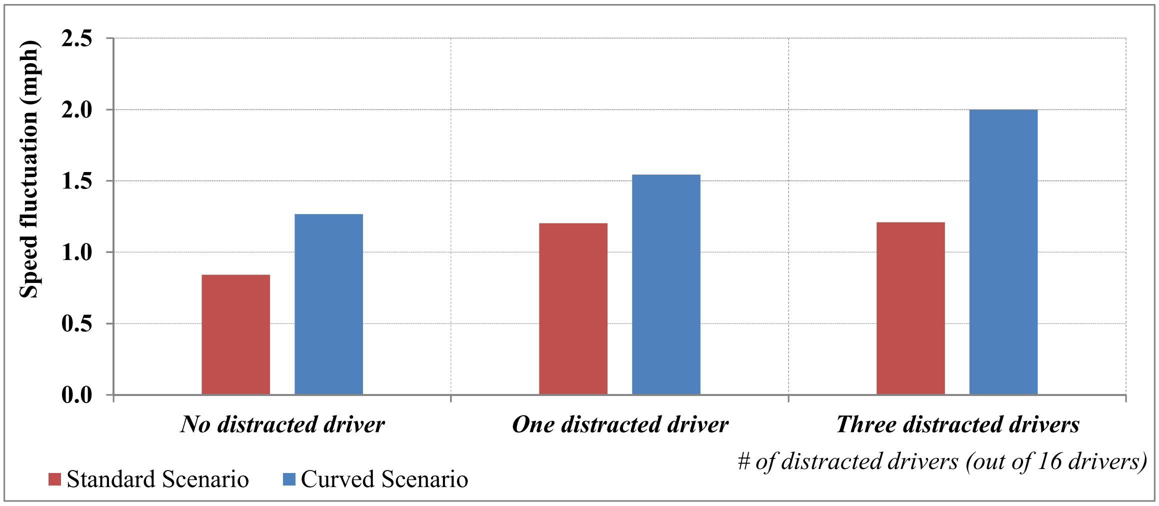


Figure 1.9. Speed Fluctuations for Different Distraction Scenarios

1.6 Dissertation Organization

This dissertation consists of five chapters including the Introduction and Conclusions. The Introduction chapter provides the overview of the research problem and describes the research questions. The Conclusions chapter discusses the research contributions, practical implications, limitations and future research directions. The remaining chapters focus on the three specific research contributions (described next).

1.6.1 Predicting Macroscopic Measures of Performance (Chapter 2)

Chapter 2 describes an approach that addresses challenges associated with capturing the relationship among (i) network performance, (ii) road structure, and (iii) traffic demand. A computational model based on multivariate statistical analysis is developed. The model is proposed for pre-screening of numerous design alternatives. The model requires network structural and dynamical attributes as inputs and outputs the macroscopic measures of performance (e.g., average and standard deviation of speed and volume). I developed the model using a database including the Greater Philadelphia road network. The computational results were validated by comparing with the simulated results of a calibrated traffic model for the real network of the Greater Philadelphia. The model led to a significant saving of time and computational resources, compared to simulation models. In the future, the proposed framework can be used on a diverse database of road networks to develop a better performance prediction model.

1.6.2 Modeling Network Vulnerability due to Critical Disruptions (Chapter 3)

Chapter 3 explains an approach that identifies the vulnerability of different areas of the network due to disruptions. In the first phase of the proposed framework, the criticality of each individual link is evaluated by considering the large-scale impacts of link's disruption. In the second phase, the critical links are clustered, leading to partitioning the network into different areas with different levels of criticality. The main contributions of the research described in this chapter are: (a) identifying a new network-wide criticality criterion based on network-science theories, (b) taking both road structure and traffic demand into the consideration for identifying the network-wide criticality, and (c) clustering the network into critical and non-critical areas to have a more realistic analysis of natural disruptions. Also, in contrast to the existing approaches, the proposed framework does not require traffic simulations to quantify the criticalities. I validated the proposed framework on the Greater Philadelphia network for four times of a day. The computational results were validated by: (1) comparing with the simulated impacts of disruptions for three different clusters, and (2) observing the real-time traffic conditions online for several real incidents across the network.

1.6.3 Modeling Network Vulnerability due to Driver Distraction (Chapter 4)

Chapter 4 presents an integrated approach that addresses the large-scale simulation of driver distraction. The major contribution of this approach is the development and validation of a framework which addresses the existing gap of driver distraction by integrating a computational driver distraction model (i.e., the Distract-R) and a microscopic traffic simulation model (i.e., the VISSIM software). The research provides a tool to answer many questions that were not easy to answer before. Examples include: (i) the traffic conditions on a network when a significant number of drivers are distracted by different types (e.g., text messaging, dialing, and conversation), (ii) the areas of the network which are more vulnerable due to the impacts of driver distraction, and (iii) the impacts of changing (adding/removing) certain features of a device (i.e., adding a button to a cellphone) on traffic conditions. I validated the framework by replicating three existing case studies. Then, I employed the developed approach for a few larger-scale case studies, extracted from the Greater Philadelphia real network. Chapter 4 describes the framework and all these case studies.

CHAPTER 2. PREDICTING MACROSCOPIC MEASURES OF PERFORMANCE

2.1 Introduction

Many phenomena related to network performance depend on both network structure and dynamics, and specifically on the interrelation between structure and dynamics [4, 51, 89]. For example, congestion propagation on road networks depends on both road capacity and traffic volume. Thus, to evaluate the performance of networks, we have to consider such interrelation.

Current studies investigated the network macroscopic performance (i.e., MOEs) using different approaches. For example, the **M**acroscopic **F**undamental **D**iagram captures the relationship between two network MOEs which are density and flow [41, 45]. However, the MFD does not capture the impacts of changes in the network structure and traffic demand. Another group of approaches evaluate the impacts of changes in a single structural attribute on the network performance [4, 50]. For example, the researchers studied the impacts of nodes' betweenness on the onset of traffic congestion [4]. In addition, in our previous research, we studied the impact of the largest eigenvalue of the adjacency matrix on the congestion propagation [50]. These existing studies do not incorporate a set of multiple structural and dynamical attributes, which are required to represent network characteristics. In addition, they do not simultaneously model multiple network MOEs, which are necessary to represent the network performance.

Transportation planners and engineers typically perform traffic simulations to evaluate various design alternatives in order to improve the traffic conditions over an existing network or to build a new road network. The planners and engineers have to modify simulation models and run multiple simulations to evaluate the impacts of various proposed changes in network structure and traffic demand. Depending on the size of a network, such evaluation process would be time-consuming and tedious, especially if they want to assess several alternatives. In this research, the objective of the first research question is to formalize and develop a model that can predict the macroscopic measures of performance (i.e., average speed and volume) for new networks without performing traffic simulations. The inputs of the model are multiple structural and dynamical attributes of the new network, and the outputs are multiple network-wide MOEs. While the proposed model does not replace the simulation models, it is useful for pre-screening process of numerous design alternatives and leads to a significant saving of time and computational resources. The result of such pre-screening process is a small subset of the long list of design alternatives which will be further analyzed using simulation models.

I used a set of the existing structural attributes, such as the weighted degree and betweenness, which are presented in [3, 12, 33, 90]. I also proposed a set of dynamical attributes to capture various travel demand patterns across the network. The proposed dynamical attributes are the largest eigenvalues of an OD matrix. Then, I ran several traffic simulations to find network MOEs for different combinations of structural and dynamical attributes. In the next step, I employed a multivariate statistical method called the Canonical Correlation Analysis to capture the relationship among multiple MOEs and network attributes. Finally, using the captured relationship, I developed a model to predict macroscopic performance (i.e., multiple MOEs) of a new network. For the prediction, the model does not need the tedious task of simulation. The framework enables transportation modelers to understand how variations in network structure and dynamics could impact the macroscopic performance of design alternatives during the prescreening process.

2.2 Background Research

This section reviews related background research studies. First, I present the overview of transportation studies. Then, I review the studies related to the structure and traffic flow dynamics of (road) transportation networks. Finally, I discuss the existing studies related to the performance of a network.

2.2.1 Transportation Networks

Various research studies have been performed to investigate structure and dynamics of different transportation networks, such as rail- and subway [19, 24-26, 34, 91-98], airport [12, 20, 22, 23, 99-101], urban transit [102], public transportation [16, 54, 103, 104], and maritime transport networks [105, 106]. A specific type of transportation networks is an urban road network, in which the merges, diverges, and crossings can be considered as the network nodes, and the physical roads as the links. This section discusses the related studies from the viewpoints of network science and transportation engineering.

From the Viewpoint of Network Science

Traditionally, networks have been studied using graph theory concepts [11]. Since the 1950s, large-scale networks have been characterized as random graphs, which were first introduced by Paul Erdos and Alfred Renyi [107]. After some decades, the researchers discovered that some real-world networks are not random [11]. Several parallel developments performed during the past few years (e.g., emergence of large databases and increased computing power) showed the presence of three classes of networks [108-111]. The first class is the random graphs (e.g., road networks) [107]. They are still the benchmark for empirical studies. The second class, called small-world network (e.g., telephone call graphs), is the network in which a few set of nodes can be reached from other nodes by a small number of steps [112]. Finally, the third class, called scale-free network (e.g., airport networks, World Wide Web links), is the network in which its degree distribution follows a power-law [111, 113].

From the Viewpoint of Transportation Engineering

In transportation engineering, an urban road network is determined by: (1) supply-side properties (e.g., geometry, number of lanes), which I refer to as structural characteristics, and 2) demand-side properties (e.g., origin-destination traffic volumes and driving habits of population), which I refer to as dynamical characteristics. The traffic demand is assigned to a physical road network by two types of traffic assignment methods: 1) static and 2) dynamic. The static traffic assignment methods assume that link flows and link travel times are constant over a modeling period. Conversely, the **D**ynamic **T**raffic **A**ssignment (DTA) methods assume that link flows and link travel times are time-variant [114, 115].

In the DTA, the time-varying performance of a transportation network for a given traffic flow pattern is estimated by modeling movement of vehicles as they travel from their origins to destinations [42]. The simulation-based DTA methods were developed for different applications, such as optimize control, or real-world transportation planning without interrupting real-world traffic [44, 115, 116]. The examples of real-world applications of the simulation-based DTA models include: DynusT [117], DynaMIT [118], DYNASMART [119], VISTA [44], Dynameq [120], AIMSUN [121], TransModeler [38], INTEGRATION [122], METROPOLIS [123]. Based on the level of details, the current simulation-based DTA models can be categorized into three classes: 1) macroscopic, 2) mesoscopic, and 3) microscopic.

Macroscopic models: They employ aggregate concepts (e.g., fluids theory) without considering individual vehicle [124, 125]. In other words, they trade off the ability to model individual vehicles at small time intervals for the ability to model entire metropolitan areas at a single aggregated time. They use a set of differential equations to represent the evolution of traffic over time and space. Their outputs are static and aggregated, such as those found in the trip assignment step in the traditional four-step process.

Microscopic models: They model individual vehicle entities, decisions and interactions at time steps as small as one-tenth of a second. However, the microscopic models require a large amount of data and the detailed calibration of model parameters, and have high computational resource demands. The examples of the microscopic models include VISSIM [126], AIMSUN/2 [121], Paramics [127], and MITSIMLab [39].

Mesoscopic models: They combine different elements from microscopic and macroscopic approaches. They represent individual vehicles with a high degree of detail (such as microscopic models), but depict the activities and interactions of each vehicle with fewer details (such as macroscopic models). The mesoscopic simulation models do not necessarily locate vehicles precisely on the links of a network [128]. One mesoscopic approach is to group vehicles into packets that act as one entity and share speed [129]. The packets can be either discrete packets or continuous packets. In continuous packets, vehicles are distributed inside each packet, defined by the head and the tail points. In discrete packets, all vehicles belonging to a packet are grouped and represented by a single point [130, 131]. Another mesoscopic approach is the queue-server approach, such as DynaMIT [39] and DYNASMART [119]. In this approach, each road segment is modeled with two parts: a queuing and a moving part. Vehicles travel through the moving part with the speed calculated using a macroscopic model until they reach the queues on the current segment representing congestion [39].

2.2.2 Structure of Transportation Networks

Chan et al. (2011) studied 20 largest German cities for the year 2005 and proposed a variety of node-, link-, and cell-based attributes based on the structural and spatial characteristics of a road network. Their results, in addition to recent empirical studies [37, 132, 133], have shown that topological quantitative similarities exist between road networks of different cities at a network level. The resultant structural attributes of the existing studies are summarized in Table 2.1, Table 2.2, Table 2.3, and Table 2.4. In general, the structural attributes can be divided into two levels:

1) Network-wide attributes, which describe characteristics of the entire network (Table 2.1).

Table 2.1. The Network-wide Structural Attributes of Road Networks

Attribute	Description
Network size	Number of nodes and links
Network diameter	Longest shortest path
Average shortest path	Average of all shortest paths
Gamma index	The gamma index is a measure of the density of the network. It is the ratio of the number of links divided by maximal number of links for a given number of nodes [16]
Alpha index	The alpha index is another measure of the density of the network; it is the ratio of the number of elementary cycles divided by maximal number of elementary cycles [16]

Attribute	Description
Compactness	A measure of how much a city is ‘filled’ with roads. If we denote the area of a city by A and the total length of roads by l_T, the compactness $\Psi \epsilon [0, 1]$ is defined as [134]: $\Psi = 1 - \frac{4A}{(l_T - 2\sqrt{A})^2}$
Ringness	A measure of the importance of a ring and the extent that the arterials are organized as trees. If we denote the total length of arterials on rings by l_{ring} and the total length of all arterials by l_{tot}, the ringness $\varphi \epsilon [0, 1]$ is defined as [134]: $\phi_{ring} = \frac{l_{ring}}{l_{tot}}$
Route factor	The ratio is larger than one; the closer the route factor to one, the more efficient the network. If we denote the natural Euclidean distance by $d_E(i,j)$, and the total ‘route’ distance as the length of the shortest path between i and j by $d_R(i,j)$, the route factor (also referred to as the detour index or the directness [135] for this pair of nodes (i, j) is then given by $$Q(i,j) = \frac{d_R(i,j)}{d_E(i,j)}$$
Eigenvalues of Adjacency	Another important global view based on eigenvalues is referred to as spectral theory. Spectral theory studies the adjacency matrix (or the Laplacian) of network and connects its eigenvalues to network properties [136]. For instance, the largest eigenvalue of the network is a measure of network connectivity.

2) Local attributes, which describe the characteristics of either groups or individual nodes within a network. Such attributes can be categorized into: (a) node-based (Table 2.2), (b) link-based (Table 2.3), and (c) cell-based (Table 2.4) attributes [90, 92].

Table 2.2. The Local Node-based Structural Attributes of Road Networks

Attribute	Description
spatial density	Nodes with high degrees are usually located in dense urban areas, often close to the city centers.
degree distribution	The node degree is the number of links it has to other nodes and the degree distribution is the probability distribution of node degrees over the entire network. Researchers have studied the degree distribution in different road networks. For instance, Buhl et al. (2006) studied 41 road networks and concluded that the average node degree ranges between 2.02 and 2.86 [35]. The reported average node degree for the US interstate highway network is 2.86 [33]. Recently, Chan et al. (2011) noted that the average node degrees of the individual cities lie within a narrow range between 3.17 and 3.31 [21].
Betweenness centrality	The betweenness centrality of a node is equal to the number of shortest paths from all nodes to all others that pass through that node. The betweenness centrality is found to have a power law exponent in the range [1.279, 1.486]. The results indicate a strong heterogeneity of the network with the existence of a few central roads [137, 138].
Average nearest neighbors degree	The average nearest neighbors degree shows whether the node is surrounded by large or small nodes
Clustering coefficient	The clustering coefficient is a measure of tightness and density of links, i.e., the measure of degree to which nodes tend to cluster together.

Table 2.3. The Local Link-based Structural Attributes of Road Networks

Attribute	Description
Link length distributions	The local relationships between link and node characteristics indicate that, on average, link lengths decrease with increasing of involved node degree. This signals a possible relationship with the spatial distribution of nodes and links within cities.
Link angle distributions	A measure of rectangularity of intersections. The link angle distributions have two general peaks at 90◦ (which corresponds to perpendicular intersections of pairs of roads) and 180◦ (which shows that the corresponding nodes are formed by a straight road from which a secondary one splits perpendicularly).
Double-angle distributions	A measure of straightness of the crossing roads. In general, nodes with degree $k_n = 4$ have only one sharp peak around 180◦, while nodes with degree $k_n = 3$ have two peaks at 180◦ and 270◦.
curvature	The ratio of link length divided by Euclidean distance between the two connected nodes.

Table 2.4. The Local Cell-based Structural Attributes of Road Networks

Attribute	Description
cell area	Area that is formed by closed loops consisting of different links.
topological cell degree	The number of neighboring cells [139], where neighboring cells of a cell are the ones that have a common edge with the cell.
geometric cell degree	Also known as the cycle length [140], the number of straight road segments forming the cell.
cell diameter	The maximum diameter of cells.

Attribute	Description
cell perimeter	Perimeter of the cell that is formed by closed loop consisting of different links.
form factor	The measure of the shape of a cell; combination of cell area and diameter [37]; is defined as the ratio between the actual cell area A_c and the area of the smallest possible circumscribed circle ($\pi . d_c^2$).

2.2.3 Dynamics of Transportation Networks

In general, networks have dynamical processes running on top of them. A variety of possible dynamical processes was investigated, such as synchronization [141], search and random walk [142], and spread of ideas, infection, or computer viruses on different types of networks [3, 11, 82, 143, 144]. For a road transportation network, the dynamical process is the traffic flow [21, 55, 145]. Basically, traffic flow patterns depend on drivers' behavior and ways of thinking, socio-economic environment, and time constraints [90]. Existing studies focused on traffic flow dynamics, such as flow of people within city and commuting traffic flow between different cities [37]. However, these studies did not consider the physical topology of a network. For instance, the link with the largest betweenness cannot be detected without considering the topology of the network.

Different approaches have been proposed to identify the traffic flow pattern for a network. One simple approach is to consider a constant generation rate for all the nodes of a network. This approach assumes that a constant number of vehicles are generated at each time step for every node with a certain probability [4]. This approach simplifies the definition of traffic flow patterns, but it does not fit real-world traffic patterns. In reality, traffic flow is neither random nor constant for all nodes. Another approach is to predict traffic volume of different links using sparse traffic counts and additional information, such as local land use data, time-steps, local employment, population attributes, roadway details [146, 147]. The limitation of such techniques is that they rarely use the topological attributes of a network. Instead, such approaches used statistical estimation techniques, such as weighted regression, or universal Kriging models [147-149].

Another approach is to estimate **O**rigin–**D**estination (OD) matrix which provides traffic volumes between all pairs of geographical areas (i.e., the origins and destinations) [150]. The first method to estimate the OD matrices is from household interviews or partial traffic counts [151, 152] In general, such method fails to estimate detailed OD matrices because of its high data acquisition cost and low accuracy due to sparse data [151, 152]. The second method for estimating OD matrices is to record the number of vehicles passing by a section of the roads by leveraging different sensors, such as road cameras and loop detectors. This method is also expensive and prone to malfunctioning [151, 152]. The third method is to use location traces of probe vehicles at high resolutions (up to one Hz) based on GPS data [153]. However, they are often degraded on purpose due to privacy issues, and thus cannot provide detailed OD matrices at large scales. The fourth method is to utilize mobile phone data which can be used wherever the geographical locations of communications (e.g., phone calls and texts) are recorded [8].

Recently, Wang et al. (2012) presented a method to estimate an OD matrix with mobile phone data [29]. They counted the number of trips between all pairs of zones to obtain the distribution of travel demands. The resultant distribution is based on the number of the phone users. Therefore, they rescaled the distribution proportional to the population size and possible transportation modes (e.g., car, carpool, public transportation, bicycle and walk) of each zone to find the OD matrix.

Wang et al. (2012) considered the temporal variation of the OD matrices over a day [29]. Based on the observed distribution of daily traffic (Figure 2.1), they divided a day into four periods: (1) morning: 6 am–10 am, (2) noon & afternoon: 10 am–4 pm, (3) evening: 4 pm–8 pm, and (4) night: 8 pm–6 am. Then, for each period, they estimated the distribution of travel demands. Therefore, the spatiotemporal travel pattern over a day was presented by four OD matrices, for morning, noon & afternoon, evening, and night.

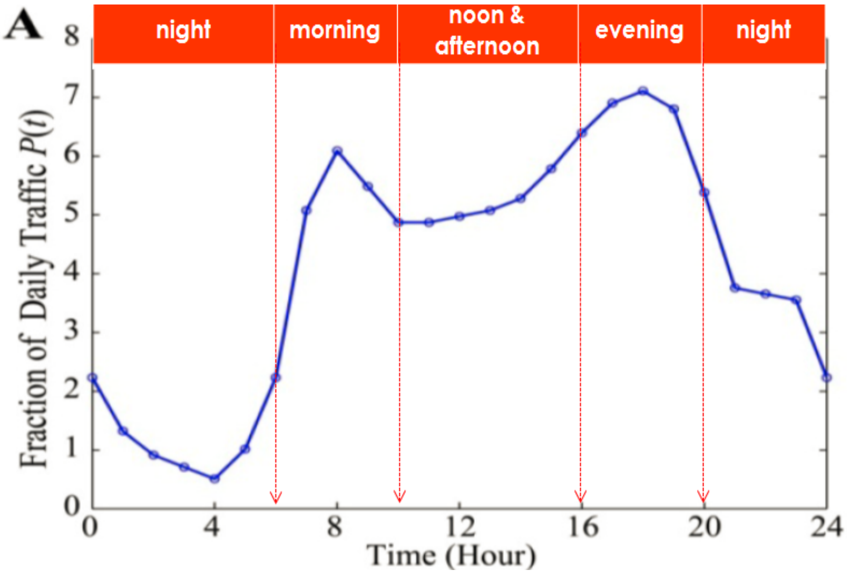


Figure 2.1. The Observed Distribution of Daily Traffic Presented in [29]

2.2.4 Performance of Transportation Networks

The performance of road networks has been studied from different viewpoints. A group of researchers has focused on the macroscopic performance of road networks [41, 45, 46]. For instance, Geroliminis and Daganzo (2008) verified the existence of a **M**acroscopic **F**undamental **D**iagram for an urban area. The MFD shows network performance by linking its measures of performance (i.e., flow and density) (Figure 2.2). They showed that a well-defined MFD exists if the congestion is evenly distributed, which is not valid for real-world networks. Recently, Geroliminis and Sun (2011) explored the impacts of spatial distribution of vehicles on the shape of an MFD [41]. They concluded that if the spatial distribution of traffic density is same for two different time intervals, the average flows of these two time intervals should be equal.

The second group of researchers has studied the prediction of threshold behavior of networks based on a single structural attribute (e.g., betweenness) [4, 32, 47, 51]. For instance, Zhao et al. (2005) discussed that the node with largest betweenness can be easily congested, and such congestion can propagate throughout a network. Therefore, they defined the phase-transition as the point when the node with largest betweenness is congested. Sun et al. (2008) studied the dynamics of traffic congestion by deriving critical flow generation rate (i.e., number of generated vehicle at each time step) based on the characteristics of links (e.g., road length, maximum speed) [47]. Based on their results, the critical flow generation rate is a function of: (a) average shortest path length, (b) number of nodes, (c) betweenness, and (e) characteristics of the link with largest betweenness (e.g., road length, maximum speed).

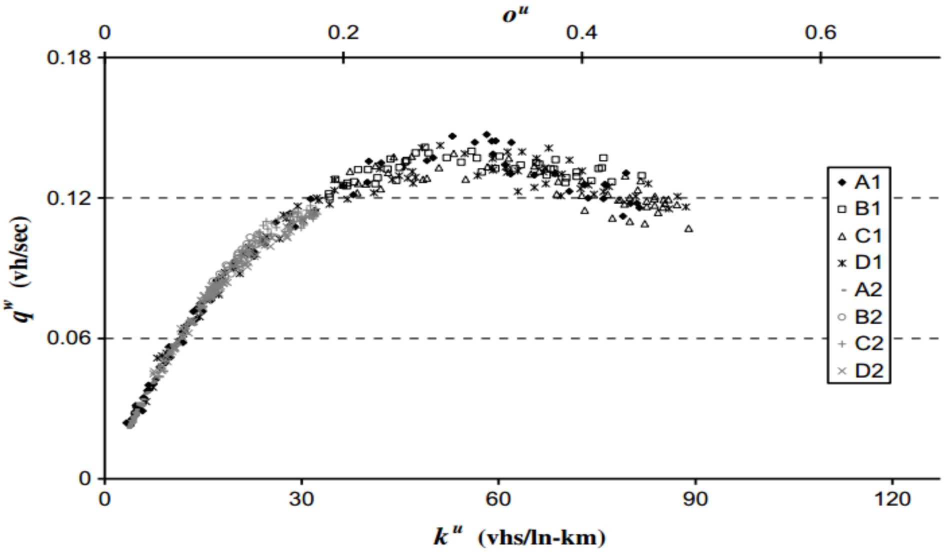


Figure 2.2. An Example of MFD presented in [45]

Although the current solutions for modeling network performance (e.g., MFDs or phase-transition models) are capable of characterizing macroscopic performance of a network, there exists a critical gap. The MFDs depict the relationship between two MOEs at a time. However, they do not show the impacts of variations in structure and dynamics on network performance. Similarly, the phase-transition models consider a few structural attributes (e.g., betweenness and number of nodes) which do not capture the characteristics of a network. Therefore, a framework is needed to capture the relationship among multiple MOEs (at a time) and multiple structural and dynamical attributes. The advantage of having such a framework over the existing models (i.e., MFDs and phase-transition models) is the ability to predict performance of a new network based on its structural and dynamical attributes without performing traffic simulations.

2.3 Proposed Approach

This section describes the proposed approach for modeling the performance of a road network under normal condition without considering disruptions or distractions. In this research, a framework is developed that depicts multiple macroscopic MOEs (e.g., average speed, delay, and volume) based on a combination of multiple structural and dynamical attributes. The formalization of the proposed framework is motivated by the previous works of Helbing (2009), and Geroliminis and Daganzo (2008; 2011) [41, 45, 46]. From a macroscopic viewpoint, Helbing (2009) showed that a network can be in three different conditions: (1) *undersaturated,* where the capacities of links and intersections are sufficient to accept all vehicles, (2) *congested,* where the capacities of intersections are exceeded, or (3) *oversaturated,* where the capacities of links and intersections are exceeded. They showed that although link-based results were different, the average over different road sections led to a smooth relationship between speed and density. Helbing (2009) concluded that one should study the relationship between network structure and traffic flow dynamics to understand network performance.

2.3.1 Framework

The proposed approach is presented in Figure 2.3. The main objective is to ascertain the impacts on network performance due to the variations in its structural and dynamical attributes through several traffic simulations. The approach is divided into four steps (Figure 2.3). In the first step, real road networks are assimilated. In the second step, various structural and dynamical attributes are identified to represent network structure and dynamics (i.e., traffic demand), respectively. In the third step, different traffic flow patterns are loaded on each network through several traffic simulations. The output is a database including multiple network MOEs, structural and dynamical attributes. Finally, in the fourth step, the relationship among different network MOEs and attributes is captured by a multivariate statistical method. Based on this relationship, I propose a model to predict multiple MOEs for new networks based on their structural and dynamical attributes.

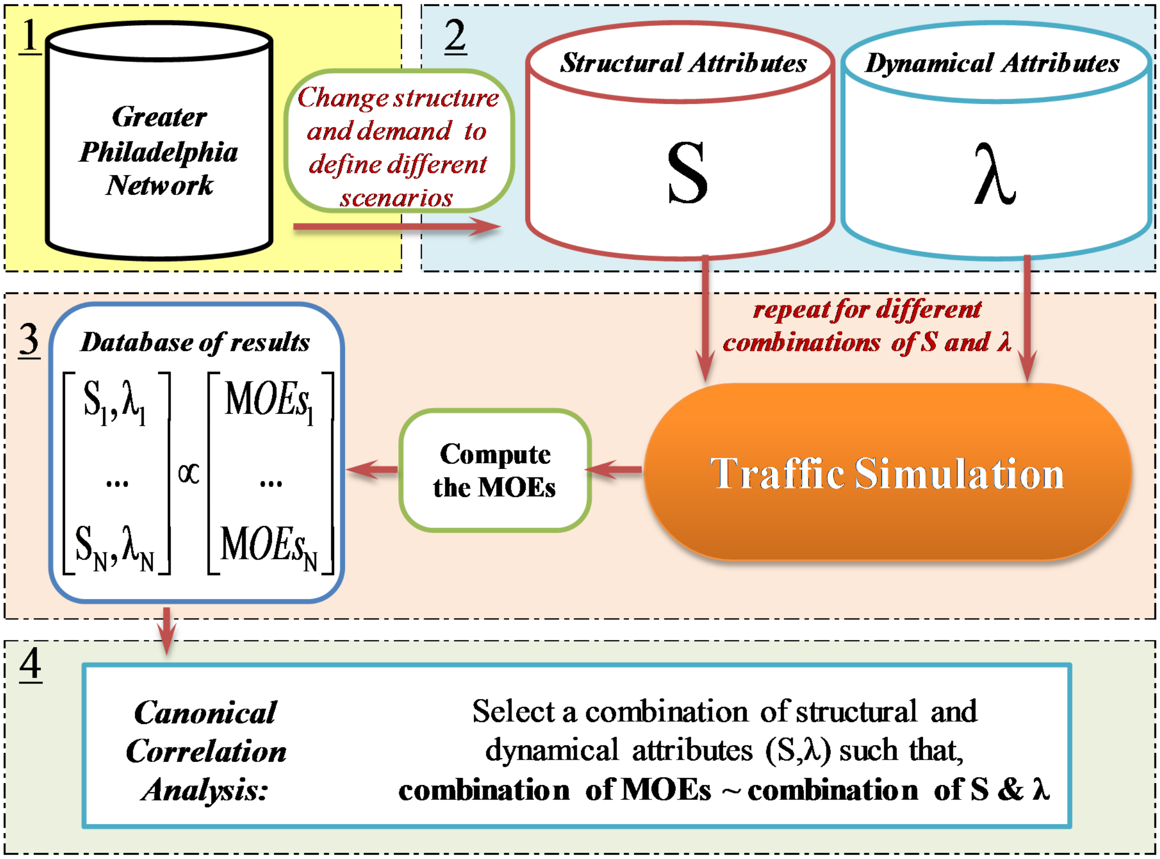

1
Greater Philadelphia Network
Change structure and demand to define different scenarios
2
Structural Attributes
S
Dynamical Attributes
λ
repeat for different combinations of S and λ
3
Database of results
$\begin{bmatrix} S_1,\lambda_1 \\ \dots \\ S_N,\lambda_N \end{bmatrix} \propto \begin{bmatrix} MOEs_1 \\ \dots \\ MOEs_N \end{bmatrix}$
Compute the MOEs
Traffic Simulation
4
Canonical Correlation Analysis:
Select a combination of structural and dynamical attributes (S,λ) such that,
combination of MOEs ~ combination of S & λ

Figure 2.3. Flowchart of the Proposed Approach for Research Question 1

Step 1: Acquiring Real Road Networks

The first step is to build a database including several road networks from different urban areas. To develop a comprehensive model, the database should represent a diverse range of road networks representing different network structures and traffic demands. So, several road networks with different sizes should be acquired from various urban areas. The road networks should be modeled and calibrated by comparing with real traffic data. Such simulations can be performed using a macroscopic simulation software because the focus of the first research is to predict network-wide (i.e. macroscopic) measures of performance.

Step 2: Identifying Network Attributes

The second step is to determine the proper representations of network structure and traffic demand. The structural and dynamical attributes are defined as the representations of network structure and traffic demand. I developed an algorithm which: (i) takes required information (i.e., weighted adjacency matrix) of a given network as an input, (ii) processes the information using different tools (e.g., MATLAB, iGraph and ArcGIS), and (iii) returns a set of its structural attributes (e.g., weighted degree and betweenness distribution). To calculate these structural attributes (shown in Table 2.6), I employed iGraph which is a free software package for creating and manipulating different types of graphs [154]. The proposed structural attributes can be grouped into: (a) local attributes, such as link's capacity, weighted degree, and weighted betweenness, and (b) network-wide attributes, such as largest eigenvalue of the weighted adjacency. The local attributes were identified for all individual links of the network. So, the means and standard deviations of local attributes are considered as the corresponding network-wide attributes. Finally, the proposed algorithm returns a set of network-wide structural attributes.

The next task is to identify the representations of traffic demand. In this research, I used an OD matrix that captures traffic volumes between all Origin-Destination pairs for four different times of a day. However, the size of the OD matrix can grow quadratically with the number of OD pairs, and thus the OD matrix requires significant memory for a large network. Thus, I proposed a subset of largest (positive) eigenvalues of the OD matrix to represent the traffic demand. The subset will be selected based on a sequential selection approach similar to the forward-search sequential feature selection presented in [155]. In this approach, a subset of eigenvalues is selected by sequentially adding an eigenvalue (to the selected subset of eigenvalues) until no improvement can be seen in performance prediction.

Step 3: Running Traffic Simulations

The third step is to assess the traffic conditions on different road networks via several traffic simulations. For each road network, four simulation iterations were executed by loading four OD matrices representing four different times of a day. Since network-wide MOEs (e.g., average speed, delay, and volume over capacity) are needed, there is no need for high-fidelity microscopic models. A macroscopic model is required to compute the MOEs for all links. We discussed different existing macroscopic package in Research Background Section. In this research, I employed the VISUM software because: (i) the VISSUM software returns the needed MOEs, and (ii) the calibrated traffic model of the Greater Philadelphia is developed in the VISUM.

Each simulation run returns four sets of MOEs each of which is associated with a set of structural and dynamical attributes for the AM, MD, PM, and NT periods. The output of this step is an excel database that includes network MOEs for various combinations of structural and dynamical attributes.

Step 4: Developing the Performance Prediction Model

The resultant database of the third step is used as the input for developing the performance prediction model using the Canonical Correlation Analysis method. In this research, independent variables (aka predictors) are the network structural and dynamical attributes, and dependent variables (aka observers) are the network MOEs. So, there exists multiple independent and multiple dependent variables (Figure 2.4).

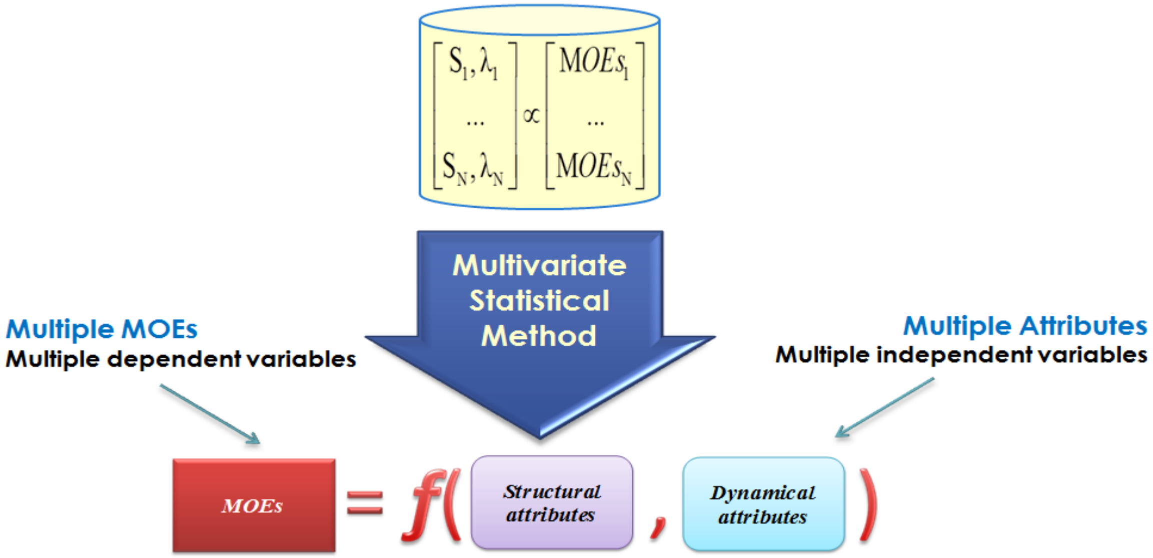

$\begin{bmatrix} S_1, \lambda_1 \\ \dots \\ S_N, \lambda_N \end{bmatrix} \propto \begin{bmatrix} MOEs_1 \\ \dots \\ MOEs_N \end{bmatrix}$
Multivariate Statistical Method
Multiple MOEs
Multiple dependent variables
Multiple Attributes
Multiple independent variables
MOEs
= f(
Structural attributes
,
Dynamical attributes
)

Figure 2.4. Process of Performance Prediction

In multivariate statistics, multivariate regression analysis is used to predict a single dependent variable from multiple independent variables [156]. However, when both dependent and independent variables are multivariate, the **C**anonical **C**orrelation **A**nalysis (CCA) is used to simultaneously predict multiple dependent variables from multiple independent variables. The CCA has been widely used to measure linear relationship between two sets of multivariate variables [157, 158]. Generally, the CCA can be used as:

(a) an explanatory tool to investigate whether two sets of variables are related, (b) a prediction tool to predict dependent variables based on independent variables, and (c) a tool to investigate whether one set of variables relates longitudinally across two time points [157].

The CCA is aimed at identification and quantification of the interrelations between a p-dimensional variable X and a q-dimensional variable Y [159]. The CCA seeks for linear combinations of the original variables, $a^T X$ and $b^T Y$, that have maximal correlation. In mathematical terms, the CCA selects vectors $\alpha \in R^p$ and $\beta \in R^q$ such that,

$$(\alpha, \beta) = \underset{a,b}{\operatorname{argmax}} |Corr(a^T X,\ b^T Y)|$$

The selected univariate variables, $U = X.\alpha$ and $V = X.\beta$, are referred as canonical variates. The number of pairs of canonical variates is equal to the minimum of the dimensions of X and Y. In the CCA, each pair of canonical variates may provide an interpretation of the relationship. The ones with the highest correlations are the most important ones.

