ABSTRACT

SONG, TAI-JIN. Recurrent Freeway Bottlenecks Identification and Applications (under the direction of Dr. Billy M. Williams and Dr. Nagui M. Rouphail).

Traffic congestion and safety concerns are contributing to the many challenges facing transportation agencies as population, motorization, and urban densities continue to rise. Congestion is contributing to longer travel times, higher fuel consumption, and increased emissions of air pollutants. Nearly 40% of roadway congestion in the United States stems from recurrent bottlenecks, which can cost agencies both time and money if not properly addressed.

Identification of recurrent bottlenecks is an effective way to hone in on appropriate investment in current facilities to relieve congestion. Furthermore, it would enable the ranking or prioritization of bottlenecks since bottleneck removal and its associated impact alleviation are hampered by limited sources. It is imperative that transportation jurisdictions identify the basis for ranking bottlenecks by exploring: how often they activate; how long it takes the congestion to disappear; and how many miles of road are affected.

In this thesis, a data-driven approach for identifying recurrent bottlenecks spatiotemporally is introduced, using probe vehicle speed reports. Historical spatiotemporal characteristics of bottlenecks are investigated through a comprehensive analysis of statewide interstates in North Carolina. Using the characteristics discovered, the recurrent bottleneck locations with a historical time span of bottleneck activation are revealed and tested. The findings of the proposed identification scheme generate critical information in order to quantify and diagnose a bottleneck and its associated impact area.
There is also a chance that those same bottlenecks can be exacerbated when there happen to be a crash within their area of impact. In other words, the crash results in additional congestion on top of the recurrent congestion. Accordingly, this effect should be identified and quantified separately when it comes to identifying and monitoring bottlenecks and their associated impacts more accurately. Previous bottleneck studies have estimated bottleneck impact with little attention to distinguishing the underlying source of congestion. In addition, studies that have attempted to distinguish between recurrent and non-recurrent congestion have focused on separating non-recurrent congestion from recurrent congestion only for the purpose of estimating the intensity of congestion using parameters for the speed distribution in a time of day in a segment or point. A data-driven methodology for quantifying recurrent bottleneck impact is introduced, which quantifies recurrent and collision-induced congestion impacts separately.

In order to quantify collision-induced congestion impacts, there needs to be a thorough understanding of the conditions in which vehicles occur. An easily implementable methodology that can classify all reported crashes in terms of the operational conditions under which each crash occurred is presented. Unlike previous secondary crash classification schemes, the proposed methodology requires neither a-priori identification of the precipitating incident nor definition of the precipitating incident’s impact area.

Using 2014 statewide interstate data in North Carolina, a total of 95 bottleneck segments were identified with the selected thresholds. Of these, an average two recurrent bottlenecks occurred within a bottleneck impact area. Six bottlenecks were finally selected by a systematic process and then used for identifying the statistical characteristics of congestion type in order to segregate recurrent congestion impact from collision-induced
impacts. Recurrent bottleneck spatiotemporal impacts were calculated using the proposed quantification scheme. The historical impacts varied from 1 to 3 mile-hours of impact per activation except for the third study site which yielded 13 mile-hours per activation, and was thus identified as the worst bottleneck among all the study sites. The impact factor can be directly applied for ranking freeway bottlenecks and facilitates monthly and/or annual analysis on a large scale network. In addition, it allows for identification of degraded or improve recurrent bottleneck impact. The proposed approaches are robust and represent a significant improvement in the understanding and monitoring of mobility and reliability.
Recurrent Freeway Bottlenecks Identification and Applications

by
Tai-Jin Song

A dissertation submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Civil Engineering

Raleigh, North Carolina

2016

APPROVED BY:

__________________________________________  ______________________________________
Billy M. Williams, Professor  Nagui M. Rouphail, Professor
Co-Chair of Advisory Committee  Co-Chair of Advisory Committee

__________________________________________  ______________________________________
George F. List, Professor  Justin Post, Assistant Professor
DEDICATION

To heavenly farther, my lovely wife, Aeri, and my two children, Ryan and Ian.

Without their support and endless love, none of this would be possible.
BIOGRAPHY

Tai-Jin Song is from Seoul, South Korea. He received his Bachelor and Master of Transportation Systems Engineering from Hanyang University in 2008 and 2010, respectively. Tai-Jin began working on his Doctor of Philosophy in Civil Engineering at North Carolina State University in 2013 and anticipates completion in 2016. Before seeking a PhD, Tai-Jin worked professionally for 4 years with the Korea Transport Institute. As a key researcher, Tai-Jin conducted more than fifteen research and development government projects with a variety of topics related to transportation safety and operation.
ACKNOWLEDGMENTS

I would like to express a heartful gratitude to Dr. Billy M. Williams and Dr. Nagui M. Roupail. They gave me invaluable lessons in life and research as well as continuous support for my research. Their guidance and passion made this work possible. I would also like to thank Dr. George F. List and Dr. Justin Post for the advice and support provided as members of my advisory committee.

A special thanks to Dr. Sankey Kim, Shannon Warchol, Kihyun Pyo, Juwoon Baek, J. Lake Trask, Thomas Chase, and all other staff at the Institute for Transportation Research and Education (ITRE) at North Carolina State University for helping and supporting my work all the time.
TABLE OF CONTENTS

LIST OF TABLES .................................................................................................................. VIII
LIST OF FIGURES ................................................................................................................ IX
LIST OF SYMBOLS ............................................................................................................... XI
LIST OF ABBREVIATIONS ................................................................................................... XIII
CHAPTER 1 INTRODUCTION ............................................................................................... 1
  1.1 Research Objectives and Key Tasks .......................................................................... 4
  1.2 Research Scope .......................................................................................................... 6
  1.3 Research Contribution .............................................................................................. 7
  1.4 Organization of the Thesis ........................................................................................ 7
CHAPTER 2 LITERATURE REVIEW ....................................................................................... 9
  2.1 Traditional Congestion and Bottleneck Identification ............................................. 9
  2.1.1 Breakdown ........................................................................................................... 10
  2.1.2 Bottleneck ........................................................................................................... 11
  2.2 Recurrent Congestion Versus Non-Recurrent Congestion ....................................... 17
  2.3 Secondary Crash ...................................................................................................... 21
  2.4 Quantification of Bottleneck Impact ....................................................................... 25
  2.5 Summary .................................................................................................................... 27
CHAPTER 3 DATA-DRIVEN SPATIOTEMPORAL “RECURRENT” BOTTLENECK IDENTIFICATION .................................................................................................................. 29
  3.1 Introduction ................................................................................................................ 29
  3.2 Review of Relevant Studies ....................................................................................... 30
  3.2.1 Bottleneck, Breakdown, and Congestion Definitions ............................................ 30
  3.2.2 Recurrent Congestion .......................................................................................... 31
  3.3 Methodology ............................................................................................................. 32
  3.3.1 Overall Framework .............................................................................................. 32
  3.3.2 Data Source ......................................................................................................... 33
  3.3.3 Congestion Identification ..................................................................................... 34
  3.3.3.1 Congestion, bottleneck, and breakdown phenomenon .................................... 34
  3.3.3.2 Congestion value and index .......................................................................... 34
  3.3.4 Average Historic Congestion Index .................................................................... 36
  3.3.4.1 Spatial patterns of AHCI .............................................................................. 38
  3.3.4.2 Temporal patterns of AHCI .......................................................................... 40
  3.3.5 Recurrent Bottleneck Identification .................................................................... 43
  3.3.5.1 Identification of Recurrent Bottleneck Location ............................................ 43
  3.3.5.2 Identification of historical time span of recurrent bottleneck activation ........... 44
  3.4 Case Study .................................................................................................................. 45
  3.4.1 Site Description .................................................................................................... 45
  3.4.2 Congestion Identification and AHCI Contours ..................................................... 45
  3.4.3 Sensitivity Analysis and Illustrated Examples ...................................................... 46
  3.4.4 Recurrent Bottlenecks in NC .............................................................................. 50
  3.5 Conclusions ............................................................................................................... 51
CHAPTER 4 A NOVEL COLLISION CLASSIFICATION METHODOLOGY BASED ON TEMPORAL LINK SPEED DATA AND CONGESTION THRESHOLDS .................................................................................................................. 53
CHAPTER 5 RECURRENT CONGESTION IMPACT BASED ON SPATIOTEMPORALLY HISTORIC CONGESTED INFORMATION.....73
5.1 Introduction.......................................................... 73
5.2 Methodology .......................................................... 76
  5.2.1 Overall Framework .............................................. 76
  5.2.2 Spatiotemporal Congestion Impact ......................... 77
  5.2.3 Separation of Congestion Types ............................. 78
  5.2.4 Recurrent Congestion Area ................................. 81
  5.2.5 Quantification of Recurrent Bottleneck Impacts ............ 83
5.3 Case Study .......................................................... 84
  5.3.1 Data Description ................................................ 84
  5.3.2 Study Site Identification ..................................... 84
  5.3.3 Probability Distribution of Congestion Spatiotemporal Impact Index .... 87
  5.3.4 Capturing and Quantifying Recurrent Congestion Impact ........ 89
5.4 Applications ....................................................... 91
5.5 Conclusions ........................................................ 92

CHAPTER 6 CONCLUSIONS AND RECOMMENDTIONS..................94
6.1 Data-Driven Spatiotemporal “Recurrent” Bottleneck Identification .... 94
6.2 A Novel Collision Classification Methodology based on Temporal Link Speed Data and Congestion Thresholds ......................................................... 96
6.3 Recurrent Congestion Impact based on Spatiotemporally Historic Congested Information ................................................................. 97
6.4 Future Research ....................................................... 99
REFERENCES.....................................................................101
APPENDICES ................................................................................................................................................. 106
   A. Collision Contour and Crash Classification Tool (4CT) ................................................................. 107
   B. Collision Contour and Crash Classification Tool (4CT) – Java code .......................... 114
LIST OF TABLES

Table 2-1 Comparison of Secondary Incidents Identification Studies ........................................ 24
Table 4-1 Classification of the Crashes Based on the Proposed Methodology ......................... 70
Table 4-2 Reported vs. Unreported Primary Incidents or Crashes on Crash in Non-recurrent Congested Areas .................................................................................................................. 71
Table 5-1 Congestion Type Occurring in Recurrent Bottleneck Impact Area ..................... 81
Table 5-2 Bottleneck Characteristics for Study Sites Selected in NC .................................. 85
Table 5-3 Number of Type 1 Crashes Observed under Time and Location Cell .................. 86
Table 5-4 Descriptive Statistics of Congestion Spatiotemporal Impact Index (CSII) ............. 87
Table 5-5 Computed Statistics Values by Distribution ............................................................ 88
Table 5-6 Quantification Result of Recurrent Congestion Impact ......................................... 91
LIST OF FIGURES

Figure 1-1 Overall Framework ........................................................................................................... 5
Figure 1-2 Spatiotemporal Traffic State Matrix for Key Tasks ............................................................... 6
Figure 2-1 Critical Speed and LOS C/D on I-880 Speed Flow Data (Jia et al. 2010) .................. 11
Figure 2-2 Bottleneck Activation Sites (with Darkness Proportional to Frequency of
Recurrence) (Wieczorek 2010) ...................................................................................................... 14
Figure 2-3 Illustration of Congested Roadways and Bottlenecks (Florida, 2011) ................. 15
Figure 2-4 Bottlenecks Identification on the Vehicle Probe Project Suite (RITIS 2016) ..... 16
Figure 2-5 Methodology to Assess Recurrent and Non-recurrent Congestion (Medina 2010)
...................................................................................................................................................... 18
Figure 2-6 Reduced Speed by Accident (left: $D_i(t_{m}) = 1$, right: $D_i(t_{m}) = 0$) (Chung 2011) ... 19
Figure 2-7 Classifying a Secondary Incident (Chou and Miller-Hooks 2010) ....................... 21
Figure 2-8 Linguistic State Space and Rules of the Bottleneck Severity Diagnosis (Liu and
Fei 2010) ...................................................................................................................................... 26
Figure 3-1 Overall Recurrent Bottleneck Identification Framework ............................................... 33
Figure 3-2 Schematic Definition of Breakdown, Bottleneck, and Congestion ......................... 34
Figure 3-3 Congestion Index Contour Map (3 January 2014, on I-85 Southbound) ............. 36
Figure 3-4 AHCI Contour Map (260 days-Weekdays of I-40 Westbound in 2014) ............ 37
Figure 3-5 Spatial Patterns of AHCI in 2014 .................................................................................. 40
Figure 3-6 Temporal Patterns of AHCI across Multiple TMC’s .................................................... 42
Figure 3-7 A Screenshot of 2014 Statewide AHCI Contour (on a portion of I-40 Westbound
in North Carolina) ....................................................................................................................... 46
Figure 3-8 Bottlenecks Identified under Various Cut-off for AHCI ............................................ 47
Figure 3-9 Number of Bottlenecks Identified under Various Cut-off Threshold of AHCI and
$\gamma$ .............................................................................................................................................. 48
Figure 3-10 Schematic Temporal Boundary of a Recurrent Bottleneck (I-85 Northbound in
Mecklenburg County, NC) ............................................................................................................... 49
Figure 3-11 Statistics of Time Span of Historical Recurrent Bottleneck Activation on
Interstates in NC (2014) .................................................................................................................. 50
Figure 3-12 Spatiotemporal Recurrent Bottleneck Identification in NC ....................................... 51
Figure 4-1 Methodology Flow Procedure ......................................................................................... 59
Figure 4-2 Contours of (a) Congestion Value and (b) Traffic Flow Data Format (Congestion
and Congestion Index) .................................................................................................................. 61
Figure 4-3 Bottleneck Segment Identification Using an AHCI Contour Map ......................... 63
Figure 4-4 The Crash Classification Process ................................................................. 64
Figure 4-5 Number of Case 2 under Various Upper and Lower Bound of AHCI ........ 66
Figure 4-6 Schematic Representative a Case 1 Crash Condition ............................. 68
Figure 4-7 Examples of TMCc Classified as Cases 2 and 3 in Step 3 and 4 .............. 69
Figure 5-1 Overall Historic Congestion Impact Quantification Framework .......... 77
Figure 5-2 Estimation of Congestion Spatiotemporal Impact Index (CSII) .............. 78
Figure 5-3 Types of Congestion Occurring within a Recurrent Congestion Area .... 80
Figure 5-4 Crash Location and Time Corresponding to Congestion ..................... 86
Figure 5-5 Normalized CSII distribution of recurrent congestion (Type 0) with Type 1 88
Figure 5-6 Recurrent Congestion Spatiotemporal Extent .......................................... 90
Figure 5-7 A Degraded Recurrent Bottleneck on Westbound I-40 (Site 2) .............. 92
### LIST OF SYMBOLS

**α**  The cut-off speed threshold to separate congested or uncongested state: the ratio of speed at capacity to free flow speed in the US Highway Capacity Manual.

**β**  The threshold (%) of AHCI for clearly historically congested

**γ**  The threshold of spatial congestion difference in AHCI

**δ**  The threshold (%) of AHCI for time span of recurrent bottleneck activations at bottleneck $i$

**η**  The threshold (%) of AHCI for clearly non-recurrent congested

**λ**  The cut-off threshold (%) of AHCI to identify a recurrent bottleneck spatiotemporal impact area

**ω**  Extra spatiotemporal congestion impact (miles·hours) due to a crash within an active bottleneck

**Ω**  Total extra spatiotemporal congestion impacts (miles·hours) due to crashes within an active bottleneck in a study period

**b**  A bottleneck segment

**B**  Segments with time span of bottleneck activation

**c**  Congestion caused by an event or active bottleneck

**d**  Weekday $\in$ (Monday, Tuesday, Wednesday, Thursday, Friday)

**f**  Congestion frequency

**i**  A TMC segment

**I**  TMCs within a recurrent bottleneck impact area

**$L_i$**  length of segment $i$ (miles)

**m**  Index for a day in the study period
The number of days in the study period

The number of time intervals identified in recurrent bottleneck time span on the bottleneck segment $i$

Non-recurrent congestion

Reporting Period (typically one or more years)

Recurrent congestion

Time resolution (15 min for this thesis)

Types of congestion occurring in recurrent bottleneck impact area; type 0 = congestion due to an active bottleneck, type 1 = congestion due to a crash occurring within a bottleneck but bottleneck is inactive before the crash, type 2 = congestion due to an active bottleneck and extra congestion due to a crash occurring within an active recurrent bottleneck

Specified time interval in a day (e.g., 8:00-8:15, 15 min); typically $t$ will vary from 1 to 96 for a single day

Specified time intervals in a day within time span of bottleneck activation

Weekend $\in$ (Saturday, Sunday)
LIST OF ABBREVIATIONS

AHCI – Average Historic Congestion Index at for segment $i$ at time $t$ ($\%$)

ATMS – Advanced Traffic Management Systems

C – Congestion value

CI – Congestion Index

CSII – Congestion Spatiotemporal Impact Index (miles · hours)

FFS – Free Flow Speed (mi/h) for segment $i$

GPS – Global Positioning System

NCDOT – North Carolina Department of Transportation

PTI – Planning Time Index

RBI – Recurrent Bottleneck Identification

RBSII – Recurrent Bottleneck Spatiotemporal Impact Index (miles · hours per activation)

RBLI – Recurrent Bottleneck Location Identification

RBTI – Recurrent Bottleneck Time span Identification

RITIS – Regional Integrated Transportation Information System

RSCM – Representative Speed Contour Map

SBSIF – Simulation-Based Secondary Incident Filtering Method

SND – Standard Normal Deviate

TEAAS – Traffic Engineering Accident Analysis System

TMC – Traffic Message Channel

TMCc – a TMC segment where a crash occurs

TTI – Travel Time Index

VSL – Variable Speed Limit
CHAPTER 1  INTRODUCTION

Traffic congestion and safety issues are increasingly becoming major problems for transportation agencies. Both are increasing as population growth, increased motorization, and change in population density become more prevalent. Congestion is one of the major factors affecting longer travel times, higher fuel consumption, and increased emissions of air pollutants. Nearly 40% of roadway congestion in United States stems from recurrent bottlenecks (Cooner et al., 2011), which can cost both agencies time and money if not properly addressed. Additionally, vehicular crashes endanger life and limb, damage property, and cause severe congestion, thereby presenting an obstacle to the goal of improving the safety, efficiency, and sustainability of the transportation system. Therefore, it is imperative to understand and investigate the detailed characteristics of both issues.

Traffic jurisdictions should identify and understand where, when, how long, how frequent, and to what extent congestion occurs by recurrent bottleneck activations on freeways due to their characteristics of recurrence. Although an active bottleneck is “a physical point on the network upstream of which one finds a queue and downstream of which one finds freely flowing traffic” (Bertini and Leal 2005), most of existing studies have focused on identifying congestion with no attention to distinguishing recurrence level of at the same “bottleneck” location. Oxford’s Dictionary defines “recurrent” as being something that occurs often or repeatedly. That is, recurrent bottleneck location means a “predictable” location in time and when drivers feel like “this area on this time is often and repeatedly congested.” Therefore, a recurrent bottleneck can be defined as a location where congestion occurs frequently during specified time periods. In the real world, it is likely that several bottlenecks can be activated within a recurrent congestion-impacted area. Accordingly, the
detailed spatiotemporal characteristics of bottlenecks should be first identified in order to quantify and diagnose a bottleneck and its impact area.

To identify recurrent bottleneck clearly, distinguishing between recurrent congestion and non-recurrent congestion should be considered. There has been several studies on this theme. However, previous studies have used either the mean or the median of a speed distribution during a specified time of day to measure recurrent delay and extra delay caused by incidents as non-recurrent congestion on a highway. In addition, the distribution of speeds collected from detectors has different distributions by each location. It is hard to determine an appropriate threshold of speed to identify recurrent and non-recurrent congestion.

Congestion occurs due to either an un-expected event or an active bottleneck. It indicates that its associated impacts should be separated and labeled by types of congestion in conducting a performance monitoring of a roadway. In addition, a recurrent bottleneck has a site-specific spatiotemporal shockwave phenomenon. Accordingly, the associated links with time need to be identified in quantifying bottleneck impacts.

When it comes to reduction in congestion impacts caused by vehicular crashes, it is imperative to understand the operational conditions when crashes happen. This knowledge can and should inform how crashes are managed and which resources are allocated for incident response programs. For instance, Variable Speed Limits (VSL) may be an effective countermeasure to prevent crashes during recurrent congestion. Conversely, minimizing the incident response rate and clearance time for crashes may be the most effective strategy for crashes occurring outside of recurrent congestion periods.

Another motivation for understanding and classifying the types of congestion under which crashes occur is that it will also bring an opportunity to help reliability and safety
studies (Golob et al., 2008; Kononov et al., 2008; Lee et al., 2006; Lord et al., 2005; Marchesini and Weijermars, 2010; Wang et al., 2009; Zhou and Sisiopiku, 1997). It is generally known that vehicular collisions and other unplanned incidents increase travel time variability and decrease reliability. However, the relationship between reliability and safety is less understood. A necessary precursor in investigating this relationship involves developing a method to classify each crash in terms of whether or not it occurred during congested conditions and, if so, to further determine whether the congestion is most likely recurring or whether it is the result of an unplanned event.

Although mobility based collision classification is the key information for proactive traffic management on freeways, vehicular crashes are currently classified by injury or damage severity, vehicle type, number of vehicles involved, location, first harmful event, and other associated factors (National Safety Council, 2007). In addition, the studies for secondary incident or crash identification from primary incidents have been conducted by assuming that the secondary incidents identified are directly affected by the impact area of the primary incident, which means a crash in non-recurrent congestion area can cause secondary incidents identified by previous studies. However, to manage an ATMS effectively, the system operator should identify not only crashes in non-recurrent congestion but also crashes in recurrent congestion or in an uncongested area.

