Enhanced Traffic Incident Analysis with Advanced Machine Learning Algorithms

Zhenyu Wang
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/msve_etds

Part of the Artificial Intelligence and Robotics Commons, Civil Engineering Commons, and the Transportation Commons

Recommended Citation
https://digitalcommons.odu.edu/msve_etds/59

This Dissertation is brought to you for free and open access by the Computational Modeling & Simulation Engineering at ODU Digital Commons. It has been accepted for inclusion in Computational Modeling & Simulation Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.
ENHANCED TRAFFIC INCIDENT ANALYSIS WITH ADVANCED MACHINE
LEARNING ALGORITHMS

by

Zhenyu Wang
B.S. May 2011, Beijing University of Posts and Telecommunications, China
M.E. May 2014, Beijing University of Posts and Telecommunications, China

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MODELING AND SIMULATION

OLD DOMINION UNIVERSITY
December 2020

Approved by:

Hong Yang (Director)
Yuzhong Shen (Member)
Sherif Ishak (Member)
Mecit Cetin (Member)
Kun Xie (Member)
ABSTRACT

ENHANCED TRAFFIC INCIDENT ANALYSIS WITH ADVANCED MACHINE LEARNING ALGORITHMS

Zhenyu Wang
Old Dominion University, 2020
Director: Dr. Hong Yang

Traffic incident analysis is a crucial task in traffic management centers (TMCs) that typically manage many highways with limited staff and resources. An effective automatic incident analysis approach that can report abnormal events timely and accurately will benefit TMCs in optimizing the use of limited incident response and management resources. During the past decades, significant efforts have been made by researchers towards the development of data-driven approaches for incident analysis. Nevertheless, many developed approaches have shown limited success in the field. This is largely attributed to the long detection time (i.e., waiting for overwhelmed upstream detection stations; meanwhile, downstream stations show light traffic volume) and the concerns about the costly false alarms (e.g., dispatching response teams to non-incident cases). With the advancements in advanced machine learning algorithms and emerging data sources, there are opportunities to leverage such algorithms and a variety of data to significantly improve incident analysis practices.

As such, this dissertation first aims to develop an incident detection framework based on advanced machine learning algorithms that can leverage large-scale sensor data to enhance the predictive performance. Artificial neural network (ANN) is selected as a representative artificial intelligence (AI) module to predict incident occurrence based on lane-based data or station-level average loop detector data with occupancy, speed, and flow information. The memory unit and relevant knowledge database are integrated to refine the prediction result of AI module and provide
the framework the ability to evolve and learn from historical prediction records. Compared with the benchmark approach California algorithm (CA#7), the proposed framework demonstrates its augmented prediction performance in terms of the shorter time to detection (TTD), lower false alarm rate (FAR), and higher detection rate (DR).

Secondly, we notice the existence of inaccurate labeled incident occurrence time and its impact on the incident detection framework. Therefore, we propose to utilize the unsupervised learning approach, fuzzy c-means (FCM) clustering, to relabel incident occurrence times and to further examine its impact on three different incident detection approaches (i.e., CA#7, ANN, and support vector machine (SVM)). In order to better automatically relabel three types of inaccurate mapping between reported incident occurrence times and loop detector measurements, Bayesian information criterion (BIC) values and additional restriction rules are applied. The evaluation results based on simulated incident scenarios demonstrate that the proposed relabeling strategy helps improve the performance of three traffic incident detection (TID) approaches in terms of a higher DR and a lower FAR.

Finally, we propose a data-driven analysis framework for identifying secondary incidents (SI). The proposed approach intends to leverage the untapped potential of ubiquitous probe vehicle data for SI identification. The developed framework consists of three major components: detecting the impact area of a primary incident (PI), estimating the boundary for the impact area, and identifying SIs within the boundary. The proposed framework has been tested based on probe data collected from different simulation models. The results show that the impact area induced by a PI can be well represented by the estimated boundary, especially by the genetic algorithm (GA-) and ant colony optimization (ACO-) based methods.
This dissertation is dedicated to my parents, Zhao Wang and Lanying Zhang.
ACKNOWLEDGMENTS

There are many people who have contributed to the successful completion of this dissertation. I extend many, many thanks to my committee members for their patience and hours of guidance on my research and editing of this manuscript. I would like to express my sincere gratitude to my advisor, Dr. Hong Yang, for continuously providing guidance and advice with enthusiasm and patience. His mentoring helped me to expand my knowledge in transportation fields and to develop professional research skills. I am honored to have worked for such a visionary professor.

I would also like to express my gratitude to my advisory committee members: Dr. Yuzhong Shen, Dr. Sherif Ishak, Dr. Mecit Cetin, and Dr. Kun Xie for their time, as well as for their valuable guidance and suggestions in helping me finish my dissertation. I am grateful to all the support and help from my friends, faculty, and staff in the CMSE Department at ODU. Especially, I would like to thank my friend and teammate Qingyu Ma in the SenseLane Lab for all the suggestions and support while working together on multiple different projects. In addition, I would like to acknowledge that the effort on the primary incident detection in the dissertation is financially supported by the FHWA project “Effectiveness of TMC AI Applications in Case Studies.”

Finally, I would also like to express my deepest gratitude to my parents. Without their unconditional love and support, I could never fulfill my ambitions in higher education and get this done.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>LIST OF TABLES</th>
<th>LIST OF FIGURES</th>
<th>INTRODUCTION</th>
<th>INTRODUCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1.1 BACKGROUND</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.1.1 PRIMARY INCIDENT DETECTION &amp; ESTIMATING ITS IMPACT AREA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.1.2 SECONDARY INCIDENT IDENTIFICATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.2 PROPOSED WORK AND CONTRIBUTIONS OF THE DISSERTATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.3 DISSERTATION STRUCTURE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LITERATURE REVIEW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>2.1 INCIDENT DETECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.2 INCIDENT IMPACT ESTIMATION &amp; SECONDARY INCIDENT IDENTIFICATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.3 INCIDENT PREVENTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.4 SUMMARY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>29</td>
<td>PRIMARY INCIDENT DETECTION BASED ON AI APPROACH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.1 METHODOLOGY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.2 SIMULATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.3 PERFORMANCE CRITERIA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.4 RESULTS &amp; ANALYSIS</td>
</tr>
</tbody>
</table>

Page

x

xi

1

1

3

5

6

9

10

10

18

26

28

29

29

38

39

41
3.5 CONCLUSIONS .................................................................................................................. 44

INCIDENT IDENTIFICATION WITH REDEFINED EVENT LABELING .......................... 45

4.1 PROPOSED FRAMEWORK FOR RELABELING EVENTS IN TRAINING

DATASET ................................................................................................................................. 45

4.2 SPECIFICATION OF RELABELING STRATEGY WITH FCM ........................................ 51

4.3 EXPERIMENTAL DESIGNS ......................................................................................... 53

4.4 RESULTS AND DISCUSSION ......................................................................................... 56

4.5 CONCLUSIONS .............................................................................................................. 61

SECONDARY INCIDENT IDENTIFICATION ................................................................. 62

5.1 OVERVIEW OF THE PROPOSED APPROACH ......................................................... 62

5.2 DETECTING THE IMPACT AREA ................................................................................ 63

5.3 ESTIMATING BOUNDARY OF THE IMPACT AREA ..................................................... 67

5.3.1 MULTI-STAGE LINEAR APPROXIMATION ......................................................... 69

5.3.2 GENETIC ALGORITHM-BASED ESTIMATION ..................................................... 71

5.3.3 ANT COLONY OPTIMIZATION-BASED ESTIMATION ......................................... 75

5.4 AUTOMATED IDENTIFICATION OF SECONDARY INCIDENTS ................................. 79

5.5 SIMULATION TEST ......................................................................................................... 82

5.6 RESULTS AND DISCUSSION ......................................................................................... 84

5.6.1 ESTIMATION OF THE IMPACT AREA .................................................................... 84

5.6.2 EVALUATION OF PERFORMANCE ....................................................................... 90

5.7 CONCLUSIONS .............................................................................................................. 96

CONCLUSIONS AND FUTURE WORK ........................................................................... 97

6.1 CONCLUSIONS .............................................................................................................. 97
6.2 FUTURE WORK .................................................................98
REFERENCES ........................................................................100
VITA ....................................................................................114
LIST OF PUBLICATIONS ......................................................115
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Summary of contributions</td>
<td>8</td>
</tr>
<tr>
<td>2. Representative Work on Incident Detection</td>
<td>17</td>
</tr>
<tr>
<td>4. Average Predictive Performance without Specifying Incident Locations and Traffic Demands</td>
<td>42</td>
</tr>
<tr>
<td>5. Average Predictive Performance Before and After Redefined Event Labeling without Specifying Incident Locations and Traffic Demands</td>
<td>60</td>
</tr>
<tr>
<td>6. Comparisons of accuracy rates of different methods for identifying SIs.</td>
<td>94</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Flow Diagram of Analysis of Traffic Incidents.</td>
<td>3</td>
</tr>
<tr>
<td>2. Conceptual Illustration of Incident Occurrence on Highways</td>
<td>30</td>
</tr>
<tr>
<td>3. An Proposed AI Modeling Framework with Memory Units and Learning Ability.</td>
<td>32</td>
</tr>
<tr>
<td>4. Memory Unit with the Archive of Historical Information for Reference</td>
<td>34</td>
</tr>
<tr>
<td>5. Illustration of Possible Incident Prediction Results.</td>
<td>40</td>
</tr>
<tr>
<td>6. Illustration of Incident Labeling Based on Upstream &amp; Downstream Detectors.</td>
<td>46</td>
</tr>
<tr>
<td>7. Issues in Labeling Incidents.</td>
<td>50</td>
</tr>
<tr>
<td>8. Framework of Relabeling Events in Training Dataset.</td>
<td>51</td>
</tr>
<tr>
<td>9. Relabeling Training Dataset via FCM Approaches.</td>
<td>57</td>
</tr>
<tr>
<td>10. Flow Chart of the Proposed Approach.</td>
<td>63</td>
</tr>
<tr>
<td>11. Clustering the Probe Vehicle Data Points.</td>
<td>65</td>
</tr>
<tr>
<td>12. Retrieving Boundary Points.</td>
<td>68</td>
</tr>
<tr>
<td>13. Determining the Critical Points at the Boundary.</td>
<td>69</td>
</tr>
<tr>
<td>14. Multi-stage Approximation for Establishing the Boundary.</td>
<td>70</td>
</tr>
<tr>
<td>15. Genetic Algorithm for Estimating the Boundary.</td>
<td>72</td>
</tr>
<tr>
<td>16. ACO-based Algorithm for Estimating the Boundary.</td>
<td>76</td>
</tr>
<tr>
<td>17. Identifying Secondary Incidents by the Ray Casting Algorithm.</td>
<td>80</td>
</tr>
<tr>
<td>18. Examples of Simulation Test Scenarios.</td>
<td>83</td>
</tr>
<tr>
<td>19. Simulated Vehicle Trajectories.</td>
<td>86</td>
</tr>
<tr>
<td>20. Results of the K-means and FCM Clustering.</td>
<td>87</td>
</tr>
<tr>
<td>21. Extracted Original Boundary Points (left) and Estimated Singular Points (right).</td>
<td>88</td>
</tr>
</tbody>
</table>
Fig. 22. Different Methods to Estimate the Impact Area................................................................. 90

Fig. 23. Estimated Accuracy Based on Each Approach (scenarios: PI with 20-min duration. .... 93
CHAPTER 1

INTRODUCTION

This chapter describes the motivation behind the work of this dissertation, and discusses the proposed work of primary incident detection, incident impact area estimation, and secondary incident identification. Finally, the contributions are outlined.

1.1 BACKGROUND

Traffic incidents frequently disrupt traffic operations and pose significant challenges for traffic management centers (TMCs) in maintaining reliable service on highway systems. It has been estimated that they account for nearly a quarter of all delays on the highway system in the United States [1]. In addition, traffic incidents can also significantly impact the safety of both motorists and incident responders by exposing them to the risk of secondary incidents [2].

Therefore, incident management agencies are actively working on various strategies with the common goal of detecting, responding to, and removing incidents, as well as restoring traffic operations as safely and quickly as possible. Many of the implemented strategies have shown great benefits in improving safety, mobility, motorist satisfaction, etc. [3, 4]. Among all the initiatives that are being taken, traffic incident detection (TID) and reducing false alarm rate (FAR) are the most important tasks which draw attention to many traffic management centers (TMCs). TTD denotes the time interval between the incident occurrence time and the time when it is detected by
a TMC. A timely and accurate detection of the incident occurrence is critical to the time-sensitive incident management plans. A high TTD is likely to result in the increment of victim injuries involved, heavier congestion, and high risks of secondary incidents. Meanwhile, FAR represents the percentage of falsely detected incidents. A high FAR will be subject to high cost of operating incident management programs since expense on falsely detected incidents are wasted.

As shown in Fig. 1, TMCs usually will continuously collect traffic data sourced from loop detectors and use algorithms to efficiently and timely detect an incident. Once an incident is detected, its impact area will be estimated, and its paired secondary incidents will be identified. The implementation of such tasks (detecting an incident, estimating its impact area, and identifying its paired secondary incident) are challenging due to the randomness of their occurrence time and locations. Many TMCs still heavily rely on human-based detection approaches, such as visual checking by TMC operators reviewing highway surveillance cameras [2]. However, manual incident analysis is time consuming and expensive. It brings high pressure on TMCs in the shortage of operators, especially during holidays and weekends. Therefore, an automated incident analysis framework which can continuously provide the needed detection functionality with a low FAR and TTD given the limited resources is needed.
1.1.1 PRIMARY INCIDENT DETECTION & ESTIMATING ITS IMPACT AREA

It should be noted that prior to activities such as responding to and clearing incidents, the immediate question is to detect incidents in a timely manner. An extended delay to detect incidents can expand their negative influences (e.g., a heavier congestion).

Numerous data-driven approaches have been developed and deployed as models to detect incidents in past decades. In general, many of these approaches primarily rely on the tuned mapping function between loop detector data and incident occurrence times for promptly detecting incidents. However, two main considerations by TMCs hinder their widely applications in real world. The primary concern is to avoid falsely predicting incidents which will result in high incident operational costs. For example, a nationwide survey on TMCs reported that only algorithms with no more than ten false alarms per day are acceptable, while most approaches failed to meet such an requirement [4]. Thus, many TMCs still need human-based detection approaches (e.g., visually checking surveillance cameras) for the sake of a lower rate to falsely detect incidents [5]. On the other hand, another concern is about the computational complexity and implementation
cost. Given the trade-off between diverse incident detection approaches’ performances and expenses, classical and simple approaches (e.g., CA#7) have been frequently employed by TMCs. However, it should be noticed that tuning such models requires extensive involvement of traffic experts. Nevertheless, the tuned thresholds of CA#7 typically cannot be transferred to a new site or may not work well if the prevailing traffic pattern changes.