Researchers investigated the use of the CCA method as a prediction tool to predict multiple dependent variables from multiple independent variables [160-162]. Another interesting output of the CCA is the loading of each variable, which shows the contribution of that variable in the linear combination. The loading of independent variables illustrates the prediction capability of the independent variables, and shows which variables are more important for the prediction [158].

In this research, I used the CCA method to predict the network MOEs. I have p number of independent variables (i.e., network attributes) and q number of dependent variables (i.e., network MOEs). I employed MATLAB statistical toolbox functions (e.g., *canoncorr*) to perform the prediction. The CCA method captures the highest correlation between linear combinations of MOEs and linear combinations of network attributes.

The CCA method returns α and β which are the matrices of canonical coefficients for the X (i.e., network attributes) and Y (i.e., MOEs). In addition, the CCA method returns K linear relationships between pairs of $(\beta_k Y, \alpha_k X)$, where $K = \min(p, q)$. Using a regression analysis, I fitted K trend lines to these K graphs. The coefficients of these linear regressions are sorted in two vectors: $\overrightarrow{c_1}$ and $\overrightarrow{c_2}$,

$$Y.\beta = \overrightarrow{c_1}.(X.\alpha) + \overrightarrow{c_2}$$

When the canonical coefficients are computed, they can be used to predict the MOEs of a new network without performing a traffic simulation. For any new network, based on the network structure and dynamics (i.e., X_0), I can calculate:

$$\overrightarrow{c_1}.(X_0.\alpha) + \overrightarrow{c_2} = C_0$$

$$Y_0.\beta = C_0 \xRightarrow{\times\ \beta^{-1}\ (which\ is\ the\ inverse\ matrix\ of\ \beta\,)} Y_0.\beta.\beta^{-1} = C_0.\beta^{-1}$$

With that, I can predict network MOEs (Y_0) for the new network. The developed performance prediction model is as follows:

$$Y_0 = C_0.\beta^{-1}$$

In addition, the CCA method helps us to calculate the prediction capability (i.e., loading) of the independent variables. For that, I need to calculate the internal correlations of X (i.e., ρ_X, which is the matrix of the correlation coefficients among each pair of attributes X). Then, the loadings vector (i.e., $Loading_X$) of individual independent variables are:

$$Loading_X = \rho_X\,.\alpha$$

2.4 Results

This section presents the results of the proposed framework for the Greater Philadelphia region. To develop a performance prediction model that can predict the MOEs of any new network, one should use a diverse database including calibrated traffic models of various urban road networks. However, building a database of calibrated models for different networks requires significant amount of time, and is not within the scope of the proposed research.

In this research, I obtained the road network data of the Greater Philadelphia region from the Delaware Valley Regional Planning Commission (DVRPC), which is a federally designated Metropolitan Planning Organization. The DVRPC is responsible for coordinating the transportation planning process conducted in the region, including collecting and providing data for regional studies, conducting research, and developing the long range transportation plan and the short-term transportation improvement program.

As shown in Figure 2.6, the Greater Philadelphia region includes following counties: Bucks, Chester, Delaware, Montgomery and Philadelphia (in Pennsylvania State) and Burlington, Camden, Gloucester and Mercer (in New Jersey State). It is a region including 352 municipalities in 2,439,899 acres, which serve total population of 5,626,186.

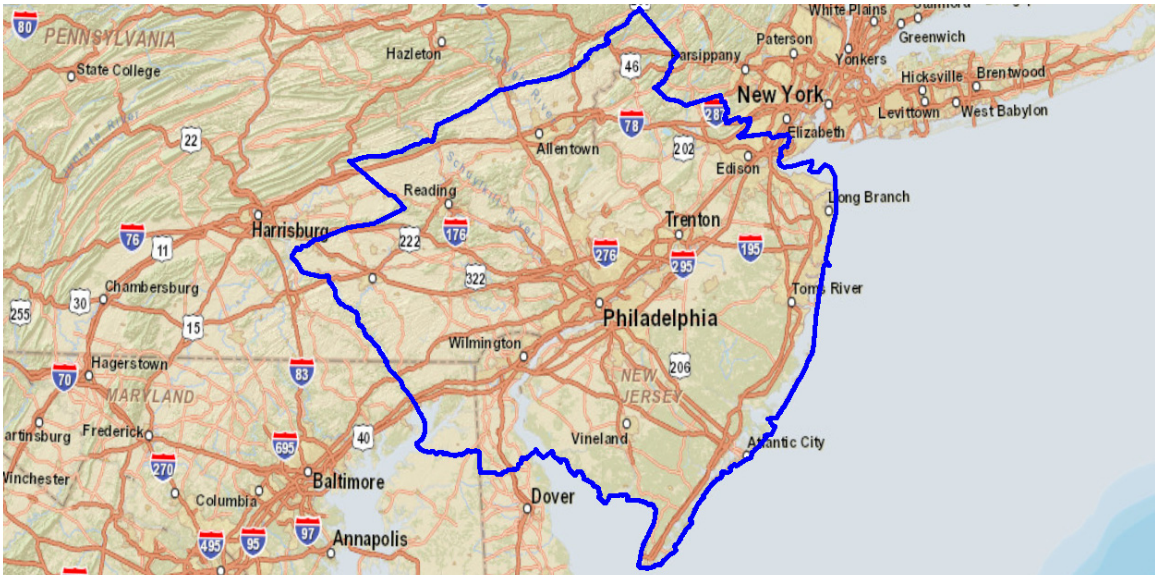


Figure 2.5. The Study Area (The Greater Philadelphia Region)

The acquired traffic model is a calibrated model developed in the VISUM regional traffic simulation software. The model includes four Origin-Destination (OD) matrices for four different times of a day: AM Peak (6:00 AM to 10:00 PM), Midday (10:00 AM to 3:00 PM), PM Peak (3:00 PM to 7:00 PM), and Night Time (7:00 PM – 6:00 AM). The DVRPC calibrated the model at a regional level by comparing the traffic simulated results with real traffic counts for different times of a day. Originally, the model contained around 254,800 links. In this research, I considered important roadways, such as highways, arterials, and major collectors to create a medium size network of 101,909 links. This is done to speedup simulation run, which on average took 60 hours on our machine.

I tested several settings (of network structure and traffic demand) by changing the links' capacities and OD matrices of the Greater Philadelphia network. So, the predicted results are assumed to be valid for different similar variations in network structure and traffic demand. Though the proposed framework is validated on one network, it is scalable to other networks by building a database of multiple road networks.

I used different strategies to change the roads' capacities and the traffic demand. Table 2.5 lists my strategies, in which I altered either capacities or traffic demand of either all links or a subset of the links based on their volume over capacity ratio (V/C). For example, in scenario S2, I changed the capacity of the roads whose V/C were greater than 1. In this case, I multiplied the capacity of these links by their V/C, i.e., I increased the capacity of the links whose volumes were more than their capacities.

Table 2.5. Different Scenarios for Changing Links' Capacities

Scenario	**Links that are subjected to change:**	**Capacity change**	**OD matrices**
S1	None	No change	Default ODs
S2	Links whose V/C ≥ 1.0	New Capacity = V/C. Capacity	Default ODs
S3	Links whose V/C ≥ 0.6	New Capacity = 0.6 Capacity	Default ODs
S4	All links	New Capacity = 1.5 Capacity	Default ODs
S5	All links	New Capacity = 0.6 Capacity	Default ODs
S6	All links	New Capacity = V/C. Capacity	Default ODs
S7	Randomly selected links (50%)	New Capacity = 0.6 Capacity	Default ODs
S8	Randomly selected links (50%)	New Capacity = V/C. Capacity	Default ODs
S9	Randomly selected links (50%)	New Capacity = 1.5 Capacity	Default ODs
S10	None	No change	Transpose of default ODs

Scenario	Links that are subjected to change:	Capacity change	OD matrices
S11	Links whose V/C ≥ 0.5	New Capacity = 1/(V/C). Capacity	20% increase
S12	All links	New Capacity = V/C. Capacity	25% increase
S13	All links	New Capacity = 2.0 Capacity	100% increase

The weights of a weighted adjacency matrix represent the links' capacities. So, the variation in the capacities led to the variation of all weighted attributes, such as weighted degree and betweenness. However, other structural attributes, such as number of nodes and links, form factor, diameter, and Gama index, remained the same. Hence, among the structural attributes discussed in the Background Research section, I used the eight structural attributes that are listed in Table 2.6.

Table 2.6. Proposed Structural Attributes

Structural Attributes	
1	Average of links' capacities
2	Standard deviation of links' capacities
3	Skewness of links' capacities
4	Largest eigenvalue of weighted adjacency
5	Average of weighted degree
6	Standard deviation of weighted degree
7	Average of weighted betweenness
8	Standard deviation of weighted betweenness

In the “Proposed Approach” section, I discussed that the dynamical attributes are represented as a subset of largest eigenvalues obtained using the forward-search sequential selection approach. In this research, I implemented this forward-search sequential selection approach and selected six largest eigenvalues.

I ran the traffic simulations for each of the 13 scenarios (shown in Table 2.5) using the VISUM software. Each simulation run took an average of 2 days. For each road network listed in Table 2.5, there were four OD matrices representing four different times of a day. Therefore, each simulation run returned four combinations of structural and dynamical attributes for four different times of a day. By performing simulations, I created a database including 52 sets of MOEs (e.g., speed, delay and V/C) and network attributes (e.g., weighted degree and betweenness, capacity, and largest eigenvalues of the OD matrix).

Finally, I used the CCA method to capture the relationship among different MOEs and network attributes. In this research, the number of variables in X and Y were 14 (including eight structural attributes and six dynamical attributes) and six (including the average and standard deviation of: speed, delay, and V/C). Therefore, the number of the captured canonical variates (K) was six:

$$p = 14 \quad and \quad q = 6 \ \rightarrow K = \min(p, q) = 6$$

I calculated α and β which were the matrices of canonical coefficients for the X (i.e., network attributes) and Y (i.e., MOEs). The α and β matrices had six columns. Each column of α (and β) contained the coefficients of individual X_i (and Y_j) in the canonical variates.

$$\alpha = \left[\begin{array}{cccccc}
17526.17 & -201528 & -1217.24 & 256947.2 & 894677.9 & -533 \\
0.007171 & -0.02404 & 0.044648 & -0.00897 & 0.332846 & -0.16 \\
-0.0149 & -0.0016 & -0.10003 & 0.038081 & -0.52401 & 0.228 \\
2.52E-06 & 1.71E-05 & 1.03E-05 & -1.44E-05 & -3.55E-05 & 3.01E \\
-4451.84 & 51190.26 & 309.1921 & -65267.4 & -227258 & 13551 \\
-0.00281 & 0.007474 & -0.0159 & 0.003274 & -0.11595 & 0.056 \\
-1.22E-05 & -2.03E-06 & -3.62E-05 & 2.93E-05 & -9.60E-05 & 2.44E \\
1.51E-06 & -3.79E-08 & 4.24E-06 & -3.44E-06 & 1.12E-05 & -2.91E \\
-1.72E-04 & 3.85E-05 & -4.35E-05 & 9.32E-06 & -1.44E-04 & -5.36E \\
2.05E-04 & -0.00015 & 2.90E-04 & 4.69E-05 & 1.48E-04 & -0.00 \\
4.94E-06 & 5.33E-05 & -1.89E-04 & 0.000194 & 1.89E-04 & 2.61E \\
-4.16E-05 & 8.31E-05 & -0.00028 & -4.21E-04 & -1.31E-05 & 6.82E \\
-1.41E-04 & -9.48E-05 & 1.40E-04 & 2.77E-04 & 2.12E-04 & -0.00 \\
7.58E-05 & 1.31E-04 & 3.88E-05 & -0.00016 & -4.78E-04 & 0.00
\end{array}\right.
\quad
\beta = \left[\begin{array}{cccccc}
1.287565 & -0.95647 & 3.091475 & -4.27142 & -0.57272 & 1.496 \\
0.974016 & 0.768594 & 3.582141 & -0.64788 & 1.573034 & 1.874 \\
0.2338 & -0.22104 & -0.15783 & -2.27359 & 0.074721 & 0.028 \\
-0.00085 & 0.000834 & 0.000828 & 0.009093 & -1.31E-04 & -1.55E \\
0.045036 & -0.17978 & 0.282873 & -0.25663 & -0.09826 & 0.197 \\
-0.15127 & 0.168688 & -0.3196 & 0.135361 & -0.27265 & -0.44
\end{array}\right.$$

The six sets of canonical variates are the linear relationships between six linear combinations of X and six linear combinations of Y (i.e., $X.\alpha$ and $Y.\beta$). While one can define infinite sets of canonical variates (i.e., infinite linear combinations of X and Y), these canonical variates are the six sets of linear combinations that have the largest correlations among these infinite combinations. I plotted these six linear combinations in Figure 2.6 in which the x-axis shows a linear combination of the network attributes (i.e., $X.\alpha$) and the y-axis shows a linear combinations of the MOEs (i.e., $Y.\beta$). Using a linear regression analysis, I fitted six trend lines to these six sets of data. The equations of these trend lines are also shown on the graphs on Figure 2.6.

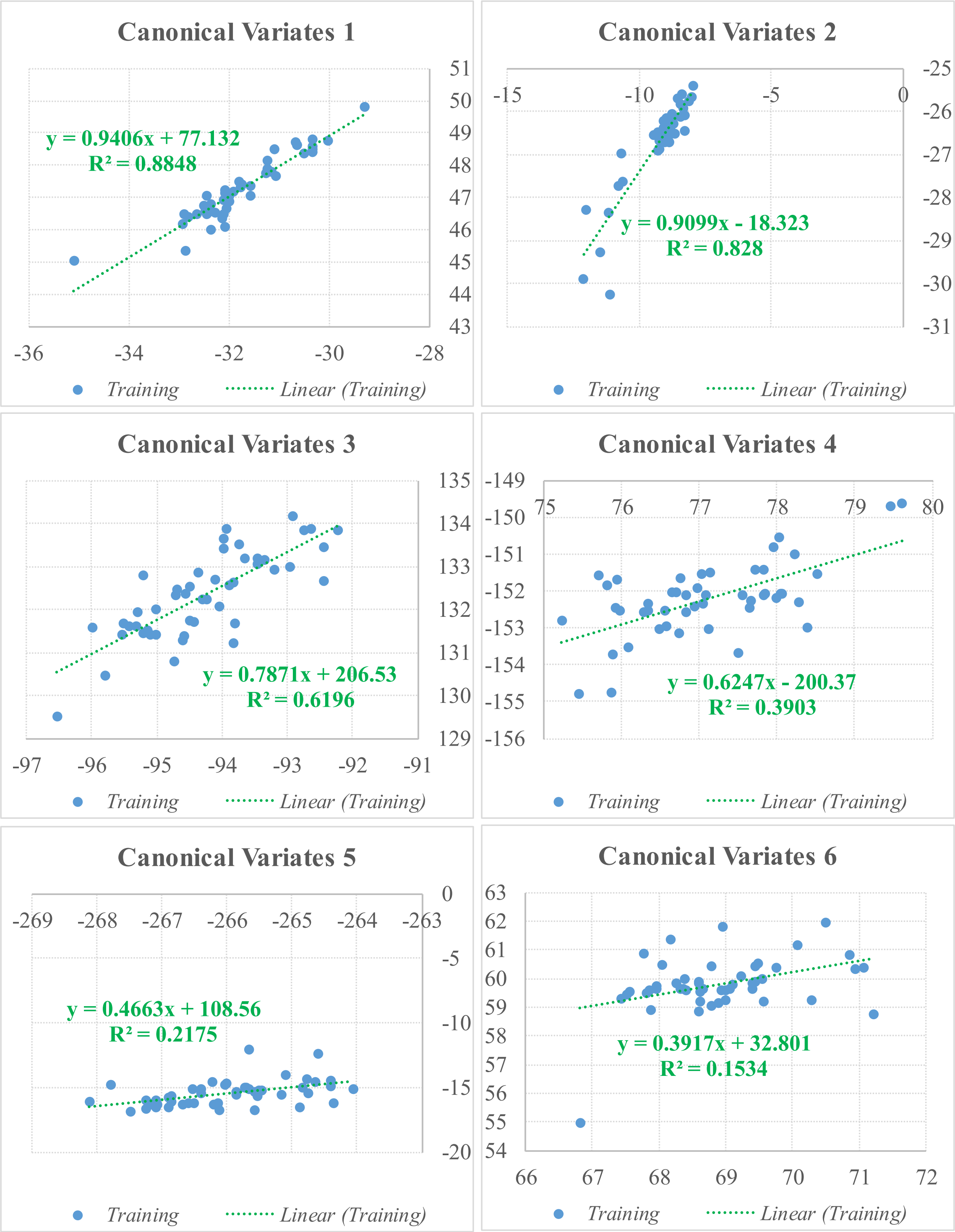


Figure 2.6. Relationships between Canonical Variates

The six equations of these trend lines (shown on Figure 2.6) are the basis on which I developed the performance prediction model. In this research, I have six unknowns (i.e., averages and standard deviations of the speed, V/C, and delay) and six equations. So, I can predict the unknowns using the system of these six linear equations. However, the effectiveness of a prediction depends on the goodness of fit of the captured linear relationships. Based on the r-squared values (i.e., measures of goodness of fit) shown on Figure 2.6, the first set of canonical variates had the best fit (r-squared=0.88), and the sixth one had the worst fit (r-squared=0.15). The r-squared values of the last three sets of canonical variates (i.e., Canonical Variates 4, 5, and 6) were lower than 0.7 which means these pairs of canonical variates were not highly correlated. Such low captured correlations would negatively affect the prediction of the MOEs. One possible way to improve such a limitation would be to assimilate a diverse database of networks which is beyond the scope of this research.

The coefficients of the linear regressions (shown in Figure 2.6) were sorted in two vectors $\overrightarrow{c_1}$ and $\overrightarrow{c_2}$, as:

$$\mathbf{c_1} = \begin{vmatrix} 0.941 \\ 0.910 \\ 0.787 \\ 0.625 \\ 0.466 \\ 0.392 \end{vmatrix} \quad \& \quad \mathbf{c_2} = \begin{vmatrix} 77.132 \\ -18.323 \\ 206.531 \\ -200.374 \\ 108.557 \\ 32.801 \end{vmatrix}$$

Based on the computed matrices of canonical coefficients (α, β) and vectors of coefficients of linear regressions ($\overrightarrow{c_1}$, $\overrightarrow{c_2}$), for any new network with known attributes (i.e., X_0), I can compute:

$$C_0 = \overrightarrow{c_1}.(X_0.\alpha) + \overrightarrow{c_2}$$

Thus, the developed performance prediction model that predicts the MOEs (i.e., Y_0) for a new network based on its attributes (i.e., X_0) is as follows:

$$Y_0 \ = C_0.\beta^{-1}$$

Although this model returns the average traffic conditions (i.e., average MOEs), it cannot completely replace the simulation models. Instead, this model can be used during the pre-screening process of numerous design alternatives for macroscopic assessments. The few selected alternatives can be then analyzed using detail simulation models. This approach (i.e., first using the proposed model for pre-screening and then using detail simulation models for accurate comparison) speeds up any planning or design processes which need the analysis of numerous alternatives.

2.5 Validation

As discussed earlier, the created database in this research contained 52 sets of MOEs and network attributes. Among these 52 sets, four sets were the simulated results of the original network of the Greater Philadelphia (i.e., S1 in Table 2.5 which represents the real scenarios) for four different times of a day. The other 48 sets were the simulated results of the altered networks (i.e., S2 to S13 in Table 2.5 which represent the altered scenarios).

To validate the framework, I first developed the performance prediction model based upon the 48 sets of the MOEs and network (structural and dynamical) attributes. Then, I used the developed model to predict the MOEs of the four real scenarios to test the effectiveness of prediction. For each of the four real scenarios, I calculated X_0 and used the model ($Y_0 = C_0.\beta^{-1}$) to predict the MOEs (i.e., Y_0).

In addition, I performed traffic simulations of these four real scenarios. By comparing the predicted results and the simulated results, I can evaluate the effectiveness of the developed model. Table 2.7 lists and compares the predicted and simulated results for the real scenarios. In Table 2.7, I presented the six macroscopic MOEs that are the means (i.e., averages) and deviations (i.e., standard deviations) of the speed, V/C, and delay for the AM, MD, PM, and NT periods. The results show that, on average, the difference between the predicted results (i.e., means and deviations of the speed and V/C) and the simulated results were around 6%. Considering the timesaving advantage of the proposed model over the simulation models (e.g., couples of seconds compared to two days), the results are satisfactory especially for pre-screening purposes.

However, the model failed to predict the mean and standard deviation of the delay. To analyze the reason, I investigated the VISUM delay calculation method. To calculate the delay, the VISUM software uses a function which expresses the travel times on a road as a function of traffic volume (aka volume-delay function, VDF) [163]. This function calculates the delay as the difference between travel time for current condition and free-flow condition. The VISUM software uses the BPR (Bureau of Public Roads) function which is a widely used volume-delay function as follows [164]:

$$t_{current} = t_{freeflow}.[1 + 0.75\ (V/C)^4]$$

However, the BPR function has some drawbacks [165]. For example, it is very easy to get large delay, if the V/C is close to or over 1. On the other hand, for the links that are used far under their capacity, the BPR functions yield always free flow times. These drawbacks have led to unreliability in simulation results for delay that were used to develop the proposed model. Therefore, the model failed to predict the mean and standard deviation of the delay. However, I did not remove the delay results to highlight the importance of training-data accuracy.

Table 2.7. Comparison of Simulated and Predicted Results for Real Scenarios

Time of the day ↓	***Speed***				***V/C***				***Delay***			
	Mean		**Deviation**		**Mean**		**Deviation**		**Mean**		**Deviation**	
	Pred	**Sim**	**Pred**	**Sim**	**Pred**	**Sim**	**Pred**	**Sim**	**Pred**	**Sim**	**Pred**	**Sim**
AM	30.3	30.2	9.6	9.7	51.4	53.6	31.5	32.2	2.0	3.3	336	13.6
MD	30.9	31.3	9.3	9.1	48.7	47.5	30.7	30.9	9.4	1.8	1752	9.2
PM	30.3	29.2	9.5	10.2	49.6	58.5	29.9	32.9	2.2	4.6	1351	18.9
NT	34.1	33.9	7.6	7.7	20.4	15.9	21.1	17.0	4.9	0.1	1132	1.8

Notes: **Pred**: *Prediction Results,* **Sim**:*Simulation Results*

I plotted the simulated results of the four real scenarios (aka test scenarios) on the same graphs of the predicted results (aka training scenarios) which were captured by the CCA method (Figure 2.7).

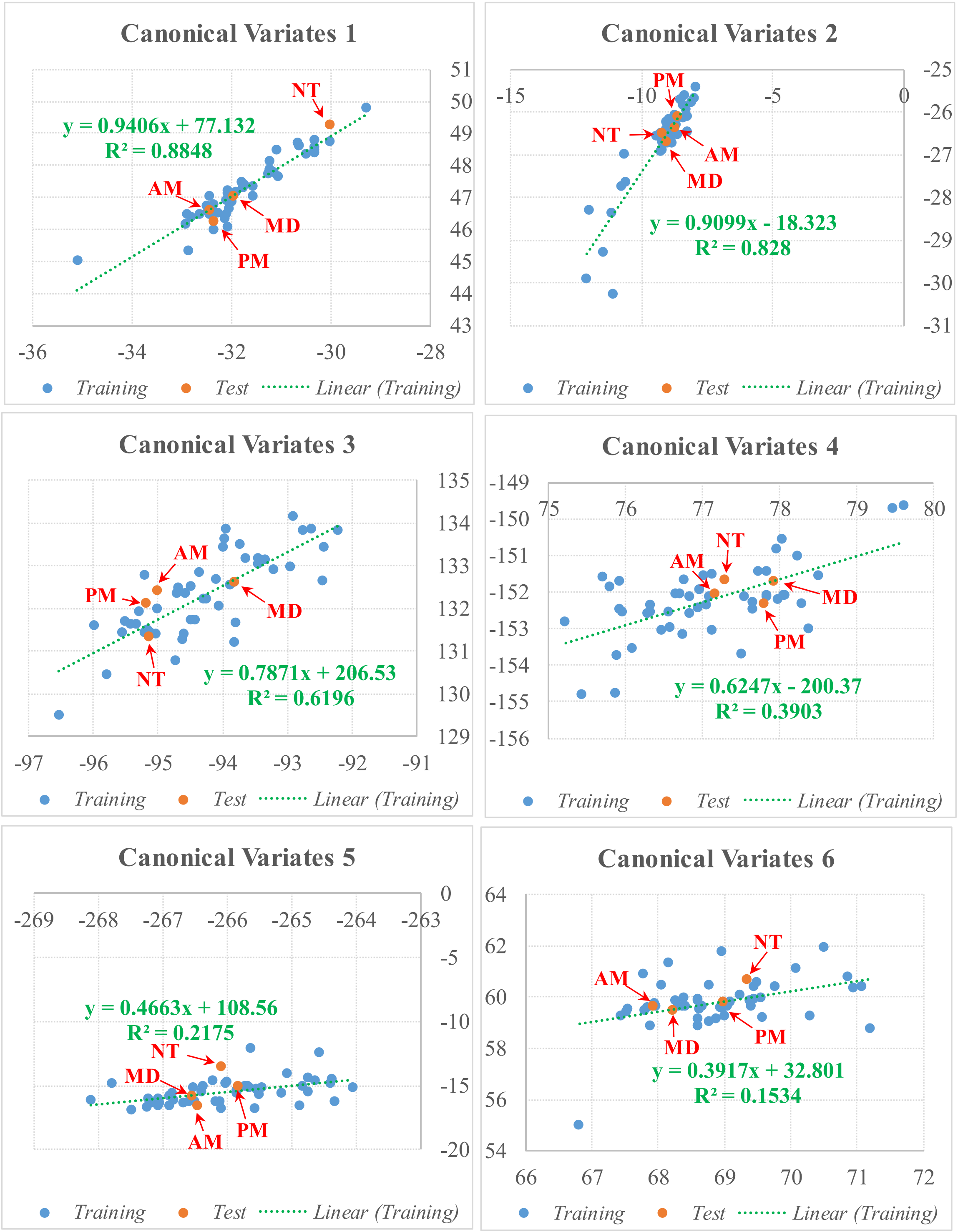


Figure 2.7. Simulated Results for Real Scenarios (i.e., test) Plotted on the Captured Canonical Variates

The closer the simulated results to the trend lines, the better the prediction performance. As shown in Figure 2.7, for the AM and MD periods, the simulated results were close to the six trend lines. This means that the model was effective in predicting the MOEs for the AM and MD periods. The Table 2.7 also verifies these results for AM and MD periods. However, for PM and NT, in some cases the simulated results were not close to the trend lines. So, the model failed to predict some of the MOEs during these periods (e.g., average V/C during the PM period).

To conclude, currently, transportation planners have to perform many time-consuming traffic simulation runs for numerous different design alternatives. In general, evaluating such alternatives require a significant amount of time and computational resources. However, for pre-screening process of these numerous alternatives, a macroscopic analysis is sufficient. Therefore, the proposed computational framework could help planners predict average MOEs without performing exhaustive simulations resulting into significant timesaving. After selection of a few alternatives, detail analysis is necessary via simulation runs to accurately assess traffic conditions.

In the future, it is recommended to assimilate a diverse database of road networks to improve the prediction model and to investigate whether this model is scalable to other network types as well. In addition, after building the diverse database, it is recommended to consider other network attributes including: structural attributes (e.g., network diameter, and form factor) and dynamical attributes (e.g., spatial distributions of traffic origins and destinations). This would also lead to a better representation of road structure and traffic demand which may improve reliable performance prediction.

CHAPTER 3. MODELING NETWORK VULNERABILITY DUE TO CRITICAL DISRUPTIONS

3.1 Introduction

Civil infrastructure systems, such as transportation systems and electrical grids, are of utmost importance to our modern societies. These systems play a fundamental role in operation and development of various aspects of our societies, such as economy, safety and security. Generally, infrastructure systems can be defined as a set of interconnected elements that, as a whole, provide critical supports. Therefore, vulnerability assessment of such systems (i.e., networks) to large-scale collapse is important [6, 31, 49, 52-56, 166]. Small local failure may propagate through these networks causing a large-scale breakdown, which is termed as a macroscopic avalanche [27, 49, 167-169].

Among different transportation networks (e.g., roads, railways, airways, and waterways), road network is the most widely used infrastructure system. Hence, any disruption that leads to a failure of single or multiple nodes or links in a road network may significantly degrade its performance by decreasing an average speed or increasing an average delay [43, 170, 171]. The cause of such disruptions can originate either within a network (e.g., car crashes, or bridge collapses) leading to blockage of a single link/node, or from external sources (e.g., floods, landslides, snowfall, storms, earthquakes or other natural hazards) leading to partial/complete blockage of multiple links at a time. An example of multiple-links failures is area-covering disruption which is defined as the disruption that degrades a substantial portion of a road network within an affected area [6]. Berdica (2002) defined '*vulnerability*' of a road network as its susceptibility to events that may result in considerable reduction in serviceability [1]. It is important to assess the vulnerability of a road network in response to both single-link blockage and area-covering disruptions [6].

Current research studies have focused on the impacts of failure of single links and ranked them based on their impacts [172, 173]. In the case of failure of multiple links, the researchers considered the top-ranked subset of single links as the most important subset of links [6]. Recently, the researchers also studied the impacts of area-covering disruptions which disrupt an area with a pre-defined shape (e.g., square) and size [6]. However, they did not investigate the impacts of disruption on a large area without any prior assumptions about the shape and size of the affected area. In reality, natural disruptions, such as flood, heavy snow, and hurricane, may simultaneously impact multiple links in an area with arbitrary size and shape. Hence, instead of finding a critical set of links, an interesting question might be to find the critical area(s) which are defined as the areas within a network whose disruption will significantly impact the connectivity of a network compared to the disruption of non-critical areas.

3.2 Background Research

3.2.1 Background Research on Road Disruption Analysis

Different criteria are proposed to evaluate the impacts of disruptions on network performance [28, 53, 56, 174-178]. They studied disruptions based on their causes, types and severity. The cause refers to the origins of disruption, and can be grouped into two categories: (a) internal and (b) external causes [6]. Car crashes, random technical failure such as bridge collapse, and incidents due to road works, are labeled as internal causes, while natural (e.g., flood, storms, hurricanes) and anthropogenic (e.g., terrorist attack) hazards are external causes. The disruptions due to internal causes typically lead to capacity reduction (or blockage) of a single link [6]. Thus, these research studies [28, 56, 169, 174-176] investigated single-link failures.

In contrast, the disruptions caused by nature may extend to large areas, and may impact multiple links [6]. In general, there are two types of multiple-link disruptions: (1) disruption of multiple links spatially scattered across a network, and (2) disruption of multiple links in a specific area within a network (aka area-covering disruption), which is typical in a transportation network. Jenelius and Mattsson (2012) studied area-covering disruptions that degrade a substantial portion of a road network within an affected area. To model an area-covering disruption, they modeled a road network by a grid of uniformly shaped and sized cells. Each cell represented a spatial coverage of a possible disrupting event. They investigated different square cell sizes to analyze the sensitivity and accuracy of their models. They concluded that the area affected by an area-covering disruption is different than that of a single link disruption [6]. However, in their models, the grid shape and cell size limit the shape and extent of a disruption. The severity of a disruption refers to the capacity reduction of links due to an event. While ordinary events, such as partial flooding and minor accidents, may partially reduce the capacity of a given link, catastrophic events such as earthquake, collapse of bridges and major accidents may completely reduce the capacity [172].

Criticality criteria

Generally, the links or nodes with higher traffic volumes are considered as critical (aka important) links or nodes. Two standard measures of criticality (also known as importance) are: (1) average annual daily traffic (AADT) and (2) volume over capacity ratio (V/C) [179]. The disadvantage of these measures is that they are localized measures that do not consider network-wide impacts of link failure. Network-wide impacts are the changes in traffic conditions over an entire network due to failure of an individual node or link. Sullivan et al. (2010) discussed that the critical links are not necessarily the links with highest traffic volume, but are links with relatively higher volume and fewer alternate routes. This conclusion highlights the importance of network structure, which is defined as the arrangement of network's various components, such as nodes and links.

The structural characteristics have been considered as the basis of several criticality criteria. The examples include: (1) the change in the shortest paths between all pairs of nodes after disruption [180], (2) gamma index of connectivity [172], and (3) Latora-Marchiori measure, which defines the efficiency of a network based on its topology [181]. These criteria are only based on network topology, and traffic flow dynamics has not been considered. Because of the spillback impacts of congestion, Knoop et al. (2008) discussed that it is important to define the criteria based on both the topology and traffic flow dynamics [176]. The examples of these criteria include: (1) Network Robustness Index (NRI) which captures an increase in delay [6, 172], (2) change in the total cost (i.e., distance, time or money) of travel [174], and (3) change in network accessibility defined based on distance, population, and traffic volume [175].

Existing studies has two major limitations. First, the current methods did not consider network-wide impacts of disruption, which are the impacts of disruption on overall network connectivity. So, any new criteria should be identified to consider the impacts of disruptions on network connectivity. Second, the existing methods for an area-covering disruption need prior assumptions about the size and shape of grid cells. These assumptions limit the occurrence of a disruption on squared-shape grid cell of a network, which may not be a realistic assumption. So, any new approach is needed to analyze disruptions which may cover multiple areas without pre-defined shape and size. It is also important to investigate whether the areas' criticality depends on the occurrence time of disruption (e.g., AM, MD, PM, or NT periods).

3.2.2 Background Research on Network Science

In general, there is a large body of research related to different types of networks [3, 11]. Boccaletti et al. (2002) reviewed the concepts and results achieved in the study of the structure and dynamics of several networks, such as social network, the Internet, World Wide Web, genetic networks, and brain networks. Also, Albert et al. (2002) discussed the main categories of networks including random graphs, small-world and scale-free networks (discussed in Chapter 2). They also studied the interrelation between topology and dynamics due to attack and failure of network nodes or links.

Application of network science in civil engineering

The structure and dynamics of network infrastructures (e.g., transportation networks) have been studied from the perspective of network science [3, 11, 52, 182, 183]. The structure defines the topology of a network including links and nodes. Chan et al. (2011) studied the road networks of 20 largest cities in Germany for the year 2005, and proposed a variety of measures based on the structural and spatial characteristics of the road networks. The examples of structural measures include: number of nodes and links, node's degree and betweenness, clustering coefficient (i.e., a measure of tightness and density of links), and link's length probability distribution. Their results, in addition to recent empirical studies [37, 132, 133], have shown that there exist topological quantitative similarities between road networks of different cities.

In addition, the dynamics of a network defines what processes are happening on the network [4]. In the case of road networks, dynamics refers to traffic flow. Different approaches are proposed to identify traffic flow dynamics based on: household interviews [151], records of road cameras and loop detectors [152], GPS data of the location traces of probe vehicles [153], and mobile phone data which can be used wherever the geographical locations of the origins of phone calls and texts are recorded [29, 184, 185]. For instance, Gonzales et al. (2008) studied two different datasets of mobile phone users from two different areas to investigate human mobility patterns. Also, Wang et al. (2012) considered the temporal variations of the OD matrices over a day. Based on the observed distribution of daily traffic, they divided a day into four periods: (1) morning: 6 am–10 am, (2) noon & afternoon: 10 am–4 pm, (3) evening: 4 pm–8 pm, and (4) night: 8 pm–6 am. Then, for each period, they estimated the distribution of travel demands. Therefore, the spatiotemporal travel pattern over a day was presented by four different OD matrices representing morning, noon & afternoon, evening, and night [29].

Furthermore, the interrelation between structure and dynamics was highlighted in many research studies [11, 49, 50, 169]. For example, the author studied the impacts of network structure (e.g., largest eigenvalue) on the propagation of traffic congestion [50]. Researchers studied the vulnerability of infrastructures to large-scale collapse in modern societies [49, 186]. Winkler et al. (2010) combined hurricane damage prediction and topological assessment of power systems to evaluate the impacts of hurricane on network vulnerability. They concluded that the network vulnerability correlates directly with topological features, such as centrality and clustering [186]. Duenas-Osorio and Vemuru (2009) considered the cascading failure of infrastructures. They concluded that regardless of the nature of the event, the additional performance loss due to cascading failures can be significantly larger than the initial loss [49]. Their conclusion indicates the importance of considering both structure and dynamics in vulnerability assessment of infrastructures.

3.3 Proposed Approach

This section discusses the research objective, approach and validation strategies for modeling transportation network vulnerability in response to an area-covering disruption.

3.3.1 Objective

The main objective of the research is to formalize and develop a framework that enables transportation modelers to identify various critical areas on a given road network. To overcome the limitations of existing studies in network-wide vulnerability assessment, the proposed framework builds upon network-science theories. Specifically, I employed a method from epidemiology [15] to define a network-wide measure of connectivity. In this method, the network connectivity is defined based on the largest eigenvalue of the network [15]. In addition, I proposed an approach based on community detection method to identify critical areas [187]. Thus, the important components of the framework are: (1) eigenvalue-based measure of connectivity, and (2) community detection methods for clustering.

3.3.2 Framework

The overall proposed approach is presented in Figure 3.1. In the first stage, the criticality of individual links is determined (discussed in section *Phase 1: Identifying links' criticality*). Then, in the second stage, a community detection (aka modularity optimization) algorithm is used to cluster links based on their different levels of criticality (discussed in *Phase 2: Clustering links*). The output is a collection of different clusters throughout the network representing different levels of criticality.