In order to invest capacity expansion and deployment of ITS technologies in current facilities to alleviate congestion efficiently and effectively, transportation jurisdictions should know the spatiotemporal impact of bottlenecks. Therefore, there is a need for developing a methodology for quantifying the impact of bottlenecks. In a bottleneck impact area, it is likely that congestion occurs due to a bottleneck activation, by an incident occurring within
temporal boundary of bottleneck activation. Furthermore, crashes occurring in recurrent congestion aggravate congested impact spatiotemporally. Accordingly, a quantification of recurrent bottleneck impact starts from separating impacts by different sources of congestion.

An approach for segregating recurrent congestion area holds the promise of quantifying impacts accurately. In addition to quantifying recurrent congestion impact, additional impacts of incidents occurring within an active bottleneck should be monitored to determine how mobility and reliability worsen. This challenge brings valuable information to traffic managers and operators by assessing where bottlenecks are vulnerable to incidents.

Finally, most of these studies used point detectors such as inductive loop detectors and overhead or side-fire radar detectors. However, an in-depth examination of recurrent bottlenecks and classification under operational conditions requires high-resolution traffic data and more accurate records to identify primary crashes. In this research, probe vehicle speed data obtained from a private vendor (INRIX) were utilized. It facilitates a more precise quantification of congestion and bottlenecks, providing a more comprehensive picture of traffic problems.

1.1 Research Objectives and Key Tasks

The primary research objectives of this effort are to identify “recurrent” bottleneck and its impact area and to develop several major components in a proactive ATMS. To satisfy the objectives, several key tasks were formulated:

- To develop a novel data-driven recurrent bottleneck identification methodology using historical speed data;
- To investigate recurrent bottlenecks and their impact areas;
➢ To develop a robust and easily implementable collision classification methodology based on operational conditions using link speed data and congestion thresholds; and,

➢ To develop a data-driven approach for quantifying recurrent congestion impacts.

Figure 1-1 Overall Framework

This thesis is conducted following the depicted framework shown in Figure 1-1. Figure 1-2 illustrates what is planned to be executed throughout the research. In the figure, the yellow box denotes a historical spatiotemporal bottleneck activation area. The red star shows a crash occurring within an active bottleneck, which will demonstrate a case of the results of collision classification under different types of congestion. Finally, the black line
drawn in the figure depicts the historical recurrent congestion-impacted area due to an active bottleneck.

<table>
<thead>
<tr>
<th>Segment</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Time period</th>
</tr>
</thead>
</table>

Historical spatiotemporal boundary of recurrent bottleneck activation

Recurrent congestion impact area

A collision occurring at a recurrent bottleneck

Figure 1-2 Spatiotemporal Traffic State Matrix for Key Tasks

1.2 Research Scope

Most traffic data used in this thesis were obtained through by the Regional Integrated Transportation Information System (RITIS) operated by the University of Maryland CATT Lab. For the collision classification, crash data were gathered from Traffic Engineering Accident Analysis System (TEAAS) provided by North Carolina Department of Transportation.

All methodologies presented here were developed based on heuristic methods using rational thresholds. The results and recommendations of this research will be valid for many
areas. However, it should be noted that both public agencies and researchers should clarify and understand their own regional unique characteristics of recurrent bottlenecks and crashes.

1.3 Research Contribution

This thesis presents a comprehensive understanding of freeway recurrent bottleneck and its impact area to researchers or agencies who are interested in analyzing freeway bottlenecks. Many studies have been conducted on bottleneck identification and understanding spatiotemporal characteristics. However, this research is unique from other studies in several ways in that it.

- Defines recurrent bottleneck locations with time span of bottleneck activation based on long-term temporal link speed data;
- Provides novel and defensible definition of recurrent and non-recurrent congestion;
- Classifies congestion types when crashes occur; and,
- Distinguishes and quantifies recurrent congestion impact separated from collision-induced congestion impact.

The results of the research proposed will support decision makers in their efforts to implement mobility, reliability, and safety treatments that are precisely targeted and effective.

1.4 Organization of the Thesis

This thesis consists of seven chapters. In the next Chapter a literature review on related topics is provided. Chapter Three provides glossary of abbreviations and terms. Chapter Four proposes a data-driven recurrent bottleneck identification method. In the chapter, bottleneck, breakdown, and congestion are discussed and defined clearly. This
chapter also introduces a novel concept of the definition of recurrent congestion. Chapter Five presents a comprehensive collision classification methodology. This is based on the result of recurrent bottleneck identification. Chapter Six provides a data-driven approach for quantifying spatiotemporal recurrent congestion impacts. Chapter Seven finally provides a summary of the research results, including findings, conclusions, and future recommendations.
CHAPTER 2  LITERATURE REVIEW

Many studies have been defined and identified congestion by using the concepts of breakdown, bottleneck, and congestion, and in fact they used these concepts interchangeably. This calls for a thorough review on them. To define and identify recurrent bottleneck, a literature review for distinguishing between recurrent and non-recurrent congestion is the key to classify congestion types of which crashes occur. With the background, the literature review in this study focuses on the definition and identification of congestion, bottleneck, breakdown, recurrent congestion, and secondary incident.

The first section of this literature review concerns the definition of congestion by exploring various congestion identification methods in numerous studies. Many times, each study had its own definition for congestion and tried to identify congestion under the definition. The second section is about distinguishing recurrent and non-recurrent congestion. Most studies also had their own definition of the classification methodology. The third section is about the secondary incident identification methods by exploring previous studies. The final section is about approaches for quantifying bottleneck impacts.

2.1 Traditional Congestion and Bottleneck Identification

There are three major concepts in identifying congestion conditions: “congestion”, “breakdown”, and “bottleneck”. These are obviously different concepts and definitions, but many studies have been conducted using a single definition alone. Therefore, these are required concepts defined in the purpose of the study.
2.1.1 Breakdown

The “breakdown” phenomenon is defined as a traffic transition state to congested states from free flow states conditions. There are two approaches to define and identify breakdown points: speed-based and other definitions such as occupancy-based. Previous studies using speed-based definitions of breakdown were based on either a pre-specified speed threshold or a precipitous speed drop. Elefteriadou et al. proposed that the average speed threshold of all lanes on a freeway be dropped below 56 mph for a period of at least five minutes (Elefteriadou 1995). Another study by Jia et al. (2010) developed a methodology to identify the breakdown phenomenon by applying a combination speed and density thresholds: 1) the observed speed is below the critical speed (different value on freeways); 2) the observed density is greater than or equal to the boundary between level of service C and C (LOS C/D). Figure 2-1 shows a schematic combination speed and density threshold. This method was proposed to filter points out in the black circle area in the figure. Breakdown was referred to as the points below the red line of critical speed and to the right of the LOS C/D line.

In addition to speed-based definition of breakdown, several studies have proposed an alternative definition of breakdown, especially which of sudden occupancy increase over flow rate. Hall and Agyemang-Duah used a ratio of the occupancy over flow rate as the breakdown indicator. In this study, a ratio of 1.1 was used as the threshold (Hall and Agyemang-Duah 1991). Zhang and Levinson also used a threshold of occupancy as the indicator of breakdown. If the occupancy from a detector is larger than 25%, this station is referred to as a condition region (Zhang and Levison 2004). Finally, the Highway Capacity
Manual 2010 (TRB 2010) defines a segment as having an equal or greater than 1 ratio of demand/capacity.

Figure 2-1 Critical Speed and LOS C/D on I-880 Speed Flow Data (Jia et al. 2010)

2.1.2 Bottleneck

A traffic “bottleneck” is well known as a “physical bottleneck” and is usually defined as any location where there is a physical reduction in the roadway width, such as lane drops. Neudorff et al. defined bottleneck as a short segment of highway with insufficient capacity (Neudorff et al. 2011). Although bottleneck is usually defined as a physical location, several studies have developed bottleneck identification methodologies with no distinction between bottleneck and congestion, using their definitions interchangeably.

Chen et al. presented an algorithm for identifying bottleneck locations in time activated (Chen et al. 2004). This study used speed difference as an indicator of bottleneck activation. The speed difference based algorithm is below:
\[ x_j - x_i < 2 \text{miles} \]  
Equation 2-1

\[ v(x_k, t) - v(x_i, t) > 0 \quad \text{if} \quad x_i \leq x_k < x_j \]  
Equation 2-2

\[ v(x_j, t) - v(x_i, t) > 20 \text{mph} \]  
Equation 2-3

\[ v(x_i, t) < 40 \text{mph} \]  
Equation 2-4

Location \( x_i \) is upstream of \( x_j \), \( x_k \), \( x_l \) are the detectors between these locations. If the above four inequalities hold, there is an active bottleneck between two locations, \( x_j, x_i \), with \( x_i < x_j \), during period \( t \). The authors defined recurrent bottleneck with the following equation:

\[
\sum_{t_{i_1} \leq t \leq t_{i_2} - N + 1} A_i(t) \geq qN, \forall t_i
\]
Equation 2-5

Where \( A_i(t) = 1 \), if there is an active bottleneck at location \( i \) in time period \( t \). In this study a recurrent bottleneck is defined as a sustained bottleneck if there are at least several active bottlenecks periods (or 25 minutes) within specified consecutive periods (or 1 hour), \( N \).

Warita et al. exhibited a method for identifying potential bottlenecks along with freeways. The ratio (85th percentile) of free-flow speed was selected as the threshold of bottleneck occurrence (Warita et al. 2006). However, the threshold value for determining bottlenecks was not clearly specified in this study. Although a case study was performed using 5-minute speed data collected from Tokyo metropolitan expressway, the accuracy of the result was not well justified.

Wiezorek et al. conducted a rigorous evaluation of a previous bottleneck identification method proposed by Chen et al. (Chen et al. 2004), in which a bottleneck was declared if the speed at upstream detectors was below the maximum upstream speed threshold (denoted as MaxUpstreamSpeed) and the speed difference between upstream and downstream detectors was above the minimum speed differential threshold (labeled as
MinSpeedDifferential) (Wieczorek 2010). This study conducted a sensitivity analysis to adjust the three parameters in Chen’s model (MaxUpstreamSpeed, MinSpeedDifferential, and Aggregation Interval) to achieve the best model performance. Using the sensitivity analysis model, five values of each parameter in a total of 125 combinations were tested using the archived dataset of the northbound I-5 corridor in Portland, Oregon. For comparison purposes, 91 bottlenecks over 24 days were extracted manually (and visually) using the oblique-curve method and were set as the baseline reference points for evaluating the outcomes of Chen’s approach. To account for the impacts of each parameter as well as the interactions between parameters, the analysis of variance (ANOVA) model was performed independently for each of the three score functions (SumScore, ProductScore, and Accuracy). Consequently, the original parameter value in Chen’s model applied for the San Diego (SA) data (20 mph minimum speed differential, 40 mph maximum upstream speed, and 5-min aggregation) were close to, but not the same as, the optical settings for this Portland freeway (15 mph differential, 35 mph maximum upstream speed, and 3-min aggregation). The authors also recommended that, for researchers and transportation managers in other cities wishing to implement a system using the Chen’s approach, it is necessary to perform a similar SA procedure to determine the optimal parameter settings for their own network. Finally, this study just introduced the historical percentile methodology to visualize years of data for the first time as seen in Figure 2-2.
Yildirimoglu and Geroliminis presented a methodology for forecasting travel time along a roadway segment using speed data at fixed loop detectors (Yildirimoglu and Geroliminis 2012). In this study, the travel time prediction problem was divided into two parts: 1) forecasting major traffic events on the roadway (e.g. bottlenecks) and 2) determining the speed profile in the time-space diagram. A previous bottleneck identification method proposed by Chen et al. was employed to determine the location and spatial extent of the bottlenecks. Since a large number of detectors and time periods in a day may result in large number of observations, the authors utilized the principal component analysis (PCA) method to reduce the dimensions of the dataset. A probability distribution map of congestion, established by clustering analysis, was further developed for calculating travel time. During the second part, the speed profile inside the bottlenecks was calculated using the minimum of instantaneous and average speed; while the travel time for non-bottleneck locations was computed directly using instantaneous travel speed. Five-minute loop detector data, covering a 60 mile section of I-5 in the district of the San Diego/Imperial area was extracted from the California freeway performance measurement system (PeMS). The experiment results
indicated that the proposed methodology provided promising travel time predictions for various traffic flow conditions.

Florida (2011) presented a methodology for identifying bottlenecks on Florida’s Strategic Intermodal System (SIS) using vehicle probe data and travel time reliability measures. The vehicle probe data, obtained from INRIX, provided travel speeds from the roadway for an entire year at five-minute intervals. To identify bottlenecks at the statewide and districtwide level, this study employed two TTR measures: 1) planning time index (PTI) and 2) frequency of congestion (FOC). The frequency of congestion is defined as the percent of time that travel speeds fall below 75% of the free-flow speed during the daytime. Bottlenecks were identified as the portion of the congested roadway which has the highest combination of PTI and FOC with Figure 2-3. The proposed can be updated annually with the latest vehicle probe data.

Figure 2-3 Illustration of Congested Roadways and Bottlenecks (Florida, 2011)
The Vehicle Probe Project Suite provides bottleneck ranking (RITIS 2016) by defining and identifying bottlenecks as a Traffic Message Channel (TMC)’s speed goes below 60 percent of the free-flow speed as seen in Figure 2-4. In contrast to other bottleneck activation algorithms discussed above, the potential bottleneck suspected is used if speed at a TMC segment drops below the reference speed of that segment and if that speed is maintained for at least 5 minutes. This method declares a termination of bottleneck if the speed rises above the reference speed and that continues for at least 10 minutes.

![Diagram of bottleneck identification](image_url)

Figure 2-4 Bottlenecks Identification on the Vehicle Probe Project Suite (RITIS 2016)
2.2 Recurrent Congestion Versus Non-Recurrent Congestion

Traffic congestion can be classified as either recurrent or non-recurrent. A consensus about the definition of non-recurrent congestion is any unexpected delay caused by an incident, a work zone, adverse weather, and so forth (Hallenbeck et al. 2003, Skabardonis et al. 2003, Kwon et al. 2006, Medina 2010, Chung 2011, Sullivan et al. 2013). However, a variety of definitions exists for recurrent congestion. For instance, Caltrans (2016) defined recurrent congestion as when the average speed drops below 35 miles per hour for 15 minutes or more on a typical weekday on a freeway. Schaefer considered a value of 1.5 for the Travel Time Index (TTI) as being the threshold of recurrent congestion (Schaefer et al. 2011). Dowling et al. defined recurrent congestion as being caused by demand surges or capacity deficiencies in peak periods (Dowling et al. 2004).

A study by Hallenbeck et al. defined non-recurrent congestion and then associated any congestion not within those non-recurrent congestions’ characteristics as recurrent congestion (Hallenbeck et al. 2003). The non-recurrent congestion in the study was defined as a condition where the lane occupancy is 5 or more percentage points higher than the median for all days during the period of interest.

Another study by Medina (2010) developed a method to distinguish non-recurrent and recurrent congestion based on delay threshold using loop detector data as seen in Figure 2-5. This study introduced the concept of “recurrent”. In this study, ‘recurrent congestion segments’ (RCS) with delays on 50% or more of the days during a specified period were defined to have recurrent congestion. Non-recurrent delays were calculated by subtracting recurrent delay to total delay at segment $i$ in time $t$. In step 8, this methodology established non-recurrent delay causes in incident databases. If no matched incidents exist with the non-
recurrent delays, it moves on to step 11 to check network effects. If there is no problem with network effects, then the incident is finally classified as another cause of non-recurrent delay.

Skabardonis et al. used the average and the probability distribution of delays to distinguish recurrent and non-recurrent congestion (Skabardonis et al. 2003). Non-recurrent delay represents the extra delay caused by any incidents, while recurrent delay means the

Figure 2-5 Methodology to Assess Recurrent and Non-recurrent Congestion (Medina 2010)
delay in the absence of those incidents. Thus, total delay is the sum of non-recurrent congestion and recurrent congestion.

Chung defined non-recurrent traffic congestion as the extra delay caused by incidents compared with the annual average section travel speed (Chung 2011). For instance, if the free-flow speed is 60 mph and the annual average section travel speed is 30 mph during peak periods, then it is assumed that recurrent congestion occurs. In the study, the difference the mean speed $\bar{s}_i(t_m)$ and the threshold, $\alpha \cdot \sigma_{si(t_m)}$, where $\alpha$ is a positive value, any particular speed $\hat{s}_i(t_m)$, mean speed $\bar{s}_i(t_m)$, and standard deviation speed $\sigma_{si(t_m)}$ was drawn from the distribution of $s_{in}(t_m)$, respectively. On the basis of such results, two discriminatory values as seen in Figure 2-6 can be assigned to all cells representing the speed $\hat{s}_i(t_m)$ as follows:

$$D_i(t_m) = \begin{cases} 0 & \text{if } \hat{s}_i(t_m) > \bar{s}_i(t_m) - \alpha \cdot \sigma_{si(t_m)} \\ 1 & \text{otherwise} \end{cases}$$

Equation 2-6

Figure 2-6 Reduced Speed by Accident (left: $D_i(t_m) = 1$, right: $D_i(t_m) = 0$)(Chung 2011)
In recent, there was one study (Sullivian et al. 2013) which used the link based speed data (the INRIX Traffic Message Channel data) for defining recurrent and non-recurrent congestion. The Standard Normal Deviate (SND) procedure, which was first proposed by (Dudek et al. 1974), was used to distinguish recurrent and non-recurrent congestion. The SND for each speed was calculated as shown in the following equation:

$$SND_{it} = \frac{|\hat{u}_{it} - \bar{u}_{i}|}{SD}$$

Equation 2-7

Where,

SND: standard normal deviate;

\(i/t\): number of row/column respectively;

\(\hat{u}_{it}\): speed at segment \(i\) in time \(t\);

\(\bar{u}_i\): average speed at segment \(i\) during the specified time period \(T (T = \sum t)\); and,

SD: standard deviation.

This procedure allows users to scan historical speed data, identify congestion, and characterize them as either recurrent or non-recurrent with 4 threshold values between -1.3 and -1.9.

Previous studies use either the mean or the median of a speed distribution during a specified time of day to measure recurrent delay and extra delay caused by incidents as non-recurrent congestion on a freeway. However, Oxford’s Dictionary defines “recurrent” as being something that occurs often or repeatedly. In other words, recurrent congestion means a “predictable” location with time and when drivers fell like “this area on this time is often and repeatedly congested.”
2.3 Secondary Crash

In this thesis, crashes which occur in non-recurrent congestion locations are classified as such. This definition approximates that of a secondary crash as found in the literature. In other words, this type of crash classification can be referred to as a secondary crash by crashes that occur during a non-recurrent congestion period. Thus, classification of crashes in a non-recurrent congestion need further investigated methodologies to identify secondary crashes. Previous regarding studies for identifying secondary crashes were two folds: 1) secondary crash or incident identification and 2) major factors associated with such a crash. This literature review focused on the previous one for the purpose of this thesis.

The key to identifying secondary crashes is to determine the boundaries of the primary incident area as seen in Figure 2-7, since secondary incidents (crashes) are defined as crashes occurring within the impact area of primary incident.

Figure 2-7 Classifying a Secondary Incident (Chou and Miller-Hooks 2010)
The methodologies used either static or dynamic thresholds to determine the boundaries of a primary incident impact area. Static thresholds employ a fixed boundary. For example, if an incident occurs within 15 minutes and 1 mile upstream of a primary incident, it is classified as a secondary incident, assuming these values meet the thresholds (Raub 1997a). Moore et al. identified secondary incidents as those that occur within two hours and two miles of a primary incident using California (CA) Highway Patrol data sources (Moore et al. 2004). The study found that the frequency of secondary crashes is about 1.5% to 3% to that of primary crashes. However, the primary incident impact area can actually be longer than the fixed threshold boundaries.

Methodologies that use dynamic thresholds can overcome some of the limitations of static approaches. In 2007, Sun and Chilukuri developed a dynamic threshold method by varying the back of a queue location throughout the entire duration of the incident (Sun and Chilukuri 2007). This study indicates that by using a dynamic approach up to a 30% difference can arise in the number of incidents classified as secondary. Chou and Miller-Hooks proposed a simulation-based secondary incident filtering method (SBSIF) using the CORSIM microscopic simulation tool (Chou and Miller-Hooks 2010). They implemented a regression model for corner point identification of the boundaries of the primary incident impact area along with the SBSIF method. In another research study performed in Virginia, Zhang and Khattak analyzed the cascading incident event duration (Zhang and Khattak 2010a). They identified and analyzed not only single-pair events (one primary and one secondary incident) but also large-scale events (one primary and multiple secondary incidents) by categorizing them as either contained or extended, using deterministic queueing analysis methods. In other words, if a secondary incident is the last one being cleared during
such an event, it is considered an extended event; otherwise, it is classified as a contained event. Later they developed an incident management integration tool to calculate dynamic incident duration predictions, secondary incident occurrences, and incident delays (Khattak et al. 2012).

Recently, Chung introduced a procedure to identify secondary crashes by different types of primary crashes in the impact area and developed a method to separate the non-recurrent congestion from any recurrent congestion using inductive loop detector data (Chung 2013). Yang et al. proposed the use of historical virtual sensor measurements to identify secondary crashes using a Representative Speed Contour Map (RSCP) with percentile speed of historical incident-free virtual sensor speed measurements of each spatiotemporal cell (Yang et al. 2013). Using the measurements, the study tried to show recurrent congestion areas and non-recurrent congestion areas in a speed contour map only with the RSCP. However, to appropriately find the precise recurrent congestion area, the bottleneck point needs to be identified first. In addition, the distribution of speed collected from detectors did not follow a specified distribution. Therefore, it is hard to determine a representative historical speed. Table 2-1 shows the comparison of secondary incident identification from previous studies.