Multiple reasons challenge the high performance of aforementioned machine learning approaches, such as the randomness of incidents’ occurrence times, severities, and locations, loop detector data’s precision (e.g., aggregation time interval), and the trial-and-error strategy to tune models. Among all such issues, the data precision issues regarding loop detector measurements and incident reports have been rarely addressed. However, TID approaches, ranging from classical comparative approaches to prevalent machine learning approaches, heavily rely on the quality of loop detector measurements and incident reports to tune specific models. Conventionally, TID models are tuned to match incident labels with loop detector measurements. Since not every anomaly in loop detector measurements is in fact an incident, it will introduce inaccurate labels if only using unsupervised approaches such as clustering algorithms to identify anomalies and assign them with them to incident labels. Meanwhile, incident reports provide the benchmark of clustering boundary and incident labels. However, incident reports are prone to several issues such as delay due to phone reports and the incident propagation time to nearby loop detectors. This will also lead to the inaccurate mapping between incident labels and loop detector measurements. In addition, an incident can occur and does not lead to any observable congestion (e.g., at midnight).
However, according to the incident report, the normal traffic condition reflected by loop detectors will be labeled as incident affected. In summary, these aforementioned inaccurate maps within training incident data will either be ignored as outliers or degrade TID performances during the interfered tuning procedure. As such, it is expected to tackle such noisy data issues to better address TID challenges well.

1.1.2 SECONDARY INCIDENT IDENTIFICATION

Other than the notable traffic congestion issue, incidents occurred on roadways also increase exposure to secondary incident hazards [6]. It was reported that the risk of having an incident can increase more than six times post the occurrence of an incident [7, 8]. In addition, the likelihood of a secondary incident (SI) increases by 2.8 percent if the primary incident presents for an additional minute [3]. This often exposes road users and incident responders to higher risk. In addition, it is difficult for the rescue crews to reach and clear the incident scene. It was estimated that secondary incidents accounted for about twenty percent of all incidents and eighteen percent of all fatalities on the US freeways [5, 9]. Thus, preventing secondary incidents can result in millions of economic benefit [10].

According to the Federal Highway Administration program (FHWA), namely “TIM Performance Measures Focus States Initiative,” the number of secondary incidents is considered a core performance measurement [11]. For example, many state agencies considered the determination and the reduction of secondary incidents in allocating funding for Road Rangers and the development of TIM programs [12-14]. To reduce the risk of secondary incidents more
successfully, the mechanism and characteristics of secondary incident occurrence need to be well understood. For instance, when, where, and how do they occur? Prior to addressing these questions, however, an immediate challenge is to identify the secondary incidents. Almost all incident report forms in practice do not have a label to mark whether an incident is an SI. As a complementary solution, a few studies have proposed several post-event analysis methods to address the issue. For example, Yang, et al. [15] used regression models for determining the corner points of the impact area for identifying secondary incidents. Others considered queuing models [16, 17] and speed contour maps [14, 18] to relate an SI to a primary incident (PI). However, these methods often require a wide range of simplified estimation procedures and assumptions (e.g., spatiotemporal windows and simplified models). In addition, the identification of secondary incidents is a challenging task due to the heterogeneous nature of day-to-day traffic conditions.

Therefore, an efficient approach to timely estimate the impact area of primary incidents, identify secondary incidents, and even prevent incident occurrences is needed.

1.2 PROPOSED WORK AND CONTRIBUTIONS OF THE DISSERTATION

First, this dissertation proposed an AI-based incident detection framework that can leverage large-scale sensor data along with advanced learning algorithms to enhance the predictive performance. It investigates the generic algorithmic problems when designing a detection approach and places more emphasis on the architecture of the AI-based detection framework with
the inclusion of learning and evolving capabilities. The proposed framework has been assessed by case studies.

Second, this dissertation examined the impact of such inaccurate event labels on different incident detection approaches, and further introduced an unsupervised learning approach, fuzzy c-means clustering (FCM), to infer accurate observable incident occurrence times reflected by loop detector measurements. Three representative approaches including California#7 (CA#7), artificial neural network (ANN), and support vector machine (SVM), are selected. VISSIM is used to collect data, and R is used to evaluate proposed approaches under numerous incident scenarios. The test results based on comparisons between TID models using relabeled training database and those using original database indicate that redefining such mislabeling database helps to improve TID performances in terms of higher detection rates, lower false alarm rates, and reasonable longer times to detect incidents.

Last, this dissertation proposed a data-driven analysis framework for innovating the identification of secondary incidents. The dissertation intends to leverage the untapped potential of ubiquitous probe vehicle data for secondary incident identification. The developed framework consists of three major components: detecting the impact area of a primary incident, estimating the boundary for the impact area, and identifying secondary incidents within the boundary. The proposed framework has been tested based on probe data collected from different simulation models. The results show that the impact area induced by a primary incident can be well represented by the estimated boundary, especially by the GA- and ACO-based methods.
A summary of the dissertation contributions is listed in Table 1. We have several peer reviewed publications: the research related to topic 1 is supported by the paper “Development of an AI-based Modeling Framework for Traffic Incident Detection” presented in TRB 2020 Annual Meeting [19]; the research related to topic 2 is supported by the paper “Augmenting Traffic Incident Detection Performance with Redefined Event Labels” presented in TRB 2020 Annual Meeting [20]; the research related to topic 3 is supported by the paper “Use of Ubiquitous Probe Vehicle Data for Identifying Secondary Crashes” published in Transportation Research Part C [21], and “Methodological Evolution and Frontiers of Identifying, Modeling and Preventing Secondary Crashes on Highways” published in Accident Analysis & Prevention [22].

<table>
<thead>
<tr>
<th>Topic</th>
<th>Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 PRIMARY INCIDENT DETECTION BASED ON AI APPROACH</td>
<td>In this research, we proposed an AI-based framework. The framework using either lane-based data or station-level average data performed relatively better than the benchmark approaches, including CA#7 and ANN. The augmented performance of the proposed approach has been primarily demonstrated in terms of the shorter detection time, lower false alarm rate, and higher detection rate. The presented results show the improved performance of the proposed AI-based framework regardless of the sensor spacing.</td>
</tr>
<tr>
<td>2 INCIDENT IDENTIFICATION WITH REDEFINED EVENT LABELING</td>
<td>In this research, we introduced an unsupervised learning approach, FCM, to relabel incident occurrence times and further examined its impact on three different incident detection approaches (CA#7, ANN, and SVM). The proposed relabeling strategy is evaluated given 360 simulated incident scenarios. The results showed that the relabeling strategy can help improve the performance of three TID approaches in terms of a higher DR and a lower FAR. Comparisons of TID performances under different scenarios also imply that TID approaches need more efforts to predict incidents with a longer loop detector gap, the incident occurrence in the middle</td>
</tr>
</tbody>
</table>
between two loop detectors, and under a lower traffic volume condition.

3 SECONDARY INCIDENT IDENTIFICATION

In this research, we proposed a data-driven analysis framework for identifying secondary incidents. The proposed approach intends to leverage the untapped potential of ubiquitous probe vehicle data for secondary incident identification. The developed framework consists of three major components: detecting the impact area of a primary incident, estimating the boundary for the impact area, and identifying secondary incidents within the boundary. The proposed framework has been tested based on probe data collected from different simulation models. The results show that the impact area induced by a primary incident can be well represented by the estimated boundary, especially by the GA- and ACO-based methods.

1.3 DISSERTATION STRUCTURE

The dissertation is organized as follows: Chapter 2 provides the literature review for existing efforts on traffic incidents identification, analysis, and prevention on highways; Chapter 3 proposes an AI-based incident detection model which can learn from historical information and thus avoid repeating false alarms; Chapter 4 elaborates on an enhanced incident labeling approach which can alleviate the impact of inaccurate and imprecise field data on incident detection; Chapter 5 explores the secondary incident identification based on advanced machine learning algorithms; and Chapter 6 provides conclusions regarding the aforementioned work on traffic incidents with advanced machine learning algorithms.
CHAPTER 2

LITERATURE REVIEW

Previous relevant works are discussed in this chapter. The aforementioned major key points that we are going to address in this dissertation include incident detection, incident impact area estimation, secondary incident identification, and incident prevention. The literature review on each of the topics is presented below.

2.1 INCIDENT DETECTION

Diverse methods on traffic incident detection can be grouped into two categories: (a) methods comparing static/dynamic thresholds, and (b) methods based on statistical/machine learning approaches.

Comparative algorithms assume that measured traffic metrics (i.e., volume, occupancy or speed) will change upon incident occurrence. Observed values will be compared with the predefined threshold values. Once the measured traffic metrics exceed the thresholds, an incident alarm is detected. Typical algorithms such as decision tree algorithms [23], pattern recognition algorithm [24], and APID algorithm [25].

The decision tree (DT) algorithms hold a hierarchal structure and obtain different output states on leaf nodes. The most widely known DT algorithms are California algorithms. California Algorithm #7 utilized occupancy data observed from two adjacent detector stations to derive traffic
metrics such as DOCC, OCCDF, and OCCRDF. OCCDF represents for the absolute difference in occupancy between the upstream and downstream detectors, OCCRDF denotes for the relative difference in occupancy between upstream and downstream detectors compared to the upstream occupancy, and DOCC is the occupancy values obtained from downstream detectors. Three thresholds $Th_1, Th_2$ and $Th_3$ are predefined. If $OccDF > Th_1$, $OccRDF > Th_2$, and $DOcc < Th_3$, a potential incident is identified. Further, if the $OccRDF > Th_2$ for two consecutive steps, then the incident occurrence will be reported. Later, the All-Purpose Incident Detection (APID) algorithm incorporates and expands the major elements of California algorithm into a comprehensive structure (e.g., heavy volume traffic conditions, light volume traffic conditions) [24, 26, 27].

Meanwhile, other traffic metrics such as travel time and vehicle speed have been utilized. For example, the PATREG algorithm was developed by the Transport and Road Research Laboratory (TRRL) [17]. Vehicle speeds are estimated by observing travel times between detectors. Once vehicle speed exceeds the pre-established threshold values for a preset number of consecutive time steps, an incident alarm will be triggered.

On the other hand, researchers also developed methods based on statistical metrics to detect potential incidents. Once the observed traffic metrics differ statistically from estimated or predicted values, incidents will be detected. Four types of typical algorithms include: standard normal deviate (SND) algorithm [26], Bayesian algorithms [27, 28], ARIMA model [29-31], and high occupancy (HIOCC) algorithm [17].
SND algorithm was developed by the Texas Transportation Institute [32], and computed the number of deviations when the 1 minute average occupancy value deviates from the mean values. When the number of SND exceeds the predefined threshold, an incident is detected. Meanwhile, Bayesian algorithm estimated the frequency distributions of upstream and downstream occupancies during incident and incident-free conditions. The conditional probability using Bayesian statistics are calculated, and one the probability exceeds the threshold, an incident is detected.

Other than using statistical approaches to estimate such metrics, time series algorithms predicted following traffic conditions based on diverse time series. If the predicted values deviate significantly from observed values, an incident is detected. The ARIMA model takes advantage of the temporal correlation between traffic variables measured in current time step $t$ and previous time step $t-k$ and learns the normal pattern of such a relationship under incident free conditions. Short-term forecasts and confidence intervals of traffic variables are calculated. Incidents are detected once observed metrics fall outside the predicted confidence intervals. Meanwhile, HIOCC algorithm monitors detector data for changes over time with a high frequency. They conduct the field test from M1 and M4 in London and prove that HIOCC can work well under congested conditions with heavy traffic.

In order to archive the information about traffic conditions with a higher accuracy and precision, some researchers have also introduced smoothing and filtering techniques to remove short-term noises or inhomogeneities. Smoothing produced weighted average traffic metrics, while
filtering discards the undesirable high frequency noises of traffic metrics. The representative smoothing/filtering algorithms consist of the double exponential smoothing (DES) algorithm [33], low-pass filter (LPF) algorithms [34, 35], and the discrete wavelet transform and linear discriminant analysis (DWT-LDA) algorithm [36, 37].

The double exponential smoothing algorithm weights historical and current traffic metrics such as volume, occupancy, and speed to reflect true traffic conditions as closely as possible. Meanwhile, LPF algorithm series get rid of high frequency fluctuations which are considered as noise. Wide or low frequency fluctuations are considered as incident conditions and remained. In addition, the DWT-LDA algorithm is proposed as a traffic preprocessor to eliminate noises and provide higher quality traffic metrics as inputs for further neural network models.

Other than using data from traffic sensors, researchers have also sought data from other sources such as GPS and images for incident detection. For example, the AIDA algorithm makes use of both temporal and spatial variations of traffic metrics [34, 35, 38, 39], and has been improved to include ancillary information provided by video detection. Once an incident is detected, traffic operators can quickly verify alarms through visually checking. Meanwhile, probe vehicles can provide more detailed information about traffic conditions with a wider roadway coverage. For example, Parkany and Bernstein [40] and Parkany and Bernstein [41] conducted an initial exploration of the use of vehicle-to-roadside communication. Three detection algorithms respectively using headway, lane switches, and lane monitoring were proposed. Later, the Texas Transportation Institute (TTI), in conjunction with the Texas Department of Transportation made
use of the probe vehicle travel time to detect incidents [42]. Vehicles with cellular phones communicate with the communication center, and thus travel time between two adjacent reference points are calculated. It should be noted that the cellular probe system served as a prelude to the AVI system installed in Houston.

Many learning algorithms consider incident detection as a task to construct a classification model according to traffic metrics. Approaches such as machine learning (ML) and deep learning (DL) have been applied to classify traffic states with and without incidents.

Artificial neural networks (ANNs) have been widely studied to detect freeway incidents and proved to provide promising performance [38, 39, 43]. Many models have been developed such as multilayer feed-forward NN (MLFNN), constructive probabilistic NN (CPNN) [44], and probabilistic neural network (PNN) [45]. Meanwhile, other machine learning algorithms have also been applied to detect traffic incidents such as support vector machine (SVM), particle swarm optimization (PSO), and random forest. For example, Yao, et al. [46] employed the tabu search algorithm to optimize parameters of SVM to detect incidents. Local optima can be avoided. In addition, ML algorithms have been coupled with ANNs to improve the detection performance [47]. Kinoshita, et al. [48] introduced a traffic state model based on a probabilistic topic model to describe traffic states for a variety of roads.

Some researchers examined the usage of DL approaches for incident detection. DL uses multiple-layer architectures or deep architecture of neural networks to extract inherent features in data of different complexities and can represent them without prior knowledge, which offers
promising functions in traffic incident detection. For example, El Hatri and Boumhidi [49] proposed a novel fuzzy DL-based detection method that considers the spatial and temporal correlations of traffic flow. The fuzzy logic is induced to avoid the slow convergence rate and trapping by local optimums during tuning learning parameters. Meanwhile, Zhang, et al. [50] used deep belief network (DBN) and long short-term memory (LSTM) in detecting traffic accidents from social media data. DBN outperforms SVMs and ANN when processing the tweet data and matching them to nearby abnormal traffic data. Meanwhile, LSTM does not achieve a good performance, since it depends on the sequential information while words (token) in the tweet posts are not well sequentially organized. With the DL approaches, the detection performance may be improved but the relatively time and space complexity will increase, and thus needs more computing resources for real-time implementation. It also should be noted that such complex models are subject to overfitting issues.