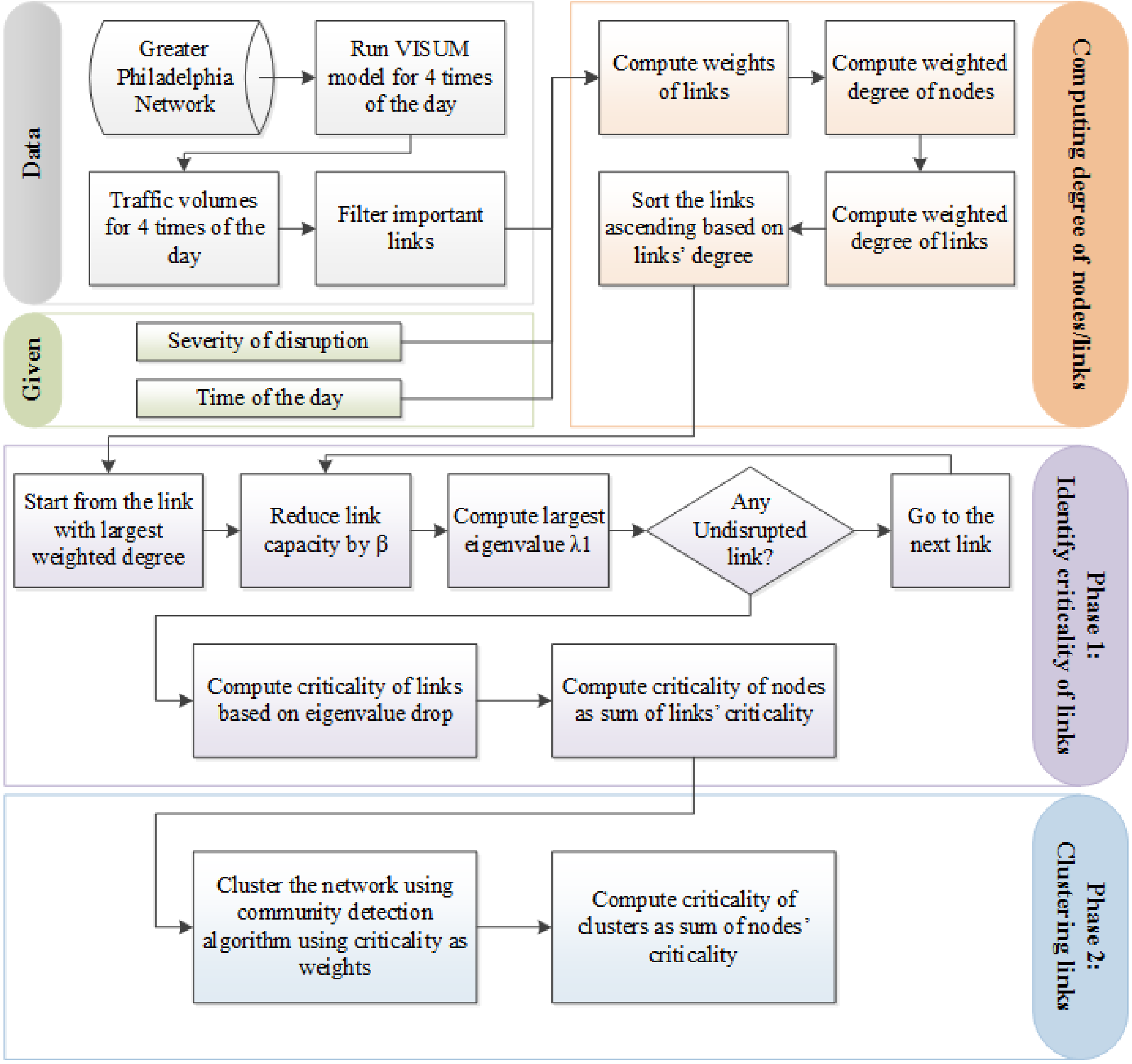

Data
Greater Philadelphia Network
Run VISUM model for 4 times of the day
Traffic volumes for 4 times of the day
Filter important links
Given
Severity of disruption
Time of the day
Compute weights of links
Compute weighted degree of nodes
Compute weighted degree of links
Sort the links ascending based on links' degree
Computing degree of nodes/links
Start from the link with largest weighted degree
Reduce link capacity by β
Compute largest eigenvalue λ1
Any Undisrupted link?
Go to the next link
Compute criticality of links based on eigenvalue drop
Compute criticality of nodes as sum of links' criticality
Phase 1: Identify criticality of links
Cluster the network using community detection algorithm using criticality as weights
Compute criticality of clusters as sum of nodes' criticality
Phase 2: Clustering links

Figure 3.1. Overview of the Proposed Framework

<u>Phase 1: Identifying links' criticality</u>

In this research, the algorithm identifies the link's criticality based on the drop in network connectivity due to the link's disruption. To do so, it defines a measure for the network connectivity (discussed in Phase1.1). Then, three different approaches are discussed to calculate the drop in the network connectivity measure and the one which is computationally efficient is selected (discussed in Phase1.2). Finally, the algorithm identifies the link's criticality based on such a drop (discussed in Phase1.3).

Phase 1.1. Define network connectivity measure

I propose the largest eigenvalue of an adjacency matrix as a connectivity measure.[14, 15]. There are several ways to define the adjacency matrix: (1) un-weighted adjacency, (2) weighted adjacency where the weights are structural measures, such as distance or capacity, (3) weighted adjacency where the weights are dynamical measures, such as traffic volume, and (4) weighted adjacency where the weights are a combination of structural and dynamical measures. I choose the latter one (i.e., combination of structural and dynamical measures) because both structure and dynamics play important roles in network performance [11]. To do so, I select “capacity of links” as their structural measure and “traffic volume” as their dynamical measure.

To define the combined weights, I needed to investigate the relationship between network connectivity and the selected measures (i.e., capacity and traffic volume). Figure 3.2(a) shows a simple example of a weighted graph, in which the links’ weights represent their capacities. Figure 3.2(b) shows the same graph, but the weight of the link with highest capacity (i.e., link 1-4) is reduced by 50%. In contrast, Figure 3.2(c) shows the graph, in which the weight of the link with lowest capacity (i.e., link 1-2) is reduced by 50%. Figure 3.2 shows the computed largest eigenvalues of adjacency matrices (λ1) for the three scenarios. The results indicate that when the highest-capacity link is disrupted, the λ1 is impacted more than the case in which the lowest-capacity link is disrupted. Therefore, there exists a direct (i.e., not inverse) relation between the capacity and λ1 which is the measure of network connectivity.

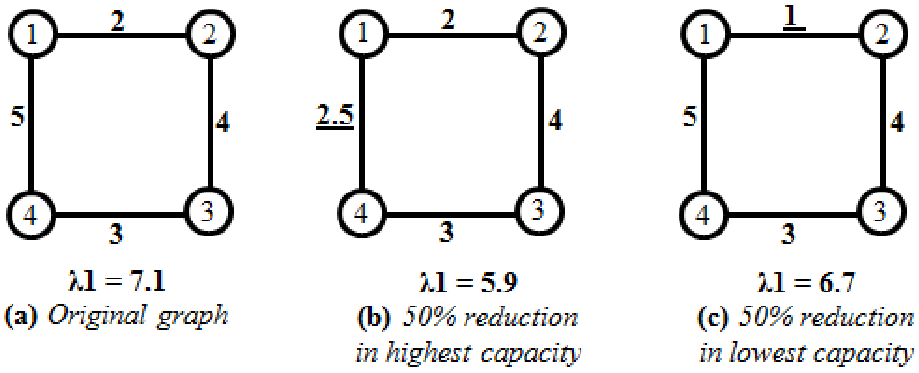

1
2
3
4
2
5
4
3
λ1 = 7.1
(a) Original graph
2.5
λ1 = 5.9
(b) 50% reduction in highest capacity
1
λ1 = 6.7
(c) 50% reduction in lowest capacity

Figure 3.2. Change in Capacities of a Small Weighted Graph

The same logic is valid for the relation between traffic volume and network connectivity. In other words, it is more critical if the links with larger traffic volumes are disrupted. Therefore, in summary, the criticality (i.e., reduction in network connectivity) has a direct (i.e., not inverse) relation with both the capacity and traffic volume. Hence, for a weighted network, I defined the weight of a link as:

$$w_l = a.C_l + b.V_l$$

where w_l is the weight of the link, C_l is the capacity of the link, and V_l is the traffic volume on the link. I assumed that a and b are both equal to ${}^1\!/_2$, i.e., the weight is the arithmetic mean of capacity and traffic volume. The traffic volumes vary based on the time of a day. Therefore, the weights of the links are also different for different times of a day.

Phase 1.2. Calculate the drop in network connectivity measure

Since the criticality of a link depends on the impact of a disruption on network connectivity, it is reasonable to calculate such criticality based on the drop in the largest eigenvalue when a given node/link is disrupted. To identify the criticality of multiple links in a given area, one has to consider the impacts of their simultaneous disruption. Prakash et al. (2013) proposed and validated an approach called Exhaustive approach (shown in

Figure 3.3) which greedily tries to find the links whose disruption minimizes the largest eigenvalue. As shown in Figure 3.3, the Exhaustive approach reduces the capacity of one link and computes the drop in largest eigenvalue due to this disruption. This process is repeated for the entire links to find the link associated with the largest drop in λ_{1A}. Once such link is found, its capacity is reduced and the algorithm searches for the next critical link, until the point that the largest eigenvalue reaches its minimum level.

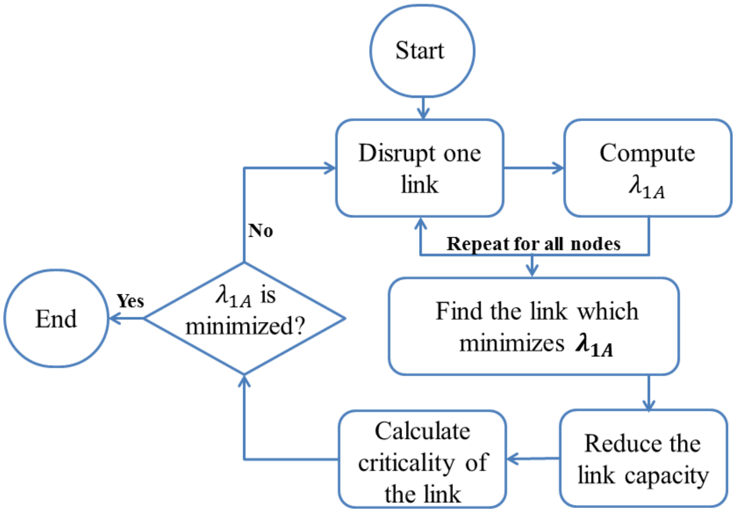
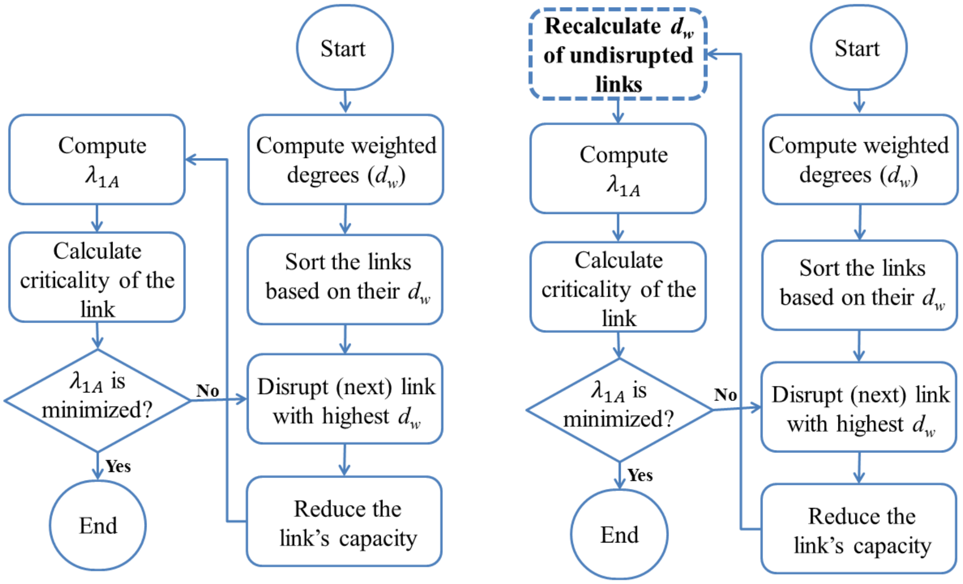

Start
Disrupt one link
Compute λ_{1A}
Repeat for all nodes
Find the link which minimizes λ_{1A}
Reduce the link capacity
Calculate criticality of the link
λ_{1A} is minimized?
No
Yes
End

Figure 3.3. Flowchart for the Exhaustive Approach

I tested the Exhaustive approach on the network of Guam Island, which has 539 nodes and 1183 links. It took 180 minutes to sort the 1183 links of Guam network in terms of their criticality. Therefore, for large networks (e.g., networks with 100,000 links) it is reasonable to develop faster alternatives compared to the Exhaustive approach. Hence, I proposed two approaches based on the weighted degrees (i.e., strength) of links. I defined the weighted degree of a link as the average of weighted degrees of its two attached nodes. For example, if the weighted degrees of the attached nodes of a link are 650 and 350, the weighted degree of the link is 500 (i.e., $650 + 350/2$). Figure 3.4(a) shows the first proposed approach (aka weighted-degree approach). In the first approach, the algorithm sorts the links based on their weighted degrees and then disrupts links sequentially in a descending order of their degrees. At each step, once the link’s weight is reduced (based on a given severity level, for example, 60%), I compute the largest eigenvalue of the new weighted network. Then, criticality of the link is defined as the drop in the largest eigenvalue due to the disruption of the link.

Figure 3.4(b) shows the second proposed approach (aka updating-degree approach). In this approach, the algorithm sorts the links based on their weighted degrees and then disrupts links sequentially. However, at each step after disrupting a link, the weighted degrees of all links are recalculated and the undisrupted links are re-sorted in a descending order of their degrees.


Start
Compute weighted degrees (d_w)
Sort the links based on their d_w
Disrupt (next) link with highest d_w
Reduce the link’s capacity
Compute λ_{1A}
Calculate criticality of the link
λ_{1A} is minimized?
No
Yes
End
Recalculate d_w of undisrupted links
Start
Compute weighted degrees (d_w)
Sort the links based on their d_w
Disrupt (next) link with highest d_w
Reduce the link’s capacity
Compute λ_{1A}
Calculate criticality of the link
λ_{1A} is minimized?
No
Yes
End

(a) weighted-degree approach **(b) updating-degree approach**

Figure 3.4. Flowcharts for the Two Proposed Approaches

I also tested weighted-degree and updating-degree approaches on the network of Guam. Figure 3.5 shows the drop in largest eigenvalue based on the size of disruption. The *y*-axis shows the largest eigenvalue of an adjacency matrix, and the *x-axis* shows different percentages of disruption. For example, the value 0.2 on the *x*-axis means the disruption of 20% of the entire links. As shown in Figure 3.5, if we disrupt 20% disruption of links using the Exhaustive approach, the value of the largest eigenvalue is minimized. This value (i.e., disruption percentages) is around 22% for the updating-degree approach and 25% for the weighted-degree approach. The differences among these values (20%, 22%, and 25%) were not significant.

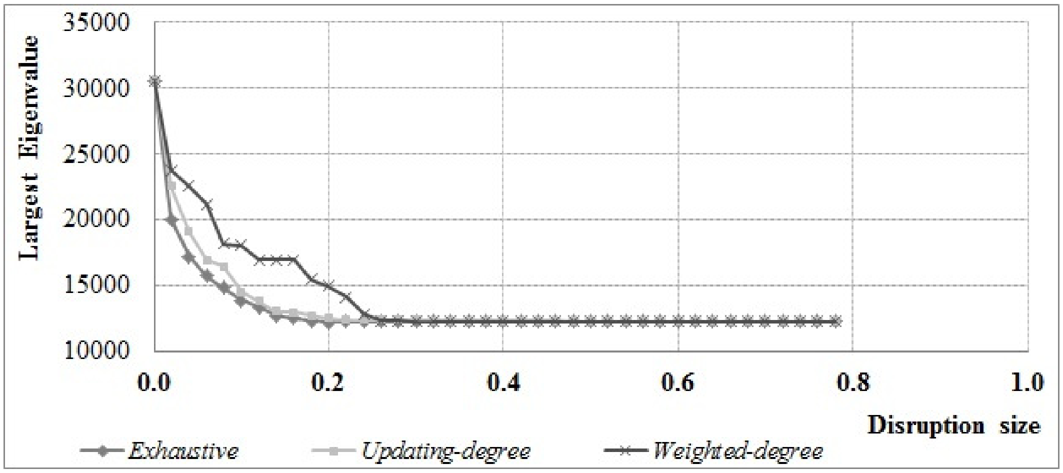


Figure 3.5. Three Methods for Finding Critical Link of Guam Island Network

Table 3.1 lists the computational complexity (i.e., running time) of the three different approaches. The main processes of the approaches are: (1) eigenvalue computation, (2) sorting, and (3) weighted degree computation. In general, the computational complexities of the eigenvalue computation, sorting, and weighted degree computation are O(n^3), O(n log n), and O(n), where n is the number of links in a network [188]. Table 3.1 shows the number of steps for each process which is defined in terms of n. The results indicate that the weighted-degree approach was computationally more efficient and faster than the updating-degree and Exhaustive approaches. Also, on my machine, the weighted-degree approach took 2 minutes to reach the minimum value, while the updating-degree and Exhaustive approaches took 30 and 180 minutes. Hence, in this research, I selected the weighted-degree approach for large networks, such as the Greater Philadelphia network.

Table 3.1. Computational Complexity (i.e., running time) of three Different Approaches

Process	Exhaustive	Updating-degree	Weighted-degree
Eigenvalue computation, $O(n^3)$	n^2	n	n
Sorting, O(n log n)	-	n	1
Weighted degree computation, O(n)	-	n	n
Total	$O(n^5)$	$O(n^4+n^2logn+n^2)$	$O(n^4+nlogn+n^2)$

Phase 1.3. Identify link's criticality based on the calculated drop

The next step is to identify the link's criticality measure based on the drop in the network connectivity measure (i.e., the largest eigenvalue). In Figure 3.6, the links that are earlier selected to be disrupted are more critical. For instance, the link l_a is more critical than the link l_b. Additionally, once the minimum level of the largest eigenvalue ($\lambda1_{min}$ in Figure 3.6) is reached, disrupting the remained links does not impact the largest eigenvalue. Therefore, I define criticality of the link l_i as follows:

$$Cr_i = \lambda1_i - \lambda1_{min}$$

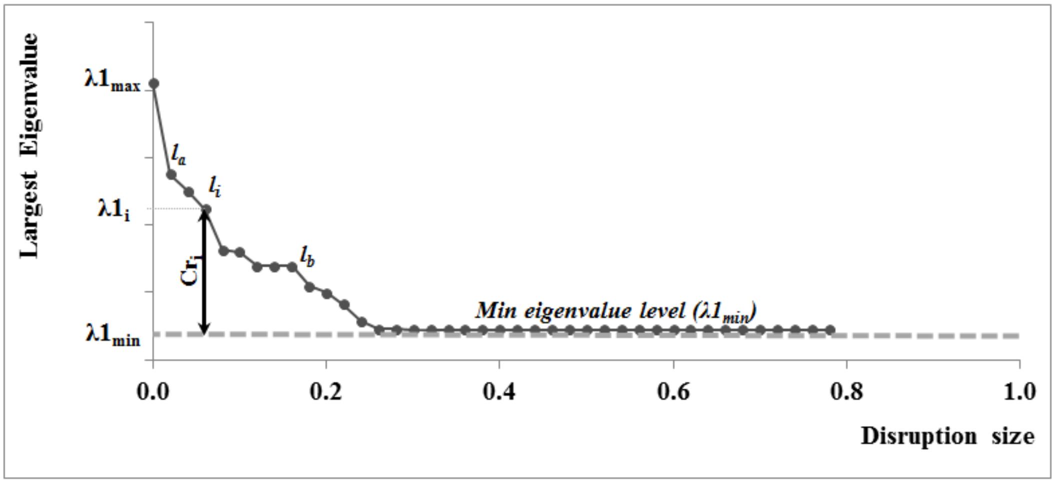


Figure 3.6. Defining Criticality of Links

Where Cr_i (shown on Figure 3.6) is the criticality of the link l_i, $\lambda 1_i$ is the largest eigenvalue after disrupting l_i, and $\lambda 1_{min}$ is the minimum level of the largest eigenvalue. It is noteworthy that given a severity of disruption (i.e., β which is the % of capacity reduction), the minimum largest eigenvalue ($\lambda 1_{min}$) for a network can be mathematically driven based on $\lambda 1_{max}$.

Proof: If the capacities of entire links are reduced by β, then:

$$A' = (1 - \beta). A$$

Where A' is the adjacency matrix after disrupting all the links and A is the adjacency matrix before disruption. Therefore, if I define $\lambda 1_{min}$ as the largest eigenvalue of A' and $\lambda 1_{max}$ as the largest eigenvalue of A:

$$\lambda 1_{min}. \vec{e} = A'. \vec{e} = (1 - \beta). A. \vec{e} = (1 - \beta). \lambda 1_{max}. \vec{e}$$

$$\xrightarrow{yields} \lambda 1_{min} = (1 - \beta). \lambda 1_{max}$$

This means that the minimum largest eigenvalue ($\lambda 1_{min}$) is a constant multiple of the initial largest eigenvalue ($\lambda 1_{max}$) shown in Figure 3.6.

Phase 2: Clustering links

The output of the first phase is a list of criticalities of all individual links. The next step is to segment the network into different clusters with different levels of criticality. The objective is to identify critical areas of the network for developing pre- and post-disaster recovery strategies. The three steps of this phase are discussed next.

Phase 2.1. Defining a weighted network

In the first step, since the objective is to find the critical clusters, the weights of the link should represent their criticality. Therefore, I define the weighted network in which the weights of the links are the criticalities of the links (calculated in the first phase).

Phase 2.2. Segmenting the network into clusters

In the second step, a community detection method is employed to detect communities (i.e., clusters) of the weighted network. Since the assigned weights are the criticality of the links, the method tends to cluster critical links together by considering their criticality, connection and geographical closeness. The advantage of this approach is that I do not limit the shape, the size, and the number of clusters.

Community detection methods

A network can be clustered into several communities if the nodes of the network can be grouped into different sets of nodes such that each set is internally connected [189]. The intuitive definition of community states that there are more links inside a community than links connecting nodes of the community with the rest of the network. Researchers have attempted to solve community detection problem with different methods [189]. One group of traditional methods are graph partitioning methods in which the nodes of a network are divided into a given number of groups of predefined size. So, the number and the size of clusters are necessary inputs of graph partitioning methods [189].

Another group of methods are modularity optimization methods in which the number and the size of clusters are automatically determined [187]. Modularity is a quality function which quantifies the difference between the number of links falling between groups minus the expected number of links in a random network with the same number of nodes and links. Positive values of modularity indicate the possible presence of community structure. Therefore, one can look for the division of a network corresponding to the large (positive) values of modularity [187]. One of the popular quality function of modularity is the Newman and Girvan modularity, which is defined as [189, 190]:

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i.k_j}{2m}).\delta(C_i, C_j)$$

Where Q is the modularity, m is the total number of links, A is the adjacency matrix, k_i is the degree of node i, and $\delta(C_i,\ C_j)$ yields one if the nodes i and j are in the same cluster, zero otherwise.

To divide a given network into two groups, the modularity can be optimized by using the eigenvalues and eigenvectors of the modularity matrix B, in which [187, 189]:

$$B_{ij} = A_{ij} - \frac{k_i.k_j}{2m}$$

One can define $\boldsymbol{s}$ as the vector that partitions the graph into two clusters, $\mathcal{A}$ and $\mathcal{B}$: $s_i = +1$ if the node i belongs to $\mathcal{A}$, and $s_i = -1$ if i belongs to $\mathcal{B}$. So, the modularity can be written as [187, 189]:

$$Q = \frac{1}{4m}\sum_{ij}(A_{ij} - \frac{k_i.k_j}{2m}).(s_i s_j + 1)$$

$$Q = \frac{1}{4m}\sum_{ij} B_{ij} s_i s_j = \frac{1}{4m} s^T B\, s$$

One can decompose $\boldsymbol{s}$ on the basis of the eigenvectors u_i $(i = 1, \ldots, n)$ of modularity matrix B as: $s = \sum_i a_i u_i$, with $a_i = u_i^T.s$. Therefore, the Newman and Girvan modularity can be rewritten as:

$$Q = \frac{1}{4m}\sum_i a_i u_i^T B \sum_j a_j u_j = \frac{1}{4m}\sum_{i=1}^{n} (u_i^T.s)^2 \beta_i$$

where β_i is the eigenvalue of B corresponding to the eigenvector u_i. To optimize the modularity, Newman (2006) suggested looking for the eigenvector of B with largest (positive) eigenvalue, u_1, and partition the nodes into two groups according to the signs of the components of u_1: the nodes with positive components are in one group, the others in the other group. However, if there is no positive eigenvalue, the network is indivisible [187].

Newman (2006) extended the abovementioned method for the cases that more than two communities exist in the network. The presented method divides the network into two groups, and repeats the subdivision until the time that the groups are indivisible.

Phase 2.3. Computing criticality of clusters

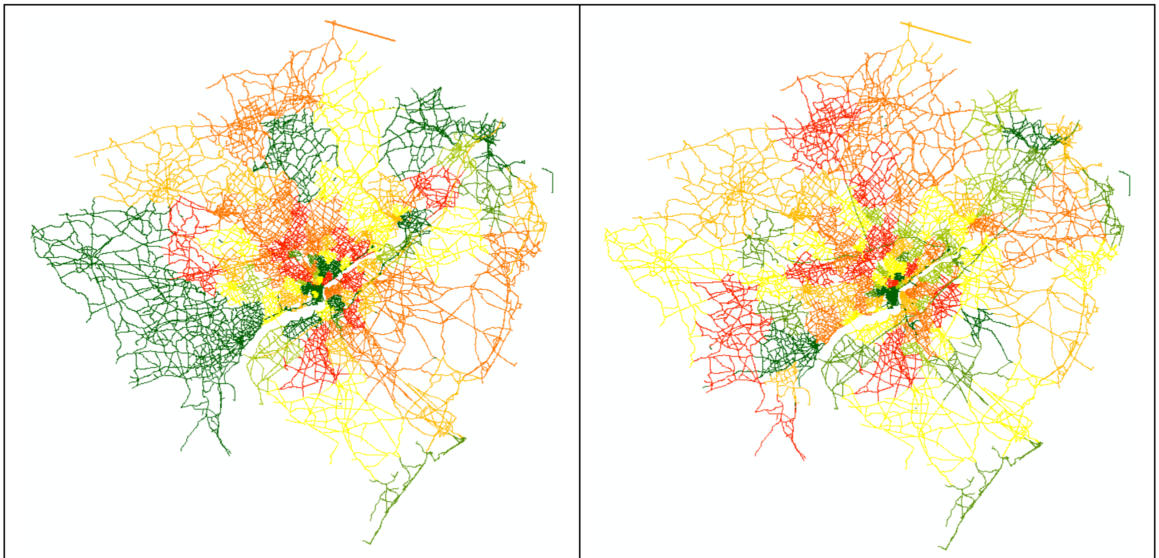
The output of the second step is a network which is segmented into multiple clusters. The criticality of each cluster is then defined as the summation of criticalities of its links. This is because if a cluster is disrupted, its entire links are impacted by the disruption. So, the impact of disrupting a cluster is equivalent to the simultaneous impacts of disrupting its links. Since the criticality of the links vary over the time of a day, the clustering results might be different for different times of a day.

3.4 Results

In this research, I present the results of the proposed framework for the Greater Philadelphia region for different times of a day. The Greater Philadelphia network and its calibrated traffic model which was developed by Delaware Valley Regional Planning Commission (DVRPC) are discussed in Chapter 2 (section 2.4). The model includes four Origin-Destination (OD) matrices for four different times of a day, defined as follows: AM Peak (6:00 AM to 10:00 PM), Midday (10:00 AM to 3:00 PM), PM Peak (3:00 PM to 7:00 PM), and Night Time (7:00 PM – 6:00 AM). The traffic volumes over the links for four different times of a day were obtained by running the VISUM simulation model using four different OD matrices. These resultant traffic volumes were used for computing links' weights for different times of a day.

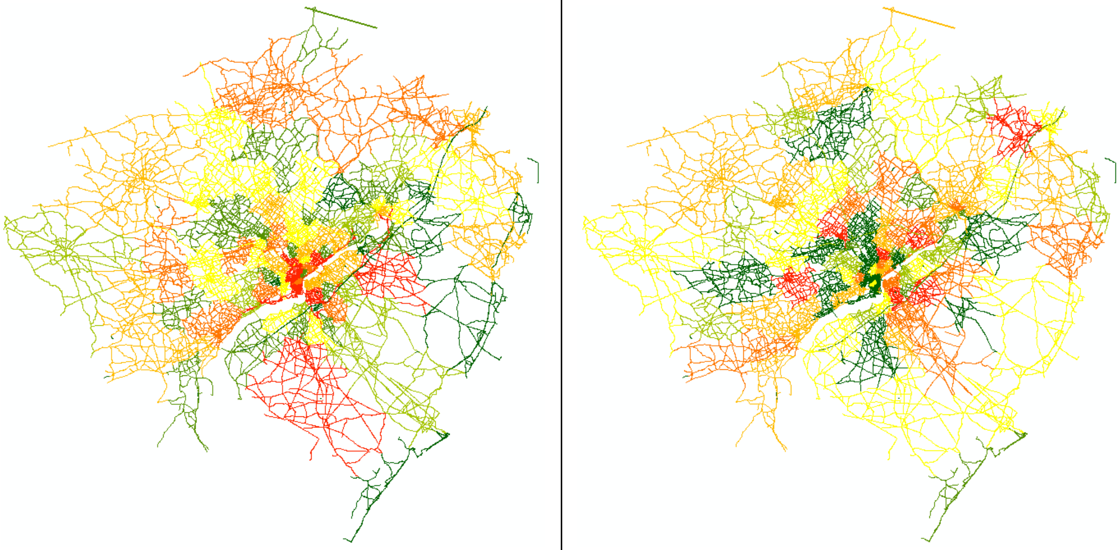
I employed iGraph which is a free software package for creating and manipulating different types of graphs [154]. It includes implementations for simple graph problems like node degree calculation, and network analysis methods, like community detection.

The inputs of the framework were: (1) road network data, (2) traffic volumes (for the given time of a day), and (3) the severity level of disruption. For any road network, the process of detecting critical clusters can be repeated for different times of a day and different severity levels of disruption. Figure 3.7 shows the results for 60% severity of disruption during the AM Peak, Midday, PM Peak, and Night Time. The plots show the differences in clustering the network into critical areas for four times of a day. The red clusters are more critical compared to the green ones.



Time: AM peak; Severity of disruption: 60%

Time: Mid day; Severity of disruption: 60%



Time: PM peak; Severity of disruption: 60%

Time: Night Time; Severity of disruption: 60%

Figure 3.7. The Results for 60% Severity of Disruption at Different Times of a Day

As shown on Figure 3.7, I observed the following findings:

1 - During the AM peak, disruption of the roads around the city of Philadelphia was more critical compared to the roads within the city of Philadelphia and the roads in outer areas of the region. This might be because, during the AM peak, many of the trips were

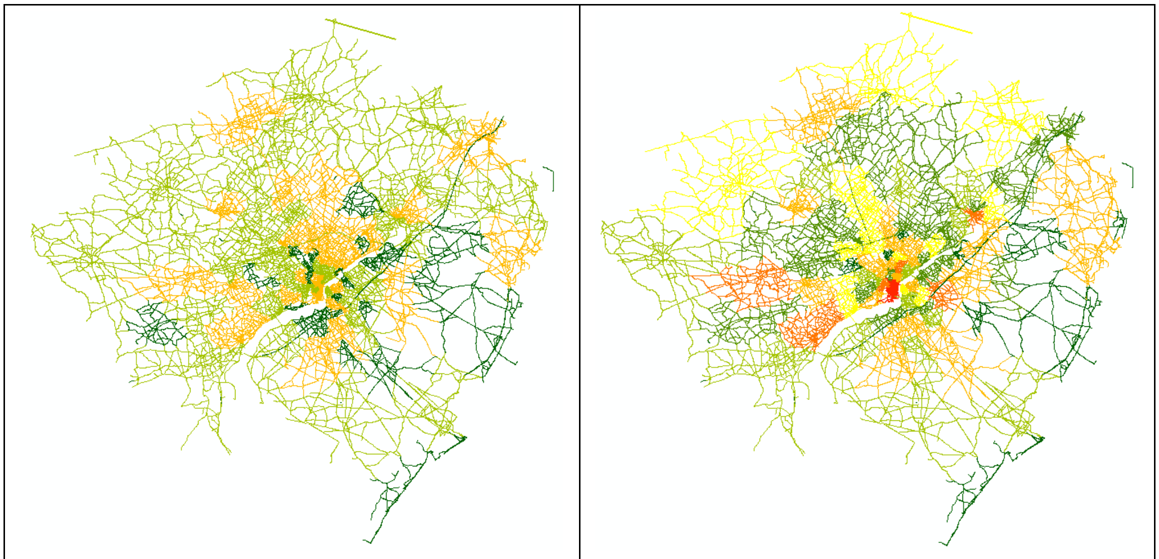
towards the city of Philadelphia and so the roads around the city acted as main arteries from the outer areas to the city.

2- During the PM peak, disruption of the roads within the city of Philadelphia was more critical than other areas. This might be because, during the PM peak, many of the trips began from this small area (i.e., city of Philadelphia) toward other areas of the region.

3- During the Midday (MD) and Night time (NT) periods, the critical clusters were not concentrated within any specific area of the network. This might be because, during the MD and NT periods, the trips were spatially scattered compared to the AM and PM peak periods.

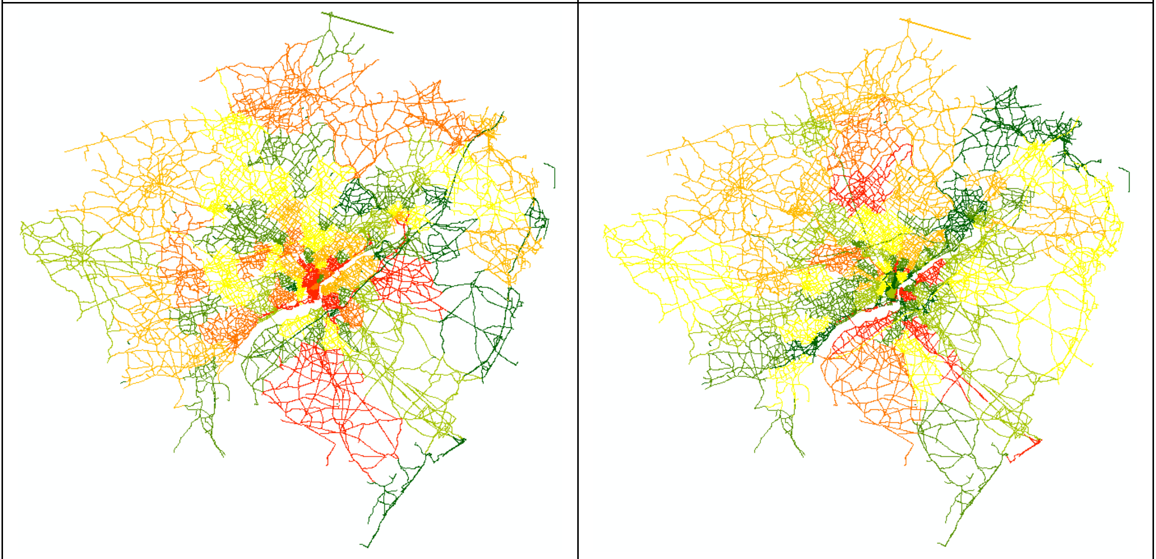
The results show that criticality of any given area varies for different time of a day. For example, a given area can be critical during the AM period while it is non-critical during the PM period. These observation, once validated, could be used in the future to identify pre- and post-disaster strategies to alleviate the negative impacts of disrupting critical areas of a network, considering the time of disruption.

In addition, the change in the severity level of disruption may also impact the clustering results. To evaluate such impacts, I investigated several scenarios for 10%, 30%, 60%, and 80% severity of disruption during PM Peak (Figure 3.8).



Time: PM peak; Severity of disruption: 10%

Time: PM peak; Severity of disruption: 30%



Time: PM peak; Severity of disruption: 60%

Time: PM peak; Severity of disruption: 80%

Figure 3.8. The Results for Different Severity of Disruptions during PM Peak Period

In Figure 3.8, the disruption of the red-colored clusters has severer impacts on overall network connectivity than the orange-, yellow-, and green-colored clusters. As Figure 3.8 shows, severe disruption (e.g., 80%) of a small area of the network significantly impacted the overall connectivity. However, in the case of slight disruptions (e.g., 10%), the impacts of disruption were not significant, even for the disruption covering a large area of the network.

3.5 Validation

In existing studies, researchers investigated their methods either on synthetic or real road networks. Researchers used different synthetic road networks to implement their methods and calculate the links' criticality [172, 181, 191]. For example, Nagurney and Qiang (2008) implemented their method on synthetic "Single Braess" and "Coupled Braess" networks [181]. Also, Scott et al. (2006) and Sullivan et al. (2010) used three synthetic networks to test their method [172, 191]. Since these networks were synthetic, no real data was available for validation. Instead, validating their methods, researchers compared their results with other existing results on the same networks.

In addition, vulnerability of several real road networks was investigated in previous research studies [28, 172, 175]. For example, Sohn (2006) examined the highway network in Maryland extracted from National Transportation Atlas Data (NTAD) [175]. Moreover, Erath et al. (2009) studied the vulnerability of Swiss national transport network containing 30,289 links and 24,316 nodes modeled in VISUM package [28]. Also, Sullivan et al. (2010) used road network of the Chittenden County Metropolitan Planning Organization (CCMPO), Vermont. The network includes 1,397 nodes and 1,791 links modeled in TransCAD software. Recently, Jenelius and Mattsson (2012) used Swedish road transport system, consisting of 32,759 nodes and 86,940 directed links modeled in EMME/2 package. In all these cases, validation was conducted by using simulated results.