Most of the studies used point detector data, such as inductive loop detectors and overhead or side fire radar detectors. However, an in-depth examination of secondary crashes requires high-resolution traffic data and more accurate incident records to identify primary crashes.
<table>
<thead>
<tr>
<th>Study</th>
<th>Thresholds for determining congested impact boundary</th>
<th>Method for determining congested impact area</th>
<th>Event type considered</th>
<th>Occurrence rate of secondary incident (direction)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Raub 1997a)</td>
<td>Fixed</td>
<td>15 min. and 1 mile</td>
<td>Crash</td>
<td>15.5% (same)</td>
</tr>
<tr>
<td>(Karlaftis <em>et al.</em> 1999)</td>
<td>Fixed</td>
<td>15 min. and 1 mile</td>
<td>Crash</td>
<td>35% (same)</td>
</tr>
<tr>
<td>(Moore <em>et al.</em> 2004)</td>
<td>Fixed</td>
<td>2 hours and 2 miles</td>
<td>Incident</td>
<td>1.5~3% (same)</td>
</tr>
<tr>
<td>(Hirunyanithiwattana and Mattingly 2006)</td>
<td>Fixed</td>
<td>1 hour and 2 miles</td>
<td>Crash</td>
<td>4.4% (same)</td>
</tr>
<tr>
<td>(Zhan <em>et al.</em> 2009)</td>
<td>Dynamic</td>
<td>Maximum queue and dissipation time of the potential lane-blockage primary incident</td>
<td>Crash</td>
<td>3.23% (same)</td>
</tr>
<tr>
<td>(Chou and Miller-Hooks 2010)</td>
<td>Dynamic</td>
<td>A regression model for corner point identification of primary incident boundary</td>
<td>Incident</td>
<td>15% (same)</td>
</tr>
<tr>
<td>(Vlahogianni <em>et al.</em> 2010)</td>
<td>Dynamic</td>
<td>A Bayesian network model using the characteristics of primary incident</td>
<td>Crash</td>
<td>16% (same)</td>
</tr>
<tr>
<td>(Zhang and Khattak 2010a)</td>
<td>Dynamic</td>
<td>Queue-based secondary incidents identification</td>
<td>Incident</td>
<td>2.7 (same), 0.38 (opposite)</td>
</tr>
<tr>
<td>(Chung 2013)</td>
<td>Dynamic</td>
<td>Binary speed contour plot on a representative speed contour map when a crash occurs</td>
<td>Crash</td>
<td>8.1 (same), 3.7 (opposite)</td>
</tr>
<tr>
<td>(Yang <em>et al.</em> 2013)</td>
<td>Dynamic</td>
<td>Congestion speed contour plot on a representative speed contour plot when a crash occurs</td>
<td>Crash</td>
<td>8.42% (same)</td>
</tr>
</tbody>
</table>
2.4 Quantification of Bottleneck Impact

There have been a few studies related to quantifying bottleneck impacts. In the RITIS database, a dashboard is provided for bottleneck ranking (RITIS 2016). The bottleneck ranking method quantifies bottleneck by an impact factor, which is the product of average duration, average maximum length of queue, and number of bottleneck occurrences for a given period of time on a given TMC per the following equation:

\[ IF = D \times L \times O \]  

Equation 2-8

Where,

- \( IF \): impact factor
- \( D \): average duration (minutes)
- \( L \): average maximum length of queue (miles)
- \( O \): number of occurrences

Another study by Liu and Fei (2010) proposed a fuzzy-logic-based approach that can diagnose the severity of bottleneck based on travel delay and frequency. This approach used nine linguistic rules for evaluating the bottleneck severity level as seen in Figure 2-8.
There are several limitations to those approaches. In order to quantify and rank bottlenecks properly, the head of the recurrent bottleneck should be identified first. However, those approaches rank and quantifies all segments with no attempt to identifying the bottleneck head location, which is critical to identifying problem area. In addition, those approaches quantify impact factors without separating the additional congestion impacts due to non-recurring incidents occurring within recurrent bottleneck impact areas. In the real
world, there is always a probability that those same bottlenecks can be exacerbated when there is a crash within their area of impact. The above methods may overestimate the impact factor.

2.5 Summary

Extensive literature reviews were mainly focused on four areas: breakdown and bottleneck identification, distinguishing between recurrent and non-recurrent congestion, secondary crash identification, and quantification of bottleneck impact. Some of the major findings are summarized as follow:

- The definition of congestion and its identification method should be clearly defined because the same field data would be likely produce much different congestion values.
- There is no detailed definition and identification of “recurrent bottleneck”. Therefore, the “recurrent bottleneck” should be clearly defined in real “recurrent” terms, where congestion occurs repeatedly in specified time periods.
- The use of a threshold from a speed distribution for each spatiotemporal cell in defining “recurrent congestion” is improper because the cells have different speed distribution. Therefore, it is hard to apply to distinguishing recurrent and non-recurrent congestion. Although a speed distribution is used to distinguish recurrent and non-recurrent congestion, we cannot say whether speed drops were caused by incidents or not.
- Previous secondary crash identification used point-based data, such as inductive loop detectors. However, hi-resolution data are required because
using the point-based data is elusive when determining congestion boundaries from the data collection points.

➢ There is no crash classification under operational conditions.

➢ It is imperative that recurrent congestion and additional congestion due to an incident occurring within a spatiotemporal historical bottleneck impact area should be identified and quantified separately when it comes to identifying and monitoring bottlenecks and their associated impacts more accurately.
CHAPTER 3  DATA-DRIVEN SPATIOTEMPORAL “RECURRENT” BOTTLENECK IDENTIFICATION

3.1 Introduction

Traffic congestion and safety issues are increasingly contributing to major challenge for transportation agencies as populations grow, motorization increases, and population densities change. Congestion is contributing to longer travel time, higher fuel consumption, and increased emissions of air pollutants. Nearly 40% of roadway congestion in the United States stems from recurrent bottlenecks (Cooner 2011), which can cost both agencies time and money if not properly addressed. Agencies should therefore identify and understand where and when congestion occurs recurrently.

Although an active bottleneck is “a physical point on the network upstream of which one finds a queue and downstream of which one finds freely flowing traffic” (Bertini and Leal 2005), most studies have focused on identifying congestion with no attention to distinguishing recurrence level of at the same “bottleneck” location. In the real world, it is likely that several bottlenecks can be activated within a congestion-impacted area. Accordingly, the detailed spatial and temporal characteristics of bottlenecks should be first identified in order to quantify and diagnose a bottleneck and its impact area.

This chapter introduces an easily implementable methodology for identifying spatiotemporal “recurrent” bottlenecks in a large-scale network. The proposed methodology uses link-based speed data to create a spatiotemporal traffic state matrix that more accurately and directly can identify bottlenecks compared to data collected from point-based detectors. Unlike previous bottleneck identification schemes, this work defines and adapts unambiguous concepts of breakdown, bottlenecks, and congestion, including a crisp
definition of “recurrent” bottlenecks. The findings from this study can help decision makers better prioritize investment in current facilities and/or develop suitable strategies to alleviate congestion.

In the remainder of this chapter, a review of relevant literature is presented followed by the methodology. In the next section, the methodology was applied to statewide interstates of North Carolina. Finally, findings and recommendations for further research are discussed.

3.2 Review of Relevant Studies

3.2.1 Bottleneck, Breakdown, and Congestion Definitions

The “breakdown” phenomenon is typically defined as a traffic transition from a free flow state to a congested state. The two approaches most frequently used to define and identify breakdown are speed-based and occupancy-based. Previous studies using speed-based definitions select either a pre-specified speed (Lorenz 2001, Jia et al. 2010) or a precipitous speed drop threshold (Elefteriadou 1995, Brilon 2005). Other studies have proposed an alternative definition of breakdown phenomenon (Hall and Agyemang-Duah 1991, Zhang and Levison 2004), based on the rate of occupancy increase with flow rate.

Various approaches were developed for identifying bottlenecks using point-based detector and probe vehicle data. Chen et al. (2004) developed an algorithm for identifying locations at activate time. In this study, a bottleneck was activated if the speed at upstream detectors dropped below a cut-off upstream speed threshold and the speed difference between upstream and downstream detectors was above a cut-off speed difference threshold. The study also defined recurrent bottleneck as a sustained bottleneck if there were at least several active bottleneck periods (e.g. 25 minutes) within specified consecutive periods (e.g. 1 hour). Several studies applied the Chen et al.’s method to identify bottlenecks to other sites and
other application (Liu and Fei 2010, Wieczorek 2010, Yildirimoglu and Geroliminis 2012). For instance, Wieczorek et al. (2010) conducted a sensitivity analysis to adjust the cut-off speed values in the Chen et al.’s method.

In recent years, probe vehicle data has been used to identify bottleneck locations. For example, researchers in Florida (2011) developed a methodology for identifying bottlenecks on Florida’s Strategic Intermodal System (SIS). Using travel time reliability measures that include the planning time index (PTI) and the frequency of congestion (FOC). The latter was defined as the fraction of time that travel speeds are below 75% of the free-flow speed in daytime. Bottlenecks were identified as the links or the congested roadway which has the highest combination of PTI and FOC. INRIX bottleneck ranking method defines bottlenecks as a condition when a Traffic Message Channel (TMC)’s speed falls below 60% of the reference speed for at least 5 minutes (INRIX 2016). The method declares a bottleneck cleared if the speed recovers above the reference speed for at least 10 minutes. Although a traffic “bottleneck” is well known as a “physical bottleneck”, usually any location where there is a capacity reduction that causes congestion, many of the studies have used bottleneck and congestion identification interchangeably. Recently, Hale et al. (2015) developed a congestion and bottleneck identification tool. In the tool, congestion was identified using cut-off speed values which the user can select using their own definition.

### 3.2.2 Recurrent Congestion

Traffic congestion can be classified into two types: recurrent and non-recurrent congestion. Non-recurrent congestion is defined as any delay occurrence caused by any unexpected events such as incidents, weather, work zone, and etc. (Hallenbeck et al. 2003, Skabardonis et al. 2003, Dowling et al. 2004, Chung 2013). However, the transportation
literature currently lacks a robust definition for recurrent congestion. Rather, authors have historically identified recurrent congestion based on their own definition for the purpose of their studies. For instance, Medina (2010) deems a segment with delays on 50% or more of the days during a specified period is defined to have recurrent congestion. Another study by Hallenbeck et al. (2003) defined the median for all days during the period of interest. Skabardonis et al. (2003) defined recurrent congestion as the delay in absence of incidents. Recurrent congestion by Chung (2011) is referred as the difference between free flow speed and the annual average section travel speed during peak periods.

Very often, previous studies used parameters from the speed distribution in a time window of day to distinguish recurrent congestion from non-recurrent congestion. Under this paradigm, bottlenecks on rural roads, which means that any congestion detected would be classified as non-recurrent.

Given these circumstances, this study develops an easily implementable approach for identifying recurrent bottleneck location with time, which does not require speed distribution to distinguish recurrent and non-recurrent congestion.

3.3 Methodology

3.3.1 Overall Framework

The overall framework is outlined in Figure 3-1. In the first step, daily congestion identification contour maps are created for a road segment in a specified time period. The congestion contour map created is a spatial temporal binary congestion state matrix (congested or free flow). In the second step, a historic congestion contour map is created by overlapping congestion identification contour maps from several days or months. The spatial recurrent bottlenecks are identified using a simple algorithm that uses the historic congestion
contour map in the third step. Finally, time spans of activation of the recurrent bottlenecks are identified. The framework presented in Figure 3-1 comprises several data-driven models, as explained next. In the remainder of this section, data sources that are used in this study are discussed, and the further details of the proposed methodology are described.

![Figure 3-1 Overall Recurrent Bottleneck Identification Framework](image)

3.3.2 Data Source

This research used the link-based speed data generated from GPS-enabled vehicle probes using INRIX technology (INRIX 2016). Those data can be directly downloaded from the Regional Integrated Transportation Information System (RITIS), a database that includes many performance measures and a user dashboard (RITIS 2016). In this study, speed and link attributes (Traffic Message Channel - TMC code) are used to identify recurrent bottlenecks.
3.3.3 Congestion Identification

3.3.3.1 Congestion, bottleneck, and breakdown phenomenon

Figure 3-2 depicts visual definition of these terms. This thesis defines “breakdown” as a transition point from an un-congested state to a congested state. A “bottleneck” is defined as a location where congestion occurs when the ratio of demand to capacity is equal to or greater than 1 from either physical characteristics or operational conditions. Finally, this thesis defines “congestion” as a traffic condition where the speed drops below the speed at capacity on a segment.

![Figure 3-2 Schematic Definition of Breakdown, Bottleneck, and Congestion](image)

3.3.3.2 Congestion value and index

Congestion value, \( C_{d\text{ or } w}(i, t, m) \), is defined as the ratio of speed to the free flow speed on segment \( i \) at time \( t \) on day \( m \). The congestion value is as follows:
\[ C_{dorw}(i,t,m) = \frac{MS(i,t,m)}{FFS(i)} \times 100\% \]  

Equation 3-1

Where,

\( i \): TMC segment \( i \),

\( t \): specified time interval in a day (e.g., 8:00-8:15, 15 min); typically \( t \) will vary from 1 to 96 for a single day,

\( m \): index for a day in the study period,

\( d \): weekday \( \in \) (Monday, Tuesday, Wednesday, Thursday, Friday),

\( w \): weekend \( \in \) (Saturday, Sunday),

\( MS(i,t,m) \): reported speed (mi/h),

\( FFS(i) \): free flow speed (mi/h) for segment \( i \), and

\( C_{dorw}(i,t,m) \): congestion value for segment \( i \) at time \( t \) on day \( m \).

Although the FFS \( (i) \) of a TMC segment is provided via INRIX.com, FFS \( (i) \) could not be applied to this thesis since different FFSs \( (i) \) are reported for different time periods. Therefore, this study uses the simple model for predicting free flow speed on a highway segment developed by the Florida DOT (Moses 2013), in which FFS \( (i) \), is simply calculated by adding 5 mph to the posted speed limit on a TMC segment. If \( C_{dorw}(i,t,m) \) is below a cut-off speed threshold, \( \alpha \), the TMC segment \( i \) is considered to be congested at time \( t \) during day \( m \). The congestion index, \( CI_{dorw}(i,t,m) \), based on \( C_{dorw}(i,t,m) \) is defined as follows:

\[
CI_{dorw}(i,t,m) = \begin{cases} 
1 & \text{if } C_{dorw}(i,t,m) < \alpha \\
0 & \text{otherwise}
\end{cases}
\]  

Equation 3-2
Where,

\( CI_{d,or-w} (i,t,m) \): Congestion Index at a spatiotemporal traffic state matrix (STM) \((i,t)\) and all other variables as defined earlier.

Figure 3-3 depicts a schematic congestion index contour map on Northbound I-85 in NC on January 3, 2014. In this study, \( \alpha \) is estimated with the ratio of speed at capacity to free flow speed in the latest version of the US Highway Capacity Manual (TRB 2010). For instance, 80\% of the cut-off speed threshold can be applied to the TMC where the posted speed limit is 60 mph (i.e. free flow speed is 65 mph).

![Congestion Index Contour Map (3 January 2014, on I-85 Southbound)](image)

**3.3.4 Average Historic Congestion Index**

The Average Historic Congestion Index, \( AHCI_{d,or-w} (i,t) \), is a key parameter for identifying a “recurrent” bottleneck. It is defined as the fraction of days in the reporting period \( T \) (typically one or more years) where a TMC segment \( i \) was congested at time \( t \), based on the specified congestion index, \( (CI_{d,or-w} (i,t,m)) \):
\[
AHCI_{d or w}(i,t) = \frac{\sum_{m=1}^{M} CI_{d or w}(i,t,m)}{M} \times 100(\%)
\]

Equation 3-3

Where,

\(M\): the number of days in the study period (e.g. 250 weekdays in a year)

\(AHCI_{d or w}(i,t)\): Average Historic Congestion Index at for segment \(i\) at time \(t\).

The AHCI proposed represents the probability that TMC segment \(i\) is congested at time interval \(t\) over \(M\) days of observation. For instance, Figure 3-4 shows that the probability of congestion for TMC 125+04965 at 7:30 AM in 2014 was 85%. An AHCI value of 20% or more would indicate congestion occurring on average more than a day a week (excluding weekends) on segment \(i\) at time \(t\).

![AHCI Contour Map (260 days-Weekdays of I-40 Westbound in 2014)](image_url)
3.3.4.1 Spatial patterns of AHCI

This thesis identifies recurrent bottlenecks based on the AHCI contour map. Therefore, the first step for identifying recurrent bottlenecks is to ascertain the spatiotemporal characteristics of the AHCI contour map. The question then is which TMCs is identified as a recurrent bottleneck?

To answer the question, the author identified three different spatial patterns observed on weekdays in 2014 on all Interstates in North Carolina. Figure 3-5 describes the patterns that are directly applicable to recurrent bottleneck identification.

The TMCs identified as a recurrent bottleneck usually have the highest AHCI (TMC A as seen in Site 1 of Figure 3-5 (a)) and are located just upstream of a significant drop in AHCI value. Therefore, these TMCs are clearly defined as recurrent bottleneck segments and are labeled as Spatial Pattern 1 in this study.

Different spatial patterns were associated with other recurrent bottlenecks in the AHCI contour map. Site 2, shown in Figure 3-5 (b), exhibits a pattern of AHCI where the TMC with the highest AHCI value (TMC C) is not located immediately upstream of a significant drop in AHCI. In this case, TMC C and the one downstream of it (TMC B) have AHCI values greater than 90% and TMC B is located upstream of a significant drop in AHCI. Therefore, TMC C is also part of the bottleneck and this pattern is called as Spatial Pattern 2. If the “physical” recurrent bottlenecks are located nearly at the end of TMC B and adjacent downstream of TMC C, the TMC B may report a travel speed based on mixed traffic flow. In this case, the AHCI value at TMC B may be lower than that at TMC C.

In Figure 3-5 (c), Site 3 describes the final spatial pattern detected in this research. No significant drop in AHCI values is observed for TMC D and the one downstream of it. If
TMCs with AHCI greater than $\beta \%$ (e.g. 50% is for Figure 3-5 (c)) are referred as recurrent bottlenecks, only TMC D can be included in a recurrent bottleneck. This conditions labeled Spatial Pattern 3.
3.3.4.2 Temporal patterns of AHCI

Another objective of this study is to identify the historical time span of bottleneck activation. Figure 3-6 depicts AHCI values over time on TMCs in the recurrent bottleneck...
impact area corresponding to Figure 3-5. Figure 3-6 (a) shows AHCI values exhibiting a bell shape curve over time. AHCI’s at 7:45 AM were at the highest values during congested conditions, then decreased gradually over time. This trend is labeled Temporal Pattern 1. Figure 3-6 (b) shows a slightly different temporal pattern. Several TMCs had AHCI values greater than 80% for 90 minutes, and then decreased sharply and this pattern is labeled Temporal Pattern 2. In Figure 3-6 (c) AHCI values on TMCs increased and then decreased slowly without any significant drop. However, the AHCI values where TMCs have greater than 50% appear similar to Temporal Pattern 1. It indicates that no difference between (a) and (c) of Figure 3-6 was found. Therefore, two temporal patterns were finally identified for the three bottleneck areas shown Figure 3-6.
Figure 3-6 Temporal Patterns of AHCI across Multiple TMC’s
3.3.5 Recurrent Bottleneck Identification

This thesis defines a “recurrent bottleneck” as a location having a spatiotemporal pattern observed over a specified study period. The Recurrent Bottleneck Identification (RBI) methodology explained later employs an exhaustive search algorithm to identify where, when, and how often congestion occurs at a bottleneck. Two constraints are imposed on the search. The first step searches recurrent bottleneck location on a contiguous sequence of TMC’s and then the latter step process the identified recurrent bottleneck locations to determine the historical time span of possible bottleneck activation.

3.3.5.1 Identification of Recurrent Bottleneck Location

The Recurrent Bottleneck Location Identification (RBLI) methodology employs an exhaustive search algorithm to identify the recurrent bottleneck locations. A TMC segment must have an AHCI value exceeding $\beta\%$. Secondly a recurrent bottleneck location can include at most two contiguous TMCs by the spatial pattern 2 of AHCI. Finally, those TMCs satisfied by above two constraints can be included in a recurrent bottleneck area if the AHCI of the first or second TMCs is greater than $\gamma$ times the AHCI value of its downstream two TMCs. The parameter $\gamma$ is defined as the threshold of spatial congestion difference in AHCI. The proposed location identification algorithm is processed using the following pseudo code:

\[
\text{For all each spatiotemporal } AHCI_{d\ or\ w}(i,t)
\]

\[
\text{If } AHCI_{d\ or\ w}(i,t) \geq \beta\%, \ 0 \leq \beta < 100, \ \gamma > 1
\]

\[
y_1 = AHCI_{d\ or\ w}(i,t) - \gamma \cdot AHCI_{d\ or\ w}(i+1,t) : \text{spatial pattern 1 and 3}
\]

\[
y_2 = AHCI_{d\ or\ w}(i,t) - \gamma \cdot AHCI_{d\ or\ w}(i+2,t) : \text{spatial pattern 2}
\]
\[ y_3 = AHCI_{d or w}(i, t) - AHCI_{d or w}(i + 1, t) : \text{spatial pattern 2} \]

\[
\text{if } y_3 \geq 0 \text{ \&\& } y_1 \geq 0 \text{ \&\& } y_2 \geq 0
\]

Then TMC (i) possibly as bottleneck

### 3.3.5.2 Identification of historical time span of recurrent bottleneck activation

A simple heuristic approach for the Recurrent Bottleneck Time span Identification (RBTI) is introduced. The time span starts at a TMC with an \( AHCI_{d or w}(i, t) \) exceeding \( \delta \% \) on a recurrent bottleneck. The \( AHCI_{d or w}(i, t) \) is then compared to the \( AHCI_{d or w}(i, t + 1) \), \( AHCI_{d or w}(i, t + 2) \), ..., \( AHCI_{d or w}(i, t + n) \) over time. If a TMC with \( AHCI_{d or w}(i, t + n) \) equals to or less than \( \delta \% \) and \( n \) is greater than 1, the cell with \( AHCI_{d or w}(i, t + n - 1) \) is referred as the last time of the time span of the recurrent bottleneck. The proposed pseudo algorithm for identifying time span of a recurrent bottleneck is as follows:

```plaintext
For all bottleneck cell (i,t)

for t=1:P (96=maximum t value)

if \( AHCI_{d or w}(i, t) > \delta \% \)

Stop if \( AHCI_{d or w}(i, t+n) \leq \delta \% \)

if \( n > 1 \)

Then the TMCs at the cell (i,t), (i,t+1)\ldots(i,t+n-1) are referred as activation time of the recurrent bottleneck
```
3.4 Case Study

3.4.1 Site Description

The proposed identification methodology is data-driven. Therefore, any data set that can be downloaded from link-based speed data such as INRIX is applicable. Statewide interstates across North Carolina (NC) identified by INRIX were selected for this case study. Aggregated 15-minute TMC data were used to create daily congestion contour plots for all weekdays in 2014. The extension of the interstates is 2,253 miles on 2,287 TMC segments, with an average TMC length of 1.11 miles. The posted speed limit on the interstates varies from 55 mph to 70 mph.