Meanwhile, with the growth of social media, crowdsourcing data such as Twitter texts have also been examined. For example, researchers extract information from social media texts using natural language processing algorithms (NLP), map such data into the high dimensional vectors in the feature space, and classify incident patterns on temporal and spatial dimensions [51-53]. For example, Gu, et al. [52] proposed a methodology to crawl, process, and filter tweets that are accessible by the public for free. Tweets were acquired from Twitter using its REST API in real time. Nevertheless, relevant social media data are often limited by time and locations and can only
serve as a supplement for incident detection since lots of incidents were not (timely and precisely) described by social media sources.
### TABLE 2. REPRESENTATIVE WORK ON INCIDENT DETECTION

<table>
<thead>
<tr>
<th>References</th>
<th>Data (Training/Test size; Duration, Location)</th>
<th>Method</th>
<th>Variables</th>
<th>DR (%)</th>
<th>TTD (s)</th>
<th>FAR (%)</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>[49]</td>
<td>30 incidents, 100 seconds interval, simulated data via SUMO</td>
<td>DSAE</td>
<td>Traffic flow count</td>
<td>98.23</td>
<td>192.44</td>
<td>0.24*</td>
<td>Complex</td>
</tr>
<tr>
<td>[38]</td>
<td>Simulated data with 100 incidents, 30 seconds interval, SR-91 Riverside Freeway, California</td>
<td>ANN</td>
<td>volume, occupancy at both upstream and downstream detectors</td>
<td>21</td>
<td>60</td>
<td>0.127*</td>
<td>Moderate</td>
</tr>
<tr>
<td>[33]</td>
<td>Simulated freeway, 150 minutes</td>
<td>ANN</td>
<td>Feature extraction of occupancy and volume</td>
<td>100</td>
<td>47.8</td>
<td>1.2*</td>
<td>Moderate</td>
</tr>
<tr>
<td>[46]</td>
<td>304 incidents, April 16-20, 2012, 30 seconds interval, 500~700-meter interval, Liaoning, China</td>
<td>SVM</td>
<td>Weather, time, occupancy, volume</td>
<td>95.7</td>
<td>72.6</td>
<td>4.82*</td>
<td>Moderate</td>
</tr>
<tr>
<td>[44]</td>
<td>45 incidents, 30 seconds interval, I880 California</td>
<td>ANN</td>
<td>Volume, occupancy, speed</td>
<td>86.96</td>
<td>228</td>
<td>0.2*</td>
<td>Moderate</td>
</tr>
<tr>
<td>[52]</td>
<td>Jan to Jul 31, 2013, 322 incidents, 5 min interval, Twitter data, Japan</td>
<td>DSAE</td>
<td>GPS, three layers</td>
<td>79.8</td>
<td>NA</td>
<td>0.04*</td>
<td>Moderate</td>
</tr>
<tr>
<td>[47]</td>
<td>138 incidents, 1,518 incident-free, Chongqing, China</td>
<td>LVQ, fuzzy logic</td>
<td>Volume, occupancy, speed, meteorological parameters</td>
<td>96.5</td>
<td>152.4</td>
<td>0.21*</td>
<td>Moderate</td>
</tr>
<tr>
<td>[50]</td>
<td>584,000 geo-tagged tweets in northern Virginia, 2,420,000 tweets in NYC from Jan 2014 to Dec 2014</td>
<td>DBN, LSTM</td>
<td>Token features extracted from tweets</td>
<td>95</td>
<td>NA</td>
<td>30*</td>
<td>Complex</td>
</tr>
<tr>
<td>[54]</td>
<td>Probe data, 1 min interval, Apr – Jul 2016, Iowa state</td>
<td>SND, outlier detection</td>
<td>Speed, probe data</td>
<td>54.1</td>
<td>887</td>
<td>0.043*</td>
<td>Simple</td>
</tr>
<tr>
<td>[55]</td>
<td>I-35W, Minneapolis</td>
<td>Filtering, CA#7</td>
<td>Occupancy</td>
<td>93</td>
<td>244</td>
<td>0.5*</td>
<td>Simple</td>
</tr>
</tbody>
</table>

**NOTE:** “*” DENOTES THAT FAR IS CALCULATED BASED ON THE NUMBER OF FALSE ALARM CASES DIVIDED BY TOTAL NUMBER OF NON-INCIDENT INSTANCES.

“**” DENOTES THAT FAR IS CALCULATED BASED ON THE NUMBER OF FALSE ALARM CASES DIVIDED BY TOTAL NUMBER OF INSTANCES.
2.2 INCIDENT IMPACT ESTIMATION & SECONDARY INCIDENT IDENTIFICATION

Once an incident is confirmed, the incident impact area estimation is necessary to measure the impact of an incident and confirm potential secondary incidents. A state-of-the-art review of the existing studies on incident impact estimation and identifying secondary incidents is provided. Most of existing studies were conducted in the recent two decades, with a focus on developing different methods/procedures for capturing the impact area of primary incidents. The available methods can be grouped into four categories, including static spatiotemporal threshold-based, queuing model-based, speed contour map-based, and the shockwave-based approaches. The relevant studies of each category were reviewed and discussed below.

Early studies defined fixed spatiotemporal thresholds to depict the impact area of an incident, and incidents falling into the impact area are defined as secondary incidents. For example, The assumption in [56, 57] is that secondary incidents are those occurred upstream within one mile and within a time frame of primary incident clearance plus 15 minutes. Many others adopted similar criteria but with some variations on the spatial and temporal thresholds. Table 3 summarizes current studies on secondary incident identification. Obviously, there is no consistent criterion to define the time-space window (thresholds). Although some studies considered the variation of each incident’s clearance time, all of them remain static (or in-kind). The subjective spatiotemporal thresholds applied to all conditions (regardless of traffic conditions, roadway
### TABLE 3. STATE-OF-THE-ART REVIEW OF METHODS FOR IDENTIFYING SECONDARY INCIDENTS.

<table>
<thead>
<tr>
<th>Study</th>
<th>Type</th>
<th>Major Data Needs</th>
<th>Method for Identifying Secondary Incident</th>
</tr>
</thead>
<tbody>
<tr>
<td>[56-58]</td>
<td>static incident</td>
<td></td>
<td>&lt; clearance time+15 min., &lt; 1 mile</td>
</tr>
<tr>
<td>[59]</td>
<td>static incident</td>
<td></td>
<td>&lt; clearance time+15 min., &lt; 3 miles</td>
</tr>
<tr>
<td>[36]</td>
<td>static incident data</td>
<td></td>
<td>&lt; 2 hours, &lt; 2 miles</td>
</tr>
<tr>
<td>[60]</td>
<td>static incident</td>
<td></td>
<td>&lt; clearance time+15 min., &lt; 2 miles, lane closure</td>
</tr>
<tr>
<td>[61]</td>
<td>static incident</td>
<td></td>
<td>&lt; actual duration, &lt; 1 mile upstream</td>
</tr>
<tr>
<td>[62, 63]</td>
<td>static incident</td>
<td></td>
<td>&lt; 2 hours, &lt; 2 miles (both directions)</td>
</tr>
<tr>
<td>[64]</td>
<td>static incident</td>
<td></td>
<td>&lt; 2 hours, &lt; 2 miles; &lt; 0.5 h, &lt; 0.5 mile (opposite)</td>
</tr>
<tr>
<td>[65]</td>
<td>static incident data</td>
<td></td>
<td>&lt; 80 min, &lt; 6,000ft; &lt; 1,000ft (other direction)</td>
</tr>
<tr>
<td>[66]</td>
<td>static incident data</td>
<td></td>
<td>&lt; 2 hours, &lt; 2 miles</td>
</tr>
<tr>
<td>[67]</td>
<td>static incident data + incident data</td>
<td></td>
<td>&lt; 2 miles; &lt;2 h, or &lt; clearance time + (15 or 30) min.</td>
</tr>
<tr>
<td>[68]</td>
<td>dynamic incident</td>
<td></td>
<td>maximum queuing model</td>
</tr>
<tr>
<td>[12, 69]</td>
<td>dynamic incident</td>
<td></td>
<td>incident progression curves</td>
</tr>
<tr>
<td>[13]</td>
<td>dynamic incident</td>
<td></td>
<td>deterministic queuing model</td>
</tr>
<tr>
<td>[11, 70]</td>
<td>dynamic incident + simulated traffic data</td>
<td></td>
<td>determine impact using simulated speed contour map</td>
</tr>
<tr>
<td>[71, 72]</td>
<td>dynamic incident + monitor + sensor data</td>
<td></td>
<td>identify influential area by ASDA model</td>
</tr>
<tr>
<td>[15, 73-75]</td>
<td>dynamic incident + sensor data</td>
<td></td>
<td>determine spatiotemporal impact by speed contour plot</td>
</tr>
<tr>
<td>[14]</td>
<td>dynamic incident + sensor data</td>
<td></td>
<td>determine incident impact region by speed contour plot</td>
</tr>
<tr>
<td>Ref</td>
<td>Type</td>
<td>Data Used</td>
<td>Methodology</td>
</tr>
<tr>
<td>-----</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>75</td>
<td>dynamic</td>
<td>incident + virtual sensor data</td>
<td>determine spatiotemporal impact by speed contour plot</td>
</tr>
<tr>
<td>76</td>
<td>dynamic</td>
<td>incident + INRIX data</td>
<td>determine spatiotemporal impact by speed contour plot</td>
</tr>
<tr>
<td>77</td>
<td>dynamic</td>
<td>incident + hourly volume data + detailed network</td>
<td>simple shockwave model to depict impact area</td>
</tr>
<tr>
<td>78</td>
<td>dynamic</td>
<td>detailed incident data + lane specific traffic sensor data</td>
<td>simple shockwave model to depict impact area</td>
</tr>
<tr>
<td>79</td>
<td>dynamic</td>
<td>detailed incident data + loop data</td>
<td>simple shockwave model to depict impact area</td>
</tr>
</tbody>
</table>
geometry, incident characteristics, weather, etc.) are deemed to be the weakness of these simple static approaches. Thus, the performance of these approaches is often questionable due to: (a) overestimation - the spatiotemporal thresholds are too large; and (b) underestimation - the spatiotemporal thresholds are too small.

To account for the dynamic progression of the impact associated with a primary incident, several studies developed queuing models to identify the influence area of a primary incident [13, 68, 69]. In general, they attempted to estimate vehicle queue length as a function of a set of explanatory variables (e.g., incident duration and number of lanes blocked). For example, Zhang and Khattak [13] estimated the maximum queue length and the recovery time as simple linear functions of assumed arrival rate, diversion rate, full capacity, lane closures, and the capacity reduction factor from the Highway Capacity Manual (HCM). Likewise, Zhang and Khattak [80] developed the deterministic queuing models to calculate the influence area of a primary incident. Sun and Chilukuri [69] proposed the concept of the incident progression curve and modeled it as a high-order polynomial equation using time elapsed after the occurrence of the primary incident as the variable. The severity and occurrence time of the primary incident were used to obtain different incident progression curves.

Despite the improvement in misclassification of secondary incidents, these queuing model-based approaches are still questionable. One challenge is the establishment of a reliable queuing model. Each roadway segment is subject to a different queuing process due to its unique traffic patterns, geometry, speed limit, ramps, number of lanes, incident characteristics, etc. A single
queuing model cannot be applied to the entire studied highway systems, let alone the challenge of collecting all precise data for the model development. It is impractical to build separate queuing models for each segment. Many unreasonable assumptions associated with these queuing models also make it difficult to use these models to construe the prevailing traffic conditions when a primary incident occurs. For example, it was assumed that the maximum impact area of a primary incident occurs at the clearance time [69] or the spatial impact only exists within the incident duration [13, 71]. However, these types of assumptions can be easily violated if the upstream vehicle arrivals exceeded the downstream discharging flow rate after clearing the primary incident [80]. Therefore, more research endeavor is expected to improve these queuing model-based approaches for identifying secondary incidents.

Using traffic sensor data, several recent studies made effort to identify secondary incidents through data-driven approaches [11, 14, 15, 70-74, 81, 82]. For example, Chou and Miller-Hooks [11] and Haghani, et al. [70] simulated the impact of primary incidents to identify secondary incidents. Vlahogianni, et al. [71], Vlahogianni, et al. [81], Orfanou, et al. [72] and Imprialou, et al. [83] used the Automatische Staudynamikanalyse: Automatic Tracking of Moving Traffic Jams (ASDA) model based on traffic sensor data to determine the impact area by a primary incident. Our earlier works [15, 73-75] have introduced the speed contour plot-based identification methods. The proposed algorithms automatically analyze massive detector data to quantify the influence of a primary incident with the assistance of the binary speed contour map. The proposed method has been used by [84] with some minor variations. In a project for the Virginia Department of
Transportation (VDOT), they assumed that only 90 percent of the time-space intervals between
the secondary incident and primary incident need to be non-recurrent congestion cells on a speed
contour map. In addition, they included the incident duration to enable the identification of
secondary incidents that may occur at any point along the primary incident timeline. A key premise
of these methods is the estimation of the reference speeds based on archived sensor data. To make
it more practical, Yang, et al. [74] used the concept of percentile-speed to obtain reference speeds.
Other complicated models such as the Gaussian Mixture Model (GMM) were introduced by Park
and Haghani [85] to determine the reference speed for classifying congested and non-congested
periods of each link. They used the Bayesian structure equation model to determine congestion
patterns based on Inrix data [78, 86]. By synthesizing the real-time traffic information and incident
data to capture the prevailing traffic conditions, these speed contour plot-based approaches greatly
improved the model of the spatiotemporal impact area of a primary incident.

However, the wide deployment of these methods is limited by the availability of the
historical traffic data. Despite the available traffic sensors (loop detectors, remote traffic
microwave sensors, etc.), not all highway roadways are instrumented due to various reasons (e.g.
high cost, maintenance requirements, data storage, etc.) [87]. Most of the roadways only have
sparse detectors, which makes it difficult to accurately quantify the congestion area. In addition,
massive historical data are needed to estimate the reference speed. As an alternative, Yang, et al.
[88] extended the data-driven approaches by introducing the concept of the virtual sensors to
collect open source traffic data from third-party such as the Bing Map, Google Map, and
MapQuest. The proposed approach makes it more scalable to be deployed on highways. Nevertheless, additional labor work is needed to mark the virtual sensors on maps.