To do a comprehensive validation based on real world data, several real disruption instances in multiple various areas are needed. For each disruption instance, traffic conditions (e.g., average speed, delay and volume) are needed for the time of disruption and also average traffic conditions for the time that there is no disruption. However, such data are not currently available. So, I employed two methods: (1) *simulation-based validation*, which compares the results of computational framework with the simulated results, and (2) *observation-based validation*, which observes the impacts of several real incidents across the network.

3.5.1 Simulation-based Validation

In this method, once I identified critical areas of the network, I reduced the links' capacities within three samples of high-critical, medium-critical and low-critical areas in separate settings. Then, I simulated traffic conditions using a regional model (e.g., VISUM software) to compute the MOEs, such as average speed, delay and V/C. By investigating the changes in these MOEs (compared to the cases with no disruption), one is able to ascertain if the criticalities of the areas identified by the proposed method are valid.

I used the proposed computational framework to cluster the Greater Philadelphia network and compute criticalities of all clusters during the PM peak for 60% severity of disruption. Among the clusters, I selected three sample clusters representing high, medium, and low criticality (Figure 3.9). For each sample cluster, I reduced capacities of the links within the cluster by 60%. Then, I simulated traffic conditions over the three adjusted networks of the Greater Philadelphia by running the VISUM models. Figure 3.9 shows the three selected clusters in which the red, orange, and green colors are the high-, medium-, and low-critical clusters. An advantage of this approach is that it does not limit the shape and size of clusters.

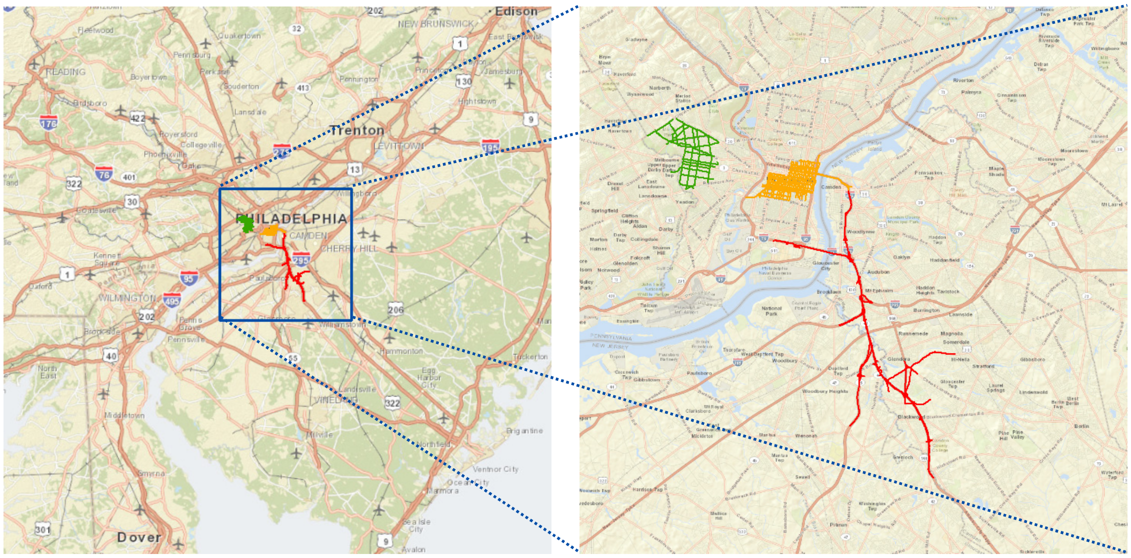
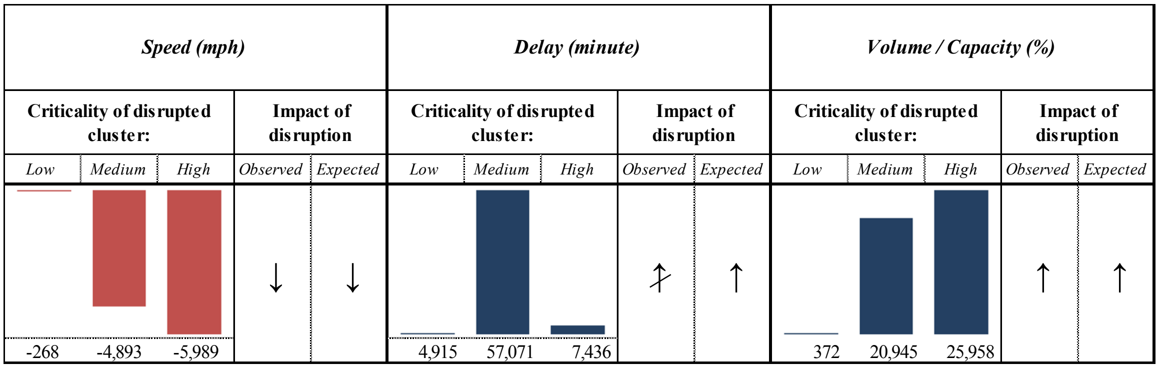


Figure 3.9. Three Selected Clusters: High-Critical (Red), Medium-Critical (Orange), and Low-Critical (Green)

Figure 3.10 presents the changes in traffic conditions of these three simulation runs, which represent the impacts of disrupting areas with high, medium, and low criticality. I calculated the changes in three measures, i.e., speed, delay, and V/C. To perform a network-wide comparison, I used the summation of the changes (in speed, delay, and V/C) over the entire network.



* Note: The values are calculated by summations over the entire network.

↓ : Decrease ↑ : Increase

Figure 3.10. Simulated Results for Validation

My hypotheses are that the more critical the disrupted area is, there is a higher probability of decrease in speed, increase in delay, and increase in V/C due to the disruption. Figure 3.10 shows that the observed simulated results were in line with these hypotheses, except for the delay due to disrupting the medium-critical cluster. To analyze the reason, I investigated the VISUM delay calculation method. To calculate the delay, the VISUM software uses a volume-delay function (VDF) which expresses the travel times on a road as a function of traffic volume [163]. This function calculates the delay as the difference between travel time for the current condition and the free-flow condition based on the V/C. The VISUM software uses the BPR (Bureau of Public Roads) function which is a widely used volume-delay function as follows:

$$t_{current} = t_{freeflow}. [1 + 0.75\ (V/C)^4]$$

However, the BPR function has some drawbacks [165]. For example, it is very easy to get large delay, if the V/C is close to or over 1. On the other hand, for the links that are used far under their capacity, the BPR functions yield always free flow times. These drawbacks of BPR function have led to some large values for delay as observed for the medium-critical cluster.

In addition, to statistically test these hypotheses, I selected a large area around these three clusters. To do so, I created a buffer zone of 5 miles around the three clusters as shown in Figure 3.11.

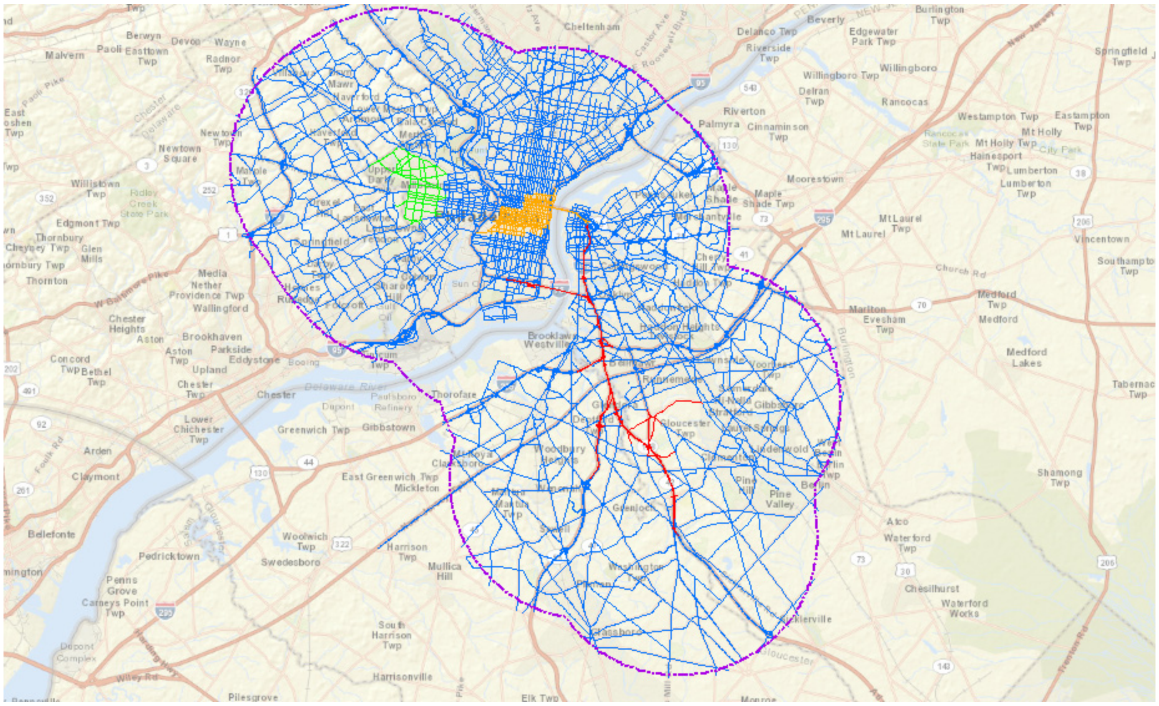


Figure 3.11. The Aggregated Buffer Region around the Three Clusters

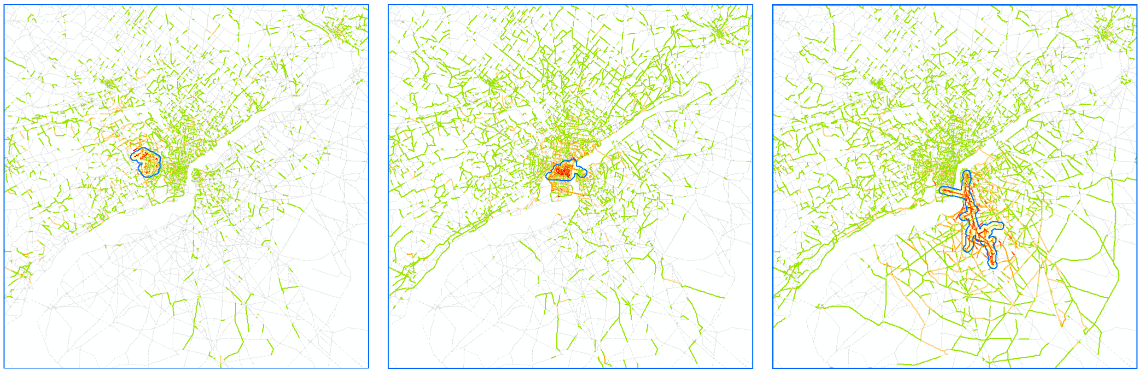
The changes in the speed and V/C of each individual link (compared to prior to disruption) within the aggregated buffer zone shown in blue color (Figure 3.11), were used to perform the two-sample Kolmogorov-Smirnov test (KS-test) and t-test at 5% significance level (Table 3.2). In the two-sample KS-test, the alternative hypothesis is that the two sample sets (e.g., the change in speed of the links within the buffer zone due to disruption in high-critical cluster versus the change in speed of the links within the buffer zone due to disruption in low-critical) are from different continuous distributions. In the case of two-sample t-test, the alternative hypothesis is that the two sample sets come from populations with unequal means.

Table 3.2. The results of the two-sample Kolmogorov-Smirnov and t-tests

Test	**Change in Speed**		**Change in V/C**	
	KS-test	*t-test*	*KS-test*	*t-test*
High-critical vs. ***Medium-critical***	Null Rejected $p < 0.001$	Null Rejected $P \approx 0.03$	Null Rejected $p < 0.001$	Null Accepted $P \approx 0.06$
High-critical vs. ***Low-critical***	Null Rejected $p < 0.001$	Null Rejected $p < 0.001$	Null Rejected $p < 0.001$	Null Rejected $p < 0.001$
Medium-critical vs. ***Low-critical***	Null Rejected $p < 0.001$	Null Rejected $p < 0.001$	Null Rejected $p < 0.001$	Null Rejected $p < 0.001$

Table 3.2 shows the test results and the p-values for KS- and t-tests. The results illustrates that except the t-test between high-critical versus medium-critical for the V/C, the null hypotheses were rejected at 5% significance level, thereby statistically validating the existence of three different clusters representing three distinct levels of criticality.

To visually investigate the validation results, one can plot the variations across the geographical area due to a disruption. For example, Figure 3.12 (a), (b), and (c) show the change in V/C for disruptions in the low-critical, medium-critical, and high-critical clusters, respectively.



(a) low-critical cluster *(b) medium-critical cluster* *(c) high-critical cluster*

Figure 3.12. Changes of V/C after Disruption Compared to before Disruption (Red: Significant Increase, Orange: Medium Increase, Green: Small Increase, Gray: No Increase)

Figure 3.12 (a), (b), and (c) illustrate that the change is observed to be larger for the high-critical cluster compared to the medium-critical and low-critical clusters. This visual comparison is in line with my initial hypothesis.

3.5.2 Observation-based Validation

The next validation method is to monitor and observe several real traffic incidents (e.g., flooding, accident, and construction zone) across the network during two time frames: (1) while there is an incident, (2) when such an incident is resolved. By comparing the traffic conditions in these two time frames, one can ascertain if the results are valid. To do so, I monitored the traffic condition during AM, MD, PM, and NT periods, using two map-based sources:

(1) The 511 Pennsylvania website (www.511pa.com), which is part of a statewide travel information service. It provides reliable traffic, weather and transit information to travelers in Pennsylvania. Using real-time traffic data from several sources across the state, this website is able to provide current traffic information.

(2) Google Maps (www.google.com/maps), which displays traffic conditions in real-time on major roads and highways. Recently, Waze traffic information (www.waze.com) have also been merged into the Google Maps. So, the real-time information about traffic incidents (e.g., accidents and construction zones) are also easily available on the Google Maps.

Wherever I found an incident (which was not necessarily within the three selected clusters in simulation-based validation), I captured the real-time traffic condition. I continued to monitor the same locations at later times until the incident was resolved. Then, I recorded another capture from the same geographical area and the same time of a day. Figure 3.13 shows the locations where I found incidents (e.g., flooding, accident, and construction zone).

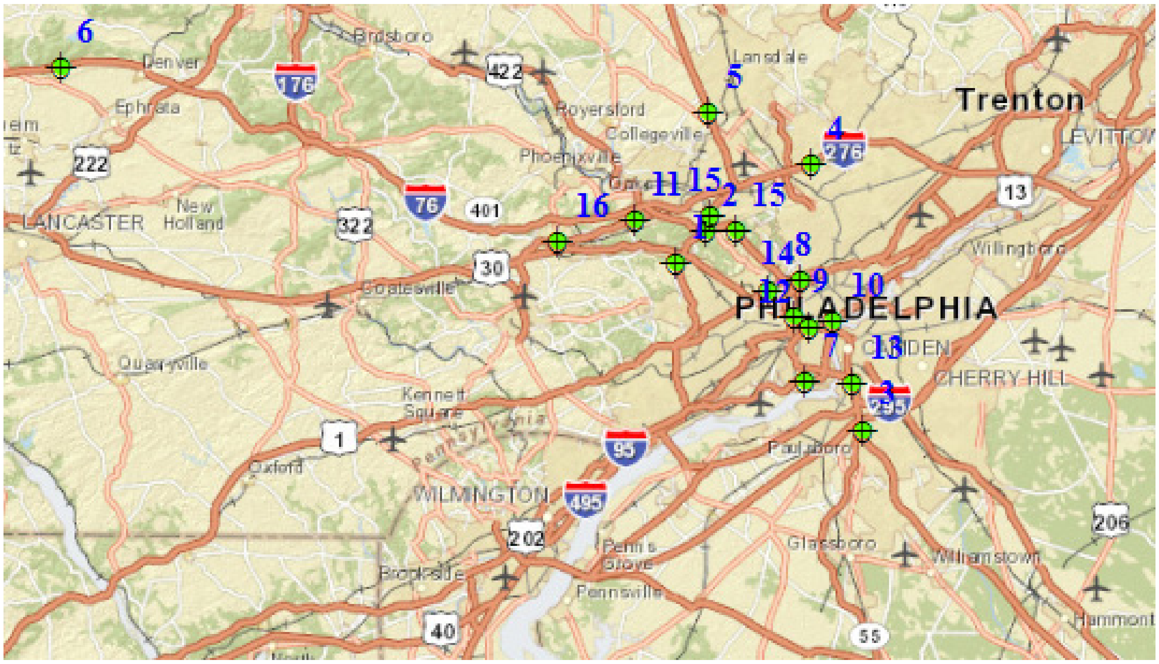
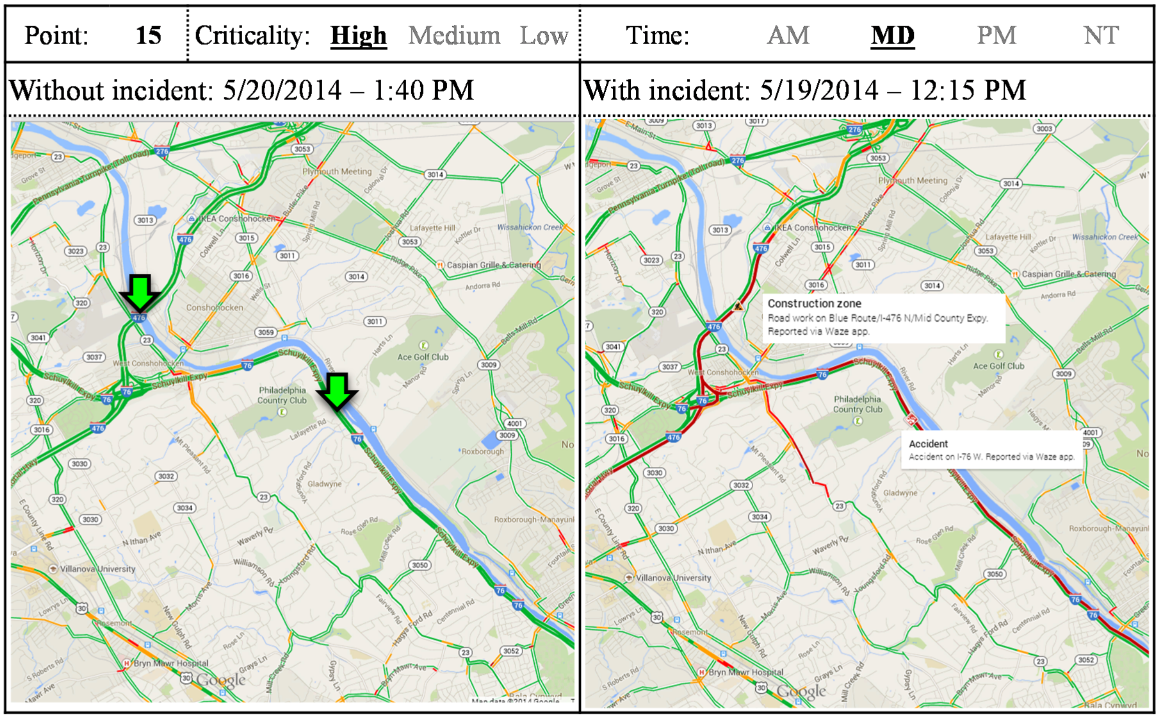


Figure 3.13. Locations of Observed Real Incidents

Figure 3.14 to Figure 3.17 show few sample observations from real incidents during different times of a day. More observations are presented in Appendix B. Each figure contains two plots, one plot shows the traffic conditions during the incident and the other one shows the traffic conditions when such an incident is resolved. Figure 3.14 to Figure 3.17 plot the traffic conditions (i.e., average speed) by colors. The green, orange, and red colors show free flow, moderate traffic, and heavy traffic, respectively.

Figure 3.14 shows an example with two simultaneous incidents (i.e., one accident on I-76 and one work zone on I-476) that occurred on a high-critical cluster during the MD period. The right plot shows the traffic condition during these disruptions while the left plot shows the traffic condition after these disruptions were resolved. As shown in Figure 3.14, the occurrence of these two disruptions led to a heavy traffic on both the Interstates 76 and 476, compared to the traffic condition after disruption (i.e., free flow condition). Therefore, the observed real traffic conditions also verify that these clusters are high-critical areas during the MD period.



Point: 15 | Criticality: **High** Medium Low | Time: AM **MD** PM NT

Without incident: 5/20/2014 – 1:40 PM

With incident: 5/19/2014 – 12:15 PM

Construction zone
Road work on Blue Route/I-476 N/Mid County Expy. Reported via Waze app.
Accident
Accident on I-76 W. Reported via Waze app.
Schuylkill Expy
Philadelphia Country Club
Ace Golf Club
West Conshohocken
Conshohocken
Gladwyne
Villanova University
Bryn Mawr Hospital
Google

Figure 3.14. Sample Observations within a High-Critical Cluster during MD Period

Wherever there was a camera, I also checked the real-time camera records for verification purposes. Figure 3.15 shows sample camera records for during and after two incidents within a high-critical cluster during the MD period. The camera records on both I-76 and I-476 show heavy traffic conditions during the disruption while they show free flow condition after disruption. This observation was in line with reported traffic conditions on Google Map, leading to the same result about high criticality of the area during the MD period.

Point: 15	Criticality: **High** Medium Low	Time: AM **MD** PM NT

Without incident: 5/20/2014 – 1:40 PM	With incident: 5/19/2014 – 12:15 PM

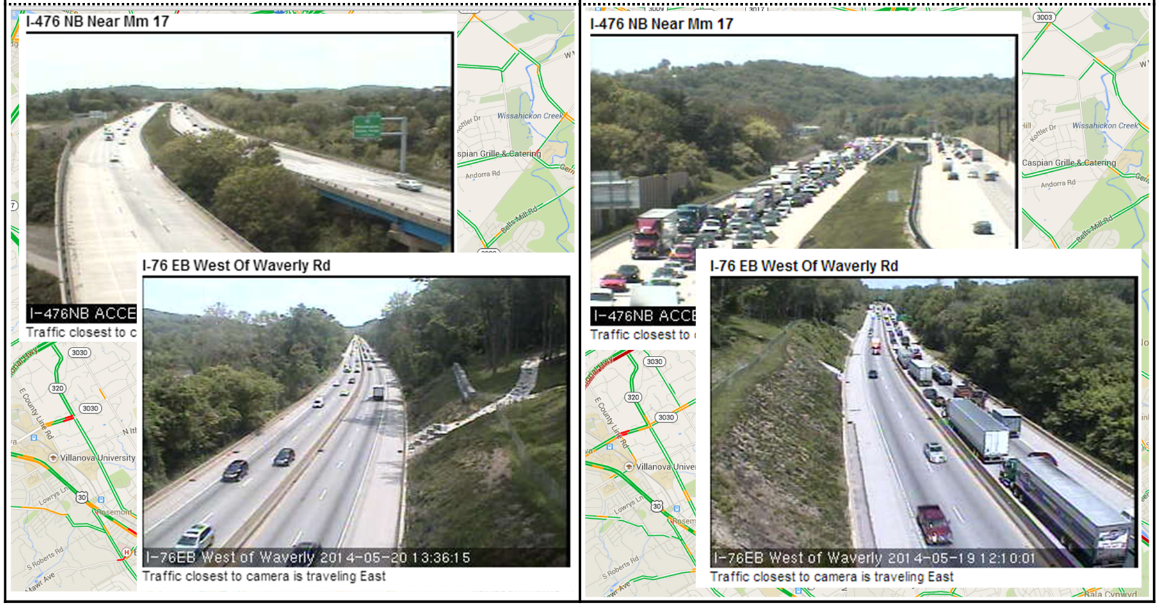
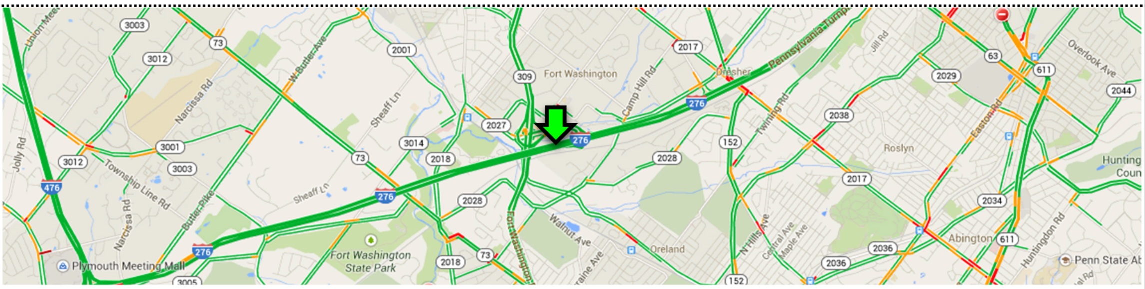

I-476 NB Near Mm 17
I-476NB ACCE
Traffic closest to c
I-76 EB West Of Waverly Rd
I-76EB West of Waverly 2014-05-20 13:36:15
Traffic closest to camera is traveling East
I-476 NB Near Mm 17
I-476NB ACCE
Traffic closest to
I-76 EB West Of Waverly Rd
I-76EB West of Waverly 2014-05-19 12:10:01
Traffic closest to camera is traveling East
Wissahickon Creek
Caspian Grille & Catering
Andorra Rd
Villanova University

Figure 3.15. Sample Camera Records within a High-Critical Cluster during MD Period

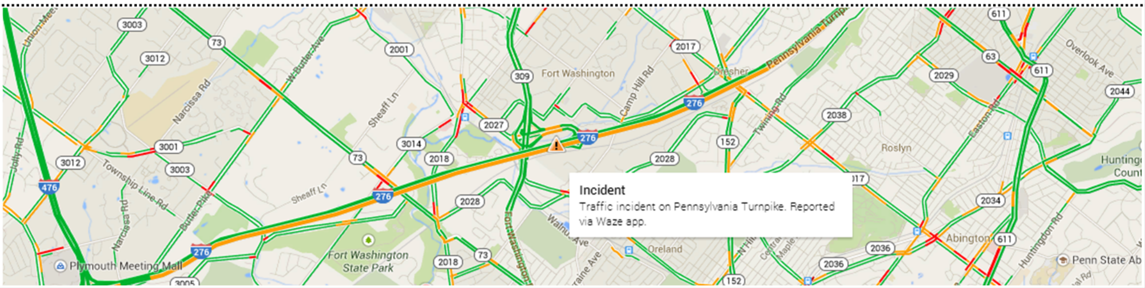
Figure 3.16 shows an example of an incident occurred on a medium-critical cluster during the PM period. The bottom plot shows the traffic conditions during the disruption while the top plot shows the traffic conditions after the disruption. As shown in Figure 3.16, the occurrence of such accident led to a moderate traffic on the Interstate 276, compared to the traffic condition after disruption (i.e., free flow condition). Therefore, the observed real traffic conditions also verify that this road segment is on a cluster which is a medium-critical cluster during the PM period.

Point: 4 | Criticality: High **Medium** Low | Time: AM MD **PM** NT

Without incident: 5/20/2014 – 4:20 PM



With incident: 5/19/2014 – 4:42 PM



Incident
Traffic incident on Pennsylvania Turnpike. Reported via Waze app.

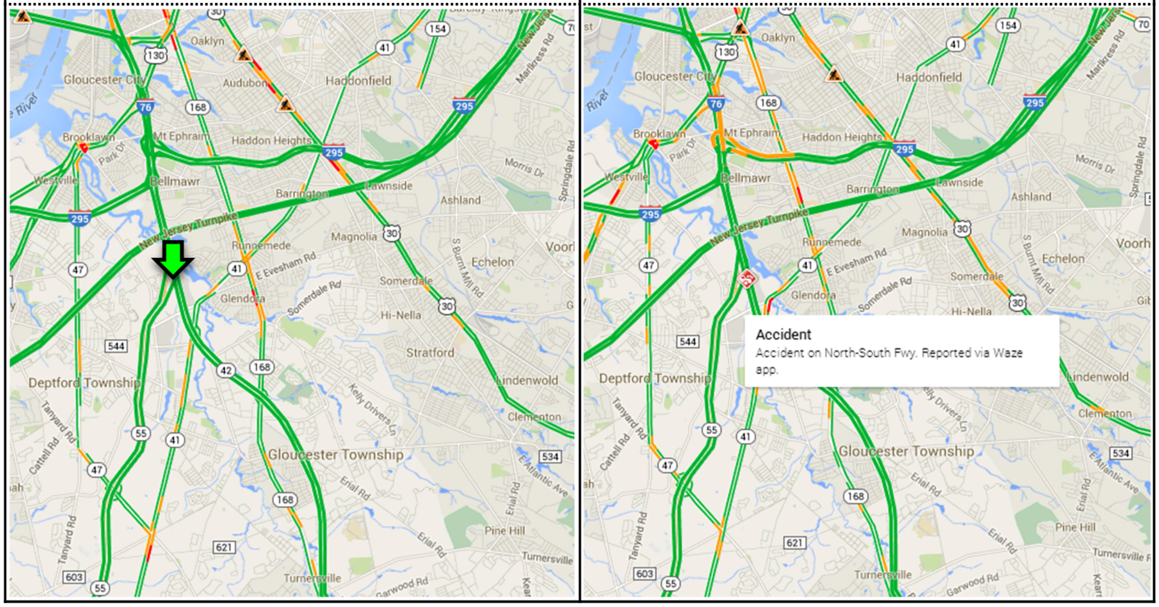
Figure 3.16. Sample Observations within a Medium-Critical Cluster during PM Period

Figure 3.17 shows another example of an accident occurred on a low-critical cluster during the MD period. The right plot shows the traffic conditions during the disruption while the left plot shows the traffic conditions after the disruption. As shown in Figure 3.17, the traffic conditions during the accident was free flow, which was the same as the time when the accident was resolved. In other words, the occurrence of such accident did not change the traffic conditions on Interstates 76. Therefore, the observed real traffic conditions also verify that this road segment is on a cluster which is a low-critical cluster during the MD period.

Point: **3** Criticality: High Medium **Low** Time: AM **MD** PM NT

Without incident: 5/19/2014 – 12:22 PM

With incident: 5/16/2014 – 12:53 PM



Accident
Accident on North-South Fwy. Reported via Waze app.

Figure 3.17. Sample Observations within a Low-Critical Cluster during MD Period

In summary, the changes in traffic conditions for these sample observations of disruptions were in line with the clustering results of the proposed computational framework. In the future, several real-time observations including observations of larger area-covering disruptions could be acquired for more validation attempts.

To conclude, it is important to identify the critical areas of a network due to area-covering disruptions. The proposed framework enables transportation modelers to divide a road network into several clusters based on the impacts of disruption on overall network connectivity. The results show that the clustering results vary based on the time of disruption as well as severity level of disruption. The proposed framework could help transportation planners to: (a) reduce vulnerability of the critical areas, (b) locate emergency service close to these areas, and (c) prioritize the emergency actions.

CHAPTER 4. MODELING NETWORK VULNERABILITY DUE TO DRIVER DISTRACTION

4.1 Introduction

According to the U.S. National Highway Traffic Safety Administration (NHTSA), driver distraction is a major cause of vehicle crashes in the United States [7]. Among the various types of distraction, cell-phone dialing and text messaging have been shown to significantly degrade driving performance [61, 81]. Therefore, it is necessary to consider vulnerability of a road network to large-scale effects of distraction. Currently, there exists a gap in large-scale study and simulation of distraction. Existing work (e.g., naturalistic experiments, laboratory experiments, statistical studies, and computational modeling) have primarily focused on driver distraction scenarios with either one or a few vehicles [60, 78, 192-195]. In contrast, available software for simulating traffic with many vehicles (e.g., CORSIM and VISSIM) has not generally considered the effects of driver distraction on driver behavior. Thus, it is critical to develop a framework to assess large-scale effects of distraction when many distracted drivers are on the network at a given time. Also, it is beneficial to evaluate real-world scenarios in which different drivers are performing different tasks, such as text messaging, dialing and conversation at the same time. For example, if 5% of drivers on road use cell-phones for text messaging and another 10% for conversation, how will it affect traffic flow dynamics? Or in the case of an emergency, what are the impacts of dramatic changes in cell-phone use? Or if a button is added to or removed from a cell-phone, how will it change traffic conditions? In this research, I formalize a computational modeling framework that integrates a cognitive model of distraction (i.e., Distract-R) and an agent-based traffic (micro-) simulation model (i.e., VISSIM). The approach is then validated using several existing experiments. The framework is employed to analyze the effects of text messaging, dialing and cell-phone conversation on several local and large-scale real-world models from China, Germany, and Philadelphia metropolitan areas.

4.2 Background Research

4.2.1 Background Research on Driver Distraction

Distraction has been typically categorized into four types: (1) visual, (2) auditory, (3) physical, and (4) cognitive distractions [7]. Visual distraction (e.g., looking at a cell phone) is the case when the driver takes his/her eyes off the road and focuses on another target for an extended time period. Auditory distraction (e.g., listening to music) is the distraction when the driver focuses on auditory tasks. Physical distraction (e.g., grabbing a pen) is the case when the driver manipulates or searches for an object by removing his/her hand(s) from the steering wheel for an extended time period. Cognitive distraction (e.g., conversation) includes any distraction in which the driver is involved in a (typically intense) cognitive task.

There are many debates regarding the forms of devices that will lead to the greatest degradation in driving performance. In general, the devices may be used for driving-related comfort (e.g., a climate-control system), driving-related information (e.g., a navigator), or entertainment (e.g., a radio or music system) [30]. In all these cases, driving is primarily a visual-manual task [7]. According to Wickens' multiple resource theory, any device that requires visual and/or manual responses will cause greater degradation in driving performance, compared to auditory or cognitive distractions [196]. Naturalistic experiments at Virginia Tech Transportation Institute (VTTI) show that manual manipulation of cell-phones (e.g., dialing and text messaging) increases the risk of crash more than auditory use of cell-phones (e.g., talking or listening). The VTTI report concluded that text messaging, which is a manual manipulation of cell phones, is associated with 23 times the risk factor compared to normal driving [58].

During last decade, driver distraction due to performing secondary tasks (especially cell-phone related tasks) has received a great deal of attention [58, 60, 61, 66, 197]. Based on the research methodology, I categorized existing studies into four different groups:

Naturalistic/Observational experiments

The naturalistic experiments record driver behavior using cameras and kinematic sensors in real-world driving conditions. The observational experiments aim at monitoring and recording driver behavior in different observation locations across an actual network. Although these experiments are more realistic and may better capture actual driving situations compared to other three groups of studies, they are expensive, time-consuming, and potentially hazardous.

Researchers at the Virginia Tech Transportation Institute have conducted a number of naturalistic experiments [58, 193]. The researchers at VTTI showed that talking and listening is not as risky as visually distracting tasks. In addition, they discussed that "Headset" cell-phones are not safer than "hand-held" cell-phones because both involve answering and dialing that require eyes to be off the road. Moreover, they showed that "true hands-free" cell-phones (e.g., voice activated phones) are less risky because the driver does not have to take his/her eyes off the road for a long period of time.

In addition, University of Massachusetts Amherst Traffic Safety Research Program (UMassSafe) has recently performed an observational experiment [76]. They observed cell-phone use of 17,677 drivers at 145 different locations throughout Massachusetts. They concluded that average cell-phone use while driving was 7.0%, among which handheld use was 5.6% and text messaging was 1.4%. They also observed that teens used their cell-phone for text messaging more often, compared to adults and elders. In addition, the observed handheld conversation rates for teens and adults were similar.

Laboratory simulations

Experimentation with a driving simulator is a common method to investigate driver distraction [7, 42, 60, 62, 64-67, 69, 71, 73, 198]. The laboratory experiments are less dangerous compared to the naturalistic experiments, but they are still time-consuming and expensive [30]. They require physical driving simulators (e.g., PatrolSim) and a database of highway/urban roadway with details (e.g. lanes, on- and off ramps, or overpasses) similar to the real world. A group of participants with different age groups based on the study goal are often included in these experiments. To simulate cell-phone conversation, as an example, participants are asked about their favorite conversations prior starting the simulation. During simulation, a research assistant maintains a dialog on topics of interest to the participant, in which the participant listens and speaks [62]. General results of laboratory simulations show that distracted drivers reduce their average speed [69], increase their speed variability [73], delay in responding to other vehicle brake [62], increase [66] or decrease [60, 74] following distance, and make fewer lane change [67].

Population-based studies

Some researchers conducted statistical studies using existing databases, such as Fatality Analysis Reporting System database, to discover the interrelation between the trend of crashes and secondary tasks (e.g., cell-phone use) [63, 70, 75, 194]. For example, Violanti (1998) studied 223,137 crashes occurring between 1992 and 1995 in the US, and concluded that the risk of fatal crash for drivers who were using a cell-phone was nine times more than other drivers. In addition, some researchers surveyed drivers who owned a cell-phone and were involved in a vehicle crash [72]. For example, Redelmeier and Tibshirani (1997) reported that the risk of crash while using a cell-phone was four times greater compared to the same drivers who were not using a cell-phone. These methods are not as expensive and time-consuming as naturalistic and laboratory experiments. However, these methods disaggregate annual nation-wide data into daily state-wide data, and such disaggregation negatively impacts their accuracy.

Computational modeling

To overcome the limitations of the abovementioned three methods, researchers analyzed the use of cognitive models in predicting distraction [78, 79]. In computer science, cognitive models deal with simulating human mental task processes in a computerized model. Such a computational model can be used to compute, simulate, and predict various aspects of human behavior. For example, the IVIS DEMAND tool was developed to integrate a behavioral model and a library of tasks from past studies [199]. The outputs of the program were measures, such as single glance time, number of glances, and number of times the driver's hand is off wheel. However, the user needs to collect data for any new task that is not in the program's library, such as text messaging. In addition, cognitive models need expert modelers to produce highly trained cognitive model of behavior [30]. To address these limitations, Distract-R system has been developed [30].