3.4.2 Congestion Identification and AHCI Contours

A total of 260 weekdays statewide congestion identification contours were created using the proposed methodology. Then, a statewide AHCI contour plot was created by overlapping those congestion identification contours. Figure 3-7 depicts a screenshot of the statewide AHCI plot from Johnston to Guilford County on I-40. Spatial information (the TMC segment) is shown on the vertical axis and temporal information (time) on the horizontal axis. AHCI values are shown in the cells and the higher AHCI value is, the deeper the red. Traffic flows from bottom to top.
3.4.3 Sensitivity Analysis and Illustrated Examples

In order to identify recurrent bottlenecks, cut-off thresholds need to first be determined. These thresholds can be selected by each state and/or local operator. Figure 3-8 depicts an example of bottleneck identification using 10% increments of $\beta$ between 20% and 60% with $\gamma = 2$ within a recurrent congestion area. This denotes how the bottlenecks are identified by different $\beta$ values. As a result, five bottleneck segments were identified using the selected thresholds for the area. This result indicated that several bottlenecks which affect congestion impacts can be activated simultaneously or continuously on contiguous TMCs in a recurrent congestion area.
This research conducted a sensitivity analysis arising $\beta$ in increments of 10% between 20% and 60%, $\gamma$ in increments of 0.5 between 1.5 and 3. Figure 3-9 shows the sensitivity for the number of bottlenecks identified to the $\beta$ on the AHCI criterion and $\gamma$. It is clear that if $\beta$ is between 40% and 60% and $\gamma$ is between 2 and 3, the number of bottlenecks becomes quite stable. However, if $\beta$ drops below 40% or $\gamma$ is decreased low 2, the number of bottlenecks changes significantly. Based on the sensitivity analysis, this thesis selected 50% as the threshold of $\beta$ and 2 for $\gamma$. In fact, the author identifies that recurrent bottlenecks can be defined conservatively if a TMC exceeding 50% of AHCI causes congestion occurrence within two days. This led the author to conclude 50% as the threshold.

---

**Figure 3-8 Bottlenecks Identified under Various Cut-off for AHCI**

<table>
<thead>
<tr>
<th>TMCCode</th>
<th>On-Road</th>
<th>Direction</th>
<th>County</th>
<th>Length (miles)</th>
<th>FR(mph)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>1.87</td>
<td>65</td>
<td>6-10 AM</td>
</tr>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>0.57</td>
<td>65</td>
<td>4%</td>
</tr>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>0.62</td>
<td>65</td>
<td>5%</td>
</tr>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>0.64</td>
<td>65</td>
<td>6%</td>
</tr>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>0.72</td>
<td>65</td>
<td>9%</td>
</tr>
</tbody>
</table>

**Identified at $\beta = 0.2$**

<table>
<thead>
<tr>
<th>TMCCode</th>
<th>On-Road</th>
<th>Direction</th>
<th>County</th>
<th>Length (miles)</th>
<th>FR(mph)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>1.41</td>
<td>65</td>
<td>55%</td>
</tr>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>0.74</td>
<td>65</td>
<td>60%</td>
</tr>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>1.17</td>
<td>65</td>
<td>61%</td>
</tr>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>0.54</td>
<td>65</td>
<td>64%</td>
</tr>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>2.13</td>
<td>65</td>
<td>67%</td>
</tr>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>0.53</td>
<td>65</td>
<td>7%</td>
</tr>
</tbody>
</table>

**Identified at $\beta = 0.6$**

<table>
<thead>
<tr>
<th>TMCCode</th>
<th>On-Road</th>
<th>Direction</th>
<th>County</th>
<th>Length (miles)</th>
<th>FR(mph)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>4.04</td>
<td>65</td>
<td>2%</td>
</tr>
<tr>
<td>125-04869</td>
<td>1-485</td>
<td>COUNTERCLOCKWISE</td>
<td>MEDICINE BURG</td>
<td>0.72</td>
<td>65</td>
<td>9%</td>
</tr>
</tbody>
</table>

Cells identified by cut-off AHCI:
- 20%
- 30%
- 40%
- 50%
- 60%
of $\beta$ is suitable for this study. In addition, the proposed $\gamma$ verified by visual observations was 2.

![Figure 3-9 Number of Bottlenecks Identified under Various Cut-off Threshold of AHCI and $\gamma$](image)

In the next step, the historical time span of recurrent bottleneck activation was identified. Figure 3-10 shows an example of a varying temporal boundary of the bottleneck activation using different $\delta$ values. The activation time span became steadily smaller with increasing $\delta$ values. In Figure 3-10, the span measured by the proposed methodology at 20% was found to be two hours (from 4:45 pm to 6:45 pm).
A sensitivity analysis was conducted for historical time span of recurrent bottleneck activation in 10% increments of $\delta$ between 20% and 60%. Figure 3-11 exhibits the sensitivity for the time span identified on recurrent bottlenecks to $\delta$ on the AHCI criterion. It is clear that if $\delta$ is between 20% and 30%, the distribution of the time spans on recurrent bottleneck activation is highly concentrated at the average boundary while for values greater than 30% of $\delta$ tend to be scattered. For this study, $\delta = 20\%$ was selected, which means that a bottleneck activated on a TMC with greater than 20% chance on any given weekday.

Figure 3-10 Schematic Temporal Boundary of a Recurrent Bottleneck (I-85 Northbound in Mecklenburg County, NC)
### 3.4.4 Recurrent Bottlenecks in NC

The following thresholds were finally applied to identify spatiotemporal recurrent bottlenecks for this study: $\beta = 50\%$, $\gamma = 2$, and $\delta = 20\%$. As a result, a total of 95 bottleneck segments were identified with the selected thresholds on all Interstates in NC. Figure 3-12 depicts a GIS-based map showing the feasible activation time span on the recurrent bottlenecks identified. As expected, most of bottlenecks were located in both Raleigh-Durham and Charlotte metropolitan areas. Of the bottlenecks identified, the historical temporal span of 60 bottlenecks varied between 2 and 3 hours, and 23 of them were exceeded 3 hours.

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. time span (hrs)</td>
<td>3.7</td>
<td>3.1</td>
<td>2.8</td>
<td>2.8</td>
<td>2.7</td>
</tr>
</tbody>
</table>

*of 95 bottlenecks identified on Interstates in NC*
Figure 3-12 Spatiotemporal Recurrent Bottleneck Identification in NC

3.5 Conclusions

This research presented a method for identifying spatiotemporal recurrent bottlenecks. The methodology has been tested on 2,253 miles of statewide interstates in North Carolina for all weekdays in 2014. In addition, this study defined clearly terms of breakdown, bottleneck, and congestion. Previous studies identified bottlenecks as congestion occurrences but did not provide consistent methods for identifying recurrent bottleneck conditions specifically. The proposed method developed through this study uses spatial and temporal concepts to establish a robust definition and framework for identifying a recurrent bottleneck based on an average congestion history using probe-reported speed data. Moreover, the data-driven proposed identification algorithm can be readily applied to other sites where link-based speeds are available. This methodology can help road infrastructure
decision makers evaluate current facilities by providing critical information about where and when congestion occurs recurrently.

There are some limitations to the study. First, the link-based speed data provides uniform traffic performance information. Thus, this study employed the starting point of TMC segments as the bottleneck location. This may cause few errors in identifying recurrent bottlenecks spatially. There is a need to develop a robust approach to establishing thresholds for identifying congestion and recurrent congestion in order to reduce the opportunity for evaluator bias. Developing such methods for identifying time spans for bottleneck activation likelihood, specifically for recurrent bottleneck conditions, can lead to more accurate projections of spans and ultimately more sound congestion interventions. The study analysis period was one year, but no verification process was conducted on it. This calls for developing a robust guideline for selecting the analysis period reasonably. Finally, there was no consideration about quantification of congested impact area because this chapter was focused on developing a robust and easily implementable methodology to identify spatial temporal recurrent bottlenecks. In chapter 6, this thesis presents research efforts to develop a robust and novel quantification methods for evaluating the locations of recurrent congested impact areas resulting due to a recurrent bottleneck.
CHAPTER 4  A NOVEL COLLISION CLASSIFICATION
METHODOLOGY BASED ON TEMPORAL LINK SPEED DATA AND CONGESTION THRESHOLDS

4.1 Introduction

The overarching role of an Advanced Traffic Management Systems (ATMS) is to improve reliability and safety through active real-time traffic management and control. Vehicular crashes endanger life and limb, damage property, and cause severe congestion, thereby presenting an obstacle to the goal of improving the safety, efficiency, and sustainability of the transportation system. To manage an ATMS effectively, it is important to understand the conditions under which crashes happen. This knowledge can and should inform how the crashes are managed and which resources are allocated for incident response programs. For instance, Variable Speed Limits (VSL) may be an effective countermeasure to prevent crashes during recurring congestion. Conversely, minimizing the incident response and clearance time for crashes may be the most effective strategy for crashes occurring outside of recurring congestion periods.

Another motivation for understanding and classifying the types of congestion under which crashes occur is that this classification makes it possible to relate reliability and safety (Zhou and Sisiopiku 1997, Lord et al. 2005, Lee et al. 2006, Golob et al. 2008, Kononov et al. 2008, Wang et al. 2009, Marchesini and Weijermars 2010). It is generally known that vehicular collisions and other unplanned incidents increase travel time variability and decrease reliability. However, the relationship between reliability and safety is less well understood. A necessary precursor to investigating this relationship is a method to classify
each crash in terms of whether or not it occurred during congested conditions, and if so, to further determine whether the congestion is most likely recurring or the result of an unplanned event.

In response, this thesis presents a robust and easily implementable methodology that can classify all reported crashes in terms of the operational conditions under which each crash occurred. Unlike previous secondary collision identification schemes, the proposed methodology requires neither identification of the precipitating incident nor a definition of the precipitating incident’s impact area. The proposed methodology can support decision makers in their efforts to implement safety and mobility treatments that are precisely targeted and effective.

In the remainder of this chapter, relevant studies are reviewed and knowledge gaps are highlighted. Then, the proposed methodology is described and applied to a 170-mile section of I-40 in North Carolina. This chapter concludes with a presentation of the findings, conclusions, and recommendations for further research.

4.2 Literature Review

4.2.1 Distinguishing Recurrent and Non-recurrent Congestion

Traffic congestion can be classified as either recurrent or non-recurrent. A consensus about the definition of non-recurrent congestion is as follows: any unexpected delay caused by an incident, a work zone, adverse weather, and so forth (Hallenbeck et al. 2003, Skabardonis et al. 2003, Medina 2010, Chung 2011, Sullivan et al. 2013). However, a variety of definitions exist for recurrent congestion. For instance, Caltrans defined recurrent congestion as when the average speed drops below 35 mile per hour for 15 minutes or more on a typical weekday on a freeway (CALTRANS 2016). Schaefer (2011) considered a value
of 1.5 for the Travel Time Index (TTI) as being the threshold of recurrent congestion. Dowling et al. defined recurrent congestion as being caused by demand surges or capacity deficiencies in peak periods (Dowling et al. 2004).

Medina developed a method to distinguish non-recurrent and recurrent congestion based on delay threshold using loop detector data. In this study, a segment with delays on 50% or more of the days during a specified period is defined to exhibit recurrent congestion (Medina 2010). Another study by Hallenbeck et al. (2003) defined non-recurrent congestion as being a condition where the lane occupancy is five or more percentage points higher than the median for all days during the period of interest. Skabardonis et al. (2003) used the average and the probability distribution of delays to distinguish recurrent and non-recurrent congestion. Non-recurrent delay represents the extra delay caused by any incidents, while recurrent delay means the delay in the absence of those incidents. Thus, total delay is the sum of the effect of non-recurrent and recurrent congestion. Chung (2011) defined non-recurrent traffic congestion as the extra delay caused by incidents compared with the annual average section travel speed. For example, if the free flow speed is 60 mph and the annual average section travel speed is 30 mph during peak periods, then it is assumed that recurrent congestion occurs. Recently, one study (Sullivian et al. 2013) used the INRIX Traffic Message Channel (TMC) data for defining recurrent and non-recurrent congestion. The study proposed to use a simple Standard Normal Deviate (SND) procedure to detect non-recurrent vehicle incidents. The procedure allows users to scan historical speed data, identify congestion, and characterize them as either recurrent or non-recurrent with 4 threshold values between -1.3 and -1.9 (Sullivian et al. 2013).
Previous studies use either the mean or the median of a speed distribution during a specified time of day to measure recurrent delay and extra delay caused by incidents as non-recurrent congestion on a freeway. However, Oxford’s Dictionary defines “recurrent” as being something that occurs often or repeatedly. In other words, recurrent congestion means a “predictable” location in a predictable time window when drivers feel “this area at this time is often and repeatedly congested.” This thesis investigates each link to define recurrent congestion area and proposes a “new” definition and methodology to identify recurrent congestion area.

4.2.2 Identification of Secondary Incidents

In this thesis, crashes which occur in non-recurrent congestion locations are classified as such. This definition approximates that of a secondary collision as found in the literature. The methodologies reviewed use static or dynamic thresholds to determine the boundaries of the primary incident impact area. Static thresholds employ a fixed boundary. For instance, if an incident occurs within 15 minutes and within 1 mile upstream of a primary incident, it is classified as a secondary incident, assuming these values meet the thresholds (Raub 1997a). However, the primary incident impact area may actually be longer than the fixed threshold boundaries.

Methodologies that use dynamic thresholds can overcome some of the limitations of static approaches. In 2007, Sun and Chilukuri (2007) developed a dynamic threshold method by varying the back of queue location throughout the entire duration of the incident. This study indicates that using a dynamic approach may yield up to a 30% difference in the number of incidents classified as secondary. Chou and Miller-Hooks (2010) proposed a simulation-based secondary incident filtering method (SBSIF) using CORSIM microscopic
simulation tool. They implemented a regression model for corner point identification of the boundaries of the primary incident impact area along with the SBSIF method. Zhang and Khattak (2010b) identified and analyzed not only single-pair events (one primary and one secondary incident) but also large-scale events (one primary and many secondary incidents) using deterministic queuing analysis methods. Recently, Chung (2013) introduced a procedure to identify secondary crashes caused by different types of primary crashes in the impact area and developed a method to separate the non-recurrent congestion from any recurrent congestion based on average and standard deviation speed for every spatiotemporal cell using loop detector data. Yang et al. (2013) proposed the use of historical virtual sensor measurements to identify secondary crashes. They used a Representative Speed Contour Map (RSCM) with percentile speed of historical incident-free virtual sensor speed measurements of each spatiotemporal cell. Using these measurements, they displayed recurrent and non-recurrent congestion areas in a speed contour map. However, to appropriately determine the precise recurrent congestion area, the bottleneck needs to be identified first. In addition, most distributions of speed collected from sensors did not follow a normal distribution. Therefore, it was difficult to determine a representative historical speed distribution.

Most reviewed studies use point detector data such as inductive loop detectors and overhead side fire radar detectors. However, an in-depth examination of secondary crashes requires high-resolution traffic data and more accurate incident records to identify primary crashes. In this chapter, link-based travel time (speed) data are used and a methodology is proposed that does not require any information on primary incidents (crashes and non-crash incidents) to identify crashes in non-recurrent congestion.
4.2.3 The relationship between Congestion and Crashes

Several studies have tested the relationship between crash rates and traffic flow or density (Hallenbeck et al. 2003, Skabardonis et al. 2003, Medina 2010, Chung 2011, Sullivian et al. 2013). However, the results have been inconsistent. Zhou and Sisiopiku (1997) found that there was a U-shaped relationship between V/C ratio and crash rate on freeways. On the other hand, Lord et al. (2005) found no relationship either between crash rate and congestion or between severity and congestion.

Insofar as crash types and congestion are concerned, there are two important questions: does congestion influence the crash type, and vice versa. This thesis is concerned with the latter. Golob et al. (2008) and Lee et al. (2006) found that rear-end crashes are more likely under unstable traffic flow conditions. On the other hand, Wang et al. (2009) found that traffic congestion has little or no impact on crash rates. However, their statistical model fails to pass statistical significance tests. The reason might be that they used a congestion index to test the relationship, and that index was based on the average congestion level across an entire year. Therefore, the method was not tested using congestion information at the time the crashes occurred. The study results presented here address that issue by using a congestion measure that is predicated on the traffic conditions at the time of the crash.

4.3 Methodology

A four-step methodology was employed in this research, as depicted in Figure 4-1. In Step 1, both mobility and crash data are integrated onto directional link segments or TMC’s. In Step 2, using this information, crashes are classified into those occurring during normally congested or uncongested periods. In Step 3, non-recurrent congestion is identified for crashes that occurred during a congested period. And in Step 4, any remaining crashes are
identified into the type of congestion (i.e. recurrent or non-recurrent). The crashes, which could not be classified, are re-identified with a supplemental classification methodology. In the remainder of this section, data sources that are used in this study are explained, and the proposed methodology for crash classification is described.

![Methodology Flow Procedure](image)

Figure 4-1 Methodology Flow Procedure

### 4.3.1 Data Sources

#### 4.3.1.1 Mobility data

INRIX.com was the source for the link based speed and travel time data employed in the study. INRIX.com uses GPS enabled probe vehicles to collect speed information. Geocoding is based on freeway and arterial Traffic Message Channel (TMC) codes, as defined by Tele Atlas and Navteq. Each TMC code corresponds to a directional roadway segment with a geo-located beginning and end points. INRIX reports average travel times,
average speeds, reference speeds, scores, and C-Values by time of day and day of the week. The reference speed is the 85th percentile measured speed, capped at 65 mph. The score indicates if the reported speed is based on historical data, real time data, or a blend of the two. The C-Value is a measure of the confidence attributable to the real time data.

The Regional Integrated Transportation Information System (RITIS) provides an integrated spatiotemporal contour map of the traffic congestion for each TMC segment by time of day (Zhan et al. 2007). RITIS also provides contour maps for comparative speeds, congestion, a travel time index, and so forth. This chapter used 15-minute aggregated congestion data. As mentioned earlier, RITIS also defines the congestion value as Equation 3-1.

Figure 4-2 (a) depicts sample congestion data obtained directly from RITIS.org. The TMC segment is shown on the vertical axis and time on the horizontal axis. Congestion values are shown in the cells. Traffic flow direction is from bottom to top from TMC segment 125-10218 to TMC segment 125N04645. Figure 4-2 (b) describes sample congestion index (CI) data which are explained in the chapter 3.
4.3.1.2 Crash data

North Carolina Department of Transportation maintains data on traffic crashes in The Traffic Engineering Accident Analysis System (TEAAS). Records in the TEAAS database indicate: injury severity, roadway characteristics, crash characteristics and location, and environmental characteristics in terms of a set of pre-specified categories (NCDOT).

4.4 Congestion Assessment Methodology

To achieve the objectives of the study, a method was developed to determine the level of congestion that was extant at the time of each crash. The one presented here makes use of a Congestion Index (CI), an Average Historic Congestion Index (AHCI), and a Recurrent Bottleneck Location Identification (RBLI).
4.4.1 Congestion Index (CI)

In chapter 3, the congestion index, $CI_{dorw}(i,t,m)$, was aimed at labeling whether segment $i$ was congested at time $t$, as illustrated in Figure 4-2 (b). For this purpose, the congestion value for that segment is determined. If the congestion value is below a threshold value $\alpha$, segment $(i)$ is classified as being congested at time $t$. In this chapter, a value of 80% was used as the congestion value threshold. This value is consistent with the ratio of speed at capacity to free flow speed in the latest version of the US Highway Capacity Manual (TRB 2010). Thus:

$$CI_{dorw}(i,t,m) = \begin{cases} 1 & \text{if } C_{dorw}(i,t,m) < 0.8 \\ 0 & \text{otherwise} \end{cases}$$

Equation 4-1

Where,

$CI_{dorw}(i,t,m)$: Congestion Index at a spatiotemporal cell $(i,t)$.

4.4.2 AHCI and RBLI

The Average Historic Congestion Index (AHCI) is used for identifying recurrent congestion for the purpose of this chapter. It denotes the probability that segment $i$ was congested at time $t$ as shown Equation 3-3.