Instead of directly modeling and visualizing the impact area of a primary incident through a speed contour map, some studies have attempted to use the traffic flow theory to assist the identification of secondary incidents. For example, Zheng, et al. [77] considered the shockwave model to estimate the impact area induced by a primary incident. This study assumed that only two simplified straight shockwaves exist under the primary incident condition, namely a queuing shockwave and a discharging shockwave. The area enclosed by these two shockwaves on the time-space diagram and the clearance time of the primary incident were then used to describe the impact area. To be implementable, they also assumed: (a) monthly average hourly traffic volume represents the traffic flow prior to the primary incident; (b) a fixed speed (65 mph) represents the prevailing speed of the link (with a length of one mile or more) prior to the occurrence of the primary incident; (c) zero flow \( q_{jam} = 0 \) \( vphpl \) for the jammed condition \( k_{jam} = 352 \) \( vpmpl \); and (d) a fixed traffic flow rate \( q_s = 1,900 \) \( vphpl \) and density \( k_s = 29 \) \( vpmpl \) for the saturated condition.

Some assumptions of the aforementioned method can be unrealistic. For example, despite the occurrence of a primary incident, the throughput at the incident site might not be reduced to zero (e.g. due to a minor incident), or the throughput might not be recovered to its full capacity immediately after the clearance of the primary incident. Thus, the fully jammed and saturated conditions are often not achievable. In addition, the day-to-day traffic flow and speed can vary
significantly even no incident occurs. Thus, the assumed reference volume and speed prior to the primary incident cannot reflect the actual traffic conditions.

To address a part of the aforementioned issues associated with the shockwave-based method, several studies have made some modifications. Recently, Sarker, et al. [78] collected lane closure information based on incident severity and number of vehicles involved to estimate the capacity reduction and subsequently the traffic flow state after the primary incident. However, they still used the back-of-queue and front-of-queue shockwaves with constant propagation speeds to identify the triangular congestion area due to the primary incident. Based on the density-flow curve, these shockwave propagation speeds were determined. It should be noted that the shape of the density-flow curve is not deterministic for all road segments due to various factors such as the proportion of truck, road geometry, weather, etc. The uncertainty associated with the hypothetical curve is deemed to affect the estimation of the shockwave speeds, which in turn leads to incorrect identification of the impact area. Instead of assuming one straight back-of-queue shockwave, Wang, et al. [79] modified the estimation of this shockwave by considering the potential shockwave induced by the rescue personnel. They assumed that another shockwave will be created upon the arrival of the responders to manage the traffic incident. Then the back-of-queue shockwave was modeled as a piecewise linear function where slope of the two segment is different. Despite the modification, it still subjects to similar issues as the two early studies.

Other than the modeling issues, the premise of the shockwave-based method relies on the archived traffic sensor data and more detailed incident information. The aforementioned studies
both used the traffic data from detectors (e.g., flow, speed and/or occupancy) to estimate the shockwaves. Unfortunately, the high-density (e.g., 0.1-mile or even shorter) installation of detectors on most of the highways is not available. The available data have typically been sparse in geographic coverage, slow in responding to changes in traffic patterns, and have only acquired crude approximations to the variables of real significance (e.g., downstream capacity and upstream demand; location of the queue between two consecutive detectors). Meanwhile, the detailed incident information (e.g., number of vehicles, lane closures, arrival of rescue crew, etc.) along the timeline of the primary incident might not be timely available for estimating the capacity reduction. In addition, all the existing shockwave-based studies only examined primary incidents that induced a queue. The scenario that a primary incident occurs within an existing queue has not been captured. All these issues bring new obstacles to implement the shockwave-based methods.

2.3 INCIDENT PREVENTION

The above sections focused on the identification and analysis of incidents and their impacts. The upcoming issue is to prevent incident occurrences. However, only a few studies have focused on incident prevention issues.

The primary countermeasures explored in existing studies include the deployment of Active Traffic Management (VMS) such as changeable message signs (CMS) or variable speed limit control (VSL) and connected vehicles (CVs). For example, Kopitch and Saphores [89] verified the effectiveness of 11 changeable message signs that provided real-time traffic
information about incidents, work zones, congestion, speed limits ahead, and alerts in reducing risk. It was found that the effectiveness of CMS increased between 2 and 11.15 miles and decreased between 11.15 and 22.3 miles. Li, et al. [90] introduced the strategy of implementing variable speed limit with both weather and traffic flow information to mitigate secondary incident risk. Two surrogate safety measures, including time exposed time-to-collision (TET) and time integrated time-to-collision (TIT), were found to be reduced by 40 to 50 percentage in a case study on I-880 in California during a heavy rain condition. Lately, Yang, et al. [91] examined the impact of connected vehicles on improving the situational awareness of drivers to mitigate secondary incident occurrences. Secondary incident risk measured by the simulated conflicts was found to be significantly reduced if the market penetration rate of connected vehicles on a highway was relatively high in dense traffic conditions.

Other than the aforementioned countermeasures, some studies also examined the benefits of service patrol programs in reducing incidents. For example, Karlaftis, et al. [92] examined the effect of the Hoosier Helper service patrol program on the Broman Expressway in Indiana. It was found that the program may help reduce secondary incident likelihood by 18.5 percent in winter and by 36.3 percent in other seasons per incident assisted. The delay savings and incident cost savings from secondary incident reduction was $568,080 in 1995 and it was 1.38 times of the service patrol program cost. Although there was no quantitative assessment, some other studies also suggested the use of service patrol programs as a helpful countermeasure to reduce secondary
incident risk. For example, Khattak, et al. [93] suggested improve the coverage of service patrols and towing service on highway chokepoints with higher incident occurrence probability.

Other researchers have also considered the usage of context-awareness mobile devices for traffic incident prevention. Martinelli, et al. [94] utilized the concept of On Board Diagnostics (OBD) to help drivers make informed decisions. The traffic scenarios based on a town of southern Italy in SUMO are simulated. The number of incidents are reported to gain the percentage variation ranging from 85.48% to 88.99% between the one with and without OBDs. Hsiao, et al. [95] conducted a descriptive analysis regarding the contributing factors of emergency vehicle incidents.

Instead of focusing on the prevention of incident, the mitigation of incident impacts has also been investigated. Compared with previous studies that only used primary incident information, Park, et al. [96] considered the evolution of primary incidents and secondary incidents over time to discuss the appropriate location of emergency response units. Linear programming approach with relaxed integrality constraint for integer variables was verified to be valid in reducing the expected total delay in a numerical study with data collected on I-695.

2.4 SUMMARY

The state-of-the-art studies regarding incident detection, estimation of its impact area, and prevention are summarized. It should be noted that there exist several shortcomings, and thus it is necessary to explore a more comprehensive approach to detect, estimate, and prevent potential incidents.
CHAPTER 3

PRIMARY INCIDENT DETECTION BASED ON AI APPROACH

Traffic incidents affect the high performance of traffic operations and result in high economic loss and safety issues. Thus, it is critical to detect incidents as early as possible to provide better transportation services. This Chapter provides a new artificial intelligence-based approach to identify traffic anomalies and detected incidents. The essential components of traffic incident detection approach are described as follows.

3.1 METHODOLOGY

Fig. 2 illustrates an incident that occurred at location $s$ and time step $t$. This incident may lead to congestion due to lane closure. From the perspective of TMCs, one key task is to timely identify the occurrence of the incident. The identification of the incident often relies on the use of surveillance cameras or sensor systems, the most common of which is measurements from loop detectors (e.g., sensor stations $L_1$ and $L_2$ in Fig. 2). This largely depends on the hypothesis that incident occurrence will be indirectly reflected by the fluctuation of traffic conditions as measured at the sensor locations. The generic problem of incident detection is reduced to the analysis of the changes in detector measurements. The following equation provides a high-level generalization of the identification problem.

$$ Y \leftarrow f(X) $$

(1)

$Y$ denotes the prevailing traffic condition with $(Y=1)$ or without $(Y=0)$ incident occurrence. $X$ represents the detector measurements, and $f()$ is a specific modeling approach that associates the detector measurements with the prevailing traffic condition. By specifying appropriate model
structure and conducting model calibration and validation, a final well-tuned model $M_x : f(X \mid \alpha)$ can be established, where $f()$ represents the finalized model with a calibrated parameter set $\alpha$. Depending on the structure of $f()$ and the number of elements in $\alpha$, the model complexity and required computational resources will be different. The tuned model $M_x$ is expected to be as efficient and accurate as possible.

A fine-tuned model with an acceptable implementation strategy will facilitate the detection of incidents with promising performance. However, it deserves to note that any specific model will have its limited capability and deficiency. Modelers often hope to maximum a model’s capability while reducing its deficiency. Nonetheless, due to a number of factors such as outliers, incomplete information, model assumptions, exclusion of some factors, etc., one can expect that the performance a developed model will be capped at a certain level. The deficiency will remain the same if no further effort was made. For example, ordinary least squares can be applied in linear regression to minimize the sum of the squares of the differences between the observed dependent variable in the given dataset and those predicted by the linear function. The calibrated coefficients

---

Fig. 2. Conceptual Illustration of Incident Occurrence on Highways.
can only help account for a certain amount of variations in the dependent variables (e.g., in terms of $R^2$). Likewise, in the context of incident detection, existing models often cannot achieve a perfect detection result either. The inherent limitations of a deployed model may not be simply addressed through the recalibration.

Instead of tweaking the model again, other approaches that may revisit the modeling results and gather related feedback or combine other processes to learn the failures would be more valuable. This motivated us to expand the detection capability through the use of artificial intelligence (AI) that does not limit itself within the fixed model framework. In other words, AI will be leveraged to imitate human-like behavior that may not be perfectly generalized by a mathematical model. In practice, an operator will learn from historical operations that s/he misclassified. The gradually accumulated lessons will be highly likely to enable the operator to avoid repeating similar mistakes. Instead of purely relying on a model, the operator will refer to the lessons or knowledge s/he learned for double checking the model result. More importantly, s/he can keep updating her/his knowledge while new cases present. Similar evolvement ability to digest and learn things has been frequently used in literature to build various expert systems, which make machines to have artificial intelligence. A well-known example is the IBM Watson question-answer computer system that successfully won the quiz show *Jeopardy!* against human champions in 2011 [97]. Watson maintains information from millions of documents such as dictionaries, encyclopedias to build its knowledge [2, 98].

Inspired by the learning and reasoning ability of those expert systems like Watson, this dissertation expands current incident detection modeling practices by framing an AI approach that assembles a memory unit with a tuned model. Fig. 3 presents the conceptual architecture of the AI approach. In a nutshell, whenever the deployed model (e.g., $M_1$) predicts the occurrence of an
incident, the input data will be stored to the memory unit and will be linked with a label as either correct or false prediction post the verification of the event. For example, the first time when the model makes a false alarm \( R_i \), the input variables \( X_i \) will be included in the memory (i.e., knowledge database) for future comparisons. At later steps, when false alarms \( R_j \) and \( R_k \) were prompted, their corresponding traffic conditions \( X_j \) and \( X_k \) will be further assessed with the reference to the memory. As they are similar to \( X_i \) that was associated with a false alarm, the initial predictions of \( R_j \) and \( R_k \) will be corrected as \( R_j \) and \( R_k \). Thus, despite the false prediction of the detection model \( M_1 \), these two later cases will be labeled as non-incident scenarios and their relevant information will also be further updated to the memory. Referring the memory at each step can be considered as the process of retrieving knowledge. Updating the memory dynamically will be comparable to accumulating the knowledge evolutionarily by the TMC operator.

**Fig. 3.** An Proposed AI Modeling Framework with Memory Units and Learning Ability.
The memory unit described in Fig. 4 consists of a historical knowledge database that archives different types of traffic information. This is similar to a dictionary that depicts different traffic profiles with relevant indexes to incident occurrence (e.g., true/false alarms in incident detection). Depending on the data source, the construction of the traffic profiles in the memory unit can be different. In terms of the typical data from loop detectors, we may gather traffic flow, speed, and occupancy, either by station-level averages or lane-by-lane measurements. Certainly, other derived metrics such as speed/flow variances, correlations, etc. can be further included. These multi-dimensional measurements together will be assembled to represent the snapshot of the traffic condition at a given time period. Fig. 4 illustrates an example of the memory unit that stores the traffic profiles. When a model $M_1$ predicts the occurrence of an incident $R_t^i=1$, the prevailing traffic condition $X_t^i$ will be further compared with the memorized information. If similar traffic profiles were identified, their indexed incident facts (i.e., with or without incidents) will be used to correct the initial prediction from $M_1$. For example, if $K$ traffic profiles in the memory units show great similarity with $X_t^i$, the corresponding incident facts $I_i, I_2, ..., I_k \in \{0=\text{without incident}; 1=\text{with incident}\}$ will be evaluated. If the incident facts show more “0,” it suggests the early prediction by $M_1$ is highly likely to be wrong. Thus, its prediction results $R_t^i=1$ will be updated as $R_t^i=0$. Otherwise, the incident facts with more “1” will further confirm the $R_t^i=R_t^n=1$. This process ensures that the model results will not be solely used as the determinant in decision. This process will help reduce the risk of false prediction. Upon the verification of $R_t^i$, the corresponding $X_t^i$ will also be included in the memory unit for future reference.
As discussed above, a key function of the memory unit is to facilitate the assessment of the current traffic profile. It is used to determine similar historical traffic profiles in each sub-unit. The similarity between the current traffic profile and the historical ones may be determined by a number of approaches such as random forest, support vector machine, and K-nearest neighbors (KNN) approach. Taking the complexity into consideration, the simplest KNN approach is adopted in this dissertation to illustrate how the similarity between traffic profiles can be quantified. For example, if a traffic profile is defined as $X_t(Q, S, O)$, the similarity can be calculated based on the Euclidean distance $D_{ix}$ between two profiles:

$$D_{it} = \sqrt{\sum_{h=0}^{t} [X_{i-t}^{h}(Q, S, O) - X_{t-h}(Q, S, O)]^2}$$

(2)

where, $X_{i-t}^{h}(Q, S, O)$ denotes the current traffic profile $X_t(Q, S, O)$’s element obtained at time $t_{x-h}$; $X_{i-t}^{h}(Q, S, O)$ is the element of the $i^{th}$ historical traffic profile $X_t(Q, S, O)$ measured at...
where, $N_F$ is the number of profiles that produced false predictions and $N_T$ is the number of profiles that produced true predictions, and $N_F + N_T = K$. In practice, an odd number $K$ (e.g., $K = 5$) is suggested so that Eq. (5) will not involve the case that $N_F = N_T$. The above procedure will expand model $M_1$’s capability by keep learning from the historical profiles. Any new profile can be rollover to the memory unit, which keeps the memory unit as fresh as possible. For practices, the number of the historical profiles in the memory unit can be limited (e.g., 2,000 records) to reflect the memory capability of an operator. If the number of historical profiles exceed the limit, the latest profiles will be kept, whereas the out-of-date ones will be phased out so that the memory unit will maintain the latest information.