Distract-R is a tool with which engineers can develop prototypes of new in-vehicle systems and evaluate them with respect to driver distraction. The tool allows a user to specify the layout of a new device interface to be placed in the central console of a vehicle. It also allows for setting of some driver and environmental parameters. Then, in a few seconds of simulation time, Distract-R generates predictions of driver performance for relevant measures of lane keeping and car following. The predictions derive primarily from a computational cognitive model of driver behavior [78], implemented in the ACT-R (Adaptive Control of Thought-Rational) cognitive architecture [200], which has been validated for various aspects of driver performance. In running such simulations, engineers can obtain quick estimates of the distraction potential of new devices, helping to winnow many ideas down to a smaller subset to be built and tested in a more rigorous manner.

4.2.2 Background Research on Distracted Drivers and Surrounding Traffic

Existing studies have mainly focused on driver distraction scenarios with either one or a few vehicles, and have generally not looked into large-scale simulation of several vehicles [67, 68]. Recently, researchers have started to investigate the impacts of driver distraction on the surrounding traffic conditions [60, 61, 81]. For instance, Stavrinos et al. (2013) used laboratory simulation to analyze behavior of young adults engaged in cell-phone conversation and text messaging. They conducted a laboratory simulation participated by seventy-five participants between 16 to 25 years of age. For simulating different traffic conditions, Stavrinos et al. (2013) changed the traffic density (i.e., number of vehicles on road) based on the various **L**evels **O**f **S**ervice (LOS) as defined in the 2000 Highway Capacity Manual [201]. The free flow, stable flow and oversaturation are labeled as LOS A, LOS B and LOS C. They reported that distraction, especially text messaging, negatively impacted traffic conditions. Greater speed deviations, fewer number of lane changes, and more lane deviations were observed [81].

In another recent study, Salvucci (2013) developed a computational cognitive model based on Distract-R to predict the impacts of dialing on other vehicles around the distracted driver's vehicle. He simulated two distinct car-following scenarios: (1) a standard car-following scenario, which involved 16 vehicles, one following the other on a straight roadway, with a lead vehicle driving at a constant speed of 48 kph, and (2) a circular car-following scenario, which involved the same vehicles but in a circular loop of traffic. He repeated the simulation with 0, 1, and 3 distracted drivers and measured the mean and deviation of headway and speed. In both scenarios, he observed that distracted drivers reduced their mean speed and increased their speed deviation and headway deviation.

4.2.3 Limitations of Existing Studies

While existing studies investigated the impacts of distraction at driver-level, they have some limitations. First, the effects of simultaneous distraction of significant number of drivers have not yet investigated. Second, the evaluation of network-level impacts of distraction needs large-scale experiments, which are not easy to set up using physical simulators. Third, real scenarios representing different combinations of distraction types are challenging to define using the existing methods. For instance, currently it is not easy to evaluate the impacts of distraction on a large network in which 10% of drivers use their cell-phone for conversation, and 5% of drivers use their cell-phone for text messaging. Moreover, currently available software programs for simulating traffic with many vehicles (e.g., CORSIM and VISSIM) have not yet considered the effects of driver distraction [202].

4.3 Proposed Approach

The main objective of the research is to develop a framework which enables transportation modelers to predict changes in a city's traffic conditions. To overcome the limitations of existing research in large-scale vulnerability assessment, this research focuses on formalizing a framework and integrating a cognitive model of distraction (i.e., Distract-R) and an agent-based traffic (micro-) simulation model (i.e., VISSIM). The integration mechanism is a necessary component for modeling large-scale impacts of driver distraction.

The proposed framework is based on two levels of abstraction (Figure 4.1). In the first phase of the framework (i.e., abstraction level I), I simulate distraction of an individual driver in one-tenth-of-second intervals without considering complexity of the road network and interaction with many drivers around the driver's vehicle. Then in the second phase (i.e., abstraction level II), I employ the high-resolution results of the first phase to simulate traffic dynamics of many vehicles on a large-scale road network for longer period of time.

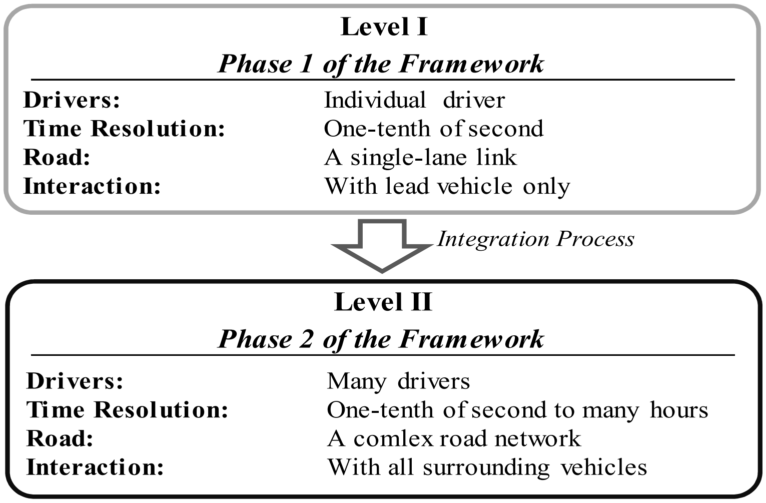

Level I
Phase 1 of the Framework
Drivers: Individual driver
Time Resolution: One-tenth of second
Road: A single-lane link
Interaction: With lead vehicle only
Integration Process
Level II
Phase 2 of the Framework
Drivers: Many drivers
Time Resolution: One-tenth of second to many hours
Road: A comlex road network
Interaction: With all surrounding vehicles

Figure 4.1. Levels of Abstraction in the Proposed Framework

4.3.1 Framework

The overall proposed approach is presented in Figure 1.1. In the first phase, Distract-R is employed to generate a time profile of distraction. Then, in the second phase, large-scale microscopic simulation of traffic is conducted by integrating the resultant distraction time profile (from phase 1) into a microscopic simulation model (e.g., VISSIM).

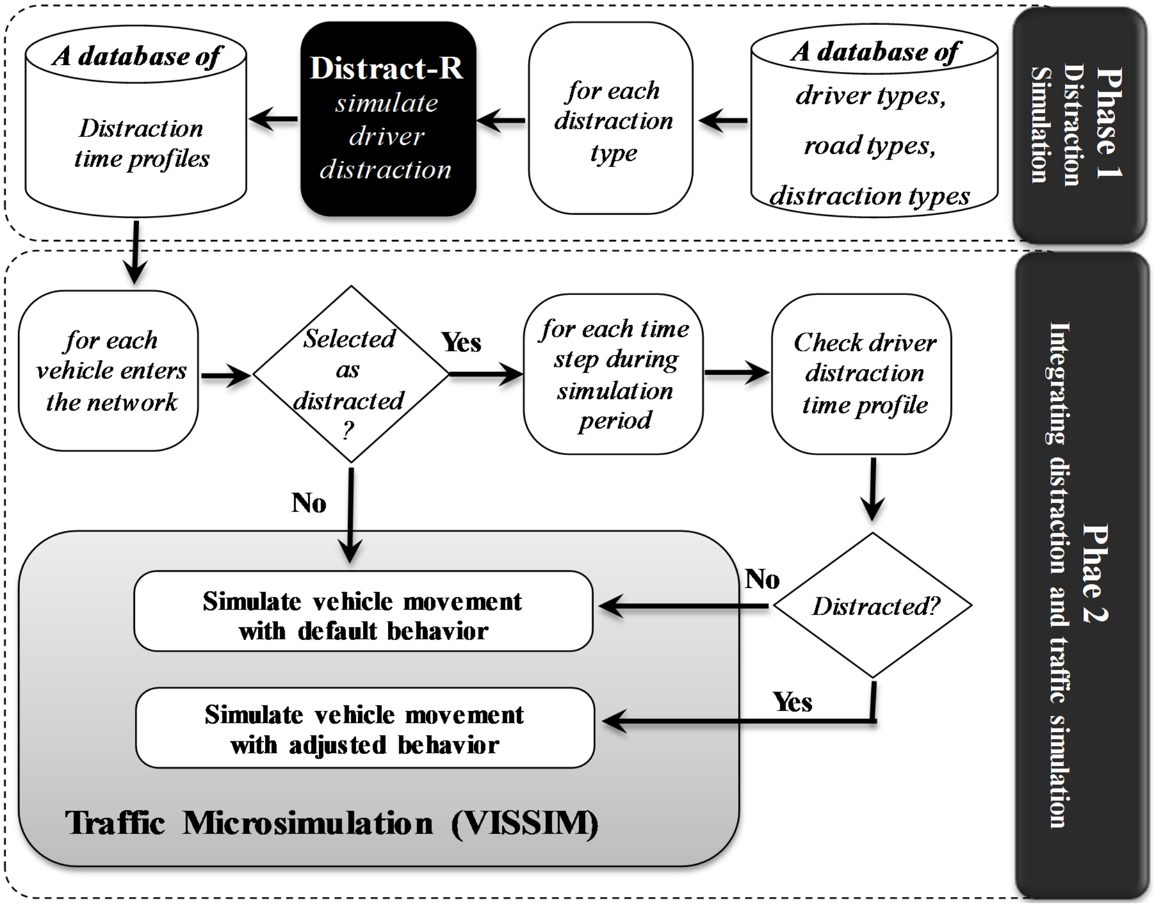

A database of
Distraction time profiles
Distract-R
simulate driver distraction
for each distraction type
A database of
driver types,
road types,
distraction types
Phase 1
Distraction Simulation
for each vehicle enters the network
Selected as distracted ?
Yes
for each time step during simulation period
Check driver distraction time profile
No
Simulate vehicle movement with default behavior
Simulate vehicle movement with adjusted behavior
Traffic Microsimulation (VISSIM)
No
Distracted?
Yes
Phae 2
Integrating distraction and traffic simulation

Figure 4.2. Flowchart of the Proposed Approach

Phase 1: Distraction simulation

Two approaches may be developed to consider the impacts of driver distraction. In the first approach, one can conduct a separate distraction simulation for each individual vehicle that enters a road network. For example if 10,000 vehicles travel across the network, one has to conduct 10,000 iterations of distraction simulations, which are computationally expensive. However, if I group drivers based on their general characteristics, the number of groups (i.e., driver types) can be restricted. Therefore, simulating multiple driver types is more computationally efficient than simulating each individual vehicle.

In this research, Distract-R is employed for driver distraction simulation because it enables us to prototype a given device. For each distraction type (e.g., text messaging), one distraction time profile is estimated which shows the distraction status (yes/no) of a driver in each time step (10 milliseconds) for a given simulation period (see Figure 4.3). To generate such profile, Distract-R takes device (e.g., cell-phone) prototype, driver characteristics (e.g., age and driving aggressiveness), distractive task (e.g., text messaging) and driving environment (e.g., straight or curved road) as the inputs. After cognitive modeling, Distract-R outputs the distraction time profile. In this research, the outputs for different distraction types (i.e., text messaging, conversation, and phone dialing) are stored in a database that could be used by the traffic simulation model during the distraction period.

Figure 4.3 shows the sample segments of three distraction time profiles. Each segment depicts the status of the distracted driver (either driving or performing secondary task) at each one-tenth of a second, between time step = 45 seconds to time step = 75 seconds. For text messaging and dialing, the distraction time profiles were generated by Distract-R. However, conversation is not a visual task and the currently available Distract-R is unable to simulate distraction due to cell-phone conversation. Instead, I had to model the distraction due to conversation using another way. Conversation is a continuous task which requires memory retrievals at frequent intervals. Hence, a simple way to model the distraction due to conversation could be the inattention from driving for a short period (e.g., 100 milliseconds) repeated in constant intervals (e.g., every 600 milliseconds).

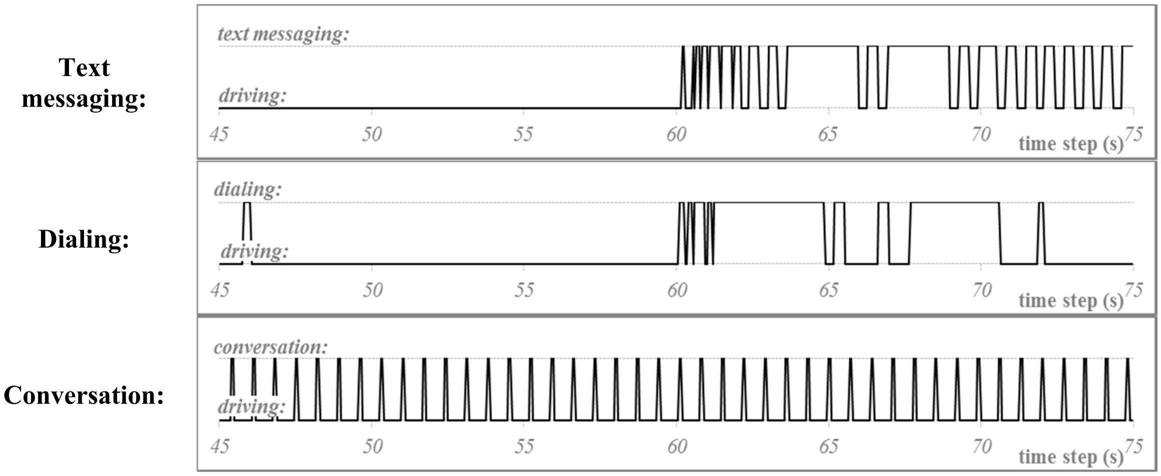


Figure 4.3. Sample 30-second-long Snapshots of Three Distraction Time Profiles

Phase 2: Integrating distraction and large-scale traffic simulation

Figure 4.2 shows the proposed approach for integrating a cognitive model for driver distraction and a microscopic traffic simulation model. For each vehicle that enters a network:

1) A random decision is made to determine whether its driver will be distracted or not. For example, for 5% distraction level, 5% of all drivers will be randomly selected as distracted.

2) If the driver of the vehicle is modeled as non-distracted, the behavior parameters of a non-distracted driver will be assigned to the driver.

3) If the driver of the vehicle is modeled as distracted:

3.1) A distraction time profile is randomly assigned to the driver. This time profile is assigned from a database containing different time profiles for various distraction types. The proposed framework is able to take different probabilities for different distraction types as an input. For example, in the case of two distraction types (i.e., phone conversation and text messaging), one may give 67% and 33% probabilities to the model for phone conversation and text messaging. This assumption means that the drivers are twice likely to be distracted by conversation than text messaging.

3.2) During a simulation period, based on the distraction time profile of the driver, whenever the status of a vehicle is distracted, the driver behavior will be adjusted. The adjustment is that the distracted driver will not respond to any acceleration/deceleration of the lead vehicle since his/her eye is off the road.

This process is conducted for all vehicles entering the network during simulation period.

Microscopic simulation basis

The Distract-R simulates driving task discretely, with 0.05-second time resolution. To incorporate its results in traffic simulation, I need to employ a traffic simulation model which simulates the movement of individual vehicles at small time steps less than a second. Thus, I should employ a microscopic simulation model which provides a small time resolution close to 0.05 of a second. Additionally, the traffic simulation model should allow us to adjust the driving behavior of each individual vehicle at each time step, if the driver is distracted. Based on these concerns, I decided to employ VISSIM traffic simulation software which is a widely used microscopic simulation model [202]. VISSIM, developed by Planung Transport Verkehr, analyzes highway and street systems, transit, and pedestrians and is designed to model individual vehicle movements up to 0.1 of a second [126]. In addition, VISSIM has an application programming interface (API) which enables us to adjust driving behavior at each 0.1 second. VISSIM can simulate traffic on networks of different sizes, from individual intersections to entire metropolitan areas.

In VISSIM, driving behaviors such as driver's car following, lane changing and lateral behaviors can be defined for different driver types. Traffic simulation in VISSIM is performed by randomly assigning driver types to individual vehicles and modeling individual vehicle movements. Therefore, each vehicle can be characterized by: (1) technical specifications of the vehicle, such as length, maximum speed, or potential acceleration and (2) driver behaviors, such as aggressiveness, desired speed, or memory of driver [126]. The basic idea is that a driver can be in one of the four driving modes. The first mode is free driving in which there is no observable influence of preceding vehicles. Hence, the driver reaches and maintains its desired speed. The second mode is approaching in which the driver adapts its speed to the lower speed of a preceding vehicle. The third mode is following in which the driver follows preceding vehicle by keeping a constant safety distant. Finally, the fourth mode is braking in which the driver decelerates when the distance is below the desired safety distance.

4.4 Validation

I validated this research by replicating the existing published experiments [61, 81]. I identified two laboratory-based studies which assessed the impacts of a single distracted driver on traffic conditions [60, 81]. These studies analyzed the impacts of distraction by a physical simulator with and without distraction that included phone conversation and text messaging. In addition, there is one recent computational study that assessed the impacts of a few distracted drivers on traffic conditions [61]. I attempted to replicate the physical experiment performed by Stavrinos et al (2013) as well as the computational experiment proposed by Salvucci (2013). Furthermore, the recent VTTI naturalistic study published by NHTSA reported overall measures of distraction (e.g., speed fluctuation) in the case of phone conversation and text messaging [193]. I compare my results with these naturalistic results as well.

4.4.1 Validation: case I

I set up several experiments to replicate the research presented by Stavrinos et al. (2013). In their research, they focused on behavior of 75 teens and young adults operating a driving simulator while engaged in various distractions, such as cell-phone conversation and text messaging. They performed experiments on a two-lane 24-mile straight highway, in three different driving conditions: LOS A (6.5 vehicles per mile in right and left lane combined), LOS C (40 vehicles per mile in right and left lane combined), and LOS E (170 vehicles per mile in right and left lane combined). In LOS A, the simulated vehicles traveled at a speed of 58 miles/h. In LOS C, the simulated vehicles traveled at 58 miles/h for the first 5,000 feet and slowed their speed to 41 miles/h for the rest of their travel. In LOS E, at the beginning they moved at 30 miles/h and after 2,000 feet they slowed down to 11 miles/h.

I generated three traffic models in VISSIM for three different traffic conditions: a) free-flow, b) stable flow and c) oversaturated flow (Figure 4.4). Based on the Pennsylvania Driver's Manual, to avoid last minute moves, the driver must look 12 to 15 seconds ahead to see and react to things early [203]. Within this distance, on average, there might be 1, 5, and 6 vehicles for LOS A, LOS C, and LOS E. Therefore, in my models, I generated six simulated vehicles in front and six vehicles behind my main driver's vehicle. For each model, I repeated simulation three times with: (1) no distraction, (2) text messaging, and (3) cell-phone conversation. For text messaging condition, I employed Distract-R. I set up Distract-R to simulate text messaging as defined by Stavrinos et al. (2013), such as "What is your favorite television show?" A sample part of text messaging distraction time profiles is presented in Figure 4.3.

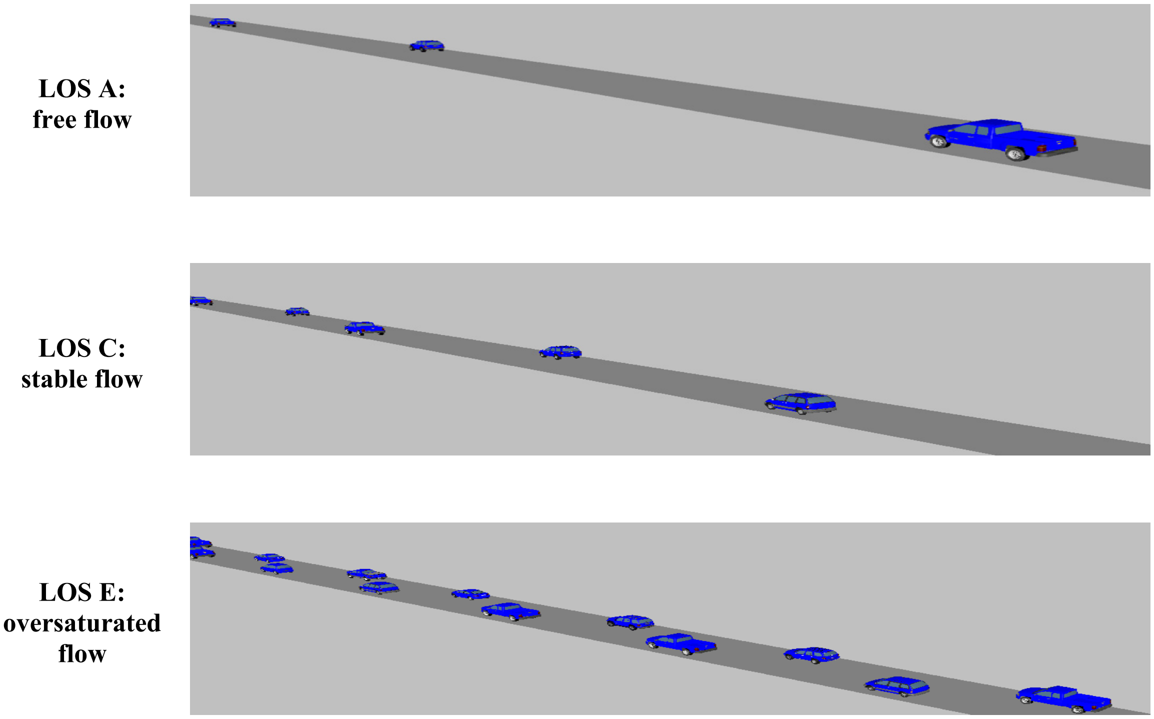


Figure 4.4. Snapshots of Three Different Models

Stavrinos et al. (2013) reported that, overall speed deviation increased for text messaging versus no distraction and for conversation versus no distraction. However, no significant difference was reported for text messaging versus conversation. Figure 4.5 compares my results for speed deviations with the results in Stavrinos et al. (2013) research and shows that the pattern of my results looks the same as the Stavrinos's results. To statistically investigate such differences among text messaging, conversation and no distraction, several hypothesis tests should be conducted.

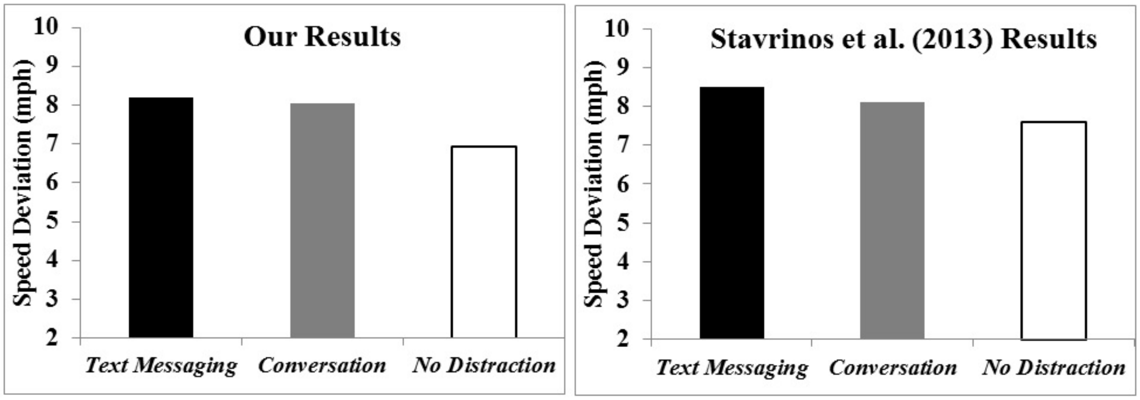


Figure 4.5. Comparison of the Results for Speed Deviation

I conducted twelve two-sample Kolmogorov–Smirnov (KS) tests to see whether the traffic measures (e.g., headway distance and speed deviation) differ for no-distraction, conversation and text messaging cases. The Kolmogorov–Smirnov test may be employed to test whether two underlying probability distributions differ [204]. If the p-value of the test turns out to be less than a certain significance level, the two distributions differ. Table 4.1 shows my KS test results for speed deviation. The results show that for free-flow conditions, distraction did not impact speed deviation. However, for stable flow and oversaturated conditions, distraction negatively impacted traffic condition by increasing speed deviation. Table 4.1 also shows that, overall, my test results are in line with Stavrinos et al. (2013) results, in which I observed an increase for text messaging versus no distraction ($p < 0.001$), an increase for conversation versus no distraction ($p < 0.001$) and no significant difference for text messaging versus conversation.

Table 4.1. KS Test Results for Speed Deviation

Tests	LOS A	LOS C	LOS E	Overall
Text messaging vs. no distraction	–	↑ $p < 0.001$	↑ $p < 0.001$	↑ $p < 0.001$
Conversation vs. no distraction	–	↑ $p < 0.001$	↑ $p < 0.001$	↑ $p < 0.001$
Text messaging vs. conversation	–	↑ $p < 0.01$	↑ $p < 0.05$	–

Note: ↑ : *increased association* – : *association not significant*

4.4.2 Validation: case II

I set up two experiments similar to the computational experiments conducted by Salvucci (2013). Figure 4.6 shows configurations of the two experiments, in which sixteen drivers drove on a straight roadway (Figure 4.6-a) and in a circular loop (Figure 4.6-b). In this case, the researcher studied distraction due to cell-phone dialing. For more details of the experiments, interested readers can refer to [3].

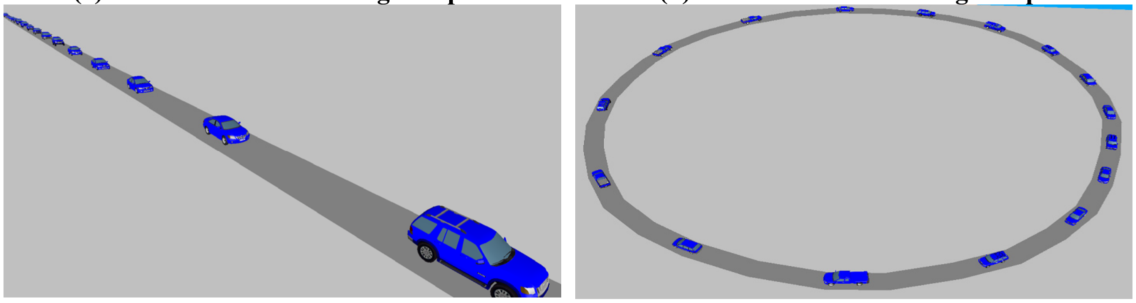
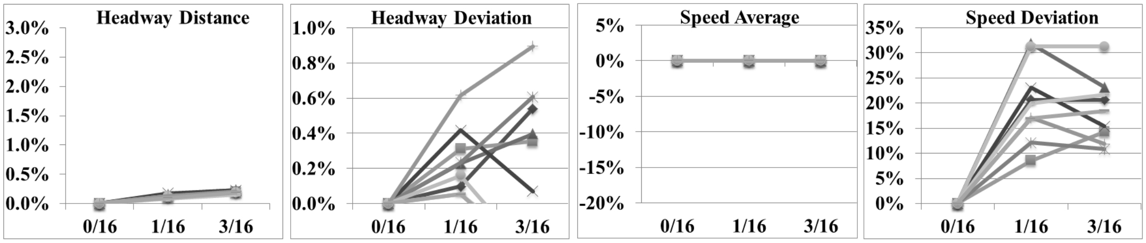

(a) Standard Car-Following Setup: **(b) Circular Car-Following Setup:**

Figure 4.6. Configurations of Two Experiments Proposed by Salvucci (2013)

Salvucci (2013) concluded that for a standard car-following scenario, headway distance average increased a small amount, headway deviation did not significantly vary, speed average decreased for some groups, and speed deviation increased. Figure 4.7(a) and Figure 4.7(b) show the results of ten groups of drivers from Salvucci (2013) and my research respectively. Each graph line represents a group of model drivers simulated across three conditions, namely with 0, 1, or 3 distracted drivers (performing the dialing task) out of 16 total drivers. To better visualize the effects of distraction, the graphs depict the changes (increase or decrease) compared to the no-distraction (i.e., 0/16) case. For example, +30% at 1/16 for speed deviation means that when there is one distracted driver, speed deviation increased by 30% compared to no distraction case. It is noteworthy that the pattern of changes (i.e., either increasing or decreasing) with respect to the number of distracted drivers was considered rather than the amount of changes. For example, I wanted to investigate whether I can observe the same increasing pattern (rather than decreading) for speed deviation that was reported in Salvucci (2013) results.

For the standard car-following scenario (Figure 4.7), my approach could capture the same pattern in speed average and speed deviation compared to Salvucci (2013) study. However, they differ in the case of headway-distance average and deviation.

(a) Standard Car-Following - Salvucci (2013) Results:



(b) Standard Car-Following – My Results:

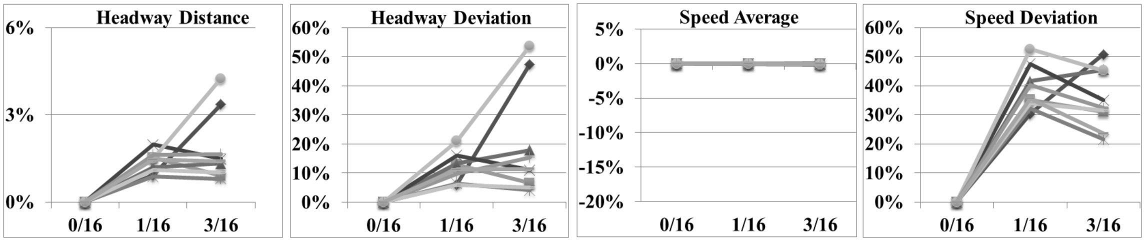
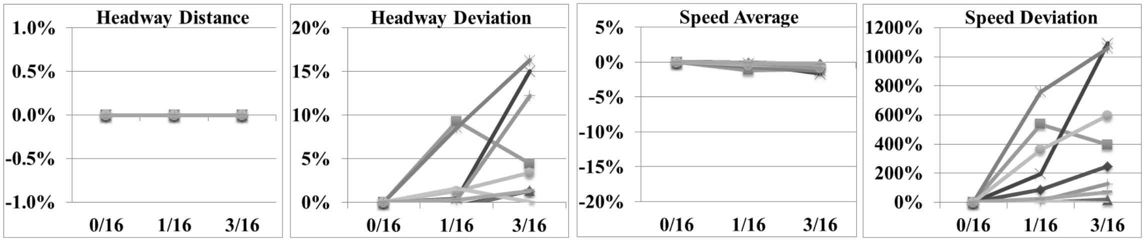


Figure 4.7. Standard Car-Following Results (Changes Compared to No Distraction). There Are Ten Graph Lines, each of Which Represents a Group of Model Drivers Simulated Across Three Conditions, Namely with 0, 1, or 3 Distracted Drivers.

For the circular car-following scenario (Figure 4.8), Salvucci (2013) reported that: (a) headway distance average remained constant, (b) headway deviation grew slightly, (c) average speed decreased, and (d) speed deviation increased. All these patterns were observed in my test except the speed deviation in which my results did not match to that of Salvucci (2013). However, the observed increase in speed deviation by Salvucci (2013) significantly differs from other observed patterns in this study. Assuming similar conditions for both standard and circular scenarios, I could not interpret the reason behind such an increase in speed deviation observed in the research by Salvucci (2013).

(a) Circular Car-Following - Salvucci (2013) Results:



(b) Circular Car-Following – My Results:

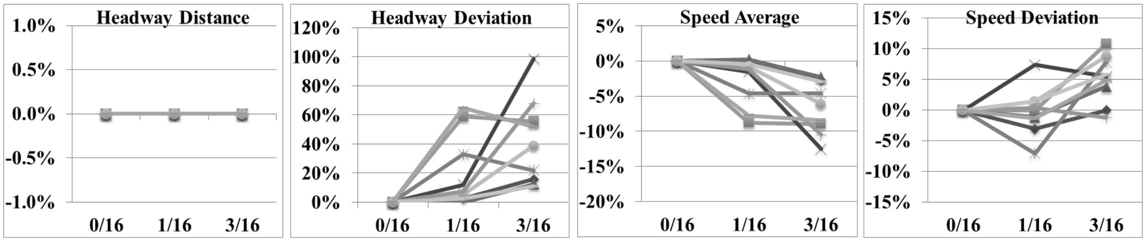


Figure 4.8. Circular Car-Following Results (Changes Compared to No Distraction). There Are Ten Graph Lines, each of Which Represents a Group of Model Drivers Simulated Across Three Conditions, Namely with 0, 1, or 3 Distracted Drivers.

The other interesting observation is the propagation of speed deviation between individual drivers. Figure 4.9 shows speed deviation of all sixteen individual drivers for standard and circular car-following scenarios. When drivers follow each other on a straight lane (i.e., standard scenario), the speed deviation starts from zero for the lead driver (because it travels with a constant speed) and increases gradually. In contrast, when drivers drive in a circular lane, the lead driver is constrained by the last driver in the loop, thus no gradual increase in speed deviation is observed. In this case, the change in speed deviation occurs around the distracted drivers. Therefore, in overall, smaller increase in speed deviation is expected over all drivers compared to the standard scenario. This observation interprets the resultant smaller increase of speed deviation in my research (Figure 4.8(b)).

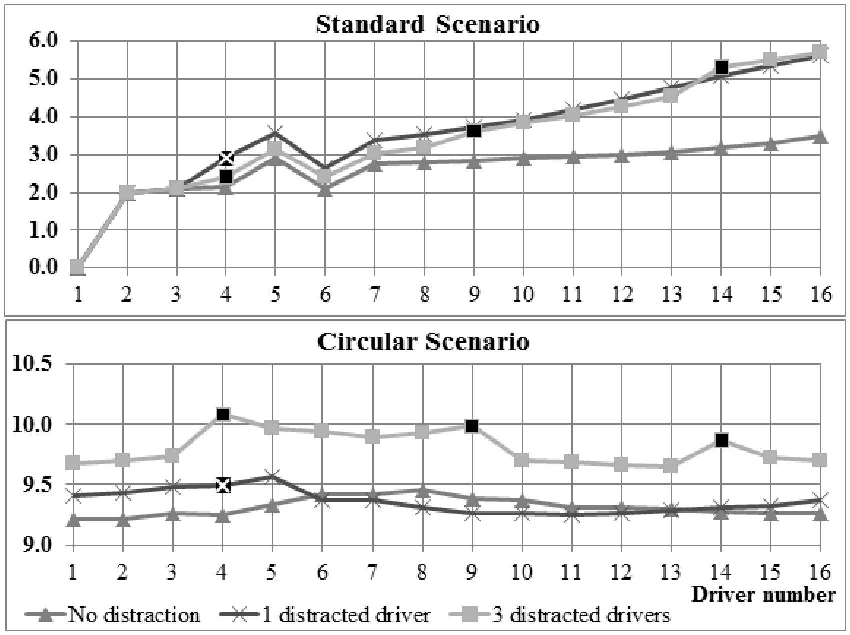


Figure 4.9. Speed Deviation for Each Individual Driver

4.4.3 Naturalistic Verification

Naturalistic experiments at Virginia Tech Transportation Institute (VTTI) show that manual manipulation of cell-phones (e.g., text messaging) increases the risk of crash more than auditory use of cell-phones (e.g., talking and listening). The VTTI study concluded that text messaging, on average, lasted 36.4 seconds and was associated with 23 times the risk factor compared to normal driving [58, 193]. In my first validation case, I simulated similar text messaging scenarios as Stavrinos et al. (2013) used. My results show that the average duration of generated time profile for text messaging is approximately 36 seconds which is in line with the VTTI naturalistic results.

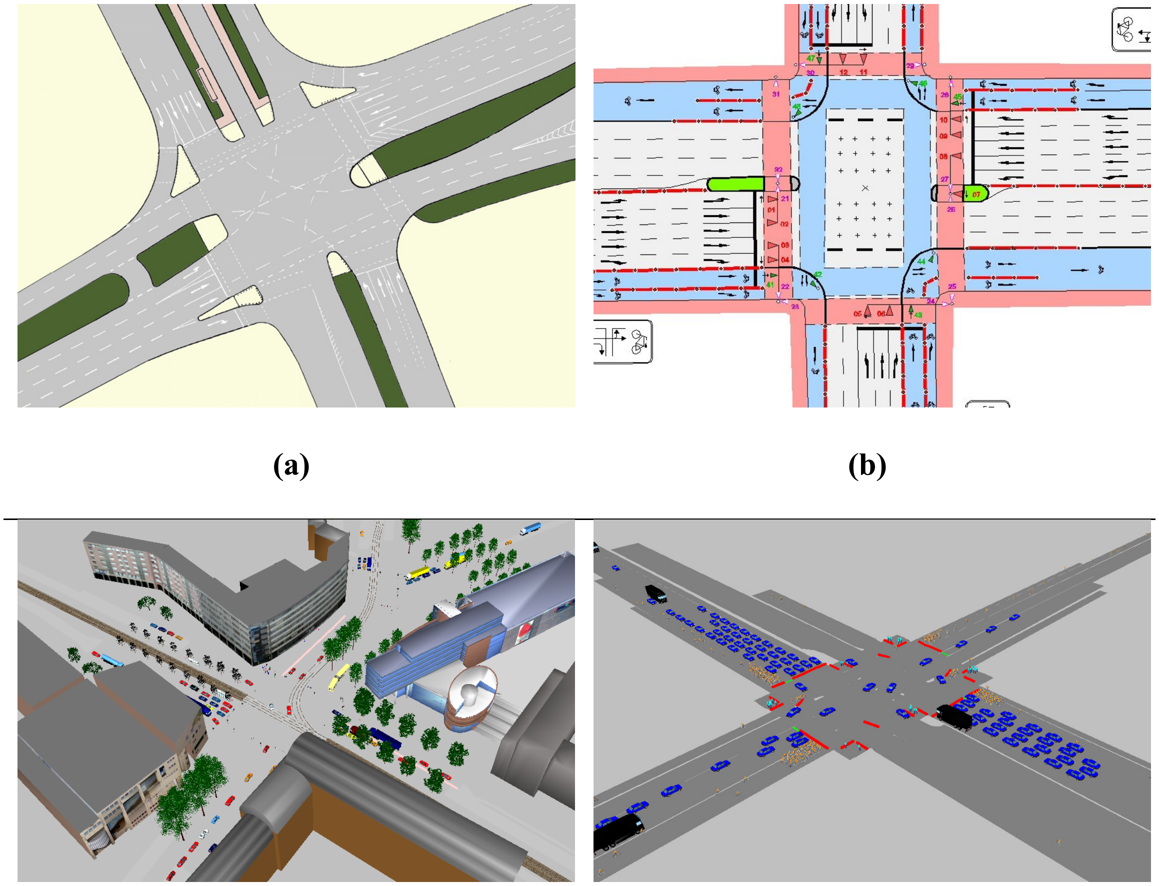
Regarding longitudinal vehicle control, the VTTI study reports significant increase in the speed deviation for hand-held and hands-free cell-phone use. For both of the validation cases in this research, I observed statistically significant increase in speed deviation, as observed in the VTTI naturalistic experiment [193].