AHCI contour maps are used to define recurrent bottlenecks as well as their influence areas. To illustrate, this research used three different spatial patterns observed as seen in Figure 3-5. Each block corresponds to an AHCI value at a TMC at a certain time of day. A TMC with the highest AHCI ($\beta \geq 50\%$) value at a given time of day is defined as a recurrent bottleneck area. In addition, bottleneck influence area includes all the TMC segments upstream of a bottleneck area with AHCI values greater than 20%.
This research used RBLI to identify the recurrent bottleneck locations. Figure 4-3 depicts a bottleneck segment in the case that TMC U which produce an AHCI value greater than 2 times (here \( \gamma = 2 \)) the AHCI value of its downstream TMC D.

<table>
<thead>
<tr>
<th>TMC Segment</th>
<th>Name</th>
<th>Miles</th>
<th>7:15AM</th>
<th>7:30AM</th>
<th>7:45AM</th>
<th>8:00AM</th>
<th>8:15AM</th>
<th>8:30AM</th>
<th>8:45AM</th>
<th>9:AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>125P04965</td>
<td>Gorman St/Exit 295 (TMC D)</td>
<td>0.38</td>
<td>1%</td>
<td>4%</td>
<td>15%</td>
<td>17%</td>
<td>11%</td>
<td>11%</td>
<td>12%</td>
<td>9%</td>
</tr>
<tr>
<td>125+04965</td>
<td>Gorman St/Exit 295 (TMC U)</td>
<td>1.26</td>
<td>7%</td>
<td>72%</td>
<td>88%</td>
<td>90%</td>
<td>73%</td>
<td>53%</td>
<td>26%</td>
<td></td>
</tr>
<tr>
<td>125P04964</td>
<td>Lake Wheeler Rd/Exit 297</td>
<td>0.74</td>
<td>8%</td>
<td>67%</td>
<td>87%</td>
<td>83%</td>
<td>64%</td>
<td>40%</td>
<td>21%</td>
<td></td>
</tr>
<tr>
<td>125+04964</td>
<td>Lake Wheeler Rd/Exit 297</td>
<td>0.39</td>
<td>6%</td>
<td>40%</td>
<td>70%</td>
<td>55%</td>
<td>26%</td>
<td>14%</td>
<td>8%</td>
<td>5%</td>
</tr>
<tr>
<td>125P04963</td>
<td>US-70/US-401/Exit 298</td>
<td>0.58</td>
<td>5%</td>
<td>16%</td>
<td>44%</td>
<td>23%</td>
<td>13%</td>
<td>8%</td>
<td>5%</td>
<td>5%</td>
</tr>
<tr>
<td>125+04963</td>
<td>US-70/US-401/Exit 298</td>
<td>0.2</td>
<td>5%</td>
<td>6%</td>
<td>16%</td>
<td>12%</td>
<td>7%</td>
<td>5%</td>
<td>3%</td>
<td>2%</td>
</tr>
</tbody>
</table>

Figure 4-3 Bottleneck Segment Identification Using an AHCI Contour Map

4.5 Identifying Crashes by Congestion Type

As previously mentioned, adding a congestion level indicator to the crash records involves four steps as shown in Figure 4-1. This section provides detailed explanations for each. The detailed flow chart is shown in Figure 4-4.
4.5.1 Step 1: Locate the crash

First, the location of the crash has to be ascertained. In this step, TMC codes are associated with the crashes. Since the crash record includes longitude and latitude information, the TMC codes can be identified easily. A link shapefile of TMCs is applied to join the two datasets. Thus, each crash becomes associated with specific TMC, which is defined as “TMCc.”

4.5.2 Step 2: Crashes in uncongested conditions

In this step, the crashes which occur during uncongested conditions are identified. As shown in Figure 4-4, if the crash occurred on a TMC that was uncongested at the time of the crash, then the crash is labeled as “Case 1 – Crash not in congested conditions”. Otherwise, it is for the assessed in Step 3.
To determine the congestion status of TMC $i$ at the time of the crash $t$, two main metrics are employed: $CI_{dorw}(i,t,m)$ and $AHCI_{dorw}(i,t)$. As mentioned earlier, if $CI_{dorw}(i,t,m)=0$, then TMC $i$ is considered to be uncongested at time $t$ and vice versa. The next set of processes involves the historical pattern $AHCI_{dorw}(i,t)$. Two breakpoints, $\delta$ and $\gamma$ are used to evaluate $AHCI_{dorw}(i,t)$. If $AHCI_{dorw}(t) < \eta$, then TMC $i$ is considered to be historically uncongested at time $t$ otherwise if $AHCI_{dorw}(t) \geq \beta$ then it is historically congested.

Thus, in Step 2, the test performed is as follows. If $CI_{dorw}(t,m) = 0$, then the crash is labeled as occurring when the segment was operating uncongested. If $CI_{dorw}(t,m) = 1$, then further analysis is required, based on $AHCI_{dorw}(t)$.

The values for $\eta$ and $\beta$ are determined through a sensitivity analysis. Figure 4-5 shows the sensitivity for the number of crash labeled as “Case 2 – Crash in non-recurrent congestion” to the upper and lower bounds on the AHCI criterion. It is clear that if the upper bound is between 50 and 70 and the lower bound is between 10 and 20, the number of Case 2 selections becomes quite stable. However, if the upper bound drops below 50 or the lower bound is increased beyond 30, the number of Case 2 classifications changes significantly. This led the author to conclude that the upper and lower bounds used in the selection process presented above represented a good combination to employ. The thesis selected 60% as the threshold of $\beta$ and 20% as that for $\eta$. 
The proposed values for $\eta$ and $\beta$ are 20% and 60% respectively, so if $AHCI_{d or w}(t) < 20\%$, then TMC $i$ is deemed to be congested at time $t$ status at less than one day in five days. If $AHCI_{d or w}(t) \geq 60\%$, then it is an a congested state at least every other day.

**4.5.3 Step 3: Crashes in congested conditions**

In Step 3, if $AHCI_{d or w}(t) < \eta$, then the crash is labeled “Case 2”. If not, it is passed to Step 4.

**4.5.4 Step 4: Classify remaining crashes**

In this step, if $AHCI_{d or w}(t) \geq \beta$, then the crash is labeled “Case 3 – Crash in recurrent congestion”. If not, which means $20\% \leq AHCI < 60\%$, then a sub-test is performed to ascertain whether there is historically consistent recurrent congestion at the time of the crash. If positive, the crash is labeled as “Case 3”. If not, it is labeled “Case 2.” In the area of

![Figure 4-5 Number of Case 2 under Various Upper and Lower Bound of AHCI](image)
uncertainty, i.e. $20% \leq \text{AHCI} < 60\%$, then a test is made to see whether this is a recurrent bottleneck downstream of queue crash location, and to assess whether the queue from the bottleneck spillback into the crash location. This algorithm is shown below:

\begin{align*}
\text{If TMC}_c \text{ is in a recurrent bottleneck area} & \quad \text{Go to Case 3} \\
\text{Elseif TMC}_c \text{ is not in a recurrent bottleneck area} & \quad \text{Go to Case 3} \\
\text{Else} & \quad \text{Go to Case 2}
\end{align*}

In the algorithm, $\text{AHCI}_{\text{TMC}_c}(i,t)$ is the AHCI of TMC$_c$, and $\text{AHCI}_{\text{TMC}_b}(i+b,t)$ is the AHCI of the recurrent bottleneck location downstream from the TMC$_c$.

### 4.6 Illustrated Case Study

The objective of the collision classification methodology is to develop a novel methodology for classifying collisions in different types of congestion. The developed methodology is based on the concepts of CI and AHCI. Therefore, it is important to illustrate these concepts in a real-world case study.

#### 4.6.1 Study Site

A 170-mile section of I-40 was used to test the methodology. It extends from Exit 259 (at the split with I-85 in Durham) to Exit 420 (Gordon Rd, at the eastern end of I-40 outside of Wilmington). This section contains 98 TMCs, with an average length of 1.66 miles. The
posted speed limit is either 65 or 70 mph. The data employed were for Tuesdays, Wednesdays, and Thursdays in April, May, September, and October of 2012 and 2013 (a total of 105 days). Aggregated 15-minute TMC data were used to create the congestion contour plots. The crash dataset comprised 500 records (234 westbound crashes and 266 eastbound crashes) during the same weekdays and timeframe.

4.6.2 Crash Classification Results

Crashes on the TMCs were classified using the methodology described above. As shown in Figure 4-6, a “Case 1 – Crash not in congested conditions” occurring during a time when congestion is not present. The number of Case 1 was 419 out of 500 (i.e., 84%). Figure 4-6 shows the contour map for one of the crashes classified as Case 1. The remaining 81 crashes (16%) were subsequently analyzed in Steps 3 and 4 of the methodology.

<table>
<thead>
<tr>
<th>TMC Segment</th>
<th>Rd. Name</th>
<th>Miles</th>
<th>7:30PM</th>
<th>7:45PM</th>
<th>8:00PM</th>
<th>8:15PM</th>
<th>8:30PM</th>
<th>8:45PM</th>
<th>9:00PM</th>
<th>9:15PM</th>
<th>9:30PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>125N04860</td>
<td>Exit 287</td>
<td>0.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>125-04860</td>
<td>Exit 287</td>
<td>1.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>125N04861</td>
<td>Exit 285</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>125-04861</td>
<td>Exit 285</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>125N04862</td>
<td>Exit 284</td>
<td>0.7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>125-04862</td>
<td>Exit 284</td>
<td>0.6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>125N04863</td>
<td>Exit 283</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>125-04863</td>
<td>Exit 283</td>
<td>0.3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>125N04864</td>
<td>Exit 282</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4-6 Schematic Representative a Case 1 Crash Condition

In Step 3, if the TMCc in the AHCI contour map had a value less than 20%, it was classified as Case 2. Of the 81 remaining crashes, 62 were classified in Case 2 in Step 3 (Case 2 is a crash that occurs in non-recurrent congestion). Parts (a) and (b) of Figure 4-7 show a TMCc that was identified as belonging to Case 2. Finally, 19 crashes were passed to Step 4. Of these, only 9 had a value more than 60% and were placed in Case 3. Parts (c) and (d) of Figure 4-7 show an example of a TMCc (AHCI \_d or w(t) ≥ 60%) that was identified as
belonging to Case 3. The remaining 10 crashes were identified as being moved on to the sub-test in Step 4 and were subsequently also classified as Case 3.

Figure 4-7 shows an instance where the Case 3 classification was based on the CI contour map. Figure 4-7 (f) shows an instance where the Case 3 classification was based on the AHCI contour map. The results of the crash classifications identified from the
methodology are summarized in Table 4-1. As can be seen, the proportions of cases 1, 2, and 3 identified in the study were 84%, 12%, and 4%.

Table 4-1 Classification of the Crashes Based on the Proposed Methodology

<table>
<thead>
<tr>
<th>Step</th>
<th>Case Description</th>
<th>I-40</th>
<th>Westbound</th>
<th>Eastbound</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of Crashes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>Case 1 (Crash not in Congestion)</td>
<td></td>
<td>192 (82%)</td>
<td>227 (85%)</td>
<td>419 (84%)</td>
</tr>
<tr>
<td>Step 3</td>
<td>Case 2 (Crash in Non-recurrent Congestion)</td>
<td></td>
<td>36 (15%)</td>
<td>26 (10%)</td>
<td>62 (12%)</td>
</tr>
<tr>
<td>Step 4</td>
<td>Case 3 (Crash in Recurrent Congestion)</td>
<td></td>
<td>6 (1%)</td>
<td>13 (3%)</td>
<td>19 (2%)</td>
</tr>
<tr>
<td></td>
<td>To Algorithm for congestion type classification</td>
<td></td>
<td>3 (1%)</td>
<td>7 (3%)</td>
<td>10 (2%)</td>
</tr>
<tr>
<td></td>
<td>Case 2 (Crash in Non-recurrent Congestion) by the sub-test</td>
<td></td>
<td>0 (0%)</td>
<td>0 (0%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td></td>
<td>Case 3 (Crash in Recurrent Congestion) by the sub-test</td>
<td></td>
<td>3 (1%)</td>
<td>7 (3%)</td>
<td>10 (2%)</td>
</tr>
</tbody>
</table>

4.6.3 Comparative Analysis

The method described here enables the analyst to identify crashes in non-recurrent congested areas. These might be crashes that occur either in the impact area of a primary crash that is not locatable in time and space or in the impact area of non-crash incidents.

Table 4-2 shows the proportion of reported incidents or crashes and those not reported for the 62 samples of crashes in non-recurrent congested TMC’s. As seen in Table 4-2, in only 58% of the cases was a primary crashes or incident was also reported. This indicates that more crashes in a non-recurrent congested area will be detected by the use of the proposed methodology. Otherwise, only those crashes tied to a primary incident will be classified, in our case 39 of the possible 62.
Table 4-2 Reported vs. Unreported Primary Incidents or Crashes on Crash in Non-recurrent Congested Areas

<table>
<thead>
<tr>
<th>Crashes in Non-recurrent Congestion Locations</th>
<th>Reported Incident or Crash</th>
<th>No Data Primary Incident or Crash</th>
<th>Total Crashes in Non-recurrent congested TMC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Primary Crash</td>
<td>Primary Incident</td>
<td></td>
</tr>
<tr>
<td>Number of Cases</td>
<td>36</td>
<td>3</td>
<td>23</td>
</tr>
<tr>
<td>Percentage</td>
<td>58%</td>
<td>5%</td>
<td>37%</td>
</tr>
</tbody>
</table>

4.7 Conclusions

This chapter presented a method for classifying crashes based on the type of congestion in which they occur. The methodology has been tested using North Carolina crash data for 170 centerline miles on I-40 in North Carolina for Tuesdays, Wednesdays, and Thursdays in April, May, September, and October of 2012 and 2013. In addition, a new way to identify recurring congestion has been presented in support of the procedure. Unlike previous studies that used the mean and median of the speed distribution to distinguish recurrent and non-recurrent congestion pattern, the proposed method used a recurring congestion definition that is based on an average congestion history using probe-reported speeds. Moreover, the proposed recurrent bottleneck searching algorithm identified recurrent and non-recurrent congestion locations.

The proportion of secondary crashes (synonym to crash in non-recurrent congestion) identified in the case study is in line with results from previous classification studies, where the secondary crash percentages ranged from 2.2% to 15.5% (Raub 1997a, Raub 1997b, Moore et al. 2004, Hirunyanitiwattana and Mattingly 2006, Zhan et al. 2007, Zhan et al. 2009, Khattak et al. 2010, Vlahogianni et al. 2010, Zhang and Khattak 2011, Vlahogianni et al. 2012, Chung 2013, Yang et al. 2013). However, the proposed methodology yields a secondary crash proportion that is near the upper end of the range reported in these earlier
studies. This is to be expected because the proposed methodology classifies crashes as secondary crashes whenever they occur in atypical congestion without the need to identify the primary crash or incident event.

There are some limitations to the study. The most important is the need to validate or verify efforts regarding secondary incidents or crash identification. It calls for identifying and defining real-world secondary events with more detailed approaches in the real world. Another issue is that the link based traffic data provide uniform traffic performance information for the entire TMC. Thus, this study employed the starting and ending points of TMC’s as the beginning and ending points for the congestion. This may cause few errors in identifying non-recurrent congestion conditions because of the limitation of the TMC segment itself (different link lengths). However, the author expects that this limitation will be addressed in the future as vendors plan to provide sub-segment based link information. In addition, as the vendors’ market penetration is expected to dramatically increase over time, data quality issue will also be less critical.

Despite the fact that only 4% of the total crashes in this case study were identified as “Crash in recurrent congestion,” the percentage of “secondary” crashes caused by a primary incident in recurrent congestion increased compared well to the results of previous studies. Therefore, identifying these crashes calls for a further detailed classification methodology, which the author are presently investigating. Finally, there was no consideration about the impact of rubbernecking because the objective of this study was to focus on developing for a robust and easily implementable methodology to classify crashes in different types of congestion.
CHAPTER 5  RECURRENT CONGESTION IMPACT BASED ON SPATIOTEMPORALLY HISTORIC CONGESTED INFORMATION

5.1 Introduction

Traffic congestion is a major obstacle to continued economic growth, deteriorating mobility, travel time reliability, and the environment. Traffic congestion is generally classified into one of two types: recurrent and non-recurrent. Recurrent congestion is well-known as the condition in which demand exceeds capacity at a facility over a specified time period. Non-recurrent congestion references any delays caused by an un-expected event (Hallenbeck et al. 2003, Skabardonis et al. 2003, Dowling et al. 2004, Chung 2013). Of these, non-recurrent congestion can also be classified by the location where it occurs in relationship to recurrent bottlenecks: 1) at recurrent bottleneck but bottleneck is inactive before the non-recurrent congestion occurrence, 2) in recurrent bottleneck impact area, and 3) outside of recurrent bottleneck impact area. The two first cases result in extra congestion from recurrent congestion and thereby deteriorate mobility and reliability of roadway systems considerably (Khattak et al. 2012, Zhang et al. 2012). Accordingly, it is imperative that those two congestion cases should be identified and quantified separately when it comes to identifying and monitoring bottlenecks and their associated impacts more accurately.

Although the literature is replete with attempts to identify bottlenecks (Chen et al. 2004, Jiang 2010, Wieczorek 2010), a few link-based approaches for identifying bottlenecks included quantifying bottlenecks’ impact based on queue length, duration, and frequency (Florida 2011, INRIX 2016). Another study by Liu and Fei (2010) proposed a fuzzy-logic-based approach that can diagnose the severity of bottleneck based on travel delay and frequency. Those approaches made no attempt to distinguishing those cases occurring in
Many studies attempting to distinguish between recurrent and non-recurrent congestion have been conducted (Hallenbeck et al. 2003, Skabardonis et al. 2003, Dowling et al. 2004, Chung 2013). They used parameters for the speed distribution in a time of day at a segment to distinguish between recurrent and non-recurrent congestion. In other words, non-recurrent congestion was noted when a specified speed falls below a representative speed threshold (Chung 2011) regardless of cause of congestion. Despite the fact that the extra congestion occurring in recurrent bottleneck impact area extends spatiotemporally, these only considered the severity of congestion. This calls for separating extra congestion from recurrent congestion spatiotemporally to quantify bottleneck impact. Furthermore, recurrent bottleneck impacts may have a site-specific spatiotemporal shockwave phenomenon (May 1990). It indicates that associated links with time need to be identified in quantifying bottleneck impacts.

With these considerations in mind, the objective of this study is to develop a dynamic data-driven approach for quantifying recurrent congestion impacts. The proposed method is based on spatiotemporally historic congestion information. In this study, congestion impacts are classified by contributing event: 1) active bottleneck, 2) crash occurring at a recurrent bottleneck but bottleneck is inactive before the crash, and 3) crash occurring within an active bottleneck period. To quantify recurrent congestion impact, this study develops an innovative approach to capture the spatiotemporal recurrent bottleneck impact areas. In addition, a new procedure is proposed to define the stochastic variation of spatiotemporal congestion.
impacts. Finally, this study introduces a formulation to calculate recurrent congestion impacts by each recurrent bottleneck impact area.

This chapter is organized as follows. Following the introductory section with review of relevant works we present the overall procedure for this study. The procedure is followed by a detailed description of each component in the procedure and a case study of applying it to freeways across North Carolina. In the next section, several possible applications are discussed. Finally, the author concludes with the key findings and recommendations for future work.
5.2 Methodology

5.2.1 Overall Framework

The overall framework for this study is outlined in Figure 5-1: the left figure describes the study processes and outlines the main components, while the right figures depict simple schematic examples of the main tasks achieved by each component (A through G). As a first step, study location and time period should be decided. In this study, link-based speed data is used to create a daily CI contour map (A). An AHCI contour map (B) is created by summing the CIs created for a specified study period. As the AHCI contour map becomes available, it is used to identify recurrent bottlenecks with time span of bottleneck (C). In the next step, spatiotemporal congestion impacts occurring at recurrent bottlenecks identified are calculated (D). This information is stored into a knowledge base and supports the remaining applications of the framework.

The key objective of this study is to isolate spatiotemporal recurrent congestion from non-recurrent congestion. As stated above, in a recurrent congestion area, non-recurrent congestion can occur either at 1) a bottleneck location before the bottleneck is inactive or 2) within the impact time of an active recurrent bottleneck. Those two types are classified in this research. This study uses crash data classified under different operation conditions to separate non-recurrent (or collision-induced) congestion and recurrent congestion.

This study performs separation of the impacts by congestion type (E). A spatiotemporal impact distribution is then generated for each type of congestion impact and is compared to the distributions of the recurrent congestion impacts in the specified study period. The distribution of the impacts of collision-induced congestion occurring at a bottleneck before bottleneck is active can be applied to segregate historical recurrent...
congestion extent in step (F). Eventually, quantification of recurrent congestion area is performed (G).

![Figure 5-1 Overall Historic Congestion Impact Quantification Framework](image)

### 5.2.2 Spatiotemporal Congestion Impact

To achieve the objective of this study, it is essential to develop an approach for quantifying the spatiotemporal congested impact area. This thesis introduces a new concept called “Congestion Spatiotemporal Impact Index (CSII), CSIIe”, which takes into account the segment length, and the duration of congestion by an event or active bottleneck. Figure 5-2 shows a schematic example of estimation of the $CSII_e$ with a CI contour. The proposed $CSII_e$ is as follows:

$$CSII_e (miles \cdot hours) = \sum_{i=1}^{T(i)} \sum_{t=1}^{L_i} CI(i,t) \cdot \frac{R}{60}, \forall CI(i,t) = 1$$

Equation 5-1

Where,

---

77
\(c\): congestion caused by an event, active bottleneck or both,

\(i\): segment id,

\(I\): total number of consecutive TMC segments in the area of the impact,

\(t\): time interval (i.e. 15 min.) id,

\(T(i)\): congestion duration expressed in terms of time intervals on segment \(i\),

\(L_i\): length of segment (miles),

\(R\): time resolution (i.e. 15 min. \(= 15\)), and

\(CI(i,t)\): congestion index on segment \(i\) in \(t\).