The AI modeling framework in Fig. 3 requires the implementation of a detection model $M_1$. There are many candidate models that can be used. Rather than the simple regression approaches and rule-based methods, the models that capture the nonlinear relationship between
traffic measurements and incident occurrence are preferred. Under the umbrella of AI, ANN is often sought. This dissertation also uses a simple ANN to demonstrate a specification of $M_i$ to support the AI-based detection framework. It should be mentioned that other methods such as SVM can also be considered. The used neural network only includes three layers, including an input layer, one hidden layer, and the output layer. Mathematically, the ANN model can be written as the follows:

$$X_a^{(c)} = f_{NN} \left( \sum_{b=1}^{b=B} (\omega_a^{(b,c)} \cdot X_a^{(b)}) + \beta_a^{(c)} \right)$$

where, $\omega_a^{(b,c)}$ is the weight parameter that links the $b^{th}$ element $X_a^{(b)}$ in the output of the $(a-1)^{th}$ layer and the element $X_a^{(c)}$ in subsequent layer. $\beta_a^{(c)}$ denotes the bias. $a = 1, 2, ..., a$ represents the index of a layer in the neural network. In the simplest scenario, $a = 3, b = 1, 2, ..., b$ denotes the element index in the preceding layer and $c = 1, 2, ..., c$ is the element index in the following layer. $f_{NN}()$ is the activation function that can be tanh(), maxout(), sigmoid(), etc., for the calculation of a specific layer’s output. For incident detection, the softmax() is used to calculate the probability of incident occurrence as the final output.

The input layer will have the elements in the current traffic profile $X_t (Q,S,O)$. Like the example shown in Fig. 4, this profile can include current and multi-step historical measurements for flow, speed, and occupancy by multiple sensors (e.g., $L_1$ and $L_2$ in Fig. 2) along the target highway section. Considering the variation of these measurements, other than the row sensor measurements, this dissertation also adopted the CUSUM (cumulative sum) control chart to include additional derived metrics in the traffic profile as the input [99, 100]. The ANN model with both the raw sensor data and the CUSUM metrics is named as NN-CUSUM in later analysis.
This dissertation uses the R program to develop the script for testing the approach with collected input data.

The CUSUM is an abbreviation of cumulative summation which depicts the variation of observed factors. Diverse categories of input variables are taken into consideration to format input x. To start with, the input x can be written as \((Q_i^1, S_i^1, O_i^1, Q_i^2, S_i^2, O_i^2)\), since traffic metrics such as flow, speed, and occupancy have been frequently used in previous literature. Q denotes the flow, O is the occupancy, and S represents the speed of downstream (i.e., 1) and upstream (i.e., 2) sensors. Since congestion will lead to potential speed variations, additional derived metrics CUSUM is introduced to depict the extent of speed variation. CUSUM is a cumulative sum control chart used to depict the cumulative sums of the deviations of the observed variables and has been successfully applied in other research areas such as automatic control. Thus, we introduced the derived CUSUM of speed given the following equations:

\[
\begin{align*}
HCS_i^t &= \max[0, x_i - (\mu_0 + k) + HCS_{i-1}^t]; \\
LCS_i^t &= \max[0, (\mu_0 - k) - x_i + LCS_{i-1}^t]
\end{align*}
\]

\(\mu_0\) denotes the mean value, and slack value k is predefined. \(HCS_i^t\) denotes high CUSUM of speed at \(t\) of \(i^{th}\) sensor (1 for upstream and 2 for downstream), while \(LCS_i^t\) represents low CUSUM of speed at \(t\) of \(i^{th}\) sensor. Whenever the speed value is out of the upper limit or lower limit, the variance will be calculated. Speed reduced at a high speed near 800s, and then remain at a stable state with a low mean value. Meanwhile, the peak point of HCS value reflects such speed change pattern, and thus is expected to be a useful information.
3.2 SIMULATION

Ideally, real-world incident and traffic data are expected to train and test the proposed model. A large volume of data is necessary to warrant the effectiveness of an incident detection approach based on machine learning or AI. Although it is not challenging to obtain real-time traffic data from sensor stations on highways (e.g., from Performance Measurement System (PeMS) of Caltrans), publicly available incident data with accurate incident information are scarce. Particularly, the precise incident occurrence time is often not well archived in the database. In addition, the incident frequency of a single segment will be relatively low due to the randomness of the events. This will cause imbalance data issues when preparing the training dataset.

Alternatively, in order to test the performance of our proposed AI approach, this dissertation designed an experiment through microsimulation to have full control over the data collection procedure. The simulation scenarios were fine-tuned to reflect traffic conditions of a typical highway section. Specifically, Fig. 2 illustrates the designed test scenarios. A 4-lane highway section with 65 mph speed limit was coded in VISSIM [101]. The section is a 4-mile straight section. Two sensor stations $L_1$ and $L_2$ with a distance of $DL_{12}$ will be placed in the middle portion of the highway section. We considered 3 levels for $DL_{12}: \{DL_{12} = 0.3\text{miles}; DL_{12} = 0.5\text{miles}; DL_{12} = 1.0\text{mile}\}$. Each sensor station has 4 detectors to continuously collect traffic flow, speed, and occupancy of each lane with time intervals of 30 seconds. An incident was simulated to occur at time $t$ between the two sensor stations, with a distance of $IL_1$ to the upstream sensor station $L_1$. Three levels were considered for $IL_1: \{IL_1 \approx 0; IL_1 = 0.5 \times DL_{12}; IL_1 \approx DL_{12}\}$, where $IL_1 \approx 0$ represents the incident occurred right after passing the sensor station $L_1$, $IL_1 = 0.5 \times DL_{12}$ means the incident occurred at the middle of the segment, and $IL_1 \approx DL_{12}$ denotes the incident occurred right before reaching sensor station $L_2$. 
The incident was simulated to last 20 minutes with either 1 lane or 2 lanes on the curbside blocked. To simulated incident occurring at different traffic conditions, 30 levels of traffic demand were considered: traffic demand varied from 5,100 vph to 8,000 vph, with an incremental of 100 vph. Together 540 experimental scenarios were created: 3 levels ($DL_2$) × 2 levels (Lane closure) × 3 levels ($IL_4$) × 30 levels (Demand) = 540. Each simulation scenario has been run 30 minutes, with first five minutes as the warmup period. The data collection was conducted between $t = 5\text{ min}$ and $t = 15\text{ min}$. This allows to gather data for 5 minutes prior to incident and 5 minutes after the incident. The incident detection approach will be continuously implemented during these 10 minutes. Each scenario was replicated with 10 different random seeds in simulations and half them will be used for training the model and half will be used for testing its detection performance. The final data include the averaged measurements at the sensor station and the lane-by-lane measurements of each detector.

### 3.3 PERFORMANCE CRITERIA

In order to evaluate the performance of incident detection approaches, four measurements that construct the confusion matrix are considered: (i) True Negative (TN): When there is no incident, an algorithm also predicts no incident (Fig. 5(a)); (ii) False Positive (FP): When there is no incident, an algorithm predicts incident occurrence (Fig. 5(b) and Fig. 5(c)); (iii) False Negative (FN): There is an incident but an algorithm predicts no incident (Fig. 5(d)); and (iv) True Positive (TP): An incident occurs and an algorithm predicts its occurrence (Fig. 5(e)). The dotted lines denote predicted incident conditions, and the line denotes for actual incident conditions. In practice, the two cases involving FP predictions are critical as these false alarms will incorrectly report the situations that incident clean efforts. Dispatching resources (e.g., responders) to these
false alarms will greatly increase the incident management cost. Thus, the incident detection algorithm that produces less FP is preferred. In addition, FN is expected to be as small as possible so that the algorithm will not miss many actual incidents. Other than these two concerns, a better algorithm is expected to detect an actual incident as early as possible. Thus, the time to detect the actual incident ($TTD$) is also considered. $TTD$ is defined as the time difference between the actual incident occurrence time and the reported occurrence time by the algorithm:

$$TTD = t_{detect} - t_{incident} \quad (8)$$

Fig. 5. Illustration of Possible Incident Prediction Results.

For comparisons, incident detection algorithms will be implemented in the study period (i.e., 10 min in each simulated scenario). If an algorithm reports the occurrence of an incident (either TP or FP), it will be terminated for evaluation in the remaining period. Otherwise, it will be run until the end of the study period. This will allow all algorithms under evaluation to have the same time horizon and fair testing scenarios. Other than the $TTD$ for each tested scenario, we
calculate two frequently used indicators to quantify the overall performance of compared incident detection algorithms. These indicators included detection rate (\(DR\)) and false alarm rate (\(FAR\)):

\[
DR = \frac{\#TP}{\#Actual\ Incidents} \times 100\%
\]

\[
FAR = \frac{\#FN}{\#Tested\ Cases} \times 100\%
\]

where, \(DR\) is the ratio of the total number of correctly detected incidents (TP) to the total number of actual incidents. This indicator reflects the accuracy towards detecting actual incidents. A greater value of \(DR\) suggests that the algorithm is capable of reporting more incidents post their occurrence. \(FAR\) is the ratio of the false alarm cases to the total number of tested cases. Algorithms with smaller \(FAR\) will receive more preference. These performance indicators are reported based on each level of sensor spacing \(DL_{12}\). More specifically, in this study, the total number of tested cases at a given level of \(DL_{12}\) is:

\[
\#Tested\ Cases = 2\ levels\ (Lane\ closure) \times 3\ levels\ (IL) \times 30\ levels\ (Demand) \times 5\ (Random\ seeds) = 900
\]

Since our experiment does not simulate incident free scenarios, the total number of the actual incidents will be the same as the total number of the tested cases at a given level of \(DL_{12}\). If some simulation scenarios without incidents were added, these two numbers will be different. The average \(TTD\) for each scenario running with five random seeds is calculated and the overall average \(\overline{TTD}\) at a given level of sensor spacing \(DL_{12}\) is also computed.

3.4 RESULTS & ANALYSIS

The designed experiment has been implemented in VISSIM to collect all the raw data. The output data from the loop detectors as well as the derived traffic metrics were organized to meet the input need of each incident detection model. As mentioned earlier, 50% of the data were used
<table>
<thead>
<tr>
<th>Lane Closure</th>
<th>Method</th>
<th>Detection Rate (DR: %)</th>
<th>False Alarm Rate (FAR: %)</th>
<th>Time to Detect (TTD: s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$DL_{x2}=0.3m$</td>
<td>$0.5m$</td>
<td>$1.0m$</td>
</tr>
<tr>
<td>1 Lane</td>
<td>CA#7-1</td>
<td>84.0</td>
<td>61.6</td>
<td>46.0</td>
</tr>
<tr>
<td></td>
<td>CA#7-2</td>
<td>76.2</td>
<td>56.7</td>
<td>42.9</td>
</tr>
<tr>
<td></td>
<td>ANN</td>
<td>99.5</td>
<td>97.3</td>
<td>77.8</td>
</tr>
<tr>
<td></td>
<td>AI –Avg.</td>
<td>99.7</td>
<td>98.4</td>
<td>78.0</td>
</tr>
<tr>
<td></td>
<td>AI -Lane</td>
<td>99.7</td>
<td>98.2</td>
<td>82.4</td>
</tr>
<tr>
<td>2 Lanes</td>
<td>CA#7-1</td>
<td>90.9</td>
<td>78.7</td>
<td>67.3</td>
</tr>
<tr>
<td></td>
<td>CA#7-2</td>
<td>85.6</td>
<td>78.0</td>
<td>65.8</td>
</tr>
<tr>
<td></td>
<td>ANN</td>
<td>99.3</td>
<td>97.1</td>
<td>93.3</td>
</tr>
<tr>
<td></td>
<td>AI –Avg.</td>
<td>99.8</td>
<td>98.4</td>
<td>93.6</td>
</tr>
<tr>
<td></td>
<td>AI -Lane</td>
<td>98.9</td>
<td>97.8</td>
<td>98.9</td>
</tr>
</tbody>
</table>
to train each model and the other 50% were used to test the tuned model. For CA#7, the optimum threshold of the three thresholds was numerically searched within the range of [0,1], with an interval of 0.05. Specifically, two models, including CA#7-1 and CA#7-2, are tuned with the aim of minimizing FAR and TTD, respectively. Additional selection criteria, such as the total number of FAs must be less than 150 and TTD cannot exceed 180s, are applied. The other models were also tuned following the necessary calibration steps. For comparison, we also include the classical ANN model built on the use of averaged sensor data. ANN approach used one hidden layer with 20 neurons and AI-lane model used one hidden layer with 40 neurons. All models were trained and tested with data aggregated in 30s time intervals.

In order to understand the sensitivity of the proposed approach with respect to traffic volume changes, incident locations, and sensor spacing, we further investigated TP, FP, and TTD under each simulated scenario. Due to space limitation, more results are provided in our paper [19]. As shown, these performance metrics will be affected by the prevailing traffic flows, sensor spacing, and locations. In particular, when an incident occurred in the middle of two loop detectors, it will be challenging for all approaches. Nevertheless, the proposed AI-based approach with lane-based data will be able to capture more incident cases with relatively shorter TTD even under lower volume conditions. When the sensor spacing increases, each model tends to promote more FN predictions. However, among the compared algorithms, the proposed approach still produced better results in terms of FN and shorter TTD.

Overall, the proposed AI-based approaches that either use lane-based detector data (AI-Lane) or the averaged sensor data (AI-Avg.) outperforms the CA#7-1 and CA#7-2 in terms of the DR, FAR, and TTD. Although the TTD between the AI-based approaches are close to the results of ANN model, comparatively their DRs are lower and FARs is higher. This suggests the benefits
of including the learning structure in the proposed detection framework as it can help correct some false alarms from initial neural network algorithm.

3.5 CONCLUSIONS

The obtained results show that the proposed AI-based framework using either lane-based data or station-level average data performed relatively better than the benchmark approaches, including CA#7 and ANN. The augmented performance of the proposed approach has been primarily demonstrated in terms of the shorter detection time, lower false alarm rate, and higher detection rate given the simulated data. The presented results show the improved performance of the proposed AI-based framework regardless of the sensor spacing.

The current study emphasizes the architecture of the AI-based framework. We did not focus on assessing whether the involved individual components (i.e., NN and KNN in our proposed AI-based framework) will be superior to other models such as SVM or deep-learning approaches. These integrated algorithms can be replaced by other models without changing the architecture of the proposed framework. Due to the inherent limitations of the simulation models, we can expect that simulated data might not fully reflect the more complex field situations. Accurate modeling of incidents in simulation models itself is a challenging task. This study was a first step towards the applications of AI technology for better performance of incident detection.
CHAPTER 4
INCIDENT IDENTIFICATION WITH REDEFINED EVENT LABELING

This chapter discusses relabeling events in the training dataset for the enhanced incident identification. Simulations and results are provided to verify the impact of enhanced incident relabeling.

4.1 PROPOSED FRAMEWORK FOR RELABELING EVENTS IN TRAINING DATASET

As shown in Fig. 6, an incident occurs at time T4, with a distance of \( IL_1 \) to upstream loop detector \( L_1 \). The incident is induced by a collision between two vehicles represented by red dotted rectangles. The correspondingly 2-lane closure leads to a change of traffic metrics (e.g., the speed drop and a congestion queue) in the road segment during the following time period. TMCs sought to detect the change as soon as possible via observing upstream and downstream loop detector measurements. Based on the premise that an incident will lead to an observable traffic pattern change, the detection of changes is equivalent to TID issues, and can be generalized the equation (1) using \( X \) and \( Y \).