4.5 Results

This section presents real-world applications of the proposed framework in two different categories: local and large-scale case studies. The objectives of this section are: (1) to demonstrate the effects of distraction in traffic, and (2) to show how quickly one can employ the proposed framework for various applications.

4.5.1 Local Case Studies

I analyzed the impacts of distraction for two different intersections from Karlsruhe, Germany, and Beijing, China (Figure 4.10). The Beijing intersection (Figure 4.10-a) represents a busy intersection. In addition to cars, these models simulate the movement of pedestrian, bikes, trucks, and buses. The Karlsruhe intersection (Figure 4.10-b) represents a complex urban intersection including signal control, transit, bicycles and pedestrians. I conducted several simulations for different percentages of distraction: 0%, 5%, 10%, and 30% of all drivers. In each case, the distracted drivers were distracted by text messaging, dialing, or conversation equally likely. I used the distraction profiles presented in Figure 4.3.



(a) (b)

(c) (d)

Figure 4.10. Snapshots of Two Intersections in: Germany (Left, (a) and (c)) and China (Right, (b) and (d))

Figure 4.11 shows simulated results for the two intersections. Each graph depicts five different measures which are the changes compared to no-distraction (i.e., 0%) case. Positive values show increase and negative values show decrease in the measure. The five measures are: speed coefficient of variation (SV), headway distance coefficient of variation (HV), speed average (SA), headway distance average (HA), and number lane change (LC). The results show that when distraction level increased, speed coefficient of variation (c.o.v.) and number of lane change increased significantly. Coefficient of variation is defined as the proportion of standard deviation to the average. However, average speed and average headway distance decreased. Additionally, the observed trend of change for Germany intersection is approximately proportional (to the distraction level) while for China intersection the 5% and 10% lines differ significantly from 30% line. This means that the negative impacts of smaller levels of distraction on busy intersections are lower than that of normal intersections.

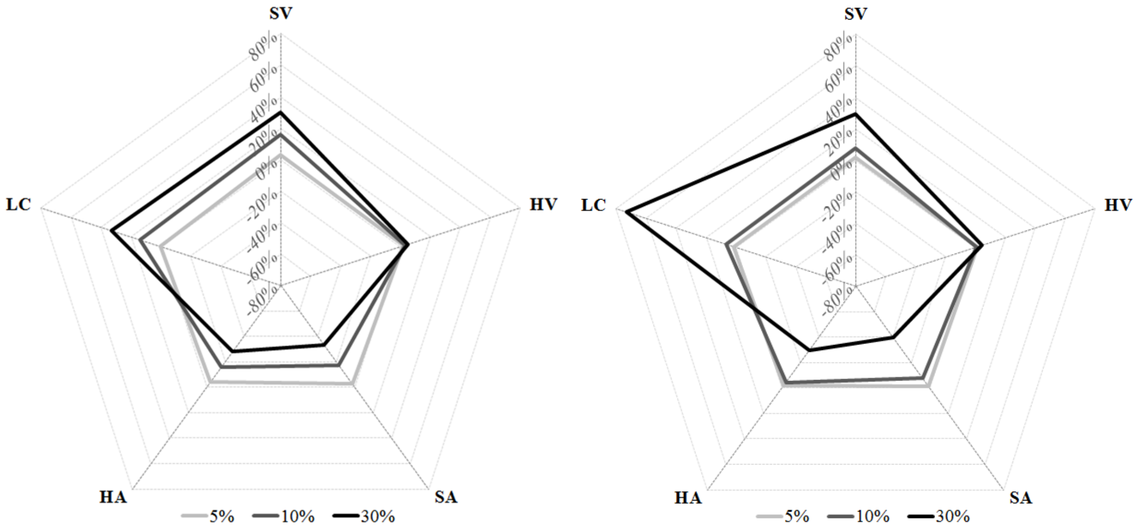


Figure 4.11. Results for Germany (left) and China (right) Including Change in: Speed c.o.v. (<u>SV</u>), Headway Distance c.o.v. (<u>HV</u>), Speed Average (<u>SA</u>), Headway Distance Average (<u>HA</u>), and Lane Change (<u>LC</u>) , for Different Levels of Distraction (5%, 10%, and 30% of drivers). Positive Values Show Increase and Negative Values Show Decrease in the Measure.

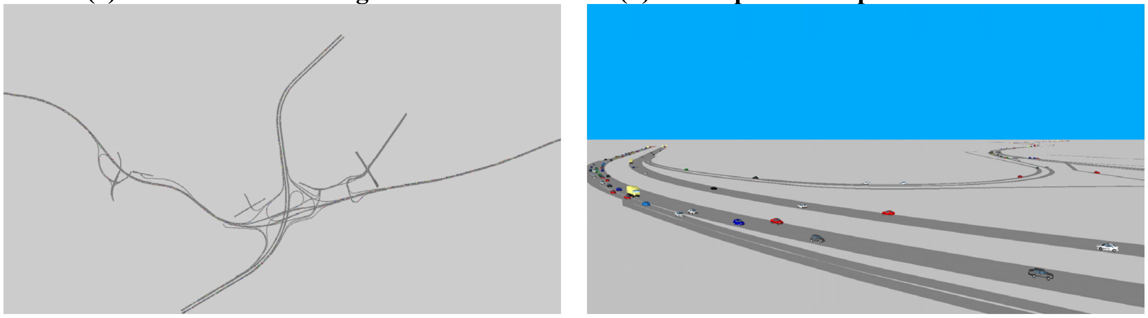
4.5.2 Preliminary Results on Large-Scale Case Studies

The proposed framework enables transportation modelers to easily set up virtual experiments and evaluate the impact of distracted drivers on traffic conditions of large-scale networks. As a preliminary example, I employed the framework for a real case study from the Philadelphia region. The model was built around the interchange of Interstates 476 and 76 including about one mile of each interstate (Figure 4.12(a)). I conducted several

10-minute traffic simulations during PM hours with: (i) 0%, (ii) 5%, (iii) 10% and (iv) 30% of drivers distracted by text messaging, dialing, or conversation.

The preliminary results in Figure 4.12-(c) show that when the number of distracted drivers increased, on average, drivers decreased their headway distance and speed. However, speed deviation, headway distance deviation, and number of lane change increased significantly. In addition to the overall results for all drivers, Figure 4.13 categorizes the results for distracted, not distracted, and all drivers, separately. Although the trends of change (either increasing or decreasing) are similar for different driver categories, distracted drivers are affected more than the surrounding vehicles. However, the overall impacts of distraction may be considered significant in the sense that a sizeable proportion of vehicles are driving in an impaired capacity.

(a) I-76/I-476 Interchange Model: **(b) A Sample 3D Snapshot from the Model:**



(c) The Results of Large-scale Distraction Simulation Including Change in: Speed c.o.v. (SV), Headway Distance c.o.v. (HV), Speed Average (SA), Headway Distance Average (HA), and Lane Change (LC), for Different Levels of Distraction (5%, 10%, and 30% of Drivers) :

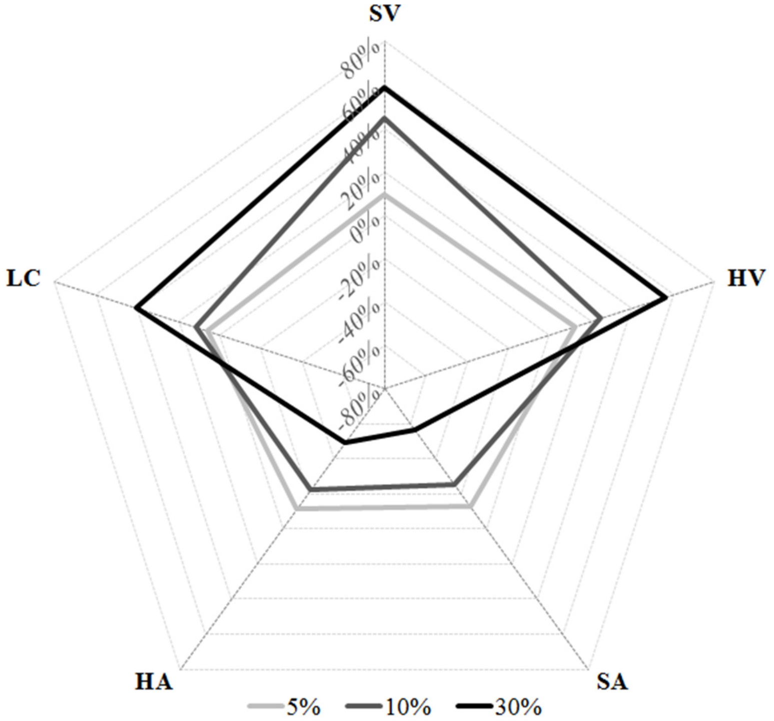


Figure 4.12. A Real-World Large-Scale Case Study from Philadelphia Region

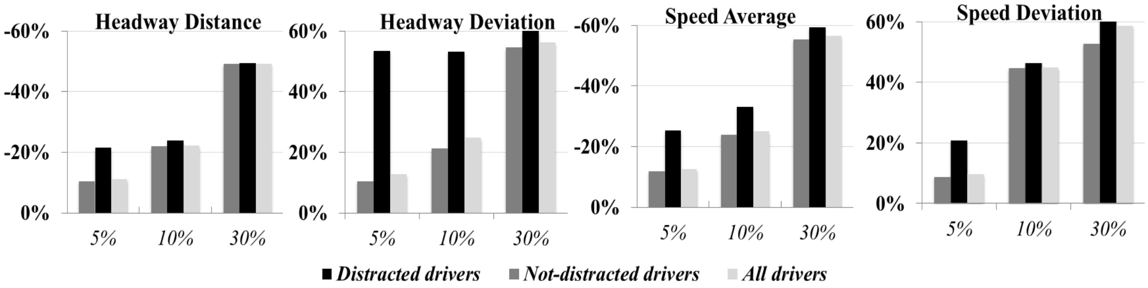


Figure 4.13. Difference between Distracted and Not-Distracted Drivers (The Values on Vertical Axis Are Relative to 0% Distracted Drivers)

Recent research has explored how traffic efficiency may be reduced by different factors, such as presence of slow vehicles and greater speed variability [5]. My preliminary results (Figure 4.13) indicate that, on average, distracted drivers proceed slower than surrounding traffic, and their presence increases speed and headway distance variability. Therefore, the presence of distracted drivers reduces traffic efficiency. In addition, speed variability in a network translates to traffic safety because deviation in speed increases the need for responding to sudden-onset events, and thus increases the risk of crash. Thus, my results may indicate a risk in traffic safety in the presence of many distracted drivers.

In addition, with smaller number of distracted drivers (Figure 12), the changes (except for headway distance deviation) are significantly higher for distracted than non-distracted drivers. However, with higher number of distracted drivers, the changes are comparable for distracted and non-distracted drivers. One may conclude that although the presence of small number of distracted drivers (e.g., less than 5%) degrades traffic efficiency and safety, it may not significantly affect the traffic conditions. However, larger number of distracted drivers (e.g., more than 10%) will cause major degradation in the behavior of not-distracted drivers. For example, Figure 4.13 shows that with 10% distraction, speed deviation of not-distracted drivers increased the same as distracted drivers.

Finally, my results contribute to the discussion in the literature regarding the change in headway distance due to distraction. Some researchers concluded that the headway distance increased in the context of distraction [13], while others reported a decrease in this measure [8]. My preliminary results indicate a significant decrease in headway distance, which again might translate to decrease in traffic safety. In addition, although I expected to see an increasing trend in headway deviation (by increasing distraction percentage), my results show a decrease in headway deviation which seems to contradict with my preliminary assumption. However, I may consider coefficient of variation of headway distance. This way, coefficient of variation of headway distance increases with the increase of distraction level. For example, headway coefficient of variation is 0.60, 0.67, 0.75, and 0.93 for 0%, 5%, 10% and 30% distraction, respectively. Generally speaking, I conclude that distraction increases the variability in traffic conditions and as a results decreases traffic safety and efficiency.

CHAPTER 5. CONCLUSIONS

This chapter presents the summary of my PhD research that includes the contributions and practical implications of the research. It also describes the limitations of the approach and concludes with the future research directions.

5.1 Research Contributions

The main objective of this research is to formalize and develop a computational framework that can: (a) predict the macroscopic performance of a transportation network based on its multiple structural and dynamical attributes (Chapter 2), (b) analyze its vulnerability as a result of man-made/natural disruption that minimizes network connectivity (Chapter 3), and (c) evaluate network vulnerability due to driver distraction (Chapter 4). An integrated framework to address these challenges—which have largely been investigated as separate research topics, such as distracted driving, infrastructure vulnerability assessment and traffic demand modeling—needs to simultaneously consider all three principal components (i.e., structure, dynamics, and external causes) of a network. In this research, the integrated framework is built upon recent developments (theories and methods) in interdisciplinary domains, such as network science, cognitive science and transportation engineering. This is the novelty of the proposed framework compared to existing frameworks and approaches. So, this PhD research has three major contributions:

5.1.1 Performance Prediction Model

Transportation engineers typically use travel modeling and traffic simulation to assess the traffic conditions. In urban road planning, planners have to evaluate various design alternatives in order to improve traffic conditions over an existing network or to build a new road network. The planners have to modify traffic models and run multiple simulations to evaluate the impacts of various proposed changes in network structure and traffic demand. Depending on the size of a network, such evaluation would be time-consuming and tedious, especially if they want to assess several alternatives.

In this research, I developed a model that can predict the macroscopic measures of performance (i.e., average speed and volume) for new alternative designs without performing traffic simulations. The inputs of the model were multiple structural and dynamical attributes of the new network, and the outputs were multiple network-wide MOEs. I used a set of the existing structural attributes, such as the weighted degree and betweenness. I also proposed a set of dynamical attributes (e.g., largest eigenvalues of an OD matrix) to capture various travel demand patterns across the network. Then, I ran several traffic simulations to find network MOEs for different combinations of structural and dynamical attributes. In the next step, I employed a multivariate statistical method called the Canonical Correlation Analysis to capture the relationship among multiple MOEs and network attributes. Finally, using the captured relationship, I developed a model to predict macroscopic performance (i.e., multiple MOEs) of a new network. For the prediction, the model does not need the tedious task of simulation.

The framework enables transportation modelers to understand how variations in network structure and dynamics could impact the macroscopic performance of design alternatives. While the proposed model does not replace the simulation models, it is useful for pre-screening process of numerous design alternatives and leads to a significant saving of time and computational resources. The result of such pre-screening process is a small subset of the long list of design alternatives which will be further analyzed using simulation models.

5.1.2 Critical Area Identification

Failure of a single link or multiple links of networks is an important problem in civil infrastructures (e.g., road transportation networks). Such failures (i.e., disruptions) might be either partial or complete, caused by either man-made or natural sources. To minimize the risks of these disruptions, it is important to identify the critical areas of a network.

I developed a framework that enables transportation modelers to identify various critical areas on a given road network. I built the proposed framework upon network-science theories. Specifically, I employed a method from epidemiology to define a network-wide measure of connectivity. In this method, the network connectivity is defined based on the largest eigenvalue of the network. In addition, I proposed a new approach based on community detection method to identify critical areas of a road network rather than critical links. Thus, the two important contributions of the proposed framework are: (1) eigenvalue-based measure of connectivity, and (2) community detection methods for clustering. In the first stage of the framework, the criticality of individual links was determined. Then, in the second stage, a community detection (aka modularity optimization) algorithm was used to cluster links based on their different levels of criticality. The output is a collection of different clusters throughout the network representing different levels of criticality.

In this research, I present the results of the proposed framework for the Greater Philadelphia region for different times of a day. The results show that the clustering results vary based on the time of the disruption as well as severity level of the disruption.

5.1.3 Driver Distraction Analysis

According to NHTSA'S National Center for Statistics and Analysis, 15,254 people were killed in distraction-affected fatal crashes across the United States between 2009 and 2012 [85-87]. On average, around 20 percent of distracted drivers were distracted by the use of cell phones [87]. At any given daylight time across the United Stated, approximately 660,000 drivers are using cell-phones or other electronic devices while driving [88]. These statistics highlight the importance of large-scale distraction simulation to quantitatively assess the impacts of distraction on traffic condition and safety.

In this research, I formalized a computational modeling framework that integrates a cognitive model of distraction (i.e., Distract-R) and an agent-based traffic (micro-) simulation model (i.e., VISSIM) to perform traffic simulation in which multiple-distracted drivers are distracted by various types (e.g., dialing, text messaging, and cell phone conversation). In the first phase of the framework, I simulated distraction of an individual driver in one-tenth-of-second intervals without considering complexity of the road network and interaction with many drivers around the driver's vehicle. Then in the second phase, I employed the high-resolution results of the first phase to simulate traffic dynamics of many vehicles on a large-scale road network for longer period of time. The approach is then validated using several existing experiments. The framework is employed to analyze the effects of text messaging, dialing and cell-phone conversation on several local and large-scale real-world models from China, Germany, and Philadelphia metropolitan areas.

In general, variability in speed and headway distance increased by increasing the number of distracted drivers. However, on average, speed and headway distance decreased. All these measures corresponded to reduction in traffic safety and efficiency. Moreover, I observed that the number of lane change for not-distracted drivers increased which could be a reaction to the reduction in speed and increase in speed variability of distracted drivers.

5.2 Practical Implications

The research is expected to have the following practical implications:

5.2.1 Performance Prediction Model

At present, transportation planners have to perform many time-consuming traffic simulation runs for pre-screening of numerous different design alternatives. Examples include identifying the locations of shopping centers and changing the operating time of public and private organizations to alleviate traffic congestion. In general, evaluating such alternatives require changes in both structural and dynamical attributes of a network. The proposed model helps them to evaluate such alternatives efficiently with a few number of simulation runs, leading to a significant saving of time and computational resources.

5.2.2 Critical Area Identification

At present, the volume over capacity ratio is used as the standard measure of links' importance (i.e., criticality) while it only considers the local impacts of the link failure. Another existing measure of importance is the total delay due to the link failure which is time-consuming to calculate for all links. The proposed framework could help transportation planners to segment a network into multiple areas based on the impacts of disruption on the network overall connectivity. This is especially helpful to: (a) reduce vulnerability of the critical areas, (b) locate emergency service close to these areas, and (c) prioritize the emergency actions.

5.2.3 Driver Distraction Analysis

Currently, driver distraction analysis is performed using physical simulators, which cannot evaluate the impacts of multiple-driver distraction and changes in the design of a distracting device, such as a cell phone. The proposed framework enables transportation engineers to model the severity and impact of multiple-driver distractions due to different distraction types (e.g., texting, phone conversation) on traffic conditions. Examples of what if scenarios that could be investigated using this framework include: if 5% of drivers on road use cell-phones for text messaging and another 10% for conversation, how will it affect traffic flow dynamics? Or in the case of an emergency, what are the impacts of dramatic changes in cell-phone use? Or if a button is added to or removed from a cell-phone, how will it change traffic conditions?

5.3 Limitations and Future Research Directions

There are a number of limitations associated with the proposed framework. The future research directions are recommended to address some of the existing limitations.

5.3.1 Performance Prediction Model

Database for Model Development and Validation: The Greater Philadelphia road network was used to develop and validate the proposed model. Though the proposed model was validated on the Greater Philadelphia network, it is scalable to other network types as well. In the future, it is recommended to assimilate a diverse database of road networks to develop the model.

Inclusion of Additional Structural and Dynamical Attributes: I used eight structural and six dynamical attributes. It is recommended to consider other network attributes including: structural attributes (e.g., network diameter, and form factor) and dynamical attributes (e.g., spatial distributions of traffic origins and destinations). This would lead to a better representation of road structure and traffic demand which may improve reliable performance prediction.

5.3.2 Critical Area Identification

Assumption for Traffic Pattern: In this research, I used the traffic patterns without considering the changes in these patterns due to disruptions. However, the traffic patterns might change during or after occurrence of the disruptions. For example, some people might either cancel their trip or change their routes. Hence, it is recommended to consider such changes in traffic patterns in the future.

5.3.3 Driver Distraction Analysis

Changes in Driver's Behaviors: We modeled driving behavior in a computational environment by focusing on the changes in drivers' longitudinal performance (e.g., speed and headway distance). Future research may consider the changes of lateral behavior (i.e., lane violation). Also, more investigations are needed to capture the real-world distraction time profile for cell-phone conversation.

Accident Analysis: In this research, I focused on the impacts of multiple-driver distraction on overall traffic conditions, such as average speed and headway distance. In the future, it is recommended to simulate accidents due to cell-phone-related distraction.

5.3.4 The Integrated Framework

Combination of Distracted and Disrupted Conditions: In this research, I formalized a framework to integrate the principal components (i.e., structure, dynamics, and external causes) of a transportation network. In three research questions, I separately investigated three approaches to: (i) predict the macroscopic performance of a transportation network,

(ii) analyze its vulnerability as a result of man-made/natural disruption, and (iii) evaluate network vulnerability due to driver distraction. Future research is recommended to further investigate the performance and vulnerability of a network due to a simultaneous combination of disruption and distraction. For example, what happens if there is a severe flooding and 30% of drivers start to use their cell-phone (e.g., for checking the weather, route finding, or to reach out to their friends and family) while driving. In such cases, it is necessary to consider the correlations between distraction and disruption. For instance, during severe flooding drivers are more cautious when they need to use their cell-phone, compared to the normal weather.

Interconnected infrastructures: While the scope of this research was to model the performance and vulnerability of a road network, it is recommended to extend this framework for modeling the performance and vulnerability of multiple interconnected infrastructures in the future. There are many real-world examples in which the occurrence of a disruption on one infrastructure network may lead to the performance degradation of other networks. For example: (i) a disruption on a power network may lead to the failure of water distribution networks and public transportation systems; or (ii) the disruption on a given area of a road network may force people to use other types of transportation (e.g., public transit) leading to performance degradation of those systems. Therefore, future research is needed to investigate the propagation of risks among multiple interconnected infrastructures.

REFERENCES

1. Berdica, K., *An introduction to road vulnerability: what has been done, is done and should be done.* Transport Policy, 2002. **9**(2): p. 117-127.
2. Maerivoet, S. and B. De Moor, *Traffic flow theory.* arXiv preprint physics/0507126, 2005.
3. Boccaletti, S., et al., *Complex networks: Structure and dynamics.* Physics Reports, 2006. **424**(4–5): p. 175-308.
4. Zhao, L., et al., *Onset of traffic congestion in complex networks.* Physical Review E, 2005. **71**(2): p. 026125.
5. Klauer, S.G., et al., *The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data*. 2006.
6. Jenelius, E. and L.G. Mattsson, *Road network vulnerability analysis of area-covering disruptions: A grid-based approach with case study.* Transportation Research Part A: Policy and Practice, 2012.
7. Young, K., M. Regan, and M. Hammer, *Driver distraction: A review of the literature.* Distracted driving. Sydney, NSW: Australasian College of Road Safety, 2007: p. 379-405.
8. Calabrese, F., et al., *Estimating origin-destination flows using mobile phone location data.* IEEE Pervasive Computing, 2011: p. 36-44.
9. Beskos, D., P. Michalopoulos, and J. Lin, *Analysis of traffic flow by the finite element method.* Applied mathematical modelling, 1985. **9**(5): p. 358-364.
10. Sharma, H.K. and B. Swami, *MOE-Analysis for Oversaturated Flow with Interrupted Facility and Heterogeneous Traffic for Urban Roads.* International Journal of Transportation Science and Technology, 2012. **1**(3): p. 287-296.
11. Albert, R. and A.-L. Barabási, *Statistical mechanics of complex networks.* Reviews of Modern Physics, 2002. **74**(1): p. 47-97.
12. Barrat, A., et al., *The architecture of complex weighted networks.* Proceedings of the National Academy of Sciences of the United States of America, 2004. **101**(11): p. 3747-3752.

13. Kolaczyk, E.D., *Statistical analysis of network data: methods and models*. 2009: Springer.

14. Xuetao, W., et al., *Competing Memes Propagation on Networks: A Network Science Perspective.* Selected Areas in Communications, IEEE Journal on, 2013. **31**(6): p. 1049-1060.

15. Prakash, B.A., et al. *Fractional immunization in networks*. in *SIAM International Conference on Data Mining (SDM13)*. 2013.

16. Ping, L., et al., *Topological Properties of Urban Public Traffic Networks in Chinese Top-Ten Biggest Cities.* Chinese Physics Letters, 2006. **23**(12): p. 3384.

17. Aktan, A.E. *Innovating Infrastructure Planning, Financing, Engineering and Management*. in *International Conference on Smart Monitoring, Assessment and Rehabilitation of Civil Structures (SMAR 2013)*. 2013. Istanbul, Turkey.

18. Executive.Order, *Critical Infrastructure Protection.* 1995, Federal Register. p. 37347-37350.

19. Doménech, A., *A topological phase transition between small-worlds and fractal scaling in urban railway transportation networks?* Physica A: Statistical Mechanics and its Applications, 2009. **388**(21): p. 4658-4668.

20. Guimerà, R., et al., *The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles.* Proceedings of the National Academy of Sciences, 2005. **102**(22): p. 7794-7799.

21. Chan, S.H.Y., R.V. Donner, and S. Lämmer, *Urban road networks — spatial networks with universal geometric features?* The European Physical Journal B, 2011. **84**(4): p. 563-577.

22. Bagler, G., *Analysis of the airport network of India as a complex weighted network.* Physica A: Statistical Mechanics and its Applications, 2008. **387**(12): p. 2972-2980.

23. Li-Ping, C., et al., *Structural Properties of US Flight Network.* Chinese Physics Letters, 2003. **20**(8): p. 1393.

24. Li, W. and X. Cai, *Empirical analysis of a scale-free railway network in China.* Physica A: Statistical Mechanics and its Applications, 2007. **382**(2): p. 693-703.

25. Sen, P., et al., *Small-world properties of the Indian railway network.* Physical Review E, 2003. **67**(3): p. 036106.

26. Wang, R., et al., *Geographic coarse graining analysis of the railway network of China.* Physica A: Statistical Mechanics and its Applications, 2008. **387**(22): p. 5639-5646.

27. Chen, A., et al., *Capacity reliability of a road network: an assessment methodology and numerical results.* Transportation Research Part B: Methodological, 2002. **36**(3): p. 225-252.

28. Erath, A., et al., *Vulnerability assessment methodology for Swiss road network.* Transportation Research Record: Journal of the Transportation Research Board, 2009. **2137**(-1): p. 118-126.

29. Wang, P., et al., *Understanding Road Usage Patterns in Urban Areas.* Scientific reports, 2012. **2**.

30. Salvucci, D.D., *Rapid Prototyping and Evaluation of In-Vehicle Interfaces.* Acm T Comput-Hum Int, 2009. **16**(2): p. 9:1-9:33.

31. Albert, R., H. Jeong, and A.L. Barabási, *Error and attack tolerance of complex networks.* Nature, 2000. **406**(6794): p. 378-382.

32. Danila, B., et al., *Optimal transport on complex networks.* Physical Review E, 2006. **74**(4): p. 046106.

33. Gastner, M.T. and M.E.J. Newman, *The spatial structure of networks.* The European Physical Journal B - Condensed Matter and Complex Systems, 2006. **49**(2): p. 247-252.

34. Kurant, M. and P. Thiran, *Layered Complex Networks.* Physical Review Letters, 2006. **96**(13): p. 138701.

35. Buhl, J., et al., *Topological patterns in street networks of self-organized urban settlements.* The European Physical Journal B-Condensed Matter and Complex Systems, 2006. **49**(4): p. 513-522.

36. Porta, S., P. Crucitti, and V. Latora, *The network analysis of urban streets: A dual approach.* Physica A: Statistical Mechanics and its Applications, 2006. **369**(2): p. 853-866.

37. Lämmer, S., B. Gehlsen, and D. Helbing, *Scaling laws in the spatial structure of urban road networks.* Physica A: Statistical Mechanics and its Applications, 2006. **363**(1): p. 89-95.

38. Balakrishna, R., et al., *Comparison of Simulation-Based Dynamic Traffic Assignment Approaches for Planning and Operations Management.* 2012.

39. Ben-Akiva, M., et al., *Real time simulation of traffic demand-supply interactions within DynaMIT.* Applied optimization, 2002. **63**: p. 19-34.

40. Chiu, Y., et al., *Dynamic traffic assignment: A primer.* Transportation Research E-Circular, 2011. **E-C153**.

41. Geroliminis, N. and J. Sun, *Properties of a well-defined macroscopic fundamental diagram for urban traffic.* Transportation Research Part B: Methodological, 2011. **45**(3): p. 605-617.

42. Lam, W.H.K. and G. Xu, *A traffic flow simulator for network reliability assessment.* Journal of advanced transportation, 1999. **33**(2): p. 159-182.

43. Zhu, S., et al., *The traffic and behavioral effects of the I-35W Mississippi River bridge collapse.* Transportation Research Part A: Policy and Practice, 2010. **44**(10): p. 771-784.

44. Ziliaskopoulos, A.K., et al., *Large-scale dynamic traffic assignment: Implementation issues and computational analysis.* Journal of Transportation Engineering, 2004. **130**(5): p. 585-593.

45. Geroliminis, N. and C.F. Daganzo, *Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings.* Transportation Research Part B: Methodological, 2008. **42**(9): p. 759-770.

46. Helbing, D., *Derivation of a fundamental diagram for urban traffic flow.* The European Physical Journal B-Condensed Matter and Complex Systems, 2009. **70**(2): p. 229-241.

47. Sun, H.J., J.J. Wu, and Z.Y. Gao, *Dynamics of traffic networks: From microscopic and macroscopic perspectives.* Physica A: Statistical Mechanics and its Applications, 2008. **387**(7): p. 1648-1654.

48. Faturechi, R. and E. Miller-Hooks, *Measuring the Performance of Transportation Infrastructure Systems in Disasters: A Comprehensive Review.* Journal of Infrastructure Systems, 2014. **0**(0): p. 04014025.

49. Duenas-Osorio, L. and S.M. Vemuru, *Cascading failures in complex infrastructure systems.* Structural safety, 2009. **31**(2): p. 157-167.

50. Nourzad, H. and A. Pradhan. *Network-Wide Assessment of Transportation Systems Using an Infection Spreading Methodology.* in *ASCE IWCCE2013*. 2013. CA, USA.

51. Wu, J., et al., *Congestion in different topologies of traffic networks.* EPL (Europhysics Letters), 2007. **74**(3): p. 560.

52. Dueñas-Osorio, L.A., *Interdependent response of networked systems to natural hazards and intentional disruptions.* 2005.

53. Bell, M., et al., *Attacker–defender models and road network vulnerability.* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2008. **366**(1872): p. 1893-1906.

54. Berche, B., et al., *Resilience of public transport networks against attacks.* The European Physical Journal B, 2009. **71**(1): p. 125-137.

55. Zhao, L., K. Park, and Y.C. Lai, *Attack vulnerability of scale-free networks due to cascading breakdown.* Physical Review E, 2004. **70**(3): p. 035101.

56. Jenelius, E., T. Petersen, and L.G. Mattsson, *Importance and exposure in road network vulnerability analysis.* Transportation Research Part A: Policy and Practice, 2006. **40**(7): p. 537-560.

57. Hendricks, D., J. Fell, and M. Freedman, *The relative frequency of unsafe driving acts in serious traffic crashes*. 2001: National Highway Traffic Safety Administration Washington, DC.

58. Box, S., *New Data from VTTI provides insight into cell phone use and driving distraction.* Virginia Tech Transportation Institute, 2009.

59. Charlton, J.L., et al., *Older driver distraction: A naturalistic study of behaviour at intersections.* Accident Analysis & Prevention, 2013.

60. Cooper, J.M., et al., *An investigation of driver distraction near the tipping point of traffic flow stability.* Human Factors: The Journal of the Human Factors and Ergonomics Society, 2009. **51**(2): p. 261-268.

61. Salvucci, D.D., *Distraction Beyond the Driver: Predicting the Effects of In-Vehicle Interaction on Surrounding Traffic.* 2013.

62. Strayer, D.L. and F.A. Drews, *Profiles in driver distraction: Effects of cell phone conversations on younger and older drivers.* Human factors, 2004. **46**(4): p. 640-649.

63. Stutts, J., et al., *Driver's exposure to distractions in their natural driving environment.* Accident Analysis & Prevention, 2005. **37**(6): p. 1093-1101.

64. Tijerina, L., et al., *Driver distraction with wireless telecommunications and route guidance systems*. 2000.

65. Alm, H. and L. Nilsson, *Changes in driver behaviour as a function of handsfree mobile phones—A simulator study.* Accident Analysis & Prevention, 1994. **26**(4): p. 441-451.

66. Alm, H. and L. Nilsson, *The effects of a mobile telephone task on driver behaviour in a car following situation.* Accident Analysis & Prevention, 1995. **27**(5): p. 707-715.

67. Beede, K.E. and S.J. Kass, *Engrossed in conversation: The impact of cell phones on simulated driving performance.* Accident Analysis & Prevention, 2006. **38**(2): p. 415-421.

68. Burns, P., et al., *How dangerous is driving with a mobile phone? Benchmarking the impairment to alcohol.* TRL REPORT 547, 2002.

69. Haigney, D., R. Taylor, and S. Westerman, *Concurrent mobile (cellular) phone use and driving performance: task demand characteristics and compensatory processes.* Transportation Research Part F: Traffic Psychology and Behaviour, 2000. **3**(3): p. 113-121.

70. Laberge-Nadeau, C., et al., *Wireless telephones and the risk of road crashes.* Accident Analysis & Prevention, 2003. **35**(5): p. 649-660.

71. Muttart, J.W., et al., *Driving without a clue: Evaluation of driver simulator performance during hands-free cell phone operation in a work zone.* Transportation Research Record: Journal of the Transportation Research Board, 2007. **2018**(1): p. 9-14.

72. Redelmeier, D.A. and R.J. Tibshirani, *Association between cellular-telephone calls and motor vehicle collisions.* New England Journal of Medicine, 1997. **336**(7): p. 453-458.

73. Reed, M.P. and P.A. Green, *Comparison of driving performance on-road and in a low-cost simulator using a concurrent telephone dialling task.* Ergonomics, 1999. **42**(8): p. 1015-1037.

74. Rosenbloom, T., *Driving performance while using cell phones: an observational study.* Journal of Safety Research, 2006. **37**(2): p. 207-212.

75. Violanti, J.M., *Cellular phones and fatal traffic collisions.* Accident Analysis & Prevention, 1998. **30**(4): p. 519-524.

76. Wenners, K.E., et al. *Large-Scale Observational Study of Drivers' Cell Phone Usage*. in *Transportation Research Board 92nd Annual Meeting*. 2013.

77. Green, P. *The 15-second rule for driver information systems*. in *Proceedings of the ITS America Ninth Annual Meeting*. 1999.

78. Salvucci, D.D., *Modeling driver behavior in a cognitive architecture.* Human Factors: The Journal of the Human Factors and Ergonomics Society, 2006. **48**(2): p. 362-380.

79. Salvucci, D.D. and N.A. Taatgen, *Threaded cognition: an integrated theory of concurrent multitasking.* Psychological Review, 2008. **115**(1): p. 101.

80. Tatano, H. and S. Tsuchiya, *A framework for economic loss estimation due to seismic transportation network disruption: A spatial computable general equilibrium approach.* Natural Hazards, 2008. **44**(2): p. 253-265.

81. Stavrinos, D., et al., *Impact of distracted driving on safety and traffic flow.* Accident Analysis & Prevention, 2013.

82. Pastor-Satorras, R. and A. Vespignani, *Epidemic spreading in scale-free networks.* Physical Review Letters, 2001. **86**(14): p. 3200-3203.

83. Wang, Y., et al. *Epidemic spreading in real networks: An eigenvalue viewpoint.* in *Reliable Distributed Systems, 2003. Proceedings. 22nd International Symposium on*. 2003. IEEE.

84. Muchnik, L., et al., *Self-emergence of knowledge trees: Extraction of the Wikipedia hierarchies.* Physical Review E, 2007. **76**(1): p. 016106.

85. NHTSA, *Distracted Driving 2009*. 2010, NHTSA's National Center for Statistics and Analysis: 1200 New Jersey Avenue SE., Washington, DC 20590.

86. NHTSA, *Distracted Driving 2010*. 2012, NHTSA's National Center for Statistics and Analysis: 1200 New Jersey Avenue SE., Washington, DC 20590.

87. NHTSA, *Distracted Driving 2011*. 2013, NHTSA's National Center for Statistics and Analysis: 1200 New Jersey Avenue SE., Washington, DC 20590.

88. NHTSA, *Driver Electronic Device Use in 2011*. 2013, NHTSA's National Center for Statistics and Analysis: 1200 New Jersey Avenue SE., Washington, DC 20590.

89. Tsekeris, T. and N. Geroliminis, *City size, network structure and traffic congestion.* Journal of Urban Economics, 2013.

90. Ducruet, C. and I. Lugo, *Structure and dynamics of transportation networks: Models, methods, and applications.* 2011.

91. Marchiori, M. and V. Latora, *Harmony in the small-world.* Physica A: Statistical Mechanics and its Applications, 2000. **285**(3–4): p. 539-546.

92. Kurant, M. and P. Thiran, *Extraction and analysis of traffic and topologies of transportation networks.* Physical Review E, 2006. **74**(3): p. 036114.

93. Latora, V. and M. Marchiori, *Efficient Behavior of Small-World Networks.* Physical Review Letters, 2001. **87**(19): p. 198701.