---

**Figure 5-2 Estimation of Congestion Spatiotemporal Impact Index (CSII)**

<table>
<thead>
<tr>
<th>(L_i) (miles)</th>
<th>Congestion occurrence</th>
<th>Time period</th>
<th>(L_i \cdot \sum_{i=1}^{R} CI(i,t) \cdot \frac{R}{60} \cdot CI(t = 1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>0.7</td>
<td>0 1 1 1 1 1 1 1 1 1 1 1</td>
<td>0</td>
<td>1.40</td>
</tr>
<tr>
<td>1.2</td>
<td>0 1 1 1 1 1 1 1 1 1 1 1</td>
<td>0</td>
<td>2.40</td>
</tr>
<tr>
<td>0.3</td>
<td>0 1 1 1 1 1 1 1 1 1 1 1</td>
<td>0</td>
<td>0.60</td>
</tr>
<tr>
<td>0.8</td>
<td>0 0 1 1 1 1 1 1 1 1 1 1</td>
<td>0</td>
<td>1.40</td>
</tr>
<tr>
<td>1.1</td>
<td>0 0 1 1 1 1 1 1 1 1 1 1</td>
<td>0</td>
<td>1.93</td>
</tr>
<tr>
<td>1</td>
<td>0 0 0 1 1 1 1 1 1 1 1 1</td>
<td>0</td>
<td>1.50</td>
</tr>
<tr>
<td>0.9</td>
<td>0 0 0 0 1 1 1 0 0 0 0 0</td>
<td>0</td>
<td>0.45</td>
</tr>
</tbody>
</table>

**Flow direction**

CSII(miles-hours) 9.68

---

**5.2.3 Separation of Congestion Types**

As stated above there are two types of collision-induced congestion occurring in the recurrent congestion area shown in Figure 5-3. A collision classification developed by Song et al. (2015) is applied for this study to separate the impact of such collision-induced congestion types. In the study, collisions were classified by three different types of congestion when a crash occurs: 1) collision not in a congested area, 2) collision in a non-recurrent congestion area, and 3) collision in a recurrent congestion area. Of these, the crash data of the third type – collision in a recurrent congestion area – was used for this study.
There is an additional process to distinguish collisions occurring at a recurrent bottleneck location with time span of bottleneck activation but bottleneck is inactive before the collision and within an active recurrent bottleneck. This is processed based on the approach for identifying recurrent bottlenecks introduced above and comparing the bottleneck locations to the crash locations. As a final process in this section, this study classifies recurrent congestion as congestion occurring within a time span of bottleneck activation with no evidence of any incident shown in Figure 5-3 (a).
(a) Type 0 – Recurrent congestion (bottleneck activation)

(b) Type 1 – Congestion due to a crash but bottleneck is inactive before the crash

(c) Type 2 – Congestion due to a crash within an active bottleneck (recurrent congestion plus extra non-recurrent congestion)

Figure 5-3 Types of Congestion Occurring within a Recurrent Congestion Area
Congestion contributions are labeled by the types shown in Table 5-1 in each day of occurrence according to the following conditions occurring at recurrent bottlenecks. Type 0 is congestion due to an active bottleneck with no evidence of identified collision (referred to as recurrent congestion). Type 1 is defined as congestion due to a crash, but the bottleneck is inactive before the crash (referred as collision-induced congestion occurring at a recurrent bottleneck). Finally, Type 2 is congestion due to a crash within an active bottleneck (referred as recurrent congestion plus collision-induced congestion occurring at a recurrent bottleneck).

Table 5-1 Congestion Type Occurring in Recurrent Bottleneck Impact Area

<table>
<thead>
<tr>
<th>Congestion type</th>
<th>Bottleneck activation</th>
<th>Crash occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 0</td>
<td>Yes</td>
<td>No evidence</td>
</tr>
<tr>
<td>Type 1</td>
<td>No evidence</td>
<td>Yes</td>
</tr>
<tr>
<td>Type 2</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

In order to distinguish the spatiotemporal impacts of recurrent and non-recurrent congestion, $CSII_{c}$ is estimated and captured for congestion type by each recurrent bottleneck activation and collision occurrence during a specified study period. The author defines $CSII_{c,s}$, where $s$ is: 0 = Type 0, 1 = Type 1, and 2 = Type 2. Distributions of $CSII_{c,s}$ are then generated from the data.

### 5.2.4 Recurrent Congestion Area

The key for quantifying recurrent congestion impacts is to identify the historical spatiotemporal recurrent congestion extent with a robust methodology. This starts by segregating recurrent congestion from the impact due to Type 1. In this study, identifying historical recurrent congestion spatiotemporal extent is based on AHCI contours. As mentioned earlier, each value of $AHCI (i,t)$ in the AHCI contour is the probability of
recurring congestion at segment \( i \) in time \( t \) during a study period. In the contour, cell \((i,t)\) which is further from the bottleneck activation, either in time and/or in space, has a lower probability of frequent congestion. This study proposes an approach for identifying the maximum recurrent congestion extent using the characteristics with the distribution of \( CSII_{c,0} \) and \( CSII_{c,1} \). In the contour, the maximum value of \( AHCI \) \((i,t)\) divided by 100 at a recurrent bottleneck, \( b \), is overall congestion frequency, \( f_b \), which in \( b \) of \( M \) days. The total congestion frequency is the summation of recurrent congestion frequency, \( f_r \), and non-recurrent congestion frequency, \( f_{nr} \). This study assumes that \( CSII_{c,0} \) (recurrent congestion) with values that exceed the minimum \( CSII_{c,1} \) values (collision-induced congestion) are considered to be exclusively incident-induced congestion. This is followed by three hypotheses. First is possible that the impact due to Type 1 may include Type 0, if a bottleneck is activated in the time span of a recurrent bottleneck activation. Second, traffic passing a recurrent bottleneck is usually high even though the bottleneck maybe inactive. Finally, AHCI contours include all no-recorded non-recurrent congestion but is assumed to be recurrent congestion. The cut-off threshold, \( \lambda \), of \( AHCI(i,t) \) needed to segregate a recurrent bottleneck impact can be derived from these assumptions alternatively it can be set at 20\% (which is the conservative value of the probability that a congestion event occurs at least once a week for recurrent congestion). This leads to the following of models:

\[
\max \frac{AHCI(i,t)}{100} \cdot M = f_b, \forall i, t \in B \tag{Equation 5-2}
\]

\[
f_b = f_r + f_{nr} \tag{Equation 5-3}
\]

\[
\frac{f_{nr}}{f_b} = [1 - F(\min CSII_{c,1})] \tag{Equation 5-4}
\]
\[
\therefore f_{nr} = f_b[1 - F(\text{min } CSII_{c,i})]
\]

\[
\lambda = \frac{f_{nr}}{M} \times 100(\%)
\]

where,

- \(B\): spatiotemporal boundary of bottleneck activation
- \(F(\text{min }CSII_{c,i})\): the minimum cumulative density function of \(CSII_{c,i}\)

### 5.2.5 Quantification of Recurrent Bottleneck Impacts

To further classify bottlenecks, this study proposes a new concept called the “Recurrent Bottleneck Spatiotemporal Impact Index (RBSII)”, which takes into account the segment length, the duration of the congestion, and the frequency of occurrences (AHCI’s) by a recurrent bottleneck area. The proposed RBSII is as follows:

\[
\text{RBSII (miles \cdot hours per activation) = } \sum_{i=1}^{I} \sum_{t=1}^{T(i)} L_i \cdot \frac{AHCI(i,t)}{100} \cdot \frac{R}{60}, \forall AHCI(i,t) \geq \lambda
\]

Equation 5-7

where,

- \(i\): segment id,
- \(I\): total number of consecutive segments in the area of the impact,
- \(L_i\): length of segments (miles),
- \(t\): time interval (e.g. 15 min.),
- \(T(i)\): congestion duration expressed in terms of time intervals on segment \(i\),
- \(R\): time resolution (e.g. 15 min. = 15), and
- \(AHCI(i,t)\): Average Historic Congestion Index for time interval \(t\) on segment \(i\).
5.3 Case Study

5.3.1 Data Description

In this chapter, processed speed data were downloaded from the RITIS.org database. This case study selected statewide interstates across North Carolina. The data covered the period from January 1, 2011, to December 31, 2014, aggregated at 15-min intervals.

In addition to speed data, crash location and time are used for separating collision-induced congestion impact from recurrent congestion. Crash data stored in TEAAS was used for this study.

5.3.2 Study Site Identification

As a basis for selecting a recurrent bottleneck as a study site, the first step is to identify recurrent bottlenecks where crashes occurred. However, a crash occurring within the spatiotemporal boundary of bottleneck activation is an extremely rare event. In fact, crash data collected by police officers/witnesses can lead to subjective and erroneous understanding regarding crash occurrences. Despite the large data set, there were recurrent bottlenecks where no crashes were observed. In addition, several recurrent congestion problems were not singular bottleneck occurrences, but were the result of several compounding bottlenecks within a recurrent bottleneck impact area. Given these constraints, a systematic process was developed for study site selection. The site selection criteria were:

- At least one crash occurred within the spatiotemporal boundary of bottleneck activation;
- At least three miles from the location (the TMC with AHCI value greater than 20%) of the impact area of the nearest downstream bottleneck as measured by the AHCI contour; and,
Presence of bottleneck activations at the recurrent bottleneck selected on at least 50% of all weekdays studied.

In order to conduct this process with above criteria, recurrent bottlenecks with time span of bottleneck activation should be identified first. Therefore, daily CI contours were generated for all interstates and thereby an AHCI contour per year was created to identify recurrent bottlenecks. In all cases, weekday data were used. The recurrent bottlenecks identified based on the approach of Step (C), shown Figure 5-1, were 67, 69, 86, and 95 for 2011, 2012, 2013, and 2014, respectively.

With the crash data sorted by Type 1 and Type 2, six bottlenecks on interstates for the study period were finally selected as the study sample according to the three criteria above. The basic information for the recurrent bottleneck site is summarized in Table 5-2.

Table 5-2 Bottleneck Characteristics for Study Sites Selected in NC

<table>
<thead>
<tr>
<th>Site</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Road</td>
<td>I-540</td>
<td>I-40</td>
<td>I-77</td>
<td>I-77</td>
<td>I-77</td>
<td>I-40</td>
</tr>
<tr>
<td>Direction</td>
<td>EB</td>
<td>WB</td>
<td>EB</td>
<td>SB</td>
<td>WB</td>
<td>WB</td>
</tr>
<tr>
<td>County</td>
<td>WAKE</td>
<td>WAKE</td>
<td>MECKL.</td>
<td>MECKL.</td>
<td>MECKL.</td>
<td>DURHAM</td>
</tr>
<tr>
<td>TMC code</td>
<td>125+05079</td>
<td>1125+04965</td>
<td>125n04792</td>
<td>125-04787</td>
<td>125-04783</td>
<td>125+04868</td>
</tr>
<tr>
<td>Length (mile)</td>
<td>1.52</td>
<td>0.39</td>
<td>0.63</td>
<td>0.48</td>
<td>0.57</td>
<td>0.64</td>
</tr>
<tr>
<td>Time span of bottleneck activation</td>
<td>17:00 – 18:15</td>
<td>7:15 – 8:45</td>
<td>6:45 – 9:15</td>
<td>7:45 – 9:00</td>
<td>7:30 – 8:45</td>
<td>17:30 – 18:15</td>
</tr>
<tr>
<td>Reported frequency of bottleneck activation (1year)</td>
<td>180</td>
<td>161</td>
<td>127</td>
<td>147</td>
<td>173</td>
<td>130</td>
</tr>
<tr>
<td>Distance to downstream bottleneck (mi)</td>
<td>NA</td>
<td>8.07</td>
<td>8.32</td>
<td>4.5</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Reported number of crashes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type 1</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Type 2</td>
<td>3</td>
<td>7</td>
<td>23</td>
<td>13</td>
<td>14</td>
<td>9</td>
</tr>
</tbody>
</table>

Note: NA = no adjacent downstream bottleneck within 10 miles
As mentioned earlier, it is possible that the crash data contains errors regarding location and time of the crash. This study identified and used crashes reported which can match Type 1 congestion starting point not only in time interval \( t \) on segment \( i \) ((2,3) as shown in Figure 5-4), but also in adjacent times \((t-1 \text{ or } t+1)\) and/or segments \((i-1 \text{ or } i+1)\) (from (1,2) to (3,4) in Figure 5-4). Table 5-3 shows the frequency of crashes reported in the nearby time and space range to congestion occurrence. Only two crashes were matched to congestion starting point. This study used crashes reported and matched to congestion within one spatiotemporal cell. A total of 20 crashes reported were used as seen in Table 5-2.

<table>
<thead>
<tr>
<th>Seg. (bottleneck)</th>
<th>Time period</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5-4 Crash Location and Time Corresponding to Congestion

<table>
<thead>
<tr>
<th>Crash location</th>
<th>Same time and location</th>
<th>Same time but next location</th>
<th>Previous or after time but same location</th>
<th>Both next time and segment cell</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Type 1 crashes</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>20</td>
</tr>
</tbody>
</table>
5.3.3 Probability Distribution of Congestion Spatiotemporal Impact Index

Next, the $CSII_{c,s}$ of each congestion, $c$, occurring for each study site was calculated. Table 5-4 shows the descriptive statistics of the $CSII_{c,s}$, which include all congestion cases for Type 0, 1 and 2. Maximum values of the $CSII_{c,s}$ for the study sites were the value for Type 2. The average $CSII_{c,s}$ varied from 1.5 to 3 with the exception of study site 3. The value of $CSII_{c,s}$ for site 3 was higher than others which indicates that it is the recurrent bottleneck where has the biggest impact area among the study sites.

Table 5-4 Descriptive Statistics of Congestion Spatiotemporal Impact Index (CSII)

<table>
<thead>
<tr>
<th>Site</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average CSII (miles·hours)</td>
<td>2.96</td>
<td>1.88</td>
<td>10.52</td>
<td>1.51</td>
<td>1.74</td>
<td>1.66</td>
</tr>
<tr>
<td>Stand deviation of CSII (miles·hours)</td>
<td>1.42</td>
<td>0.99</td>
<td>5.97</td>
<td>1.48</td>
<td>1.19</td>
<td>1.45</td>
</tr>
<tr>
<td>Min. CSII (miles·hours)</td>
<td>0.23</td>
<td>0.19</td>
<td>0.32</td>
<td>0.12</td>
<td>0.14</td>
<td>0.06</td>
</tr>
<tr>
<td>Max. CSII (miles·hours)</td>
<td>6.79</td>
<td>7.3</td>
<td>26.54</td>
<td>6.62</td>
<td>7.28</td>
<td>7.31</td>
</tr>
</tbody>
</table>

As mentioned earlier, it is hard to capture the historical recurrent congestion extent using Type 1 and Type 2 with significant samples of $CSII_{c,s}$ due to the nature of crash occurrence. In addition, it is likely that some crashes are not reported in the crash data resulting in the statistics of $CSII_{c,0}$ to include several Type 1 congestion occurrences. Therefore, this study developed normalized distributions of $CSII_{c,s}$ across all study sites in terms of Type 0, 1, and 2. The Kolmogorov-Statistical (K-S) test and Anderson-Darling (A-D) tests, which are commonly used to test goodness of fit, were conducted to determine the probability distributions that reflect the stochastic characteristics of normalized congestion severity index (Jia et al. 2010). The Weibull, gamma, and normal distributions were
considered. For each of these, a K-S and A-D statistics were calculated. Two normalized distributions were generated except for Type 1 because of its limited sample size, as mentioned above. Table 5-5 shows that the Weibull distribution yields the lowest K-S and A-D statistic values in terms of both congestion types.

Table 5-5 Computed Statistics Values by Distribution

<table>
<thead>
<tr>
<th>Tested distribution</th>
<th>Type 0 K-S statistic</th>
<th>Type 2 K-S statistic</th>
<th>Type 0 A-D statistic</th>
<th>Type 2 A-D statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weibull</td>
<td>0.028</td>
<td>0.097</td>
<td>1.26</td>
<td>0.623</td>
</tr>
<tr>
<td>Gamma</td>
<td>0.053</td>
<td>0.118</td>
<td>7.52</td>
<td>1.321</td>
</tr>
<tr>
<td>Normal</td>
<td>0.084</td>
<td>0.137</td>
<td>12.091</td>
<td>1.880</td>
</tr>
</tbody>
</table>

Figure 5-5 Normalized CSII distribution of recurrent congestion (Type 0) with Type 1

Figure 5-5 illustrates the normalized $CSII_{c,0}$ with $CSII_{c,1}$ values for all study sites. The 20 values of $CSII_{c,1}$ were superimposed onto the distribution of $CSII_{c,0}$. This figure gives the minimum percentile $CSII_{c,1}$, $F(min\,CSII_{c,1})$, compared to the percentile $CSII_{c,0}$ derived from the distribution. It shows that $F(CSII_{c,1})$ values were greater than 0.9 and $F(min\,CSII_{c,1})$ was
0.863. This can be simply converted to $\lambda$ which is derived from Equation 5-5 and Equation 5-6 for each study site in the AHCI contour to ascertain the thresholds for recurrent congestion.

5.3.4 Capturing and Quantifying Recurrent Congestion Impact

Given that the normalized CSII distributions present a representative value of $F(\min CSII_{c,i})$ among all study sites, a recurrent congestion spatiotemporal extent is ascertained using $\lambda$ for each bottleneck. Part (a) and (b) of Figure 5-6 illustrates the recurrent congestion impact identified for study sites 1 and 2. The bolded value in the AHCI contour was the maximum value of $AHCI(i,t)$, 89% and 86% for sites 1 and 2, respectively. Using these values, the value of $\lambda$ for a recurrent bottleneck was estimated as 12.2% for site 1 and 11.7% for site 2.
In Step (H) in Figure 5-1, those areas were then calculated to quantify each impact. The quantification results of recurrent congestion impact for the six sites are summarized in Table 5-6. This informs the site 3 was the worst bottleneck, while the RBSII values of others were comparatively similar.
Table 5-6 Quantification Result of Recurrent Congestion Impact

<table>
<thead>
<tr>
<th>Site</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum value of $AHCI(i,t)$ (%)</td>
<td>89</td>
<td>86</td>
<td>92</td>
<td>73</td>
<td>82</td>
<td>56</td>
</tr>
<tr>
<td>$\lambda$ (%)</td>
<td>12.2</td>
<td>11.7</td>
<td>12.5</td>
<td>10.0</td>
<td>11.2</td>
<td>7.6</td>
</tr>
<tr>
<td>$RBSII$ (miles-hours per activation)</td>
<td>2.94</td>
<td>2.78</td>
<td>12.92</td>
<td>1.39</td>
<td>1.47</td>
<td>1.21</td>
</tr>
</tbody>
</table>

5.4 Applications

The proposed RBSII can be directly applied for ranking recurrent freeway bottlenecks. This facilitates monthly and/or annual analysis on a large scale network (i.e. nation or statewide). In addition, it allows for identification of degraded or improved recurrent bottleneck impact. For instance, Figure 5-7 depicts the degraded recurrent bottleneck impact for a typical weekday using the RBSII. Part (a) of Figure 5-7 shows the AHCI contour for site 2 in 2012; while the AHCI contour from 2014 is shown in (b). The RBSII in 2014 is greater than in 2012 by approximately two mile-hours per activation which indicates that traffic factors degraded the recurrent bottleneck impact. In addition to the RBSII comparison, Figure 5-7 also shows the temporal increase in recurrent congestion duration (30 minutes increase) and AHCI values in 2014. The RBSII can be also applied to calculate the maximum queue length and the duration of congestion for each recurrent bottleneck. For example, the maximum length of queue in the AHCM of 2012 was 2.4 miles at the recurrent bottleneck, while it was 3.8 miles in 2014.
5.5 Conclusions

This chapter introduced a dynamic data-driven approach for quantifying recurrent congestion impacts based on historically spatiotemporally congested information. The methodology presented here quantifies historical recurrent congestion impact. Unlike previous studies that quantified bottleneck impact with average maximum queue length and duration, the proposed method uses the segment length, duration of the congestion, and frequency of occurrences by each spatiotemporal cell that reflects historical congested impacts well identified by a recurrent congestion definition. This definition based on an average congestion history using probe-reported speeds. This research provided a stochastic
normalized spatiotemporal congested impact distribution for distinguishing recurrent and collision-induced congestion impacts. The proposed methodology can support both road infrastructure decision makers and congestion managers in their efforts to implement mobility and reliability treatments that are precisely targeted and effective by providing critical information about which bottlenecks result in the worst mobility and reliability.