The perfect TID performance relies on the wisely selected mapping function \( f() \), accurate \( X \), and \( Y \). However, achieving such a peak performance is challenging in the real world from several aspects. Firstly, a primary concern is how to wisely select and tune \( f() \). As mentioned before, numerous approaches such as ANN and SVM can be considered as the mapping function \( f() \). Different models may require different computational efforts and result in different TID performances. Researchers need to carefully make the trade-off.
Another concern is about the accuracy of input data $X$. Ideally, $X$ is expected to be collected with a set of detectors with a small gap $DL_{x2}$, a short time interval, and a large number of traffic metrics. For example, if loop detectors are allocated along the road segment with an interval of 0.1 mile and record measurements every second, the fluctuation of $X$ induced by an incident can be promptly observed. In addition, TID algorithms can benefit from other types of input data including weather information, road pavement conditions, driver behaviors, and images of surveillance cameras. However, it is expensive and impractical to prepare such ideal input data $X$ in the real world. For example, existing literatures show that the time interval is 30s for the major TID application of TMCs, and the loop detector gap is approximately 1 mile. Meanwhile, the most frequently used data are sourced from loop detector measurements [102].
Last but not least, $Y$ serves as the benchmark for selecting and evaluating $f()$, and needs to be concerned about. It is impractical to assign TMC operators ubiquitously to report incidents every minute. Therefore, the reported $Y$ are often subject to bias and noise. For example, an incident occurred on the highway at midnight and did not lead to any heavy congestion. TMCs did not have reports about such incidents, and $Y$ during that period will be labeled as incident free. On the other hand, if an incident occurred and was later reported to TMCs via phone calls, there existed the gap between the reported time and the actual occurrence time. In addition, after the incident occurrence, it takes time for the impact of an incident to propagate to upstream and downstream loop detectors. Thus, the observed $Y$ often differs from actual $Y$. It is unsuitable to directly map such observed $Y$ with $X$. For example, as shown in Fig. 6, at time step T1, the prevailing traffic condition is incident free, and upstream and downstream loop detectors continuously record traffic metrics of volume, speed, and occupancy. $Y = 0$ and is matched with $X$ composed by Q, O, and S measured by $L_1$ and $L_2$. Later, the incident occurs at time step T4, but it takes time for its impact to propagate to $L_1$ and $L_2$. Thus, $Y = 1$ according to the incident report, while the relevant $X$ is similar to those of previous time steps. It is unreasonable to tune $f()$ with similar input data $X$ and different output labels $Y$. Then, in time step T6, the impact of this incident propagates to $L_1$ and $L_2$. On one hand, looking at measurements of downstream loop detector $L_2$, O and Q decrease from 0.2, 16 veh/min to 0.1, 8 veh/min respectively while S slightly fluctuates from 60 mph to 58 mph. On the other hand, O increases from 0.2 to 0.5, while Q and S respectively decrease from 15 veh/min, 61 mph to 6 veh/min, 32 mph measured by upstream loop detector $L_1$. Such a change of loop detector measurements implies the impact of an incident and should be matched with $Y = 1$. 
To augment TID algorithms and achieve higher performances, aforementioned issues about $f()$, $X$, and $Y$ should be carefully addressed. Previous literatures have examined different $f()$ and investigated their benefits and shortcomings but have not systematically investigate issues about $X$, and $Y$. Thus, a comprehensive study is highly expected to thoroughly explore their impacts and relevant solutions.

In this dissertation, our primary concern is about potential issues of $X$ and $Y$. As shown in Fig. 7, the observed input information and labels represented by grey rectangles and circles differ from the actual information and labels denoted by blue rectangles and circles. Actual $X$ record all measurements along the road segment, and are sampled with the loop detector gap and aggregation time interval to obtain observed $X$. Meanwhile, Actual $Y$ is represented by the spatiotemporal incident state matrix, but is discretized to be a observed temporal incident state vector $Y$. The sampling and discretization procedure to archive observed $X$ and $Y$ leads to information loss and errors and affects the performance of tuned $f()$. Thus, it is critical to preprocess the observed incident data to provide valid training dataset for tuning $f()$.

As shown in Fig. 7(a), the speed of $X$ drops after an incident occurrence, and the black color denotes a high probability for incident free given the actual $X$. To contrast, the white color represents a high probability for incident occurrence. However, such actual labels with detailed probability values are unknown since only observed incident reports with incident occurrence times can be accessed. Thus, there exists three kinds of inaccurate mapping between observed $X$ and $Y$ as illustrated in Fig. 7.

The first type of inaccurate mapping is induced by the propagation time of incidents’ impacts. As shown in Fig. 7(b), based on observed incident reports, an incident occurs between T4 and T5. Given the sampled observed input $X$, the actual probability value P4 that it belongs to an
incident free scenario at time step T4 is 0.9, while P5 equals to 0.88. The difference between P4 and P5 is only 0.02. However, since such unknown actual probability values are discretized and represented by observed incident report \( Y \), observed loop detector measurements at time step T5 will be labeled as \( Y = 1 \) denoted by the triangle. Meanwhile, the actual probability value P6 is 0.5, and thus time step T6 should be considered as the starting point based on observed \( X \). The potential reasons which lead to the gap between the actual purple line and reported red line include: the detector gap is relatively large such as 2 miles, the aggregation time interval is as long as 5 minutes, and the incident happens with a light traffic flow such as 3,000 vph on a 4-lane highway.

As shown in Fig. 7(c), the second type of inaccurate mapping occurs when incidents are identified by observed reports but there does not exist any variation in observed \( X \). As shown in Fig. 7(c), P5 =0.88, P6=0.87, and P7=0.9. Three triangles represent observed \( X \) labeled as \( Y = 1 \). A typical case can be that a trivial incident occurs at midnight and is later reported to TMC operators via phone calls. The incident only affects a few passing by vehicles and their relevant actual \( X \) and is cleaned soon. However, the sampled and observed \( X \) in upstream and downstream loop detectors remains stable. Such actual probability values of incident free are large and relevant, observed \( X \) should not be labeled with \( Y = 1 \) during the tuning process of TID approaches. Due to the lack of necessary information, it is expected that all TID models should not be capable to predict such incidents. Last but not least, the third type of inaccurate mapping exists when events (e.g., lane change behaviors of heavy trucks) potentially lead to abnormal congestions for a short time period. As shown in Fig. 7(d), the actual probability value of incident free in T3, T4 are respectively 0.4 and 0.3. Such anomalies of observed \( X \) are similar to those of \( Y = 1 \). However, based on observed \( y \), such observed \( X \) should be labeled as \( Y = 0 \) represented by circles.
In order to tackle aforementioned three types of inaccurate mapping between observed $X$ and $Y$, this study proposes a framework to relabel observed $Y$ to better match with relevant $X$. As shown in Fig. 8, conventional TID models match observed $X_1$ under different scenarios (e.g., different time intervals and loop detector gaps) with incident labels $Y_1$, and tune diverse TID models such as CA#7, ANN, and SVM. Then, testing dataset of loop detector measurements $X_2$ is used as the input of tuned models to predict incident occurrences. The prediction result $\tilde{Y}_2$ is compared with observed $Y_2$ to calculate the TID performance. By introducing relabeling strategies such as clustering approaches, $Y_{new1}$ is estimated to replace $Y_1$. It is expected that TID models will benefit from the relabeled dataset during the tuning procedure and make a more accurate incident prediction result $\tilde{Y}_{new2}$. The accordingly TID performance $B$ is expected to outperform the previous $A$. 

Fig. 7. Issues in Labeling Incidents.
4.2 SPECIFICATION OF RELABELING STRATEGY WITH FCM

The above framework requires a specific relabeling strategy for implementation. Such actual continuous probability values are unknown and need to be estimated. Based on the assumption that an incident occurrence will lead to an observable anomaly in loop detector measurements, unsupervised learning approaches can serve as the promising solutions. Such approaches have been successfully used for decades in fields including image processing and pattern recognition. The typical unsupervised clustering algorithms divide a set of given data into groups, or clusters, such that all data in the same group are similar to each other, while data from different clusters are dissimilar. This is in accordance with the assumption that loop detector measurements with $Y = 1$ should be similar and different from those with $Y = 0$.

In order to estimate actual hidden probability values and accordingly $Y$, we apply FCM algorithms as a specific case study to classify observed $X$ into two groups: incident free ($Y = 0$) and incident occurrence ($Y = 1$) [35]. It should be noted that FCM can introduces potential bias when estimating unknown actual incident membership function values. However, due to that it is
impractical to access such values, we regard the classification result of FCM as the benchmark for training TID models. In addition, since we know that incidents only exist after reported incident occurrence times, those loop detector measurements labeled as $Y = 1$ before reported incident occurrence time via FCM will be relabeled as $Y = 0$. This restriction rule aims to reduce the third type of inaccurate mapping issue mentioned in Fig. 7(d).

In FCM, with the fuzzy logic, loop detector measurements are classified into two clusters. For example, the point can be assigned to the first cluster with the degree of membership $M_{i1}$ and to the second cluster with the degree of membership $M_{i2}$. If there are $K$ clusters, all membership values satisfy the constraint that the summation of values belonging to different clusters equals to one. Computationally, the fuzzy partitioning process is carried out through an iterative optimization of the utility function, with the update of membership function values and cluster centroids. Since we only consider scenarios to be under incident affected and incident free conditions, $K$ is set to be 2. The objective function is calculated as follows:

$$\arg\min_{M} \sum_{i} \sum_{j} M_{ij} \|x_i - C_j\|^2$$

(12)

$$M_{ij} = \frac{1}{\sum_{j=1}^{K} \left( \frac{|x_i - C_j|^m}{|x_i - C_{j'}|^m} \right)^{\frac{1}{m-1}}}$$

(13)

where, $X$ represents observed input variables including $Q$, $O$, and $S$; $C$ denotes the list of cluster centroids; and $m$ is a predefined fuzzier parameter, which is commonly set to be 2. For each incident report, loop detector measurements before and after the reported incident occurrence time will be grouped and clustered. It should be noted that this study does not put all loop detector measurements with different incident reports together for clustering. This is because observed
traffic metrics can vary under different scenarios such as heavy or light traffic flow, severe or minor incidents.

In order to address the aforementioned second type of inaccurate mapping in Fig. 7(c), using static threshold values such as variation of speed values can serve as a solution determine the suitable $K$. For example, if traffic metrics remain stable with a variation lower than 10 percentage, $K=1$ and all observed $X$ should be labeled as $Y=0$. However, it will introduce new issues on defining suitable threshold values (i.e., which percentage) under different scenarios. Thus, this study introduces BIC values to automatically determine whether observed $X$ matched with an incident report should be considered as one cluster or two clusters. BIC values are the approximation to integrated (not maximum) likelihood, and the model with the greatest integrated likelihood is desired. Gaussian finite mixture models are used to calculate BIC values and determine whether the suitable cluster number $K$ should be 1 or 2 [65].

4.3 EXPERIMENTAL DESIGNS

Ideally, incident reports and loop detector data in the real world are expected to be used to tune TID models and test the impact of inaccurate labels in incident database. Although it is available to archive public online data from sources such as incident records sourced from New Jersey Department of Transportation and Performance Measurement System (PeMS) Data, there exist several inherent limitations when using such data. Firstly, public real-world data are prone to accuracy issues. For example, loop detectors will record traffic prevailing conditions with observation errors, and incident reports tend to record the incident occurrence time with an unknown delay. Secondly, incident frequency is relatively low, and thus it is impractical to expect
a sufficient large training dataset to well cover incidents and loop detector measurements under diverse scenarios.

In order to evaluate the impact of observed $X$ and $Y$ under different scenarios, this study designed an experiment to tackle the aforementioned accuracy and sparse issues. As shown in Fig. 2, a 4-lane highway road segment with 65 mph speed limit is implemented in VISSIM. Numerous simulation scenarios were fine-tuned. Two loop detectors $L_1$ and $L_2$ were placed in the middle portion of the highway section. The gap between $L_1$ and $L_2$ has 2 levels: $DL_{12} = 0.5$ miles and 1 mile. Loop detectors collected traffic flow, speed, and occupancy with a time interval of 30s. The simulated incident occurred at 10 minutes between the upstream loop detector $L_1$ and downstream loop detector $L_2$. The distance between the incident’s location and $L_1$ was represented by $IL_1$, and had three levels: $IL_1:\{IL_1 \approx 0; IL_1 = 0.5 \times DL_{12}; IL_1 \approx DL_{12}\}$. $IL_1 \approx 0$ means that the incident occurred near $L_1$; $IL_1 = 0.5 \times DL_{12}$ denotes that the incident occurred at the middle of the road segment; and $IL_1 \approx DL_{12}$ represents that the incident occurred near $L_2$. To cover different severities of incidents, 1-lane closure and 2-lane closure due to the incident were simulated. Meanwhile, to simulate incidents under light and heavy traffic flows, the traffic demand varied from 5,100 vph to 8,000 vph with an incremental of 100 vph. Thus, in total 360 unique scenarios were simulated: $2 \times 2 \times 3 \times 30 = 360$. Each unique scenario had been run 30 minutes and replicated with 10 different random seeds, while $L_1$ and $L_2$ collect data between 5 and 15 minutes. Thus, the dataset is balanced and composed of data from 5 minutes prior to an incident to 5 minutes after an incident. Among 3,600 simulated scenarios (360×10=3,600), half of them are used as the training dataset for three different models: CA#7, ANN, and SVM, while the others are used for testing TID performance. When examining the
impact of relabeling incidents, the FCM approach only preprocesses 1,800 training scenarios for tuning TID approaches. The remaining testing dataset still use the original unlabeled incident reports to calculate TID performances.

In this study, five evaluation metrics are considered: TA, FA, DR, FAR, and TTD: (a) TA, i.e., TP, denotes the number of correctly detected incidents. For example, an incident occurs at 10 minutes, and TID approaches predict the incident occurrence after 10 minutes. This will be counted as a TA case; (b) FA, i.e., FP, denotes the number of falsely predicted incidents. For example, TID approaches predict the incident occurrence before 10 minutes while no incident actually occurs during the time period. This will be regarded as a FA case; (c) DR means the ratio of TP to the total number of tested incident cases; (d) FAR represents the ratio of FP to the total number of tested incident cases; and (e) TTD is the time between incident detection confirmation and reported incident occurrence time. For example, an incident occurs at 10 minutes, and is confirmed at 11 minutes. Thus, the TTD is 11-10=1 minute. Greater values of TA, and DR imply a good TID performance since the algorithm can predict the majority of incidents post their occurrences. On the other hand, smaller values of FA, and FAR are ideal from the perspective of TMCs. Once a FA occurs, TMCs need to dispatch resources which are actually wasted, and thus suffer from unnecessary and high incident management cost. In addition, a smaller TTD is expected since it means that TID approaches can promptly detect incidents.