94. Latora, V. and M. Marchiori, *Is the Boston subway a small-world network?* Physica A: Statistical Mechanics and its Applications, 2002. **314**(1–4): p. 109-113.

95. Seaton, K.A. and L.M. Hackett, *Stations, trains and small-world networks.* Physica A: Statistical Mechanics and its Applications, 2004. **339**(3–4): p. 635-644.

96. Vragović, I., E. Louis, and A. Díaz-Guilera, *Efficiency of informational transfer in regular and complex networks.* Physical Review E, 2005. **71**(3): p. 036122.

97. Xu, Z. and D. Sui, *Small-world characteristics on transportation networks: a perspective from network autocorrelation.* Journal of Geographical Systems, 2007. **9**(2): p. 189-205.

98. Lee, K., et al., *Statistical analysis of the Metropolitan Seoul Subway System: Network structure and passenger flows.* Physica A: Statistical Mechanics and its Applications, 2008. **387**(24): p. 6231-6234.

99. Li, W. and X. Cai, *Statistical analysis of airport network of China.* Physical Review E, 2004. **69**(4): p. 046106.

100. Guimerá, R. and L.A.N. Amaral, *Modeling the world-wide airport network.* The European Physical Journal B - Condensed Matter and Complex Systems, 2004. **38**(2): p. 381-385.

101. Guida, M. and F. Maria, *Topology of the Italian airport network: A scale-free small-world network with a fractal structure?* Chaos, Solitons &amp; Fractals, 2007. **31**(3): p. 527-536.

102. WU, J., et al., *URBAN TRANSIT SYSTEM AS A SCALE-FREE NETWORK.* Modern Physics Letters B, 2004. **18**(19n20): p. 1043-1049.

103. von Ferber, C., et al., *Network harness: Metropolis public transport.* Physica A: Statistical Mechanics and its Applications, 2007. **380**(0): p. 585-591.

104. von Ferber, C., et al., *Public transport networks: empirical analysis and modeling.* The European Physical Journal B, 2009. **68**(2): p. 261-275.

105. Xu, X., J. Hu, and F. Liu, *Empirical analysis of the ship-transport network of China.* Chaos: An Interdisciplinary Journal of Nonlinear Science, 2007. **17**(2): p. 023129-9.

106. Hu, Y. and D. Zhu, *Empirical analysis of the worldwide maritime transportation network.* Physica A: Statistical Mechanics and its Applications, 2009. **388**(10): p. 2061-2071.

107. Erdos, P. and A. Renyi, *On random graphs.* Publ. Math. Debrecen, 1959. **6**: p. 290--297.

108. Kochen, M., *The Small World* 1989: Ablex, Norwood, NJ.

109. Newman, M.E.J., S.H. Strogatz, and D.J. Watts, *Random graphs with arbitrary degree distributions and their applications.* Physical Review E, 2001. **64**(2): p. 026118.

110. Barrat, A. and M. Weigt, *On the properties of small-world network models.* The European Physical Journal B - Condensed Matter and Complex Systems, 2000. **13**(3): p. 547-560.

111. Barabási, A.-L. and R. Albert, *Emergence of Scaling in Random Networks.* Science, 1999. **286**(5439): p. 509-512.

112. Watts, D.J., *Small Worlds: The Dynamics of Networks between Order and Randomness*. 1999: Princeton University Press, Princeton,.

113. Dorogovtsev, S.N., J.F.F. Mendes, and A.N. Samukhin, *Structure of Growing Networks with Preferential Linking.* Physical Review Letters, 2000. **85**(21): p. 4633-4636.

114. Ortuzar, J. and L.G. Willumsen, *Modelling transport*. 1994.

115. Peeta, S. and A.K. Ziliaskopoulos, *Foundations of dynamic traffic assignment: The past, the present and the future.* Networks and Spatial Economics, 2001. **1**(3): p. 233-265.

116. Balakrishna, R., *Off-line calibration of dynamic traffic assignment models*. 2006, Massachusetts Institute of Technology.

117. Chiu, Y.-C., E.J. Nava, and H.-H. Hu, *A Temporal Domain Decomposition Algorithmic Scheme for Efficient Mega-Scale Dynamic Traffic Assignment – An Experience with Southern California Associations of Government DTA Model*. 2012: SCAG.

118. Balakrishna, R., et al., *Simulation-based framework for transportation network management in emergencies.* Transportation Research Record: Journal of the Transportation Research Board, 2008. **2041**(-1): p. 80-88.

119. Mahmassani, H.S., *Dynamic network traffic assignment and simulation methodology for advanced system management applications.* Networks and Spatial Economics, 2001. **1**(3): p. 267-292.

120. Florian, M., *Models and software for urban and regional transportation planning: the contributions of the center for research on transportation.* INFOR: Information Systems and Operational Research, 2008. **46**(1): p. 29-50.

121. Barceló, J. and J. Casas, *Stochastic heuristic dynamic assignment based on AIMSUN microscopic traffic simulator.* Transportation Research Record: Journal of the Transportation Research Board, 2006. **1964**(-1): p. 70-80.

122. Van Aerde, M., et al., *INTEGRATION: An overview of traffic simulation features.* Transportation Research Records, 1996.

123. De Palma, A. and F. Marchal, *Real cases applications of the fully dynamic METROPOLIS tool-box: an advocacy for large-scale mesoscopic transportation systems.* Networks and Spatial Economics, 2002. **2**(4): p. 347-369.

124. Lighthill, M. and G. Whitham, *On kinematic waves. I. Flood movement in long rivers.* Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 1955. **229**(1178): p. 281-316.

125. Daganzo, C.F., *The cell transmission model, part II: network traffic.* Transportation Research Part B: Methodological, 1995. **29**(2): p. 79-93.

126. AG, P.P.T.V., *VISSIM 4.10 User Manual.* Karlsruhe, Germany, 2005.

127. Smith, M., G. Duncan, and S. Druitt. *PARAMICS: microscopic traffic simulation for congestion management*. in *Dynamic Control of Strategic Inter-Urban Road Networks, IEE Colloquium on*. 1995. IET.

128. Chiu, Y.-C., L. Zhou, and H. Song, *Development and calibration of the Anisotropic Mesoscopic Simulation model for uninterrupted flow facilities.* Transportation Research Part B: Methodological, 2010. **44**(1): p. 152-174.

129. Leonard, D., P. Gower, and N.B. Taylor, *CONTRAM: structure of the model.* Research report-Transport and Road Research Laboratory, 1989(178).

130. Celikoglu, H.B. and M. Dell'Orco, *Mesoscopic simulation of a dynamic link loading process.* Transportation Research Part C: Emerging Technologies, 2007. **15**(5): p. 329-344.

131. Dell'Orco, M., *A dynamic network loading model for mesosimulation in transportation systems.* European journal of operational research, 2006. **175**(3): p. 1447-1454.

132. Cardillo, A., et al., *Structural properties of planar graphs of urban street patterns.* Physical Review E, 2006. **73**(6): p. 066107.

133. Clark, C., *Urban population densities.* Journal of the Royal Statistical Society. Series A (General), 1951. **114**(4): p. 490-496.

134. Courtat, T., C. Gloaguen, and S. Douady, *Mathematics and morphogenesis of cities: A geometrical approach.* Physical Review E, 2011. **83**(3): p. 036106.

135. Levinson, D. and A. El-Geneidy, *The minimum circuity frontier and the journey to work.* Regional science and urban economics, 2009. **39**(6): p. 732-738.

136. Chung, F.R.K., *Spectral graph theory*. Vol. 92. 1997: Amer Mathematical Society.

137. Crucitti, P., V. Latora, and S. Porta, *Centrality measures in spatial networks of urban streets.* Physical Review E, 2006. **73**(3): p. 036125.

138. Scellato, S., et al., *The backbone of a city.* The European Physical Journal B-Condensed Matter and Complex Systems, 2006. **50**(1): p. 221-225.

139. Everitt, B.S., *Finite mixture distributions.* Encyclopedia of statistics in behavioral science, 2005.

140. Paolo Masucci, P., et al., *Random planar graphs and the London street network.* 2009.

141. Wang, X.F. and G. Chen, *Synchronization in scale-free dynamical networks: robustness and fragility.* arXiv preprint cond-mat/0105014, 2001.

142. Adamic, L.A., et al., *Search in power-law networks.* Physical Review E, 2001. **64**(4): p. 046135.

143. Kuperman, M. and G. Abramson, *Small world effect in an epidemiological model.* Physical Review Letters, 2001. **86**(13): p. 2909-2912.

144. Newman, M.E.J., C. Moore, and D.J. Watts, *Mean-field solution of the small-world network model.* Physical Review Letters, 2000. **84**(14): p. 3201-3204.

145. Szabó, G., M. Alava, and J. Kertész, *Shortest paths and load scaling in scale-free trees.* Physical Review E, 2002. **66**(2): p. 026101.

146. Tang, Y., W.H.K. Lam, and P.L.P. Ng, *Comparison of four modeling techniques for short-term AADT forecasting in Hong Kong.* Journal of Transportation Engineering, 2003. **129**(3): p. 271-277.

147. Zhao, F. and S. Chung, *Contributing factors of annual average daily traffic in a Florida county: Exploration with geographic information system and regression models.* Transportation Research Record: Journal of the Transportation Research Board, 2001. **1769**(-1): p. 113-122.

148. Zhao, F. and N. Park, *Using geographically weighted regression models to estimate annual average daily traffic.* Transportation Research Record: Journal of the Transportation Research Board, 2004. **1879**(-1): p. 99-107.

149. Selby, B. and K. Kockelman. *Spatial Prediction of AADT In Unmeasured Locations By Universal Kriging*. in *Transportation Research Board 90th Annual Meeting*. 2011.

150. Barthélemy, M., *Spatial networks.* Physics Reports, 2011. **499**(1–3): p. 1-101.

151. Herrera, J.C., et al., *Dynamic estimation of OD matrices for freeways and arterials*. 2007: Institute of Transportation Studies, UC Berkeley.

152. Herrera, J.C., et al., *Evaluation of traffic data obtained via GPS-enabled mobile phones: The< i> Mobile Century</i> field experiment.* Transportation Research Part C: Emerging Technologies, 2010. **18**(4): p. 568-583.

153. Andrienko, G., et al. *Interactive visual clustering of large collections of trajectories*. in *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*. 2009. IEEE.

154. Csardi, G. and T. Nepusz, *The igraph software package for complex network research.* InterJournal, Complex Systems, 2006. **1695**(5).

155. Kohavi, R. and G.H. John, *Wrappers for feature subset selection.* Artif. Intell., 1997. **97**(1-2): p. 273-324.

156. Hair, J.F., et al., *Multivariate data analysis*. Vol. 7. 2010: Prentice Hall Upper Saddle River, NJ.

157. Borga, M., *Canonical correlation: a tutorial.* On line tutorial http://people. imt. liu. se/magnus/cca, 2001. **4**.

158. Hardoon, D.R., S. Szedmak, and J. Shawe-Taylor, *Canonical correlation analysis: An overview with application to learning methods.* Neural Computation, 2004. **16**(12): p. 2639-2664.

159. Dehon, C., P. Filzmoser, and C. Croux, *Robust methods for canonical correlation analysis*, in *Data analysis, classification, and related methods*. 2000, Springer. p. 321-326.

160. Singh, A., et al., *Performance of Multi Model Canonical Correlation Analysis (MMCCA) for prediction of Indian summer monsoon rainfall using GCMs output.* Comptes Rendus Geoscience, 2013. **345**(2): p. 62-72.

161. Singh, A., et al., *Prediction of Indian summer monsoon rainfall (ISMR) using canonical correlation analysis of global circulation model products.* Meteorological Applications, 2012. **19**(2): p. 179-188.

162. Bonner, A. and H. Liu. *Canonical correlation, an approximation, and the prediction of protein abundance*. in *Proceedings of the Eighth Workshop on Mining Scientific and Engineering Datasets (MSD'05)*. 2005.

163. BPR, B.o.P.R., *Traffic Assignment Manual*, U.P.D. US Department of Commerce, Washington D.C., Editor. 1964.

164. GROUP, P., *VISUM 12.5 Fundamentals*. 2012: epubli.

165. Spiess, H., *Technical note—Conical volume-delay functions.* Transportation Science, 1990. **24**(2): p. 153-158.

166. West, G.B. and J.H. Brown, *The origin of allometric scaling laws in biology from genomes to ecosystems: towards a quantitative unifying theory of biological structure and organization.* Journal of Experimental Biology, 2005. **208**(9): p. 1575-1592.

167. Smith, B.L., L. Qin, and R. Venkatanarayana, *Characterization of freeway capacity reduction resulting from traffic accidents.* Journal of Transportation Engineering, 2003. **129**(4): p. 362-368.

168. Bell, M.G.H., *A game theory approach to measuring the performance reliability of transport networks.* Transportation Research Part B: Methodological, 2000. **34**(6): p. 533-545.

169. Dunn, S. and S. Wilkinson, *Identifying Critical Components in Infrastructure Networks Using Network Topology.* Journal of Infrastructure Systems, 2013. **19**(2): p. 157-165.

170. Wesemann, L., et al., *Cost-of-delay studies for freeway closures caused by Northridge earthquake.* Transportation Research Record: Journal of the Transportation Research Board, 1996. **1559**(-1): p. 67-75.

171. Dehghani, M., G. Flintsch, and S. McNeil, *Impact of Road Conditions and Disruption Uncertainties on Network Vulnerability.* Journal of Infrastructure Systems, 2014. **0**(0): p. 04014015.

172. Sullivan, J., et al., *Identifying critical road segments and measuring system-wide robustness in transportation networks with isolating links: A link-based capacity-reduction approach.* Transportation Research Part A: Policy and Practice, 2010. **44**(5): p. 323-336.

173. Li, J. and K. Ozbay, *Evaluation of Link Criticality for Day-to-Day Degradable Transportation Networks.* Transportation Research Record: Journal of the Transportation Research Board, 2012. **2284**(-1): p. 117-124.

174. Taylor, M.A.P., S.V.C. Sekhar, and G.M. D'Este, *Application of accessibility based methods for vulnerability analysis of strategic road networks.* Networks and Spatial Economics, 2006. **6**(3): p. 267-291.

175. Sohn, J., *Evaluating the significance of highway network links under the flood damage: An accessibility approach.* Transportation Research Part A: Policy and Practice, 2006. **40**(6): p. 491-506.

176. Knoop, V., H. van Zuylen, and S. Hoogendoorn, *The influence of spillback modelling when assessing consequences of blockings in a road network.* EJTIR, 2008. **4**(8).

177. Matisziw, T.C. and A.T. Murray, *Modeling s–t path availability to support disaster vulnerability assessment of network infrastructure.* Computers & Operations Research, 2009. **36**(1): p. 16-26.

178. Murray, A.T., T.C. Matisziw, and T.H. Grubesic, *A methodological overview of network vulnerability analysis.* Growth and Change, 2008. **39**(4): p. 573-592.

179. FHWA, *Our Nation's Highways 2008*. 2008, The US Department of Transportation, Federal Highway Administration.

180. Malik, K., A. Mittal, and S.K. Gupta, *The< i> k</i> most vital arcs in the shortest path problem.* Operations Research Letters, 1989. **8**(4): p. 223-227.

181. Nagurney, A. and Q. Qiang, *A network efficiency measure with application to critical infrastructure networks.* Journal of Global Optimization, 2008. **40**(1-3): p. 261-275.

182. Angeloudis, P. and D. Fisk, *Large subway systems as complex networks.* Physica A: Statistical Mechanics and its Applications, 2006. **367**: p. 553-558.

183. Dueñas-Osorio, L., et al., *Interdependent Response of Networked Systems.* Journal of Infrastructure Systems, 2007. **13**(3): p. 185-194.

184. Gonzalez, M.C., C.A. Hidalgo, and A.-L. Barabasi, *Understanding individual human mobility patterns.* Nature, 2008. **453**(7196): p. 779-782.

185. Song, C., et al., *Limits of predictability in human mobility.* Science, 2010. **327**(5968): p. 1018-1021.

186. Winkler, J., et al., *Performance assessment of topologically diverse power systems subjected to hurricane events.* Reliability Engineering & System Safety, 2010. **95**(4): p. 323-336.

187. Newman, M.E., *Modularity and community structure in networks.* Proceedings of the National Academy of Sciences, 2006. **103**(23): p. 8577-8582.

188. Cormen, T.H., et al., *Introduction to algorithms*. Vol. 2. 2001: MIT press Cambridge.

189. Fortunato, S., *Community detection in graphs.* Physics Reports, 2010. **486**(3): p. 75-174.

190. Newman, M.E. and M. Girvan, *Finding and evaluating community structure in networks.* Physical review E, 2004. **69**(2): p. 026113.

191. Scott, D.M., et al., *Network robustness index: A new method for identifying critical links and evaluating the performance of transportation networks.* Journal of Transport Geography, 2006. **14**(3): p. 215-227.

192. Chen, Z., et al., *Analysis on Vehicle Motion Parameters of Distracted Driving*, in *ICTIS 2013*. 2013. p. 1692-1697.

193. Fitch, G.M., et al., *The Impact of Hand-Held and Hands-Free Cell Phone Use on Driving Performance and Safety-Critical Event Risk*. 2013.

194. Wilson, F.A. and J.P. Stimpson, *Trends in fatalities from distracted driving in the United States, 1999 to 2008.* American Journal of Public Health, 2010. **100**(11): p. 2213-2219.

195. Donmez, B., L. Boyle, and J. Lee, *Differences in Off-Road Glances: Effects on Young Drivers' Performance.* Journal of Transportation Engineering, 2010. **136**(5): p. 403-409.

196. Wickens, C.D., *Multiple resources and performance prediction.* Theoretical issues in ergonomics science, 2002. **3**(2): p. 159-177.

197. Yannis, G., et al., *Older Drivers' Perception and Acceptance of In-Vehicle Devices for Traffic Safety and Traffic Efficiency.* Journal of Transportation Engineering, 2010. **136**(5): p. 472-479.

198. Horrey, W.J. and C.D. Wickens, *Driving and side task performance: The effects of display clutter, separation, and modality.* Human Factors: The Journal of the Human Factors and Ergonomics Society, 2004. **46**(4): p. 611-624.

199. Hankey, J.M., et al., *In-vehicle information systems behavioral model and design support: Final report*. 2000.

200. Anderson, J.R., *How Can the Human Mind Occur in the Physical Universe?* 2007: Oxford University Press, USA.

201. Manual, H.C., *Transportation research board.* National Research Council, Washington, DC, 2000. **113**.

202. Bloomberg, L. and J. Dale, *Comparison of VISSIM and CORSIM traffic simulation models on a congested network.* Transportation Research Record: Journal of the Transportation Research Board, 2000. **1727**(1): p. 52-60.

203. PennDOT, *Pennsylvania Driver's Manual*, D.o. Transportation, Editor. 2013, Pennsylvania Department of Transportation: Pennsylvania Liquor, Harrisburg, PA.

204. Peaeson, E. and H. Haetlet, *Biometrika tables for statisticians*. Biometrika Trust. Vol. 2. 1976: Cambridge University Press.

Appendix A: List of Python Codes

Build a graph in i-Graph (RQ1 & RQ2)

```
# <<<<<<<<<<< Import needed modules >>>>>>>>>>>
from igraph import *
# <<<<<<<<<<< MAIN >>>>>>>>>>>
# Build the Graph
g = Graph()
# Number of Nodes
g.add_vertices(51772)
# List of links (Node ID starts from 0)
g.add_edges([(26219, 27169),
(30551, 29357),
(29366, 30551),
(30551, 29366)
…

(25553, 25635),
(25635, 25553),
(10816, 10815)])
# Assign Weights to links:
gw = [1500,
680,
680,
680,
…

500,
500,
680]
```

Calculate the structural attributes of a road network (RQ1)

```
# <<<<<<<<<<< Import needed modules >>>>>>>>>>>
import numpy
import xlrd
from xlwt import *
import sys
from igraph import *
from operator import itemgetter, attrgetter
import matplotlib.pyplot as plt
```

```
# <<<<<<<<<< Define Path >>>>>>>>>>
if "C:\NOURZAD\RQ1Results\OD" not in sys.path:
   sys.path.append("C:\NOURZAD\RQ1Results")

# <<<<<<<<<< MAIN >>>>>>>>>>
# Import weighted graph in which the weights are a function of capacity and traffic volume:
import GraphPhillyCap11 as gr    # Import the graph which is built in another module
g = gr.g                          # g is the graph
gw = gr.gw                        # gw is the vector of weights

# Calculate the largest eigenvalue of the weighted graph:
eigen = g.evcent(directed=True, scale=True, weights=gw, return_eigenvalue=True)
eigenMax = eigen[1]

# Find the weighted degree (strength) of nodes + their Mean and St Dev:
degreeOfNodes = g.strength(weights=gw)
degreeMean = numpy.mean(degreeOfNodes)
degreeStd = numpy.std(degreeOfNodes)

# Find the weighted betweenness of nodes + their Mean and St Dev:
bw = g.betweenness(weights=gw)
bwMean = numpy.mean(bw)
bwStd = numpy.std(bw)

# Write into an excel file
w = Workbook()
ws = w.add_sheet('Structure')
ws.write(0, 0, eigenMax)
ws.write(0, 1, degreeMean)
ws.write(0, 2, degreeStd)
ws.write(0, 3, bwMean)
ws.write(0, 4, bwStd)
w.save('C:\NOURZAD\RQ1Results\Structure11.xls')
```

Calculate the criticality of links and cluster them (RQ2)

```
# <<<<<<<<<< Import needed modules >>>>>>>>>>
from igraph import *
import numpy as np
from operator import itemgetter, attrgetter
import matplotlib.pyplot as plt

# <<<<<<<<<< Define functions >>>>>>>>>>
def drange(start, stop, step):
   r = start
   while r < stop:
      yield r
      r += step

# <<<<<<<<<< 1st STEP: Calculate the criticality of individual links >>>>>>>>>>
# Import the weighted graph:
import GraphPhilly    # import the graph which is built in GraphPhilly.py
g = GraphPhilly.g
gw = GraphPhilly.gw
```

```
# Assumed severity of disruption:
severity = 0.6

# Calculate the largest eigenvalue of the graph in normal conditions:
eigen = g.evcent(directed=True, scale=True, weights=gw, return_eigenvalue=True)
currentEigenMax = eigen[1]
NumberOfLinks = g.ecount()
linkfound = 0
criticalLinks = []
Eigenvalues = []
largestEigenvalueExhaust = []
largestEigenvalue = currentEigenMax

# Determine threshold for minimum Eigenvalue, based on severity:
thresholdEigenvalue = (1-severity) * largestEigenvalue * (1 + 0.25 * severity)

# Put the links' ID (from 0 to the number of links -1 ) in linksLeft vector:
linksLeft = [e for e in range(g.ecount())]

# The adjusted weight of links:
gwadj = list(gw)

# Find the weighted degree (strength) of each link:
degreeOfNodes = g.strength(weights=gwadj)
degreeOfLinks = [0 for x in xrange(g.ecount())]
for e in range(NumberOfLinks):
    edgeNode1 = g.get_edgelist()[e][0]
    edgeNode2 = g.get_edgelist()[e][1]
    degreeOfLinks[e]=(degreeOfNodes[edgeNode1] + degreeOfNodes[edgeNode2])/2
sortedAllLinks = [i[0] for i in sorted(enumerate(degreeOfLinks), key=lambda x:x[1],
reverse=True)]

# Step-by-step, find the critical links and their associated largest eigenvalues:
done = False
for fractionOfDisruption in drange(0.0, 1.01, 0.01):
    print('fractionOfDisruption',fractionOfDisruption)

    # Compute number of links to be disrupted:
    fraction = int(fractionOfDisruption*NumberOfLinks)
    if (fraction-linkfound) == 0:
        largestEigenvalueExhaust.append([fractionOfDisruption, largestEigenvalue])
    elif(done == False):

        # Select critical links to be disrupted one-by-one:
        for j in range(linkfound, fraction):

            # Find the k which is corresponding to Max drop in largest eigenvalue:
            kcritical = int(sortedAllLinks[j])

            # Adjust the weight of the "kcritical" link and calculate its largest eigenvalue:
            gwadj[kcritical] = gwadj[kcritical] * (1-severity)
            eigentemp = g.evcent(directed=True, scale=True, weights=gwadj,
return_eigenvalue=True)
            largestEigenvalue = eigentemp[1]
            print('largestEigenvalue',largestEigenvalue)
            criticalLinks.append(kcritical)
            Eigenvalues.append(largestEigenvalue)
            # Remove it from the list of links left:
            linksLeft.remove(kcritical)

            # If I reach the threshold (for Min eigenvalue), stop the process:
            if(largestEigenvalue < thresholdEigenvalue):
                done = True

        # In the next run, start from previous fraction:
        linkfound = fraction

        # Append the largest eigenvalue to the list:
        largestEigenvalueExhaust.append([fractionOfDisruption, largestEigenvalue])
    else:

        # Once I reach threshold for Min eigenvalue, append the latest largest eigenvalue for the rest:
        largestEigenvalueExhaust.append([fractionOfDisruption, largestEigenvalue])

# If there is yet more links left, add them to the list with adding the largest eigenvalue too:
numLinksLeft = len(linksLeft)
if (numLinksLeft > 0):
    for lk in xrange(numLinksLeft):
        linklft = linksLeft[0]
        criticalLinks.append(linklft)
        linksLeft.remove(linklft)
        Eigenvalues.append(largestEigenvalue)

# X: Fractions of disruption:
x = [ row[0] for row in largestEigenvalueExhaust ]

# Y: Largest eigenvalues associated with fractions of disruption:
y = [ row[1] for row in largestEigenvalueExhaust ]

# Write disruption sizes, largest eigenvalues, sorted list of links based on their criticality, their
associated eigenvalues:
f = open(r"C:\NOURZAD\RQ2Results\DisruptionResults.txt", 'w')
f.write('disruption size:\n')
f.write(str(x) + '\n')
f.write('Largest Eigenvalue associated to each disruption size:\n')
f.write(str(y) + '\n')
f.write('Sorted list of critical links=\n')
f.write(str(criticalLinks) + '\n')
f.write('List of eigenvalues associated to links=\n')
f.write(str(Eigenvalues) + '\n')
f.close()
print('End of Phase I')

# The output of previous step is the list of critical links
######################################################################
# <<<<<<<<<< 2nd STEP: Cluster the links >>>>>>>>>>
# Community Detection
# EigenDif = Eigen - Min(Eigen)
EigenDif = [ e - min(Eigenvalues) for e in Eigenvalues ]

# this way:    0 < criticalityIndex < Severity
criticalityIndex = [ ed/max(Eigenvalues) for ed in EigenDif ]

# Find subset of critical links:
criticalSubsetOfLinks = []
```

```
# If criticalityIndex > 10% => Link is CRITICAL:
for ln in xrange(len(criticalityIndex)):
    if criticalityIndex[ln]> 0.1:
        criticalSubsetOfLinks.append(criticalLinks[ln])
#==>> Find subset of critical nodes:
# (The modularity optimization methods take the nodes' weights, rather than links' weights)
nodeIsCritical = [0 for x in xrange(g.vcount())]
for ec in criticalSubsetOfLinks:
    edgeNode1 = g.get_edgelist()[ec][0]
    edgeNode2 = g.get_edgelist()[ec][1]
    nodeIsCritical[edgeNode1] = 1
    nodeIsCritical[edgeNode2] = 1
criticalSubsetOfNodes = []
for vx in xrange(len(nodeIsCritical)):
    if(nodeIsCritical[vx] == 1):
        criticalSubsetOfNodes.append(vx)
# Write the critical subset of nodes in a file, "criticalNodes.txt":
fn = open(r"C:\NOURZAD\RQ2Results\criticalNodes.txt", 'w')
fn.write(str(criticalSubsetOfNodes))
fn.close()
print('End of Finding subset of critical nodes')
#==>> Weights of links based on their criticality
coef = 0.01
criticalityFloor = coef * max(EigenDif)
criticality = [ max(ed, criticalityFloor) for ed in EigenDif ]
# Sort the weights based on the link IDs:
WeightsBasedOnLinksId = [0 for x in xrange(len(criticality))]
for ln in xrange(len(criticalLinks)):
    WeightsBasedOnLinksId[criticalLinks[ln]] = criticality[ln]
# Calculate the criticality of nodes = Sum of criticality of connected links:
criticalityOfNodes = g.strength(weights=WeightsBasedOnLinksId)
# Detect modules of the weighted network:
modules = g.community_multilevel(weights=WeightsBasedOnLinksId)
# Number of detected modules:
modulesNum = len(modules)
# Write the modules in a file, "DetectedModules.txt":
fm = open(r"C:\NOURZAD\RQ2Results\DetectedModules.txt", 'w')
for ln in xrange(len(modules)):
    fm.write(str(modules[ln]) + '\n')
fm.close()
print('End of Modularity Detection')
#==>> Check whether the modules separate critical and non-critical nodes:
moduleCriticality = list(modules)
for mj in xrange(len(modules)):
    for li in xrange(len(modules[mj])):
        if modules[mj][li] in criticalSubsetOfNodes:
            moduleCriticality[mj][li] = 1
        else:
         moduleCriticality[mj][li] = 0
# Write the criticality status (0 or 1) of each node in a module:
fy = open(r"C:\NOURZAD\RQ2Results\DetectedModulesCriticality.txt", 'w')
for ln in xrange(len(modules)):
   fy.write(str(moduleCriticality[ln]) + '\n')
fy.close()
# Write the criticality ratios of all modules:
numberOfCriticalNodes = [sum(moduleCriticality[i]) for i in range(len(moduleCriticality))]
numberOfTotalNodes = [len(moduleCriticality[j]) for j in range(len(moduleCriticality))]
modulesCriticalityRatio = [float(numberOfCriticalNodes[k])/numberOfTotalNodes[k] for k
in range(len(numberOfCriticalNodes))]
fz = open(r"C:\NOURZAD\RQ2Results\DetectedModulesCriticalityRatio.txt", 'w')
fz.write(str(modulesCriticalityRatio))
fz.close()
print('End of Finding Criticality Ratio')
#==>> Calculate the absolute value of criticality for each modules:
moduleAbsCriticality = list(modules)
for mj in xrange(len(modules)):
   for vc in xrange(len(modules[mj])):
      moduleAbsCriticality[mj][vc] = criticalityOfNodes[modules[mj][vc]]
sumOfMjCriticality = [sum(moduleAbsCriticality[i]) for i in
range(len(moduleAbsCriticality))]
fx = open(r"C:\NOURZAD\RQ2Results\DetectedModulesCriticalityAbs.txt", 'w')
fx.write(str(sumOfMjCriticality))
fx.close()
print('End of Finging absolute value of criticality for each modules')
#==> Assign the Absolute value of criticality of modules to their nodes:
nodeAbsCriticalityOfItsMj = [0 for x in xrange(g.vcount())]
nodeNumberOfItsMj = [0 for x in xrange(g.vcount())]
for nd in xrange(g.vcount()):
   for mj in xrange(len(modules)):
      if(nd in modules[mj]):
         nodeAbsCriticalityOfItsMj[nd] = sumOfMjCriticality[mj]
         nodeNumberOfItsMj[nd] = mj
print('End of Assigning Criticality to Nodes')
# Assign the Absolute value of criticality of modules from nodes to the links:
# Link Criticality of Module = Max(Abs Crt assigned to two nodes of the link):
LinkAbsCriticalityOfItsMj = [0 for x in xrange(g.ecount())]
LinkNumberOfItsMj = [0 for x in xrange(g.ecount())]
for e in range(NumberOfLinks):
   edgeNode1 = g.get_edgelist()[e][0]
   edgeNode2 = g.get_edgelist()[e][1]
   LinkAbsCriticalityOfItsMj[e]=max(nodeAbsCriticalityOfItsMj[edgeNode1],
nodeAbsCriticalityOfItsMj[edgeNode2])
   if(nodeAbsCriticalityOfItsMj[edgeNode1]==LinkAbsCriticalityOfItsMj[e]):
      LinkNumberOfItsMj[e] = nodeNumberOfItsMj[edgeNode1]
   else:
      LinkNumberOfItsMj[e] = nodeNumberOfItsMj[edgeNode2]
f1 = open(r"C:\NOURZAD\RQ2Results\LinkAbsCriticalityOfItsMj.txt", 'w')
```

```
f1.write(str(LinkAbsCriticalityOfItsMj))
f1.close()
print('End of Assigning Criticality to Links')
# Write the Link' Criticality of its Module on excel:
from xlwt import *
w = Workbook()
ws = w.add_sheet('linkcriticalityOfMj')
for l in xrange(65536):
    ws.write(l, 0, LinkAbsCriticalityOfItsMj[l])
for l in xrange(65536, len(LinkAbsCriticalityOfItsMj)):
    ws.write(l-65536, 1, LinkAbsCriticalityOfItsMj[l])
w.save('linksCrtOfMj.xls')
# Write module number of the links on excel:
wm = Workbook()
wms = wm.add_sheet('linkNumOfMj')
for l in xrange(65536):
    wms.write(l, 0, LinkNumberOfItsMj[l])
for l in xrange(65536, len(LinkNumberOfItsMj)):
    wms.write(l-65536, 1, LinkNumberOfItsMj[l])
wm.save('linksNumOfMj.xls')
# Write the node' Criticality of its Module on excel:
wv = Workbook()
wsv = wv.add_sheet('nodesCriticalityOfMj')
for l in xrange(len(nodeAbsCriticalityOfItsMj)):
    wsv.write(l, 0, nodeAbsCriticalityOfItsMj[l])
wv.save('nodesCrtOfMj.xls')
```

Replicate Stavrinos's model for LOS A in VISSIM (RQ3)

```
##### Vehicle Type description:
    # Vehicles enter the network are either from Car or CarToBeDistracted
    # Car: Vehicles that will not be distracted
    # CarToBeDistracted: Vehicles that will be distracted in a time frame during simulation period
    # CarDistracted: The vehicles with distracted parameters
# Three vehicle classes are defined with the same names: Car, CarToBeDistracted, CarDistracted
# Link behavior type description:
    # Behavior of different classes of vehicles are different for each link type:
        # Driving behavior of Car: Freeway NonDistracted
        # Driving behavior of CarToBeDistracted: Freeway NonDistracted
        # Driving behavior of CarDistracted: Freeway Distracted

# <<<<<<<<<< Import needed modules >>>>>>>>>>
import numpy
import xlrd
import win32com.client
import random
# <<<<<<<<<< FUNCTIONS >>>>>>>>>>
# Function for randomly selection of weighted items
# Input: list of weights of items
# Output: index of selected item
def weighted_choice(weights):
    totals = []
    running_total = 0
    for w in weights:
        running_total += w
        totals.append(running_total)
    rnd = random.random() * running_total
    for i, total in enumerate(totals):
        if rnd < total:
            return i

# <<<<<<<<<< Read Distraction Table (Distract-R output) >>>>>>>>>>
# The profile durations are assumed to be (at least) twice the simulation duration (2*UBound)
# When a vehicle enters, I randomly assign a number 1 =< k <= UBound as the starting point
# So, its distraction time period is (k, k+UBound)
# Open the excel file which contains distraction results from Distract-R
workbook = xlrd.open_workbook('DistractionTable.xlsx')

# Open the worksheet 'Sheet1' which contains the data
worksheet = workbook.sheet_by_name('Sheet1')

# Read the distraction data and store it in a matrix
num_rows = worksheet.nrows - 1
num_cells = worksheet.ncols - 1
# Initiate DistractionTimeProfiles matrix (whose rows are distraction types, and columns are time steps)
# Distract-R output resolution is 0.05 sec and Vissim resolution is 0.1 sec, so I need to change the resolution
DistractionTimeProfiles = [[0 for x in xrange(num_rows/2)] for x in xrange(num_cells)]
# Store the data in DistractionTimeProfiles matrix: either distracted (1) or non-distracted (0) at the time step
curr_row = 0
while curr_row < num_rows-1:
            curr_row += 2
            curr_cell = 0
            while curr_cell < num_cells:
                        curr_cell += 1

                        # HINT1: During storing, I change the rows to columns and columns to rows
                        # HINT2: I remove the title row (row=0) and the time step column (column=0)
                        # So, each row is a distraction type, and each column is a time step
                        # Distract-R output resolution is 0.05 sec and Vissim resolution is 0.1 sec,
                        #  so I use 2*curr_cell to change the resolution.
                        if worksheet.cell_value(curr_row, curr_cell) == 1:
                    DistractionTimeProfiles[curr_cell-1][curr_row/2-1] = 0
                else:
                    DistractionTimeProfiles[curr_cell-1][curr_row/2-1] = 1

# Different types of distracted drivers
distractedTypes = len(DistractionTimeProfiles)

# % of each distraction type: A vector of weights (probabilities) of different distraction types
```

```
distProbabilities = [1]
# <<<<<<<<<< Initiate VISSIM model >>>>>>>>>>
Vissim = win32com.client.Dispatch("VISSIM.vissim")
Vissim.LoadNet(r"C:\NOURZAD\RQ3Results\PlatformStavrinos\Platform.inp")
# Enable evaluation parameters
Vissim.Evaluation.SetAttValue("NETPERFORMANCE", True)
Vissim.Evaluation.SetAttValue("LANECHANGE", True)
Vissim.Evaluation.SetAttValue("DELAY", True)
Vissim.Evaluation.SetAttValue("EXPORT", True)
Vissim.Evaluation.SetAttValue("VEHICLERECORD", True)
# <<<<<<<<<< Initiate parameters and variables >>>>>>>>>>
# Define max number of time steps and max number of vehicles
UBound = 6000
MaxNumVehicles = 15
# Status of vehicle: column 0: vehicle status (either distracted=1, or non-distracted=0),
#                    column 1: distraction type, column 2: distraction Start Step
distractionStatus = [[0 for x in xrange(3)] for x in xrange(MaxNumVehicles)]
# <<<<<<<<<< Set up multiple runs >>>>>>>>>>
NumberOfRuns = 51
Vissim.Simulation.RunIndex = 0
Vissim.Simulation.RandomSeed = 10
for numrun in range(0, NumberOfRuns):
   # Temporary for check:
   CheckNumEntered = 0
   CheckNumEnteredToBeDist = 0
   CheckChangetoDist = 0
   CheckChangetoNonDist = 0
   VehPrev = []
   VehCur = []
# <<<<<<<<<< Add vehicles >>>>>>>>>>
   Vissim.Simulation.RunSingleStep()
   # Vehicle #1
   Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,  3500)
   # Vehicle #2
   Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 2,  3250)
   # Vehicle #3
   Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,  3000)
   # Vehicle #4
   Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 2,  2750)
   # Vehicle #5
   Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,  2500)
   # Vehicle #6
   Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 2,  2250)
   # Vehicle #7 - Might be distracted
   Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (200, 58, 1, 1,  2000)
```

```
    # Vehicle #8
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 2,  1750)
    # Vehicle #9
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,  1500)
    # Vehicle #10
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 2,  1250)
    # Vehicle #11
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,  1000)
    # Vehicle #12
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 2,   750)
    # Vehicle #13
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,   500)
# <<<<<<<<<< Run Traffic simulation in VISSIM step-by-step >>>>>>>>>>
    for i in range(0, UBound):

        # For each vehicle enters the network in this time step:
        VehicleEntered = list(set(VehCur) - set(VehPrev))
        NumVehEnterInCurStep = len(VehicleEntered)
        if NumVehEnterInCurStep > 0:
            for number in VehicleEntered:
                Veh = Vissim.Net.Vehicles.GetVehicleByNumber(number)
                if Veh <> None:
                    CheckNumEntered = CheckNumEntered + 1

                    # If the vehicle type is "ToBeDistracted":
                    if Veh.AttValue("TYPE")== 200:
                        CheckNumEnteredToBeDist = CheckNumEnteredToBeDist + 1

                        # Assign a distraction time profile to the vehicle
                        disTypeNo = weighted_choice(distProbabilities)
                        distractionStatus[Veh.AttValue("ID")][1] = disTypeNo

                        # Assign a distraction start step to the vehicle
                        disStartStep = numpy.random.randint(1,UBound)
                        distractionStatus[Veh.AttValue("ID")][2] = disStartStep


                        # Define the distraction status array: 1=distracted
                        distractionStatus[Veh.AttValue("ID")][0] = 1
                    else:

                        # Define the distraction status array: 0=non-distracted
                        distractionStatus[Veh.AttValue("ID")][0] = 0

        # Get the number of vehicles in the simulation
        NumVehicles = Vissim.Net.Vehicles.Count

        # For each vehicle in the network
        if NumVehicles > 0:
            for Vehicle in Vissim.Net.Vehicles:
                Veh = Vissim.Net.Vehicles.GetVehicleByNumber(Vehicle.ID)
                if Veh <> None:

                    # If the vehicle type is "ToBeDistracted":
```

```
                if distractionStatus[Veh.AttValue("ID")][0] == 1:
                    VehDisType = distractionStatus[Veh.AttValue("ID")][1]
                    VehDisStart = distractionStatus[Veh.AttValue("ID")][2]
                    # If i is in its distracted time (based on its distraction time profile)
                    if DistractionTimeProfiles[VehDisType][i + VehDisStart-1] == 1:
                        # change the type of vehicle to "CarDistracted"
                        Veh.SetAttValue("TYPE", 300)
                        CheckChangetoDist = CheckChangetoDist + 1
                    else:
                        # change the type of vehicle to "CarToBeDistracted"
                        Veh.SetAttValue("TYPE", 200)
                        CheckChangetoNonDist = CheckChangetoNonDist + 1
    VehPrev = VehCur
    # Run simulation one single step
    Vissim.Simulation.RunSingleStep()
    NumVeh = Vissim.Net.Vehicles.Count
    if NumVeh > 0:
        VehCur=[]
        for Vehicle in Vissim.Net.Vehicles:
            if Vehicle <> None:
                VehCur.extend([Vehicle.ID])
Vissim.Simulation.RunIndex = Vissim.Simulation.RunIndex + 1
Vissim.Simulation.RandomSeed = Vissim.Simulation.RandomSeed + 1
print("End of Run#",Vissim.Simulation.RunIndex)
```