Future research is required to enhance quantification of bottleneck impacts. Congested speed for each congestion event should be considered as a variable for intensity in quantifying bottleneck impact. This was not considered for this study because this study was focused on developing an easily implementable methodology that quantifies spatiotemporal congested impacts quickly. However, considering such intensity variable can lead to more accurate quantification of congested impacts in the world. This study was based on the link-based speed data that provides uniform traffic performance information spatially. Thus, this may cause some errors in quantifying congested impacts. In addition, there is a need to scientifically select an appropriate time resolution. Although this study employed normalized time resolution, an appropriate aggregated time resolution needs to be found to reduce errors temporally.
CHAPTER 6 CONCLUSIONS AND RECOMMENDATIONS

This thesis developed data-driven methodologies for identifying recurrent bottlenecks and for quantifying their associated impacts using link-based speed data. The models enable decision makers to invest in current facilities for capacity improvements and deployment of ITS technologies with appropriate ATM strategies to alleviate congestion effectively and efficiently. As fundamental information, spatiotemporal characteristics of recurrent bottlenecks were first identified using historical congestion information. Using fundamental information, this thesis developed the scheme for identifying recurrent bottlenecks spatiotemporally. Based on the recurrent bottleneck identification, two applications were introduced. The first was a collision classification methodology that can classify operational conditions when and where crashes occur. The second was an approach for quantifying recurrent congestion impacts. Those approaches were applied to a case study. A summary of outcomes in each of these key areas is presented in the following sections.

6.1 Data-Driven Spatiotemporal “Recurrent” Bottleneck Identification

In the literature synthesis, previous studies identified bottlenecks as congestion occurrences but did not provide consistent methods for identifying recurrent bottleneck conditions specifically. An active bottleneck is a physical point on the network upstream of which one finds a queue and downstream of which one finds freely flowing traffic.

The reviewed bottleneck identification schemes have used ambiguous concepts for defining breakdown, bottlenecks, and congestion. However, these terms should be clearly defined and adapted. To invest and deploy appropriate operational strategies to mitigate severe recurrent congestion area to mitigate its associated impacts, temporal as well as spatial information on bottlenecks need to be identified.
As such, an easily implementable approach for identifying recurrent bottlenecks was introduced using spatial and temporal concepts to establish a definition and framework for identifying a recurrent bottleneck based on expected congestion history. Three spatial and two temporal characteristics were identified in the case study sites. Based on the characteristics of recurrent bottlenecks, users and analysts can modify the cut-off thresholds for their purpose of their studies. The proposed methodology used link-based speed data to create a spatiotemporal traffic state matrix that more accurately and directly facilitates the identification of bottlenecks.

A summary of key findings and results from the proposed identification scheme is presented next. With the selected thresholds for recurrent congestion – including a threshold for congested speed (varying <75% to <90% of Free Flow Speed or FFS on a link) and a threshold for bottleneck recurrence (>50% of weekdays), a total of 95 bottlenecks were identified on all statewide highways in North Carolina using 2014 probe data. As expected, most bottlenecks were located in the highly urbanized areas in the state, namely Raleigh-Durham and Charlotte. The number of bottlenecks identified was sensitive to different value of the threshold parameters, especially the average historic congestion index value. In addition, an average of two bottleneck activations was within a recurrent bottleneck impact area. This finding supports the assertion and the proposed quantification methodology indicating that the ranking of bottlenecks is complex and should be based on their impact rather than the bottleneck location itself. Finally, more than 60% of all bottlenecks had a temporal span that varied between 2 and 3 hours, which indicates that bottlenecks located in NC are activated within comparatively similar congestion duration.
6.2 A Novel Collision Classification Methodology based on Temporal Link Speed Data and Congestion Thresholds

This thesis introduced a link-based method to classify operational conditions at locations where crashes occur. In contrast to traditional secondary incident identification schemes, the proposed method can identify whether the collision occurred in a normally non-congested area or collision occurred in a recurrent or non-recurrent congestion area. This approach informs how crashes can be managed and which resources should be allocated for incident response programs. Crashes and speed data are associated with a specific segment onto a GIS link file. It should be noted that the crash data used for this thesis have more than two directional values, while link speed data have only two directional values in segments. Accordingly, only crash data in the same direction as links were applied to the case study. This involved the analysis of 500 crash records on a 170-mi section of I-40 in North Carolina from Exit 259 to Exit 420. The data employed focused on Tuesday, Wednesday, and Thursday in April, May, September, and October of 2012 and 2013 (a total of 105 days).

In the case study, the collision classification methodology classified 500 crashes in terms of the operational conditions under which each crash occurred (case 1: in normally uncongested areas, case 2: in non-recurrent congested areas, and case 3: in recurrent congested areas). These proportions were estimated at 84%, 12%, and 4%, respectively. This information can allow traffic safety analysts to focus on collisions not in congested areas to reduce collision frequency effectively and, where appropriate, focus on incident response programs that should be deployed by identifying hot-spot locations for each crash classification type. The advantage of the proposed collision classification method is that it requires no identification of precipitating incident or definition of precipitating
incident’s impact area. As the result of the comparative analysis, 37% of collisions in non-
recurrent congested area were identified with no primary incidents or crashes from the
incident data.

In this research, crashes in non-recurrent congestion are assumed to be a surrogate for
secondary crashes. The proportion of secondary crashes identified in the case study was 12%.
This finding was consistent with results from previous classification studies, where the
secondary crash percentages ranged from 2.2% to 15.5%. Despite the fact that several
secondary crashes initiated by a primary incident occur in recurrent congestion, there was no
consideration to classify such case as a separate condition. The percentage of “secondary”
crashes likely to increase if future research includes such a case.

6.3 Recurrent Congestion Impact based on Spatiotemporally Historic
Congested Information

The research also developed an approach for quantifying recurrent congestion
impacts. Several novel components were introduced to generate the approach. To separate
recurrent congestion impact from collision-induced congestion, spatiotemporal congestion
impacts were calculated by congestion type, separately. In contrast to state of practice
studies, the proposed method uses the segment length, frequency of occurrences, and
duration of the congestion in each spatiotemporal cell, reflecting the historical congested
impacts for a recurrent bottleneck definition. A normalized spatial and temporal congestion
impact distribution was introduced to distinguish recurrent and collision-induced congestion
which occur within recurrent congestion impact areas. By segregating recurrent congestion
impact from collision-induced congestion impact, historical recurrent congestion impacts can
be calculated. The proposed methods facilitate a transportation agency long-term monitoring of the impacts of recurrent bottlenecks.

The summary of key findings and results from the quantification methodology includes:

- Most isolated recurrent bottleneck segments (i.e. those that exhibited no intervention from other nearby bottleneck activations) had no crashes reported over the course of one year.
- All spatiotemporal congestion impacts due to crashes occurring at a recurrent bottleneck were greater than the impact of the 86 percentile of the strictly recurrent congestion impacts for one year.
- The Weibull distribution was found to be the most suitable distribution with the lowest K-S and A-D statistic values in terms of generating the distribution of congestion types which occur within a recurrent bottleneck impact area.
- The quantification results of recurrent congestion impact for the six test sites varied from 1 to 3 mile-hours per activation except for the third study site. That site had 13 mil-hours of impact per activation and was identified as the worst bottleneck among the study sites. This parameter can be directly applied for ranking freeway bottlenecks on a large scale network.
- Approximately 70 crashes were identified as occurring within an active bottleneck impact area. Such a crash resulted in average additional congestion (of about 67%) on top of the recurrent congestion impact. This effect needs to be identified and estimated for monitoring mobility of road systems in a future research.
The recurrent bottleneck spatiotemporal impacts for the six sites degraded over time (for example the data indicated an increase of 0.67 mile-hours of impact per activation over one year for site two).

When it comes to bottleneck ranking, give that several bottlenecks can be activated within a bottleneck impact area, the proposed quantification methodology is more robust, when compared with a strict bottleneck location-based methodology.

6.4 Future Research

This research identified several gaps in developing recurrent bottleneck identification and its applications. Future studies in that line of work should focus on the following aspects of recurrent bottleneck research:

- Link-based traffic data were used for this thesis, which provide uniform traffic performance information for the entire link or the Traffic Message Channel (TMC). This study employed the starting point of TMC segments as the bottleneck location and the starting and end points of TMC’s as the beginning and ending points for the congestion. Future research is recommended to be conducted with more in-depth traffic data such as vehicle trajectory data;

- The intensity of congestion (characterized by the threshold speed) was not considered in this thesis is recommended to be considered as an additional factor when identifying congestion and quantifying bottleneck impact;

- Although the approaches presented here were based on historic congestion information and are data-driven, several factors that might affect congestion such as road characteristics (curves, lane drops, etc.), environmental
characteristics, etc. might be necessary to generate more elaborate historic congestion contour;

- This research considered only crash events but did not consider non-crash events such as disabled vehicles, weather, special event, etc. The author expects that those factors also affect congestion impact spatiotemporally. Therefore, future research should explore other incidents that cause congestion in recurrent bottleneck impact areas and model the approach for quantifying recurrent bottleneck impact;

- Data aggregated at the 15 minutes level were used in this research. However, using high resolution traffic data would be more valuable to apply the approaches presented here, in order to improve the understanding and identification of recurrent bottleneck and its impacts;

- Finally, only annual case studies were tested in this thesis. Thus, the findings need to be verified at other temporal scales. Therefore, a sensitivity analysis is advised to select a reasonable study period in order to apply the approaches presented.
REFERENCES


Caltrans, (2016). “How congested are the freeways in california?”


Florida Department of Transportation (2011). “Bottlenecks on florida sis.”


Kwon, J., Mauch, M., Varaiya, P., (2006). “Components of congestion delay from incidents, special events, lane closures, weather, potential ramp metering gain, and excess
demand.” Transportation Research Record: Journal of the Transportation Research Board 1959, pp. 84-91.


NCDOT, “Traffic engineering accident analysis system (teaas)”. NCDOT, Raleigh.


A. Collision Contour and Crash Classification Tool (4CT)

This thesis used a tool titled “Congestion Contour and Collision Classification Tool (4CTool)” that enables users to generate daily and historical link-based congestion contour and classify collisions by types of congestion. The tool has a screen for user input. Users will enter all threshold information and input data for the running the tool correctly.

As a first step, users should select an approach to distinguish between congested and un-congested state by the specified speed during a time stamp. Three options can be selected: an alpha value, a specified speed threshold, and alpha value by posted speed limit. In case of using an alpha value, users can insert the value from 0 to 1. The default value for the alpha value is 0.8 for the tool. With respect to using a specified speed threshold, users can insert a specified speed threshold (miles per hour). The default value for the speed threshold is 45 mph in the tool. Finally, if users select the dynamic alpha value, alpha values should inserted by speed limit varying from 55 mph to 75 mph.
Users click on “Calculate CI” button once all the thresholds are entered in the alpha value setup. The tool requests users select the posted speed limit file of each segment (file format: csv file). As seen in a screenshot of the posted speed limit file, it includes TMC code, route class, posted speed limit, and order information. The mandatory information for this file should contain TMC code in the first column and the speed limit data in the third column.
As a next step, the tool requests users select a folder including link speed data for a specified study period. The speed data can be downloaded directly through ritis.org and should include TMC code, time stamp, speed, reference speed, and travel time information. The author recommends using monthly data since it makes run time shorter than using annual data.
After selecting the folder including all speed data interested, the tool runs automatically and provides calculating process in the tool.
With respect to calculating AHCI and create AHCI contour map, users can generate by clicking ‘Calculate AHCI’ on the tool. The tool requests same process with ‘Calculate CI’. After the calculating AHCI, users can find the output of AHCI calculation in the same folder where the tool is.

Finally, users can classify crashes by types of congestion using the tool. In order to classify crashes, users must extract outputs for both ‘Calculate CI’ and ‘Calculate AHCI’. In addition, the ordered AHCI contour map is required for classifying crashes. TMCs in the AHCI output file calculated from ‘Calculate AHCI’ will be not in order. To impose order in each TMC, the TMC identification file downloaded automatically when downloading speed file via ritis.org can be used. TMCs in the identification file are in order on each road. By using this order, users can put number in the AHCI file.

Before clicking ‘Classify crash’, three thresholds are needed to be determined. Beta is the threshold of spatial difference with AHCI values of between the segment identified bottlenecks and its downstream segment. Gamma value in this tool is the threshold clearly
defined recurrent congestion, while eta value is the threshold clearly defined non-recurrent congestion. This project proposed using the following defaults for those thresholds:

- Beta value (>1): 2
- Gamma value (0-1): 0.6
- Eta value (0-1): 0.2

As a next step, the tool request users insert a TEAAS crash data file. The format of the crash data file is csv and the file includes attributes shown in the screenshot of TEAAS crash data.

<table>
<thead>
<tr>
<th>CRSH_ID</th>
<th>CRASH_DATE</th>
<th>CYEAR</th>
<th>WKDAY</th>
<th>CTIME</th>
<th>ON_ROAD</th>
<th>SPDLMT</th>
<th>Tmc</th>
<th>Miles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.04E+08</td>
<td>6/30/2014</td>
<td>2014</td>
<td>2</td>
<td>3:44:00 PM</td>
<td>85</td>
<td>55</td>
<td>125+04621</td>
<td>2.02</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>6/14/2014</td>
<td>2014</td>
<td>7</td>
<td>4:04:00 PM</td>
<td>85</td>
<td>65</td>
<td>125P04650</td>
<td>0.79</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>1/5/2014</td>
<td>2014</td>
<td>1</td>
<td>7:44:00 PM</td>
<td>85</td>
<td>70</td>
<td>125+05024</td>
<td>2.84</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>1/6/2014</td>
<td>2014</td>
<td>2</td>
<td>7:29:00 AM</td>
<td>77</td>
<td>55</td>
<td>125+04788</td>
<td>0.68</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>1/24/2014</td>
<td>2014</td>
<td>6</td>
<td>2:47:00 PM</td>
<td>85</td>
<td>55</td>
<td>125P04650</td>
<td>2.15</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>1/21/2014</td>
<td>2014</td>
<td>3</td>
<td>3:40:00 AM</td>
<td>95</td>
<td>65</td>
<td>125P05256</td>
<td>0.63</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>1/2/2014</td>
<td>2014</td>
<td>5</td>
<td>6:39:00 PM</td>
<td>77</td>
<td>55</td>
<td>125P04779</td>
<td>0.56</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>1/27/2014</td>
<td>2014</td>
<td>2</td>
<td>11:01:00 AM</td>
<td>85</td>
<td>60</td>
<td>125P04640</td>
<td>0.7</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>1/13/2014</td>
<td>2014</td>
<td>2</td>
<td>6:34:00 PM</td>
<td>85</td>
<td>70</td>
<td>125P05026</td>
<td>0.61</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>12/2/2014</td>
<td>2014</td>
<td>4</td>
<td>2:42:00 PM</td>
<td>95</td>
<td>70</td>
<td>125+05265</td>
<td>3.75</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>1/9/2014</td>
<td>2014</td>
<td>5</td>
<td>9:09:00 AM</td>
<td>77</td>
<td>65</td>
<td>125P04792</td>
<td>0.71</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>2/7/2014</td>
<td>2014</td>
<td>6</td>
<td>5:30:00 PM</td>
<td>85</td>
<td>60</td>
<td>125P04637</td>
<td>0.57</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>1/31/2014</td>
<td>2014</td>
<td>6</td>
<td>8:45:00 PM</td>
<td>77</td>
<td>55</td>
<td>125P04783</td>
<td>0.26</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>1/16/2014</td>
<td>2014</td>
<td>5</td>
<td>6:55:00 AM</td>
<td>85</td>
<td>65</td>
<td>125+04652</td>
<td>2.09</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>2/7/2014</td>
<td>2014</td>
<td>6</td>
<td>8:12:00 PM</td>
<td>77</td>
<td>70</td>
<td>125+05361</td>
<td>4.44</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>3/19/2014</td>
<td>2014</td>
<td>4</td>
<td>6:54:00 AM</td>
<td>77</td>
<td>55</td>
<td>125+04784</td>
<td>0.85</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>3/6/2014</td>
<td>2014</td>
<td>5</td>
<td>8:08:00 PM</td>
<td>85</td>
<td>60</td>
<td>125P04640</td>
<td>0.7</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>2/5/2014</td>
<td>2014</td>
<td>4</td>
<td>2:12:00 PM</td>
<td>77</td>
<td>65</td>
<td>125+04793</td>
<td>1.76</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>3/3/2014</td>
<td>2014</td>
<td>2</td>
<td>3:23:00 PM</td>
<td>85</td>
<td>60</td>
<td>125+04587</td>
<td>5.25</td>
</tr>
<tr>
<td>1.04E+08</td>
<td>2/25/2014</td>
<td>2014</td>
<td>3</td>
<td>11:01:00 AM</td>
<td>95</td>
<td>70</td>
<td>110+05549</td>
<td>3.71</td>
</tr>
</tbody>
</table>

After running ‘Classify crashes’, users can find the output in the same folder where the tool is as well. The title name is similar with ‘ResultFile1454151515.csv’. In the result file, CRASH TYPE should be created and the types are classified into four types: 1 = crashes
in not congested condition, 2 = crashes in non-recurrent congestion, 3 = crash in recurrent congestion, and 99 = no available. 99 values can be generated when the crash date occurred on weekend but an AHCI contour was created using only weekday information.
B. Collision Contour and Crash Classification Tool (4CT) – Java code

1. AHCI calculator

```java
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class AHCICalculator {
    private static String[] crashRecord;
    private static String line = "";
    private static String cvsSplitBy = ",";
    private static String commaDelimt = ",";
    private static float alpha = (float) 0.8;
    private static float defSpdLmt = (float) 65.0;
    private static JPanel parentPanel;
    private static Thread thread;
    private static ArrayList<String> tmcList = new ArrayList<String>();

    public static void createAHCI(JPanel mainPanel) throws FileNotFoundException, IOException {
        alpha = MainPanel.getAlphaValue();
        parentPanel = mainPanel;
        thread = new Thread() {
            public void run() {
                try {
                    readTMCSpeedLimit();
                    readCSV();
                    JOptionPane.showMessageDialog(parentPanel, "AHCI_Output.csv created. Process complete");
                } catch (IOException v) {
                    System.out.println(v);
                }
            }
        };
    }

    private void readTMCSpeedLimit() {
        // Read TMCSpeedLimit from file
    }

    private void readCSV() {
        // Read CSV file
    }

    private void setTableModel(ArrayList<String> tmcList) {
        // Set table model
    }
}
```
thread.start();

private static void readTMCSppedLimit() throws FileNotFoundException, IOException {
    JOptionPane.showMessageDialog(parentPanel, "Select the speedlimit TMC file");
    JFileChooser filechooser = new JFileChooser(System.getProperty("user.dir"));
    filechooser.showOpenDialog(parentPanel);
    File selectedFile = filechooser.getSelectedFile();
    String csvFile = selectedFile.getAbsolutePath();
    BufferedReader br = null;
    try {
        if (br == null) {
            try {
                br = new BufferedReader(new FileReader(csvFile));
                line = br.readLine();
                while ((line = br.readLine()) != null) {
                    // use comma as separator
                    String[] crashRecord = line.split(cvsSplitBy);
                    String tmc = crashRecord[0];
                    String spdLimit = crashRecord[2];
                    CrashDataMap.addTMCMap(tmc, Float.parseFloat(spdLimit));
                }
            } catch (FileNotFoundException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            } catch (NumberFormatException e) {
                System.out.println(e);
            }
            finally {
                if (br != null) {
                    try {
                        br.close();
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                }
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } catch (NumberFormatException e) {
            System.out.println(e);
        }
    } finally {
        if (br != null) {
            try {
                br.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
public static void readCSV()
{
    HashMap<String, Float> tmcMap = CrashDataMap.getTMCMAP();
    ArrayList<String> tmcInOutput = new ArrayList<String>();
    ArrayList<String> tmcValues = new ArrayList<String>(tmcMap.keySet());
    int[][] CIArray = new int[tmcValues.size() * 2][100];
    int[][] CICounter = new int[tmcValues.size() * 2][100];

    JOptionPane.showMessageDialog(parentPanel, "Select the crash incident directory containing all speed data");
    JFileChooser filechooser = new JFileChooser(System.getProperty("user.dir"));
    filechooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
    filechooser.setAcceptAllFileFilterUsed(false);
    filechooser.showOpenDialog(parentPanel);
    File selectedFile = filechooser.getSelectedFile();
    String directoryPath = selectedFile.getAbsolutePath();
    File[] filesInDirectory = new File(directoryPath).listFiles();
    for (File f : filesInDirectory)
    {
        String filePath = f.getAbsolutePath();
        String fileExtension = filePath.substring(filePath.lastIndexOf("") + 1, filePath.length());
        String fileName = filePath.substring(filePath.lastIndexOf("\") + 1, filePath.length());
        System.out.println(fileName);
        if ("csv".equals(fileExtension))
        {
            MainPanel.setMsg("CSV file calculating -> " + filePath);
        }
        String csvFile = filePath.trim();
        BufferedReader br = null;
        try
        {
            br = new BufferedReader(new FileReader(csvFile));
            line = br.readLine();
            while ((line = br.readLine()) != null)
            {
                // use comma as separator
                crashRecord = line.split(",");
                if (crashRecord[0].length() != 0 && crashRecord[1].length() != 0 && crashRecord[2].length() != 0)
Date currentDate =
new SimpleDateFormat("yyyy-MM-dd
hh:mm:ss"),parse(crashRecord[1]);
String tmc = crashRecord[0];
if (!tmc.equals("125+04595"))
{
    continue;
}
if (tmcValues.indexOf(tmc) == -1)
{
    CrashDataMap.addTMCMap(tmc,
defSpdLmt);
tmcValues.add(tmc);
}
tmcInOutput.add(tmc);
int day = currentDate.getDay();
int hours = currentDate.getHours();
int minutes = currentDate.getMinutes();
int minIndex = minutes / 15;
minIndex = (hours * 4) + minIndex;
float speed =
Float.parseFloat(crashRecord[2]);
if (day != 6 && day != 7 &&
tmcValues.indexOf(tmc) != -1)
{
    Float speedLmt = tmcMap.get(tmc);
    if (speedLmt != null && speed /
speedLmt < alpha)
    {
        CIArray[tmcValues.indexOf(tmc)][minIndex] += 1;
    }
    CICounter[tmcValues.indexOf(tmc)][minIndex] +=
1;
}
else
{
    System.out.println(\"Problem\);
}
}
}
catch (ParseException e)
{
// TODO Auto-generated catch block
e.printStackTrace();
}
catch (FileNotFoundException e)
{
 e.printStackTrace();
}
catch (IOException e) {
    e.printStackTrace();
}
catch (NumberFormatException e) {
    System.out.println(e);
}
finally {
    if (br != null) {
        try {
            br.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
System.gc();