For comparison, TID algorithms are implemented during the collected 10-minute data collection period in each simulated scenario. Once an incident is reported (either TA or FA), TID algorithms will be terminated. Otherwise, TID algorithms will run for the whole 10-minute period. Only under TA cases, TTD will be recorded, and the final average TTD value is calculated for the comparison of TID performances. Meanwhile, to guarantee that all TID approaches will be
evaluated with the same criteria, original incident reports of the testing dataset are used for calculating TID performances with both original and relabeled training datasets. Thus, it should be noted that there exist some potential unexpected TAs. For example, when an incident occurs under a light traffic flow, and does not lead to any observable traffic condition change, TID approaches are expected to be incapable of detecting such an incident. However, TID approaches such as CA#7 can occasionally predict such an incident by sacrificing the TID and FA performances under other similar scenarios. In this study, we still consider such an unexpected detection as a TP.

### 4.4 RESULTS AND DISCUSSION

As mentioned before, an incident occurrence will not necessarily lead to an observable congestion reflected by upstream and downstream loop detector measurements. As shown in Fig. 9, an observable speed fluctuation exists when $IL_1 \approx DL_{12}$. The loop detector gap $DL_{12} = 0.5$ miles, and traffic demand is 5,100 vph. The speed drops from 57 mph to around 20 mph after an incident occurrence reported at 600s represented by the red vertical line. One lane is blocked due to the incident. Accordingly, FCM membership function values of incident free decreases from 1 to 0 after 600s. The relabeled information of the incident via FCM approaches is in accordance with the original label information via the incident report. However, when $IL_1 = 0.5 \times DL_{12}$, the propagation of the impact of incidents takes time, and specifically, the congestion queue cannot arrive at downstream loop detector $L_2$. Thus, there does not exist an observable speed change based on observed measurements of $L_2$. Speed remains stable near 60 mph while FCM membership function values of incident free is 1. Incidents actually are not expected to be detected and such observed $X$ are labeled as incident free in the training dataset. This will efficiently
reduce the number of aforementioned unexpected TPs. Training errors due to misleading information would be alleviated, and TID performances are expected to be improved.

Fig. 9. Relabeling Training Dataset via FCM Approaches.

The designed experiment and comparison have been implemented in VISSIM and R, respectively, to collect and process all raw data needed. The data (flow, occupancy, and speed) from loop detectors are organized to fit input requirements of CA#7, ANN, and SVM models. 50% of the dataset is selected to tune models, while the remaining part is used to test relevant TID performances. Optimum thresholds of CA#7 models are numerically searched ranging from 0 to 1, with an interval of 5%. Since ANN and SVM models are optimized based on the highest accuracy, i.e., the detection rate, the selection criteria of CA#7 thresholds is the highest DR. An additional restriction that TTD cannot exceed 180s is also applied. For example, the calibrated thresholds $Th_1$, $Th_2$ and $Th_3$ for CA#7 approach are respectively 0.05, 0.2, and 0.25 when $DL_{12} = 1$ mile.

The sensitivity of three TID approaches regarding traffic volume changes, incident occurrence locations, sensor spacing are illustrated in terms of TP, FP, and TTD. Due to the limited space, we only show the heat map under scenarios with 2-lane closure. Color codes are provided; blue denotes for well TID performances, while red represents for bad TID performances. Each cell
represents the count of relevant metrics given certain volumes, loop detector gaps, and TID approaches. Overall, with a larger loop detector gap and lower traffic volume, TID performance tends to be inferior compared with that of a smaller loop detector gap and higher traffic volume.

In addition, incidents of $IL_1 = 0.5 \times DL_{12}$ hold a longer average TTD compared with incidents of $IL_1 \approx DL_{12}$ and $IL_1 \approx 0$.

The number of FPs is one of the top priorities for TMC operators. Both CA$\#7$, ANN, and SVM approaches under numerous incident scenarios obtain a reduced number of FPs given relabeled training dataset. This is due to that inaccurate labels within raw data have been correctly relabeled, and thus incident detection models will not hold the similar inputs but produce different outputs, especially when the impact of an incident does not propagate to upstream and downstream loop detectors. For example, incidents (5,100 vph, 1-lane closure, and $DL_{12} = 1$ mile) actually do not result in observable traffic metric changes reflected by upstream and downstream loop detector measurements. However, according to the reported incident occurrence time at 600s, TID models are sought to make predictions of incident free and incident occurrence given similar inputs before and after 600s. This will result in that tuned TID models are prone to make false alarms given similar inputs before 600s in testing dataset. As shown in Table 5, the relabeled data eliminates such cases, and thus FARs of ANN models are respectively 11.1% and 12% with $DL_{12} = 0.5$ and 1 mile but reduce to 5.6% and 6.4% using redefined labels.

It should be noted that CA$\#7$ is inherently limited by the linear structure to compare upstream and downstream loop detectors’ occupancy data, and thus its TID performance significantly degrades when $IL_1 \approx DL_{12}$. Meanwhile, ANN and SVM use speed, occupancy, and volume instead of only using occupancy information. Thus, TID performances of ANN and SVM degrade when $IL_1 = 0.5 \times DL_{12}$. ANN and SVM can benefit from the relabeled data for 1- and 2-
lane closures, and different sensor spaces, while CA#7 cannot achieve the improved performance when $DL_{12} = 1$ mile. The main reason is that incidents need relabeling mainly distributed on scenarios when $IL_4 = 0.5 \times DL_{12}$, while the calibration of CA#7 thresholds is dominantly affected by the imperfect performance when $IL_4 \approx DL_{12}$. On the other hand, ANN and SVM are machine learning approaches that assume nonlinear mapping relationships between incident occurrence and loop detector measurements, and can well tackle the high number of FAs and low number of TAs, especially when $IL_4 = 0.5 \times DL_{12}$. Last but not least, the increment of TTD after relabeling is reasonable considering the tradeoff between FAR and TTD. For example, whenever there exists a slight congestion reflected by loop detector measurements, an incident is detected by the designed approach. This will result in a high FAR since slight variations of loop detector measurements are prone to be falsely predicted as incidents.
### TABLE 5. AVERAGE PREDICTIVE PERFORMANCE BEFORE AND AFTER REDEFINED EVENT LABELING WITHOUT SPECIFYING INCIDENT LOCATIONS AND TRAFFIC DEMANDS

<table>
<thead>
<tr>
<th>Lane Closure</th>
<th>Method</th>
<th>Detection Rate (DR: %)</th>
<th>False Alarm Rate (FAR: %)</th>
<th>Time to Detect (TTD: s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$DL_{12}=0.5\text{mi.}$</td>
<td>$DL_{12}=1.0\text{mi.}$</td>
<td>$DL_{12}=0.5\text{mi.}$</td>
</tr>
<tr>
<td>1 Lane</td>
<td>CA#7</td>
<td>62.0</td>
<td>52.8</td>
<td>10.0</td>
</tr>
<tr>
<td></td>
<td>CA#7(R)</td>
<td>67.1</td>
<td>52.8</td>
<td>4.7</td>
</tr>
<tr>
<td></td>
<td>ANN</td>
<td>88.0</td>
<td>83.1</td>
<td>11.1</td>
</tr>
<tr>
<td></td>
<td>ANN(R)</td>
<td>92.4</td>
<td>85.6</td>
<td>5.6</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>89.8</td>
<td>87.8</td>
<td>9.8</td>
</tr>
<tr>
<td></td>
<td>SVM(R)</td>
<td>91.3</td>
<td>86.4</td>
<td>5.6</td>
</tr>
<tr>
<td>2 Lanes</td>
<td>CA#7</td>
<td>82.2</td>
<td>75.3</td>
<td>8.7</td>
</tr>
<tr>
<td></td>
<td>CA#7(R)</td>
<td>83.1</td>
<td>75.3</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>ANN</td>
<td>87.6</td>
<td>82.4</td>
<td>12.0</td>
</tr>
<tr>
<td></td>
<td>ANN(R)</td>
<td>93.6</td>
<td>84.0</td>
<td>6.4</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>89.7</td>
<td>87.1</td>
<td>10.0</td>
</tr>
<tr>
<td></td>
<td>SVM(R)</td>
<td>94.4</td>
<td>94.9</td>
<td>5.6</td>
</tr>
</tbody>
</table>
4.5 CONCLUSIONS

This chapter introduced an unsupervised learning approach, FCM, to relabel incident occurrence times and further examined its impact on three different incident detection approaches (CA#7, ANN, and SVM). In order to better automatically relabel three types of inaccurate mapping between reported incident occurrence times and loop detector measurements, BIC values and additional restriction rules are applied. The proposed relabeling strategy is evaluated given 360 simulated incident scenarios. The results showed that the relabeling strategy can help improve the performance of three TID approaches in terms of a higher DR and a lower FAR. Comparisons of TID performances under different scenarios also imply that TID approaches need more efforts to predict incidents with a longer loop detector gap, the incident occurrence in the middle between two loop detectors, and under a lower traffic volume condition. In addition, more improvements were obtained through relabeling strategies for machine learning approaches (i.e., ANN and SVM).
CHAPTER 5
SECONDARY INCIDENT IDENTIFICATION

This chapter discusses the secondary incident identification based on probe vehicle data. Three advanced machine learning algorithms are introduced to examine the proposed framework for secondary incident identification.

5.1 OVERVIEW OF THE PROPOSED APPROACH

Instead of using data from infrastructure-based sensors (e.g., inductive-loop detectors), this dissertation introduces a methodological framework that exploits the ubiquitous probe vehicle data for identifying secondary incidents. The proposed framework consists of three major components: (a) the detection of the impact area caused by a PI, (b) the estimation of the boundary of the impact area, and (c) the identification of secondary incidents within the impact area. The proposed approach only needs the probe data points. The exclusion of the need for detailed incident records (e.g., severity) as well as traffic information (e.g., lane closure) makes it more practical for implementation. The use of individual vehicles’ probe data and the proposed identification algorithm aims to identify secondary incidents more reliably. Fig. 10 shows the flow chart of the proposed approach. The following sub-sections describe the details of the proposed approach.
5.2 DETECTING THE IMPACT AREA

The core connection between a PI and a SI is the impact area induced by the PI. Given the occurrence of a PI, the incoming vehicles may slow down. The velocity reduction can vary due to incidents with different magnitudes of impact. If all lanes were closed, all vehicles must stop. Otherwise, fewer vehicles may pass the incident scene but with lower speeds (depending on many factors, e.g., temporary speed control). Moreover, day-to-day traffic conditions are different. Consequently, a fixed speed (e.g., 40 mph) to define the congestion threshold is not reliable. For example, a PI occurred in the traffic flow of 65 mph (e.g., off-peak hours), it may reduce the speed to 50 mph or less. Likewise, if a PI occurred in the traffic flow of 40 mph (e.g., peak hours), it may
reduce the speed to 30 mph or less. Instead of subjectively defining a reference speed to distinguish the impact area, this dissertation proposes to use clustering methods to capture the traffic pattern changes under incident conditions.

Intuitively, incident occurred on roadways will change regular traffic conditions. The prevailing “regular traffic condition” herein can be a free flow state, congested state, etc. The occurrence of an incident is expected to degrade/deteriorate such a condition by further reducing the speed and/or inducing longer queues. The changed condition is likely to distinguish itself from the pre- or post-event conditions. This naturally leads to the thought of describing the traffic states by different clusters (groups). \( K - \text{means} \) clustering technique can be a viable solution to achieve the goal. As shown in Fig. 11, assume \( Veh_j \) is the \( j^{th} \) probe vehicle and each probe data point is \( TR_{ij} = (t_{ij}, s_{ij}, v_{ij}) \), where \( s_{ij} \) represents the physical location of the vehicle at the time step \( t_{ij} \); \( v_{ij} \) is the speed; \( i = 1, 2, ..., T \); \( j = 1, 2, ..., J \); \( T \) and \( J \) are the total number of time steps and probe vehicles, respectively. We may find the optimum \( K \) centroids such that one probe data point \( TR_{ij} \) is belong to a specific cluster. Assume the \( K \) centroids are denoted as \( C_{k-mean} = \{C_1, C_2, ..., C_K\} \). By clustering different probe vehicle data points, one can summarize the dynamic traffic flow conditions by multiple states (e.g., congested vs. non-congested) that change over time and space.
Fig. 11. Clustering the Probe Vehicle Data Points.

The $K-means$ clustering assumes that every probe vehicle data point belongs to a specific cluster. In other words, we have to have clear criteria to separate traffic flow states. However, the transient traffic flow states changing between free flow conditions and fully jammed condition cannot be easily discretized. For example, it would be difficult to say that a speed of 46 mph represents a non-congested status whereas a speed of 44 mph denotes a congested status. Considering the indistinct borders of different traffic states, we propose to consider the $Fuzzy c-means$ (FCM) clustering technique to group the probe vehicle data points. FCM improves the original $K-means$ method by introducing a fuzzy logic and overcomes the limitations of the local optimum issue [100].

With the fuzzy logic, FCM allows one probe vehicle data point to belong to two or more clusters. For example, the point $TR_{ij}$ in Fig. 11(b) can be assigned to the first cluster with the degree of membership $MS_{ij}^1$ and to the second cluster with the degree of membership $MS_{ij}^2$. If there are $K$ clusters, all memberships satisfy constraint $\sum_{k=1}^{K} MS_{ij}^k = 1$. Computationally, the fuzzy
partitioning process is carried out through an iterative optimization of the utility function, with the update of membership \( MS^k_{ij} \) and the cluster centroid \( C_k \) (\( k = 1, 2, ..., K \)). The major steps to implement the FCM method are presented below.

**Step 1:** set the initial values for \( K \), a real number \( m_F \) (\( m_F \geq 1 \)), and \( MS^k_{ij} \); and calculate the cluster centroids of the probe vehicle data points using the following equation:

\[
C_k = \frac{\sum_{i=1}^{T} \sum_{j=1}^{J} \left[ \left( MS^k_{ij} \right)^{m_F} \times v_{ij} \right]}{\sum_{i=1}^{T} \sum_{j=1}^{J} \left[ \left( MS^k_{ij} \right)^{m_F} \right]}
\]  

(14)

where \( MS^k_{ij} \) is the membership function that represents the degree of \( v_{ij} \) belonging to the \( k \)th cluster. Only the speed is considered as the feature of each point.

**Step 2:** calculate the generalized distance \( d^k_{ij} \) between a centroid \( C_k \) and a probe data point \( v_{ij} \):

\[
d^k_{ij} = \| v_{ij} - C_k \|^2
\]

(15)

**Step 3:** define the utility function as follows:

\[
U^{r}_{FCM} = \sum_{i=1}^{T} \sum_{j=1}^{J} \sum_{k=1}^{K} \left( MS^k_{ij} \right)^{m_F} \times \| v_{ij} - C_k \|^2
\]

(16)

**Step 4:** check whether the maximum allowable iteration \( r_{max} \) is reached or the termination criterion based on the utility change is met:

\[
\left\| U^{r}_{FCM} - U^{r-1}_{FCM} \right\| < \varepsilon
\]

(17)

If the termination criteria are met, use the latest \( U^{r}_{FCM} \) and return \( C_k \) obtained in Step 1 as the final cluster centers. Otherwise, go to Step 5.