Replicate Stavrinos's model for LOS C in VISSIM (RQ3)

The same as the code for LOS A except for the vehicle volume as follows:

```
# <<<<<<<<<< Add vehicles >>>>>>>>>>
    Vissim.Simulation.RunSingleStep()
    # Vehicle #1
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,  1609)
    # Vehicle #2
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 2,  1477)
    # Vehicle #3
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,  1345)
    # Vehicle #4
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 2,  1213)
    # Vehicle #5
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,  1081)
    # Vehicle #6
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 2,  949)
    # Vehicle #7 - Might be distracted
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (200, 58, 1, 1,  817)
    # Vehicle #8
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 2,  685)
    # Vehicle #9
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,  553)
    # Vehicle #10
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 2,  421)
    # Vehicle #11
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,  289)
    # Vehicle #12
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 2,  157)
    # Vehicle #13
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,   25)
```

Replicate Stavrinos's model for LOS E in VISSIM (RQ3)

The same as the code for LOS A except for the vehicle volume as follows:

```
# <<<<<<<<<< Add vehicles >>>>>>>>>>
    Vissim.Simulation.RunSingleStep()
    # Vehicle #1
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 30, 1, 1,  398)
    # Vehicle #2
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 30, 1, 2,  367)
    # Vehicle #3
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 30, 1, 1,  336)
    # Vehicle #4
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 30, 1, 2,  305)
    # Vehicle #5
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 30, 1, 1,  273)
    # Vehicle #6
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 30, 1, 2,  242)
    # Vehicle #7 - Might be distracted
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 58, 1, 1,  211)
    # Vehicle #8
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 30, 1, 2,  180)
    # Vehicle #9
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 30, 1, 1,  140)
    # Vehicle #10
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 30, 1, 2,  118)
    # Vehicle #11
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 30, 1, 1,   87)
    # Vehicle #12
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 30, 1, 2,   56)
    # Vehicle #13
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (500, 30, 1, 1,   25)
```

Replicate Salvucci's model for Standard Scenario in VISSIM (RQ3)

```
##### Vehicle Type description:
    # Vehicles enter the network are either from Car or CarToBeDistracted
    # Car: Vehicles that will not be distracted
    # CarToBeDistracted: Vehicles that will be distracted in a time frame during simulation period
    # CarDistracted: The vehicles with distracted parameters
# Three vehicle classes are defined with the same names: Car, CarToBeDistracted, CarDistracted
# Link behavior type description:
    # Behavior of different classes of vehicles are different for each link type:
        # Driving behavior of Car: Freeway NonDistracted
        # Driving behavior of CarToBeDistracted: Freeway NonDistracted
        # Driving behavior of CarDistracted: Freeway Distracted


# <<<<<<<<<< Import needed modules >>>>>>>>>>
import numpy
import xlrd
import win32com.client
import random
# <<<<<<<<<< FUNCTIONS >>>>>>>>>>
# Function for randomly selection of weighted items
# Input: list of weights of items
# Output: index of selected item
def weighted_choice(weights):
    totals = []
    running_total = 0
    for w in weights:
        running_total += w
        totals.append(running_total)
    rnd = random.random() * running_total
    for i, total in enumerate(totals):
        if rnd < total:
            return i
# <<<<<<<<<<  Read Distraction Table (Distract-R output) >>>>>>>>>>
# The profile durations are assumed to be (at least) twice the simulation duration (2*UBound)
# When a vehicle enters, I randomly assign a number 1 =< k <= UBound as the starting point
# So, its distraction time period is (k, k+UBound)
# Open the excel file which contains distraction results from Distract-R
workbook = xlrd.open_workbook('DistractionTable.xlsx')
# Open the worksheet 'Sheet1' which contains the data
worksheet = workbook.sheet_by_name('Sheet1')
# Read the distraction data and store it in a matrix
num_rows = worksheet.nrows - 1
num_cells = worksheet.ncols - 1
```

```
# Initiate DistractionTimeProfiles matrix (whose rows are distraction types, and columns are time steps)
# Distract-R output resolution is 0.05 sec and Vissim resolution is 0.1 sec, so I need to change the resolution
DistractionTimeProfiles = [[0 for x in xrange(num_rows/2)] for x in xrange(num_cells)]

# Store the data in DistractionTimeProfiles matrix: either distracted (1) or nondistracted (0) at the time step
curr_row = 0
while curr_row < num_rows-1:
        curr_row += 2
        curr_cell = 0
        while curr_cell < num_cells:
                curr_cell += 1

                # HINT1: During storing, I change the rows to columns and columns to rows
                # HINT2: I remove the title row (row=0) and the time step column (column=0)
                # So, each row is a distraction type, and each column is a time step
                # Distract-R output resolution is 0.05 sec and Vissim resolution is 0.1 sec,
                #  so I use 2*curr_cell to change the resolution.
                if worksheet.cell_value(curr_row, curr_cell) == 1:
             DistractionTimeProfiles[curr_cell-1][curr_row/2-1] = 0
          else:
             DistractionTimeProfiles[curr_cell-1][curr_row/2-1] = 1

# Different types of distracted drivers
distractedTypes = len(DistractionTimeProfiles)

# % of each distraction type: A vector of weights (probabilities) of different distraction types
distProbabilities = [1]

# <<<<<<<<<< Initiate VISSIM model >>>>>>>>>>
Vissim = win32com.client.Dispatch("VISSIM.vissim")
Vissim.LoadNet(r"C:\NOURZAD\RQ3Results\PlatformSalvucci\PlatformSmall.inp")

# Enable evaluation parameters
Vissim.Evaluation.SetAttValue("NETPERFORMANCE", True)
Vissim.Evaluation.SetAttValue("LANECHANGE", True)
Vissim.Evaluation.SetAttValue("DELAY", True)
Vissim.Evaluation.SetAttValue("EXPORT", True)
Vissim.Evaluation.SetAttValue("VEHICLERECORD", True)

# <<<<<<<<<<  Initiate parameters and variables >>>>>>>>>>
# Define max number of time steps and max number of vehicles
UBound = 6000
MaxNumVehicles = 20

# Status of vehicle: column 0: vehicle status (either distracted=1, or nondistracted=0),
#                 column 1: distraction type, column 2: distraction Start Step
distractionStatus = [[0 for x in xrange(3)] for x in xrange(MaxNumVehicles)]

# <<<<<<<<<< Set up multiple runs >>>>>>>>>>
NumberOfRuns = 11
Vissim.Simulation.RunIndex = 0
Vissim.Simulation.RandomSeed = 40
for numrun in range(0, NumberOfRuns):
    # Temporary for check:
    CheckNumEntered = 0
    CheckNumEnteredToBeDist = 0
    CheckChangetoDist = 0
    CheckChangetoNonDist = 0
    VehPrev = []
    VehCur = []
    # <<<<<<<<<<<  Add vehicles >>>>>>>>>>>
    Vissim.Simulation.RunSingleStep()
    # Lead vehicle
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 29.83, 1, 1, 1049.869)
    # Vehicle #1
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  984.252)
    # Vehicle #2
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  918.635)
    # Vehicle #3
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  853.018)
    # Vehicle #4
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  787.402)
    # Vehicle #5
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  721.785)
    # Vehicle #6
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  656.168)
    # Vehicle #7
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  590.551)
    # Vehicle #8
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  524.934)
    # Vehicle #9
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  459.318)
    # Vehicle #10
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  393.701)
    # Vehicle #11
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  328.084)
    # Vehicle #12
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  262.467)
    # Vehicle #13
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  196.850)
    # Vehicle #14
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,  131.234)
    # Vehicle #15
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 1, 1,   65.617)
# <<<<<<<<<< Run Traffic simulation in VISSIM step-by-step >>>>>>>>>>
    for i in range(0, UBound):
        # For each vehicle enters the network in this time step:
```

```
VehicleEntered = list(set(VehCur) - set(VehPrev))
NumVehEnterInCurStep = len(VehicleEntered)
if NumVehEnterInCurStep > 0:
    for number in VehicleEntered:
        Veh = Vissim.Net.Vehicles.GetVehicleByNumber(number)
        if Veh <> None:
            CheckNumEntered = CheckNumEntered + 1

            # If the vehicle type is "ToBeDistracted":
            if Veh.AttValue("TYPE")== 200:
                CheckNumEnteredToBeDist = CheckNumEnteredToBeDist + 1

                # Assign a distraction time profile to the vehicle
                disTypeNo = weighted_choice(distProbabilities)
                distractionStatus[Veh.AttValue("ID")][1] = disTypeNo

                # Assign a distraction start step to the vehicle
                disStartStep = numpy.random.randint(1,UBound)
                distractionStatus[Veh.AttValue("ID")][2] = disStartStep


                # Define the distraction status array: 1=distracted
                distractionStatus[Veh.AttValue("ID")][0] = 1
            else:

                # Define the distraction status array: 0=non-distracted
                distractionStatus[Veh.AttValue("ID")][0] = 0

# Get the number of vehicles in the simulation
NumVehicles = Vissim.Net.Vehicles.Count

# For each vehicle in the network
if NumVehicles > 0:
    for Vehicle in Vissim.Net.Vehicles:
        Veh = Vissim.Net.Vehicles.GetVehicleByNumber(Vehicle.ID)
        if Veh <> None:

            # If the vehicle type is "ToBeDistracted":
            if distractionStatus[Veh.AttValue("ID")][0] == 1:
                VehDisType = distractionStatus[Veh.AttValue("ID")][1]
                VehDisStart = distractionStatus[Veh.AttValue("ID")][2]

                # If i is in its distracted time (based on its distraction time profile)
                if DistractionTimeProfiles[VehDisType][i + VehDisStart-1] == 1:

                    # change the type of vehicle to "CarDistracted"
                    Veh.SetAttValue("TYPE", 300)
                    CheckChangetoDist = CheckChangetoDist + 1
                else:

                    # change the type of vehicle to "CarToBeDistracted"
                    Veh.SetAttValue("TYPE", 200)
                    CheckChangetoNonDist = CheckChangetoNonDist + 1
VehPrev = VehCur

# Run simulation one single step
Vissim.Simulation.RunSingleStep()
NumVeh = Vissim.Net.Vehicles.Count
if NumVeh > 0:
```

```
            VehCur=[]
            for Vehicle in Vissim.Net.Vehicles:
                if Vehicle <> None:
                    VehCur.extend([Vehicle.ID])
    Vissim.Simulation.RunIndex = Vissim.Simulation.RunIndex + 1
    Vissim.Simulation.RandomSeed = Vissim.Simulation.RandomSeed + 1
    print("End of Run#",Vissim.Simulation.RunIndex)
```

Replicate Salvucci's model for Circular Scenario in VISSIM (RQ3)

The same as the code for LOS A except for the vehicle volume as follows:

```
# <<<<<<<<<< Add vehicles >>>>>>>>>>
    Vissim.Simulation.RunSingleStep()
    # Lead vehicle
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1, 1017.06)
    # Vehicle #1
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1,  951.44)
    # Vehicle #2
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1,  885.83)
    # Vehicle #3
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (200, 40, 2, 1,  820.21)
    # Vehicle #4
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1,  754.59)
    # Vehicle #5
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1,  688.98)
    # Vehicle #6
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1,  623.36)
    # Vehicle #7
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1,  557.74)
    # Vehicle #8
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (200, 40, 2, 1,  492.13)
    # Vehicle #9
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1,  426.51)
    # Vehicle #10
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1,  360.89)
    # Vehicle #11
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1,  295.28)
    # Vehicle #12
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1,  229.66)
    # Vehicle #13
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (200, 40, 2, 1,  164.04)
    # Vehicle #14
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1,  98.43)
    # Vehicle #15
    Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate (100, 40, 2, 1,  32.81)
```

Developed interface for VISSIM – Real scenarios (RQ3)

This code is for the real interchange in Beijing, China. The other scenarios are the same with a difference in their VISSIM input file.

```
##### Vehicle Type description:
    # Vehicles enter the network are either from Car or CarToBeDistracted
    # Car: Vehicles that will not be distracted
    # CarToBeDistracted: Vehicles that will be distracted in a time frame during simulation period
    # CarDistracted: The vehicles with distracted parameters
# Three vehicle classes are defined with the same names: Car, CarToBeDistracted, CarDistracted
# Link behavior type description:
    # Behavior of different classes of vehicles are different for each link type:
        # Driving behavior of Car: Freeway NonDistracted
        # Driving behavior of CarToBeDistracted: Freeway NonDistracted
        # Driving behavior of CarDistracted: Freeway Distracted
# <<<<<<<<<< Import needed modules >>>>>>>>>>
import numpy
import xlrd
import win32com.client
import random

# <<<<<<<<<< FUNCTIONS >>>>>>>>>>
# Function for randomly selection of weighted items
# Input: list of weights of items
# Output: index of selected item
def weighted_choice(weights):
    totals = []
    running_total = 0
    for w in weights:
        running_total += w
        totals.append(running_total)
    rnd = random.random() * running_total
    for i, total in enumerate(totals):
        if rnd < total:
            return i

# <<<<<<<<<< Read Distraction Table (Distract-R output) >>>>>>>>>>
# The profile durations are assumed to be (at least) twice the simulation duration (2*UBound)
# When a vehicle enters, I randomly assign a number 1 =< k <= UBound as the starting point
# So, its distraction time period is (k, k+UBound)
# Open the excel file which contains distraction results from Distract-R
workbook = xlrd.open_workbook('DistractionTable.xlsx')

# Open the worksheet 'Sheet1' which contains the data
worksheet = workbook.sheet_by_name('Sheet1')

# Read the distraction data and store it in a matrix
num_rows = worksheet.nrows - 1
```

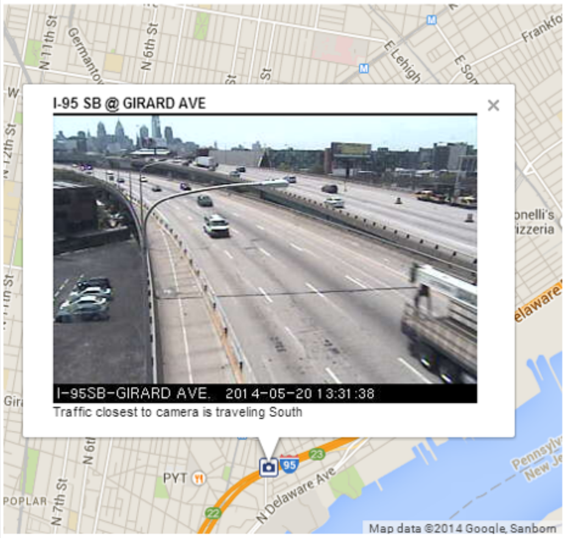
```
num_cells = worksheet.ncols - 1

# Initiate DistractionTimeProfiles matrix (whose rows are distraction types, and columns are time
steps)
# Distract-R output resolution is 0.05 sec and Vissim resolution is 0.1 sec, so I need to change the
resolution
DistractionTimeProfiles = [[0 for x in xrange(num_rows/2)] for x in xrange(num_cells)]

# Store the data in DistractionTimeProfiles matrix: either distracted (1) or nondistracted (0) at the
time step
curr_row = 0
while curr_row < num_rows-1:
    curr_row += 2
    curr_cell = 0
    while curr_cell < num_cells:
        curr_cell += 1

        # HINT1: During storing, I change the rows to columns and columns to rows
        # HINT2: I remove the title row (row=0) and the time step column (column=0)
        # So, each row is a distraction type, and each column is a time step
        # Distract-R output resolution is 0.05 sec and Vissim resolution is 0.1 sec,
        #  so I use 2*curr_cell to change the resolution.
        if worksheet.cell_value(curr_row, curr_cell) == 1:
      DistractionTimeProfiles[curr_cell-1][curr_row/2-1] = 0
     else:
      DistractionTimeProfiles[curr_cell-1][curr_row/2-1] = 1

# Different types of distracted drivers
distractedTypes = len(DistractionTimeProfiles)

# % of each distraction type: A vector of weights (probabilities) of different distraction types
distProbabilities = [1,1,1]

# <<<<<<<<<<<  Initiate VISSIM model >>>>>>>>>>>>
Vissim = win32com.client.Dispatch("VISSIM.vissim")
Vissim.LoadNet(r"C:\NOURZAD\RQ3Results\IntChina\beijing.inp")

# Enable evaluation parameters
Vissim.Evaluation.SetAttValue("NETPERFORMANCE", True)
Vissim.Evaluation.SetAttValue("LANECHANGE", True)
Vissim.Evaluation.SetAttValue("DELAY", True)
Vissim.Evaluation.SetAttValue("EXPORT", True)
Vissim.Evaluation.SetAttValue("VEHICLERECORD", True)

# <<<<<<<<<<< Initiate parameters and variables >>>>>>>>>>>
# Define max number of time steps and max number of vehicles
UBound = 6000
MaxNumVehicles = 10000

# Status of vehicle: column 0: vehicle status (either distracted=1, or nondistracted=0),
#                 column 1: distraction type, column 2: distraction Start Step
distractionStatus = [[0 for x in xrange(3)] for x in xrange(MaxNumVehicles)]

# <<<<<<<<<<< Set up multiple runs >>>>>>>>>>>
NumberOfRuns = 3
Vissim.Simulation.RunIndex = 0
Vissim.Simulation.RandomSeed = 95
```

```
for numrun in range(0, NumberOfRuns):
    # Temporary for check:
    CheckNumEntered = 0
    CheckNumEnteredToBeDist = 0
    CheckChangetoDist = 0
    CheckChangetoNonDist = 0
    VehPrev = []
    VehCur = []
# <<<<<<<<<< Run Traffic simulation in VISSIM step-by-step >>>>>>>>>>
    for i in range(0, UBound):
        # For each vehicle enters the network in this time step:
        VehicleEntered = list(set(VehCur) - set(VehPrev))
        NumVehEnterInCurStep = len(VehicleEntered)
        if NumVehEnterInCurStep > 0:
            for number in VehicleEntered:
                Veh = Vissim.Net.Vehicles.GetVehicleByNumber(number)
                if Veh <> None:
                    CheckNumEntered = CheckNumEntered + 1
                    # If the vehicle type is "ToBeDistracted":
                    if Veh.AttValue("TYPE")== 200:
                        CheckNumEnteredToBeDist = CheckNumEnteredToBeDist + 1
                        # Assign a distraction time profile to the vehicle
                        disTypeNo = weighted_choice(distProbabilities)
                        distractionStatus[Veh.AttValue("ID")][1] = disTypeNo
                        # Assign a distraction start step to the vehicle
                        disStartStep = numpy.random.randint(1,UBound)
                        distractionStatus[Veh.AttValue("ID")][2] = disStartStep
                        # Define the distraction status array: 1=distracted
                        distractionStatus[Veh.AttValue("ID")][0] = 1
                    else:
                        # Define the distraction status array: 0=non-distracted
                        distractionStatus[Veh.AttValue("ID")][0] = 0
        # Get the number of vehicles in the simulation
        NumVehicles = Vissim.Net.Vehicles.Count
        # For each vehicle in the network
        if NumVehicles > 0:
            for Vehicle in Vissim.Net.Vehicles:
                Veh = Vissim.Net.Vehicles.GetVehicleByNumber(Vehicle.ID)
                if Veh <> None:
                    # If the vehicle type is "ToBeDistracted":
                    if distractionStatus[Veh.AttValue("ID")][0] == 1:
                        VehDisType = distractionStatus[Veh.AttValue("ID")][1]
                        VehDisStart = distractionStatus[Veh.AttValue("ID")][2]
                        # If i is in its distracted time (based on its distraction time profile)
                        if DistractionTimeProfiles[VehDisType][i + VehDisStart-1] == 1:
                            # change the type of vehicle to "CarDistracted"
                            Veh.SetAttValue("TYPE", 300)
                            CheckChangetoDist = CheckChangetoDist + 1
                        else:
                            # change the type of vehicle to "CarToBeDistracted"
                            Veh.SetAttValue("TYPE", 200)
                            CheckChangetoNonDist = CheckChangetoNonDist + 1
        VehPrev = VehCur
        # Run simulation one single step
        Vissim.Simulation.RunSingleStep()
        NumVeh = Vissim.Net.Vehicles.Count
        if NumVeh > 0:
            VehCur=[]
            for Vehicle in Vissim.Net.Vehicles:
                if Vehicle <> None:
                    VehCur.extend([Vehicle.ID])
    Vissim.Simulation.RunIndex = Vissim.Simulation.RunIndex + 1
    Vissim.Simulation.RandomSeed = Vissim.Simulation.RandomSeed + 1
    print("End of Run#",Vissim.Simulation.RunIndex)
```

Appendix B: Samples of Observation-Based Validation

Point: 10 Criticality: High Medium **Low** Time: AM **MD** PM NT

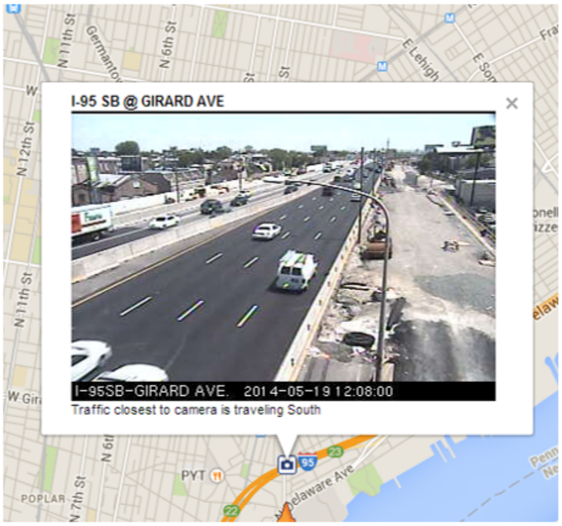
Without incident: 5/20/2014 – 1:37 PM


I-95 SB @ GIRARD AVE

I-95SB-GIRARD AVE. 2014-05-20 13:31:38

Traffic closest to camera is traveling South

With incident: 5/19/2014 – 12:13 PM


I-95 SB @ GIRARD AVE

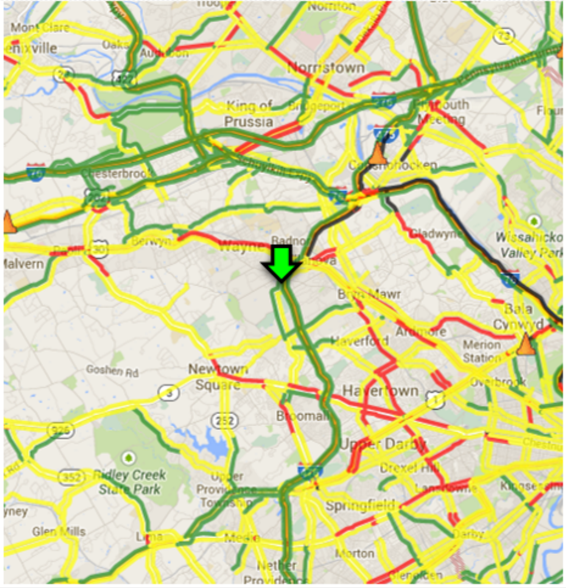
I-95SB-GIRARD AVE. 2014-05-19 12:08:00

Traffic closest to camera is traveling South

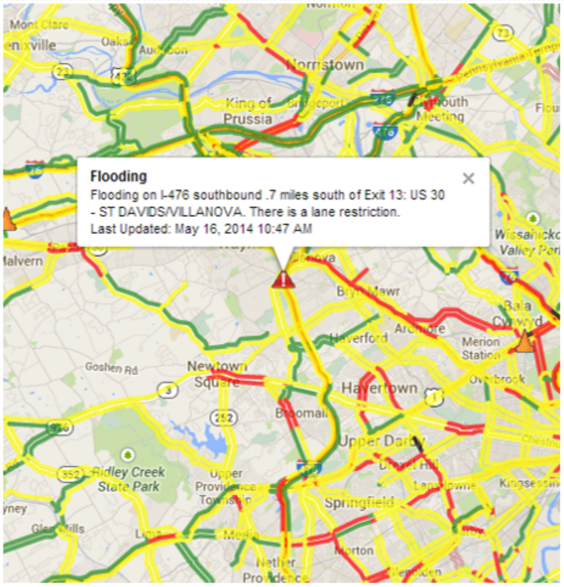
Sample Observations within a Low-Critical Cluster during MD Period

Point: 1 Criticality: High **Medium** Low Time: AM **MD** PM NT

Without incident: 5/19/2014 – 12:20 PM



With incident: 5/16/2014 – 12:40 PM


Flooding

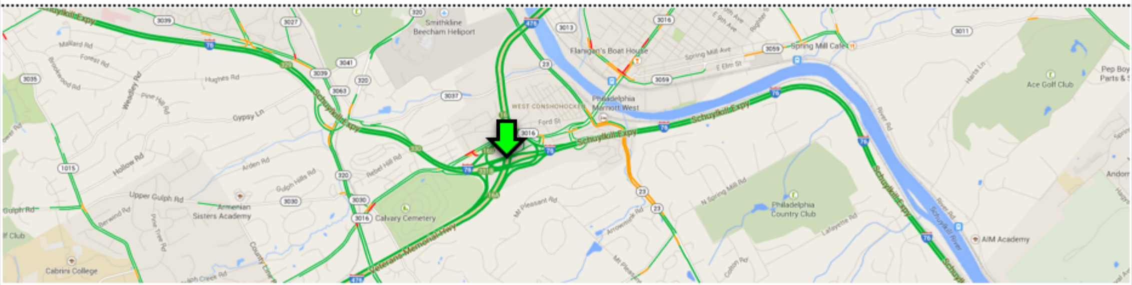
Flooding on I-476 southbound .7 miles south of Exit 13: US 30 - ST DAVIDS/VILLANOVA. There is a lane restriction.

Last Updated: May 16, 2014 10:47 AM

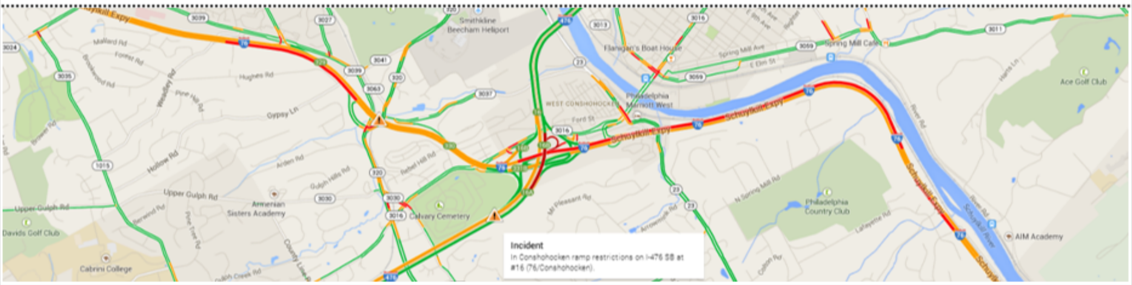
Sample Observations within a Medium-Critical Cluster during MD Period

Point:	2	Criticality:	**High**	Medium	Low	Time:	AM	**MD**	PM	NT

Without incident: 5/20/2014 – 1:30 PM



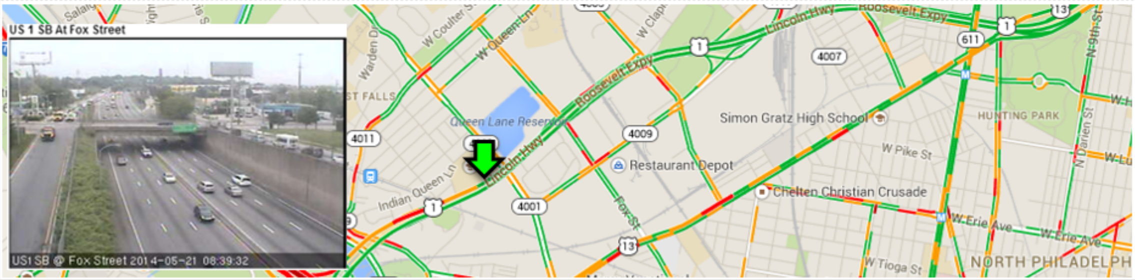
With incident: 5/16/2014 – 12:49 PM


Incident
In Conshohocken ramp restrictions on I-476 SB at #16 (76/Conshohocken).

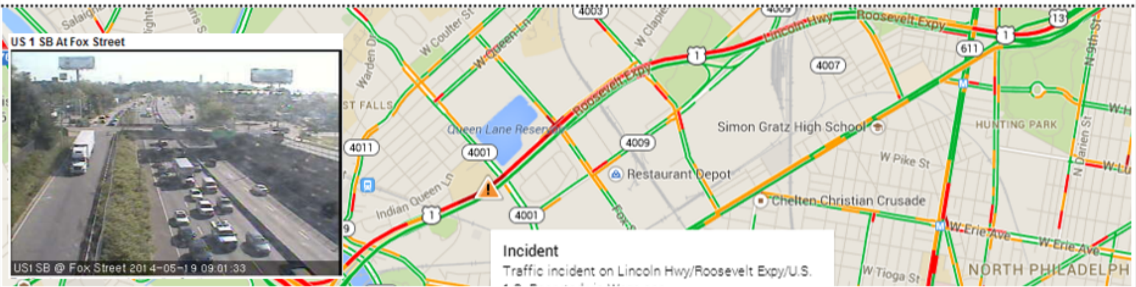
Sample Observations within a High-Critical Cluster during MD Period

Point:	8	Criticality:	**High**	Medium	Low	Time:	**AM**	MD	PM	NT

Without incident: 5/21/2014 – 8:43 AM



With incident: 5/19/2014 – 9:04 AM


US 1 SB At Fox Street
US1 SB @ Fox Street 2014-05-19 09:01:33
Incident
Traffic incident on Lincoln Hwy/Roosevelt Expy/U.S.

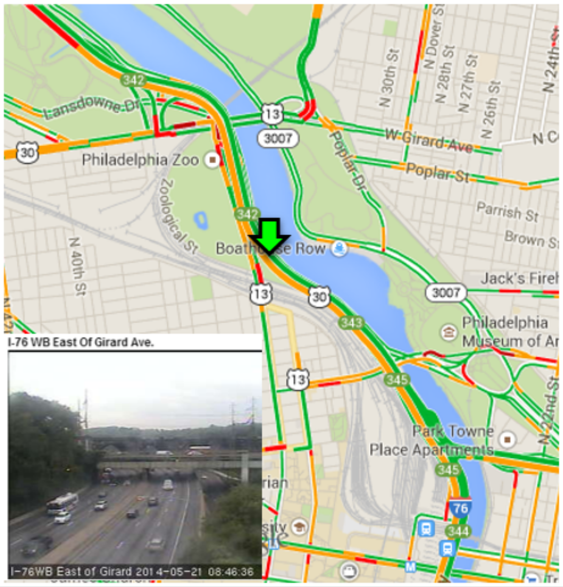
Sample Observations within a High-Critical Cluster during AM Period

Point: **9**

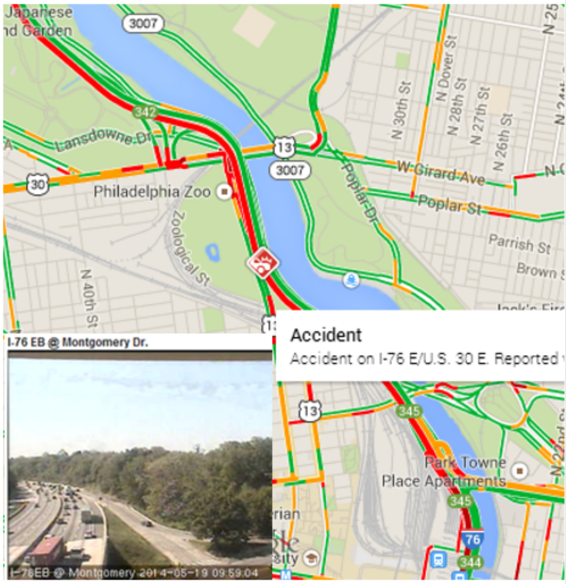
Criticality: **High** Medium Low

Time: **AM** MD PM NT

Without incident: 5/21/2014 – 09:46 AM



With incident: 5/19/2014 – 10:03 AM



Accident

Accident on I-76 E/U.S. 30 E. Reported

Sample Observations within a High-Critical Cluster during AM Period

Vita

Author: Seyed Hossein Hosseini Nourzad
Degree: Doctor of Philosophy
Date: June 2014
Date of Birth: April 5, 1983
Place of Birth: Tehran, Iran

UNDERGRADUATE AND GRADUATE EDUCATION

Drexel University	**Philadelphia, PA**
Doctor of Philosophy in Civil Engineering	*September 2010 – June 2014*
Iran University of Science & Technology	**Tehran, Iran**
Master of Sciences in Civil Engineering	*September 2005 - July 2008*
University of Tehran	**Tehran, Iran**
Bachelor of Science in Civil Engineering	*September 2001 - September 2005*

AWARDS

Continued Teaching Excellence Award 2014	**Drexel University**
Teaching Excellence Award 2012	**Drexel University**
Highly Commended Teaching Assistant 2011	**Drexel University**
PMI Delaware Valley Chapter Scholarship 2012	**Project Management Institute**

SELECTED PUBLICATION

- *"Vulnerability of Infrastructure Systems: Macroscopic Analysis of Critical Disruptions on Road Networks"*
- *"Computational Modeling of Road Network Macroscopic Performance: A Multivariate Approach"*
- *"Computational Modeling of Driver Distraction: Toward Large-scale Vulnerability Assessment"*
- *"Ensemble Methods for Binary Classifications of Airborne LIDAR Data"*
- *"Network-Wide Assessment of Transportation Systems Using an Epidemic Spreading Methodology"*
- *"Mathematical Representation of Control Strategies on Congestion Propagation within Roadway Networks: A Macroscopic Viewpoint"*
- *"Industrial Distribution System Simulation for Optimal Water Resource Assignment Using Probabilistic Tabu Search"*
- *"Automated Detection of Damaged Areas after Hurricane Sandy using Aerial Color Images"*