CsvFileWriter writer = new CsvFileWriter("AHCI_Output.csv");
for (int i = 0; i < tmcValues.size(); i++) {
    if (!tmcInOutput.contains(tmcValues.get(i))) {
        continue;
    }
    StringBuffer buffer = new StringBuffer();
    buffer.append(Integer.toString(i));
    buffer.append(commaDelimt);
    buffer.append(tmcValues.get(i));
    for (int j = 0; j < 96; j++) {
        buffer.append(commaDelimt);
        if (CIArray[i][j] != 0 && CICounter[i][j] != 0) {
            float val = (float) CIArray[i][j] / (float) CICounter[i][j];
            if (val == 1) {
                System.out.println("test");
            }
            buffer.append(Float.toString(val));
        } else {
            buffer.append(Float.toString(val));
        }
    }
    writer.append(buffer.toString());
}
2. Classify crash data
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.LinkedHashMap;
import javax.print.attribute.standard.JobMessageFromOperator;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
public class ClassifyCrashData {
    private static LinkedHashMap<String, double[]> ahciVals = new LinkedHashMap<String, double[]>();
    private static String csvSplitBy = "\,";
    private static String semiColon = ";";
    private static String[] crashRecord;
    private static String line = ""
    private static double[][] ahciArray;
    private static ArrayList<String> tmcList = new ArrayList<String>();
    private static JPanel mainPanel;
    private static String ahciFolder = "";
private static Thread thread;
public static void classifyCrash(JPanel mainpanel)
{
    mainPanel = mainpanel;
    thread = new Thread()
    {
        public void run()
        {
            readAHCIFile();
            createAHCIArray();
            classifyData();
        }
    };
    thread.start();
}

private static void createAHCIArray()
{
    int size = ahciVals.size();
    ahciArray = new double[size][97];
    int i = 0;
    for (String key : ahciVals.keySet())
    {
        // ...
        double[] ds = ahciVals.get(key);
        for (int j = 1; j < 97; j++)
        {
            ahciArray[i][j - 1] = ds[j];
        }
        i++;
    }
}

private static void readAHCIFile()
{
    JOptionPane.showMessageDialog(mainPanel, "Select the ordered AHCI file");
    JFileChooser filechooser = new JFileChooser(System.getProperty("user.dir"));
    filechooser.showOpenDialog(mainPanel);
    File selectedFile = filechooser.getSelectedFile();
    String csvFile = selectedFile.getAbsolutePath();
    String tmc = ";
    BufferedReader br = null;
    try
    {
        br = new BufferedReader(new FileReader(csvFile));
        line = br.readLine();
        while ((line = br.readLine()) != null)
        {
            try
            {

120
double[] ahci = new double[97];
// use comma as separator
String[] crashRecord = line.split(cvsSplitBy);
tmc = crashRecord[1];
ahci[0] = Double.parseDouble(crashRecord[6]);
for (int i = 0; i < 96; i++)
{
    ahci[i + 1] = Double.parseDouble(crashRecord[i + 8]);
}
tmcList.add(tmc);
ahci(vals).put(tmc, ahci);
}
catch (NumberFormatException e)
{
    System.out.println("Number format exception");
}
}
}
catch (FileNotFoundException e)
{
    e.printStackTrace();
}
catch (IOException e)
{
    e.printStackTrace();
}
catch (NumberFormatException e)
{
    e.printStackTrace();
}
finally
{
    if (br != null)
    {
        try
        {
            br.close();
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
    }
}
private static void classifyData()
{
    SimpleDateFormat df = new SimpleDateFormat("MM/dd/yyyy hh:mm:ss aa");
    JOptionPane.showMessageDialog(mainPanel, "Select the TEAS crashData File");
}
JFileChooser filechooser = new JFileChooser(System.getProperty("user.dir"));
filechooser.showOpenDialog(mainPanel);
File selectedFile = filechooser.getSelectedFile();
String csvFile = selectedFile.getAbsolutePath();

JOptionPane.showMessageDialog(mainPanel,
"Select the crash incident directory containing all speed data");
filechooser = new JFileChooser(System.getProperty("user.dir"));
filechooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
filechooser.setAcceptAllFileFilterUsed(false);
filechooser.showOpenDialog(mainPanel);
selectedFile = filechooser.getSelectedFile();
ahciFolder = selectedFile.getAbsolutePath();

String tmc = "";
BufferedReader br = null;
int counter = 0;

try
{
    br = new BufferedReader(new FileReader(csvFile));
    line = br.readLine();
    line = line + semiColon + "Type";
    CsvFileWriter.setFileHeader(line);
    CsvFileWriter writer = new CsvFileWriter("ResultFile.csv");
    while ((line = br.readLine()) != null)
    {
        try
        {
            // use comma as separator
            crashRecord = line.split(";");
            tmc = crashRecord[89];
            int type = -1;
            float spdLimit = Float.parseFloat(crashRecord[68]);
            String cdate = crashRecord[19];
            String ctime = crashRecord[23];
            Date crashTime = df.parse(cdate + " " + ctime);
            if (crashTime.getYear() == 113)
            {
                float speed = getSpeed(tmc, crashTime);
                type = getCrashType(tmc, crashTime, speed, spdLimit, counter);
            }
            line = line + semiColon + type;
            writer.writeCsvFile(line);
            MainPanel.setMsg("Total records classified = " + counter);
            counter++;
        }
        catch (NumberFormatException | ParseException e)
        {
        }
    }
}
catch (NumberFormatException | ParseException e)
private static int getCrashType(String tmc, Date crashTime, float speed, float spdLimit, int counter)
{
    if (speed / spdLimit >= 0.8)
    {
        return 1;
    }
    int hours = crashTime.getHours();
    int minutes = crashTime.getMinutes();
    int minIndex = minutes / 15;
    minIndex = (hours * 4) + minIndex;
    minIndex++;
    double[] ds = ahciVals.get(tmc);
    if (ds != null)
    {

if (ds[minIndex] >= 0.6)
{
    return 3;
}
else if (ds[minIndex] < 0.2)
{
    return 2;
}
else
{
    int i = tmcList.indexOf(tmc);
    if (i > 1)
    {
        if (isRecurringBottleneckArea(ds, minIndex, i))
        {
            return 3;
        }
        else
        {
            int curr = minIndex;
            int prev = curr - 1;
            while (prev != 0 && ds[curr] < ds[prev])
            {
                curr--;
                prev--;
            }
            if (isRecurringBottleneckArea(ds, curr, prev))
            {
                return 3;
            }
            else
            {
                return 2;
            }
        }
    }
}
return 0;

private static boolean isRecurringBottleneckArea(double[] ds, int minIndex, int i)
{
    double[] y1Tmc = ahciVals.get(tmcList.get(i - 1));
    double[] y2Tmc = ahciVals.get(tmcList.get(i - 2));
    double y1 = ds[minIndex] - 2 * y1Tmc[minIndex];
    double y2 = ds[minIndex] - 2 * y2Tmc[minIndex];
    if (ds[minIndex] >= 0.5 && (y1 > 0 || y2 > 0))
    {
        return true;
    }
}
return false;
}
private static float getSpeed(String tmc, Date crashTime)
{
    float speed = 0f;
    String monthName = Utils.getMonthName(crashTime.getMonth());
    String csvLine = "";
    String[] crashRec;
    String directoryPath = ahciFolder;
    File[] filesInDirectory = new File(directoryPath).listFiles();
    for (File f : filesInDirectory)
    {
        String fileName = f.getName();
        String fileExtenstion =
            fileName.substring(fileName.lastIndexOf("."), fileName.length());
        if (!"csv".equals(fileExtenstion))
        {
            continue;
        }
        if (!isMonthInRange(f.getAbsolutePath(), monthName))
        {
            continue;
        }
        String csvFile = f.getAbsolutePath().trim();
        BufferedReader br = null;
        try
        {
            br = new BufferedReader(new FileReader(csvFile));
            csvLine = br.readLine();
            while ((csvLine = br.readLine()) != null)
            {
                // use comma as separator
                crashRec = csvLine.split(",");
                if (crashRec[0].length() != 0 && crashRec[0].equals(tmc)
                    && crashRec[1].length() != 0 && crashRec[2].length() != 0)
                {
                    Date currentDate =
                        new SimpleDateFormat("yyyy-MM-dd hh:mm:ss").parse(crashRec[1]);
                    int diff =
                        (int)((currentDate.getTime() - crashTime.getTime())
                        / (1000 * 60));
                    if (diff < 15 && diff > -15)
                    {
                        return Float.parseFloat(crashRec[2]);
                    }
                }           
            }
        }
        finally
        {
            if (br != null)
            {
                br.close()
            }
        }
    }
}
catch (FileNotFoundException e) {
    e.printStackTrace();
}
catch (IOException e) {
    e.printStackTrace();
}
catch (NumberFormatException e) {
    System.out.println(e);
}
catch (ParseException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
finally {
    if (br != null) {
        try {
            br.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
return speed;
}
private static boolean isMonthInRange(String fileName, String monthName) {
    BufferedReader br = null;
    try {
        br = new BufferedReader(new FileReader(fileName));
        String csvLine = br.readLine();
        String[] crashRec;
        if ((csvLine = br.readLine()) != null) {
            // use comma as separator
            crashRec = csvLine.split(",");
            if (crashRec[1].length() != 0) {
                // TODO Auto-generated method stub
            }
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
    return true;
}
Date currentDate =
new SimpleDateFormat("yyyy-MM-dd
hh:mm:ss").parse(crashRec[1]);
String fileMonth =
Utils.getMonthName(currentDate.getMonth());
if (fileMonth.equals(monthName))
    { return true; }
else
    {
        return false;
    }
}
}

try
{
    br.close();
}
catch (IOException e)
    {
        e.printStackTrace();
    }
finally
{
    if (br != null)
        {
            try
            {
                br.close();
            }
catch (IOException e)
            {
                e.printStackTrace();
            }
        }
return false;
}
3. Crash classifier

```java
import java.awt.BorderLayout;
import java.awt.Component;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Toolkit;
import java.io.File;
import javax.swing.BorderFactory;
import javax.swing.BoxLayout;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.border.EtchedBorder;

public class CrashClassifier extends JFrame {
    private static CrashClassifier mainFrame;
    private static File selectedExcel;
    private JPanel basePanel;
    private JPanel mainPanel;
    private JLabel myLabel;

    public CrashClassifier() {
        this.setTitle("Crash Classifier");
        initComponents();
    }

    private void initComponents() {
        basePanel = new JPanel();
        basePanel.setLayout(new BoxLayout(basePanel, BoxLayout.Y_AXIS));
        myLabel = new JLabel();
        Font font = myLabel.getFont();
        myLabel.setText("Crash Classification Algorithm");
        Font boldFont = new Font(font.getFontName(), Font.BOLD, 20);
        myLabel.setFont(boldFont);
        mainPanel = MainPanel.getMainPanel();
        basePanel.add(myLabel);
        basePanel.add(mainPanel);
        basePanel.setBorder(BorderFactory.createEtchedBorder(EtchedBorder.RAISED));
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().add(basePanel);
        setPreferredSize(new Dimension(800, 300));
    }
}
```

128
final Toolkit toolkit = Toolkit.getDefaultToolkit();
final Dimension screenSize = toolkit.getScreenSize();
setLocation(screenSize.width / 2 - 400, screenSize.height / 2 - 150);
pack();

public static void main(String args[]) {
    showMainForm();
}
private static void showMainForm() {
    // Create mainFrame in EDT thread
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                mainFrame = new CrashClassifier();
                mainFrame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
public static void setSelectedExcelFile(File selectedFile) {
    selectedExcel = selectedFile;
}
public static File getSelectedExcelFile() {
    return selectedExcel;
}

4. Crash data
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

129
import java.util.HashMap;

public class CrashData {

    private static HashMap<Integer, HashMap<String, ArrayList<CrashInfo>>> monthlyMapping = new HashMap<Integer, HashMap<String, ArrayList<CrashInfo>>>();

    private static Date currentDate;
    private static int month;

    public static void setData(String tmc, String t_stamp, float speed, float refSpeed, float travelTime) {
        try {
            currentDate = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss").parse(t_stamp);
            month = currentDate.getMonth();
            CrashInfo info = new CrashInfo(currentDate, speed, refSpeed, travelTime);
            HashMap<String, ArrayList<CrashInfo>> tmcMap = monthlyMapping.get(month);
            if (tmcMap == null) {
                HashMap<String, ArrayList<CrashInfo>> tmcMapping = new HashMap<String, ArrayList<CrashInfo>>();
                ArrayList<CrashInfo> crashInfoList = new ArrayList<CrashInfo>();
                crashInfoList.add(info);
                tmcMapping.put(tmc, crashInfoList);
                monthlyMapping.put(month, tmcMapping);
            } else {
                ArrayList<CrashInfo> crashList = tmcMap.get(tmc);
                if (crashList == null) {
                    crashList = new ArrayList<CrashInfo>();
                }
                crashList.add(info);
                tmcMap.put(tmc, crashList);
                monthlyMapping.put(month, tmcMap);
            }
        } catch (ParseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

}
5. Crash data contour
import java.util.HashMap;
import java.util.LinkedHashMap;
public class CrashDataMap
{
    private static LinkedHashMap<String, Float> tmcMap = new LinkedHashMap<String, Float>();

    public static void addTMCMap(String tmc, float speedLimit)
    {
        Float spdLmt = tmcMap.get(tmc);
        if (spdLmt == null)
        {
            tmcMap.put(tmc, speedLimit);
        }
        else
        {
            System.out.println("Duplicate");
        }
    }

    public static HashMap<String, Float> getTMCMap()
    {
        return tmcMap;
    }
}

6. Crash Information
import java.util.Date;
public class CrashInfo
{
    private Date crashDate;
    private float crashSpeed;
    private float referenceSpeed;
    private float travelTimeMin;
    public CrashInfo(Date date, float speed, float refSpeed, float travelTime)
    {
        crashDate = date;
        crashSpeed = speed;
        referenceSpeed = refSpeed;
        travelTimeMin = travelTime;
    }
}

7. Crash file writer
import java.io.FileWriter;
import java.io.IOException;
public class CsvFileWriter
{
// Delimiter used in CSV file
private static final String COMMA_DELIMITER = "\,";
private static final String NEW_LINE_SEPARATOR = "\n";
private static FileWriter fileWriter = null;

// CSV file header
private static String FILE_HEADER = "FID,TMC,12:00 AM,12:15 AM,12:30 AM,12:45 AM,1:00 AM,1:15 AM,1:30 AM,1:45 AM,2:00 AM,2:15 AM,2:30 AM,2:45 AM,3:00 AM,3:15 AM,3:30 AM,3:45 AM,4:00 AM,4:15 AM,4:30 AM,4:45 AM,5:00 AM,5:15 AM,5:30 AM,5:45 AM,6:00 AM,6:15 AM,6:30 AM,6:45 AM,7:00 AM,7:15 AM,7:30 AM,7:45 AM,8:00 AM,8:15 AM,8:30 AM,8:45 AM,9:00 AM,9:15 AM,9:30 AM,9:45 AM,10:00 AM,10:15 AM,10:30 AM,10:45 AM,11:00 AM,11:15 AM,11:30 AM,11:45 AM,12:00 PM,12:15 PM,12:30 PM,12:45 PM,1:00 PM,1:15 PM,1:30 PM,1:45 PM,2:00 PM,2:15 PM,2:30 PM,2:45 PM,3:00 PM,3:15 PM,3:30 PM,3:45 PM,4:00 PM,4:15 PM,4:30 PM,4:45 PM,5:00 PM,5:15 PM,5:30 PM,5:45 PM,6:00 PM,6:15 PM,6:30 PM,6:45 PM,7:00 PM,7:15 PM,7:30 PM,7:45 PM,8:00 PM,8:15 PM,8:30 PM,8:45 PM,9:00 PM,9:15 PM,9:30 PM,9:45 PM,10:00 PM,10:15 PM,10:30 PM,10:45 PM,11:00 PM,11:15 PM,11:30 PM,11:45 PM";

public CsvFileWriter(String fileName)
{
try
{
  fileWriter = new FileWriter(fileName);
  fileWriter.append(FILE_HEADER.toString());
  fileWriter.append(NEW_LINE_SEPARATOR);
}
catch (IOException e)
{
  // TODO Auto-generated catch block
  e.printStackTrace();
}

public void closeFile()
{
try
{
  fileWriter.flush();
  fileWriter.close();
}
catch (IOException e)
{
  System.out.println("Error while flushing/closing fileWriter !!!");
  e.printStackTrace();
}

public void writeCsvFile(String data)
{
try {
    fileWriter.append(data);
    fileWriter.append(NEW_LINE_SEPARATOR);
} catch (Exception e) {
    e.printStackTrace();
} finally {
}

public static void setFileHeader(String head) {
    FILE_HEADER = head;
}

8. Main panel
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import javax.swing.BorderFactory;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
public class MainPanel {
    private static JPanel mainpanel;
    private static javax.swing.JButton calculateAhci;
    private static javax.swing.JButton classifyType;
    private static javax.swing.JButton plotGraph;
    private static javax.swing.JLabel alphaValue;
    private static javax.swing.JLabel msgLabel;
    private static javax.swing.JTextField alphaField;
    public static JPanel getMainPanel() {
        if (mainpanel != null) {
            return mainpanel;
        }
        mainpanel = new JPanel();
        initComp();
        return mainpanel;
    }
    private static void initComp() {
    }
mainpanel.setPreferredSize(new Dimension(500, 200));
mainpanel.setLayout(new BoxLayout(mainpanel, BoxLayout.Y_AXIS));
mainpanel.add(getButtonPanel());
}
private static JPanel getButtonPanel()
{
    JPanel allButtonPanel = new JPanel();
    JPanel plotPanel = new JPanel();
    plotPanel.setLayout(new BoxLayout(plotPanel, BoxLayout.X_AXIS));
    plotPanel.setBorder(BorderFactory.createTitledBorder("Setup information"));
    allButtonPanel.setLayout(new BoxLayout(allButtonPanel, BoxLayout.Y_AXIS));
    JPanel selectFilePanel = new JPanel();
    selectFilePanel.setLayout(new BoxLayout(selectFilePanel, BoxLayout.X_AXIS));
    selectFilePanel.setMaximumSize(new Dimension(500, 40));
    calculateAhci = new javax.swing.JButton();
    calculateAhci.setText("Calculate AHCI");
    calculateAhci.addActionListener(new ActionListener()
    {
        @Override
        public void actionPerformed(ActionEvent e)
        {
            try
            {
                AHCICalculator.createAHCI(mainpanel);
            }
            catch (IOException e1)
            {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }
        }
    });
    selectFilePanel.add(Box.createHorizontalStrut(10));
    selectFilePanel.add(calculateAhci);
    classifyType = new javax.swing.JButton();
    classifyType.setText("Classify crash data");
    classifyType.addActionListener(new ActionListener()
    {
        @Override
        public void actionPerformed(ActionEvent e)
        {
            ClassifyCrashData.classifyCrash(mainpanel);
        }
    });
    selectFilePanel.add(Box.createHorizontalStrut(75));
    selectFilePanel.add(classifyType);
    JPanel calibValPanel = new JPanel();
    calibValPanel.setLayout(new BoxLayout(calibValPanel, BoxLayout.X_AXIS));
    calibValPanel.setMaximumSize(new Dimension(500, 25));
}
alphaValue = new javax.swing.JLabel();
alphaField = new javax.swing.JTextField();
alphaField.setMaximumSize(new Dimension(200, 40));
calibValPanel.add(Box.createHorizontalStrut(10));
calibValPanel.add(alphaValue);
calibValPanel.add(Box.createHorizontalStrut(70));
calibValPanel.add(alphaField);
JPanel SpeedPanel = new JPanel();
SpeedPanel.setLayout(new BoxLayout(SpeedPanel, BoxLayout.X_AXIS));
SpeedPanel.setMaximumSize(new Dimension(500, 25));
msgLabel = new JLabel("test");
SpeedPanel.add(Box.createHorizontalStrut(10));
SpeedPanel.add(msgLabel);
plotGraph = new JButton();
alphaValue.setText("Alpha Value (0-1)\)
alphaField.setText("0.8\)
plotGraph.setText("Plot I2D Graph");
allButtonPanel.add(Box.createVerticalStrut(10));
alIButtonPanel.add(Box.createVerticalStrut(10));
alIButtonPanel.add(calibValPanel);
alIButtonPanel.add(Box.createVerticalStrut(10));
alIButtonPanel.add(calibValPanel);
alIButtonPanel.add(Box.createVerticalStrut(10));
alIButtonPanel.add(calibValPanel);
alIButtonPanel.add(SpeedPanel);
alIButtonPanel.add(Box.createHorizontalStrut(20));
plotPanel.add(allButtonPanel);
plotPanel.add(Box.createHorizontalStrut(20));

return plotPanel;
}
public static float getAlphaValue()
{
return Float.parseFloat(alphaField.getText());
}
public static void setMsg(String msg)
{
msgLabel.setText(msg);
mainpanel.revalidate();
mainpanel.repaint();
}

9. Utils
public class Utils
{

public static int getMonthIndex(String month)
{
for (int i = 0; i < 12; i++)
{
if (month.contains(months[i]))
    {
        return i;
    }
}
return -1;

public static String getMonthName(int index)
{
    return months[index];
}