**Step 5:** update parameter \( MS^k_{ij} \), and repeat the above steps:
Using above procedure, the degree of each probe vehicle data point belongs to each cluster is computed. These degrees of membership are used as the weighted factors to adjust the distance of each point to a cluster center. The final clusters can be used to represent traffic flow states.

5.3 ESTIMATING BOUNDARY OF THE IMPACT AREA

The clustering methods presented in previous section help distinguish different traffic flow states. If there are congested and non-congested periods, the probe vehicle data points with lower speed can be easily highlighted. Visually, we can see the congested area on the space-time diagram, for example, Fig. 12(a). However, given the limited resources, one can be exhausted to review the evolving space-time diagram for a long period of time. Therefore, an efficient approach that can help capture the impact area automatically is needed.

As shown in Fig. 12(a), if we can obtain the boundary points of the congested area, we can quantitatively describe evolution of the impact area over time and space. Simply, we can filter the points between the upper and lower points at each time step \( t_i \) and obtain the boundary points as shown in Fig. 12(b). In addition, a moving average filter is used to effectively remove the noise spikes of these boundary points.
By connecting all neighboring points, we can build the boundary of the impact area. However, there might be a large time gap between two consecutive probe data points. Thus, the link between these two points might not well reflect the changes of the boundary during these time steps. In order to obtain a better representation of the boundary, we first identify the critical (singular) points that significantly affect the shape of the boundary. These singular points are determined by the rotation angle $\theta$ of each boundary points with respect to its previous neighbor link, see Fig. 13(a) for illustration. $\theta$ can be calculated using the following equation.

$$\theta = \left| \arctan \left( \frac{s_{i+1} - s_{i}}{t_{i+1} - t_{i}} \right) - \arctan \left( \frac{s_{i} - s_{i-1}}{t_{i} - t_{i-1}} \right) \right|$$

(19)
If a point is associated with a rotation $\theta' \geq \max(\theta) \times \tau$, then the point will be selected as a singular point, see Fig. 13(b) for illustration. $\theta$ is a vector of all rotations. $\theta'$, is one of the elements in $\theta$ that helps determine a singular point, for example the circle points in Fig. 13(b). The adjustment factor $\tau \in (0,1)$ determines the size $M$ of these critical points. A large $\tau$ means more singular points, and vice versa.

With all the singular points and the points in between, we introduce three methods to estimate the boundary (edge) between two consecutive singular points. The proposed methods include the multi-stage linear approximation, genetic algorithm-based and ant colony optimization-based approaches, which are described below.

5.3.1 MULTI-STAGE LINEAR APPROXIMATION

With the singular points, one can directly link them to establish the boundary. However, the link might not be representative for all points in between two consecutive singular points (see the points in the rectangular box in Fig. 14(a)). Alternatively, we propose to use least square
regression models to estimate the edge (boundary) $L_h$ between two consecutive singular points $TR_{0}^{h} = (t_{(O)}, s_{(O)}^{h})$ and $TR_{E}^{h} = (t_{(E)}^{h}, s_{(E)}^{h})$. This takes into account the impact of all points between these singular points and the optimum edge will be the one that offers the minimum total residual $\xi$. The estimated regression line function $L_h$ can be described:

$$s^{h} = \beta_{0}^{h} + \beta_{1}^{h} t, \ t \in [t_{(O)}^{h}, t_{(E)}^{h}]$$

where $h$ denotes the $h^{th}$ regression line and $h = 1, 2, ..., H$.

Fig. 14. Multi-stage Approximation for Establishing the Boundary.

Since we are using the regression models, the singular points might not be the intersections of the estimate lines. If a singular point $TR_{E}^{h} = (t_{(E)}^{h}, s_{(E)}^{h})$ is not the intersection of lines $L_h$ and $L_{h+1}$, we can link points $(t_{(E)}^{h}, s_{(E)}^{h}) = \beta_{0}^{h} + \beta_{1}^{h} t_{(E)}^{h})$ and $(t_{(O)}^{h+1}, s_{(O)}^{h+1}) = \beta_{0}^{h+1} + \beta_{1}^{h+1} t_{(O)}^{h+1})$ as a supplementary edge (e.g., the yellow line in Fig. 14(b)). Note that $t_{(E)}^{h} = t_{(O)}^{h+1}$. By connecting the estimated lines and supplementary edges, we can obtain the final boundary like the one shown in Fig. 14(b).
5.3.2 GENETIC ALGORITHM-BASED ESTIMATION

The multi-stage linear approximation in Section 5.3.1 is largely shaped by the number and positions of the critical points. Since the lines were independently estimated, most likely we need to add a number of supplementary edges. However, the vertical supplementary edges (e.g., the yellow line in Fig. 14(b)) might be too long (because of a low market penetration rate of probe vehicles) to represent a reasonable propagation of a shockwave. In addition, the total residuals obtained by summing up the least squares of individual regression lines might not be minimized due to the independent regression processes. As an alternative, this dissertation proposes a genetic algorithm (GA)-based approach to estimate the edges. GA is a stochastic search algorithm which searches over a population of probe vehicle data points to find optimum solutions based on three mechanisms: selection, crossover, and mutation.

As shown in Fig. 15(a), the population is constituted of several chromosomes while each chromosome consists of several probe data points denoted as genes. For example, genes 1 and 2 of chromosome 1 represent the points with relevant time and milepost information chosen to estimate a candidate edge in generation 1. Genes 3 and 4 of chromosome 2 help establish another candidate edge. Children in Fig. 15(b) are generated from the selected parent chromosomes in the first generation with probabilistic mutation and crossover operations. Theoretically, after multiple generations, the final edge will be built by genes of the chromosome with the optimum utility function.
The detailed steps to implement the GA-based approach for estimating the boundary of the impact area are presented below:

**Step 1**: Set up the initial values for the maximum allowable iteration $r_{max}$, a tolerance value $\varepsilon$ for utility function change, the initial chromosomes with a population size of $Z$, and the probabilities $p_c$ and $p_m$ for crossover and mutation, respectively.

**Step 2**: Selection – Choose chromosomes from current generation’s population to be included as the parents in the next generation’s population. First of all, establish the line equation for a consecutive pair of genes (e.g., genes 1 and 2 in Fig. 15(a)) in a given chromosome (e.g., chromosome 1 in Fig. 15(a)). Calculating $\xi_h$, the sum of the residual for all points between the two genes with respect to the established line equation. We can establish $H_z$ line equations for a given chromosome, then calculate the total residual $\xi_z$:

$$\xi_z = \sum_{h=1}^{H_z} \xi_h$$  \hspace{1cm} (21)

Fig. 15. Genetic Algorithm for Estimating the Boundary.
The roulette selection scheme is used to select chromosomes with the probability \( p_z \) proportional to the fitness \( f_z \) of each chromosome:

\[
p_z = \frac{f_z}{\sum_{z=1}^{Z} f_z}, \quad f_z = 1/\bar{z}_z
\]

(22)

The selected chromosomes are stored as the parents for the next generation.

**Step 3**: Crossover – Combining two of the chromosomes (parents) obtained in Step 2 to generate new chromosomes, for example, crossing over genes 2 and 4 to obtain new chromosomes 1 and 2 in Fig. 15(b). Each gene in a parent is uniformly selected for crossover.

**Step 4**: Mutation – The genes in a parent will be mutated by adding a random variation. For example, genes 2, 3, and 4 in Fig. 15(a) were mutated to obtain new genes 4, 3, and 2 in Fig. 15(b). The new genes avoid the permanent fixation and allow the GA algorithm to jump out of the local optimum [102].

**Step 5**: Check the utility function changes and the number of iterations to repeat previous steps or terminate the process.

To perform above steps, pseudo code in Algorithm 1 can be used.

**Algorithm 1.** Pseudo Code for Boundary Estimation based on GA

**Input:** boundary set \( TR_{ij} = (t_{ij}, s_{ij}, v_{ij}) \) with the size of \( M \), number of lines is \( H \)

**Initialization:** Population size \( Z \), chromosome size \( 2H \), crossover probability \( p_c \), mutation probability \( p_m \), maximum generation number \( r_{max} \), and tolerance value \( \varepsilon \) for utility function change

Iteration \( r = 1 \);
While \((r \leq r_{max} \& \varepsilon^r > \varepsilon)\) do:

for each chromosome do:

if \(p_c > rand(0,1)\) do:

crossover new chromosome from parents of generation \(r - 1\)

end if

if \(p_m > rand(0,1)\) do:

mutate new chromosome from parents of generation \(r - 1\)

end if

calculate the residual \(\xi_h\) for each line

calculate the total utility function \(U_{GA}^r = \sum_{h=1}^{H} \xi_h\)

end for

for each chromosome do:

calculate \(p_z\)

if \(p_z > rand(0,1)\) do:

select this chromosome as a parent of generation \(r\)

end if

end for

find the minimum \(U_{GA}^r\) in generation \(r\) among all chromosomes

calculate change of utility function \(\varepsilon^r = |U_{GA}^r - U_{GA}^{r-1}|\)

\(r = r + 1\)

end while
5.3.3 ANT COLONY OPTIMIZATION-BASED ESTIMATION

Ant Colony Optimization (ACO) -based approach is also proposed to estimate the boundary of the impact area [103]. Unlike the two methods discussed in Section 5.3.1 and 5.3.2, ACO-based approach does not require the information such as the singular points and predetermined number of edges. The ACO-based approach was originally inspired by the social insect behavior of ants, such as ant reproduction, foraging, nest building, garbage cleaning and territory guarding. ACO is a population-based optimization technique and can achieve good performance by exploiting the swarm intelligence and imitating the pheromone nature of real ant system. This method has been successfully applied to solve many transportation issues (e.g., the travel salesman problem).

The ACO-based approach is illustrated with the assistance of Fig. 16. As shown in the figure, ants are randomly initialized and each ant walks through the probe vehicle data points. For every visited point, an ant leaves a pheromone signal according to its utility function. The pheromone level of the point will be updated (increased) with the number of visitors. In addition, the pheromone on each point will also naturally disperse with a certain rate. Subsequent ants are more likely to find the point with stronger pheromone as the target for next step (see Fig. 16(b)). The final boundary will be built by the optimum trace of an ant that visited points of strong pheromone. The step-by-step deployment of the ACO-based approach is presented below.

**Step 1:** Set up initial ants with a population size of $n_A$, the maximum iteration $r_{\text{max}}$, pheromone disappear rate $\rho$, enhance parameter $\varphi$, pheromone parameter $\alpha_0$ and heuristic parameter $\alpha_1$. Let $M$ be the number of points along the boundary. Calculating the heuristic matrix $HM$ that has $M$ rows and $M$ columns. It is a reference matrix for ant movement. In this
dissertation we assume that an ant can only move forward (e.g., see Fig. 16(a)). Thus, $HM$ is an upper triangular matrix.

![Fig. 16. ACO-based Algorithm for Estimating the Boundary.](image)

For $HM[m_r,m_c]$, establishing the line equation between the $m_r^{th}$ point and the $m_c^{th}$ point, and calculate the summed residual $\xi_{m_r,m_c}$ for all points between the $m_r^{th}$ point and the $m_c^{th}$ point with respect to the established line equation. Thus, we can obtain $HM[m_r,m_c]$ as follows:

$$HM[m_r,m_c] = 1/\xi_{m_r,m_c}$$ (23)

Likewise, set up the initial pheromone matrix $PM$ which has the same form of $HM$ but with all valid elements to be one.

**Step 2**: Given the $m_v^{\text{steps}}$ point an ant has visited, update the ant’s trace as $RT = \{m_1^v,m_2^v,...,m_v^{\text{steps}}\}$. For example, if ant 1 in Fig. 16(a) performed the third step of movement, then the point it reached will be added to establish its latest trace: $RT = \{m_1^v,m_2^v,m_3\}$. The possible
destination points for next step will be \([m_{\text{steps}} + 1, M]\). Now calculate the probability for choosing each new candidate destination point \(m_{\text{new}}\) according to the pheromone matrix \(PM\) and \(HM\).

\[
p[m_{\text{new}}] = (PM[m_{\text{steps}}, m_{\text{new}}])^{\alpha_p} \times (HM[m_{\text{steps}}, m_{\text{new}}])^{\alpha_h} \tag{24}
\]

where \(m_{\text{new}}\) is the index of the candidate destination point and \(m_{\text{steps}} + 1 \leq m_{\text{new}} \leq M\).

According to the roulette selection scheme, the ant will choose to move to the new candidate destination point with a probability of \(\tilde{p}[m_{\text{new}}]\).

\[
\tilde{p}[m_{\text{new}}] = \frac{p[m_{\text{new}}]}{\sum_{m_{\text{new}} = m_{\text{steps}} + 1}^{M} p[m_{\text{new}}]} \tag{25}
\]

Considering the uncertainty of finding the exact destination, the \(m_{\text{steps}+1}\) point to be visited will be the one that is randomly selected from a buffer area around the candidate destination point.

**Step 3:** Repeat Step 2 until all ants have passed the last point in the searching space.

**Step 4:** For each ant, calculate the utility function \(U^r\) for each ant, record the trace \(RT\), and update the corresponding pheromone matrix \(PM\). \(U^r\) equals to the summed residual \(\xi_{RT}\) for all points in the searching space with respect to the trace.

\[
PM[m^{\text{visits}}, m^{\text{visits}+1}] = (1 - \rho) \times PM[m^{\text{visits}}, m^{\text{visits}+1}] + \varphi / U^r, \quad \text{visits} \in [1, \text{steps} - 1] \tag{26}
\]

**Step 5:** Compare the utility functions of all ants and find the optimum one \(U^r_{\text{opt}}\) at current iteration \(r\). Check the termination criteria based on the maximum allowable iteration \(r_{\text{max}}\) and the tolerance of the update for the utility function. Return to step 2 if the criteria are not met. Otherwise, use the \(U^r_{\text{opt}}\) as the final utility function and its corresponding trace to establish the boundary of the impact area. Algorithm 2 presents the pseudo code for the ACO-based approach.

**Algorithm 2.** Pseudo Code for Boundary Estimation based on ACO
**Input:** boundary set $TR_g = (s_j, t_j, v_j)$ with the size of $M$

**Initialization:** Ant size $n_A$, maximum iteration number $r_{\text{max}}$, pheromone disappear parameter $\rho$, pheromone enhance parameter $\varphi$, pheromone parameter $\alpha_0$, heuristic parameter $\alpha_1$, pheromones matrix $PM$, heuristic matrix $HM$, flexible time range $a_{ACO}$, flexible milepost range $b_{ACO}$, and a tolerance value $\varepsilon$.

Iteration $r = 1$;

**While** ($r \leq r_{\text{max}} \& \& \varepsilon > \varepsilon$) **do:**

**for** each ant **do:**

- **record** points visited: $RT = \{ m^1, m^2, ..., m^{\text{steps}} \}$
- **update** available candidate destination points for next step $[m^{\text{steps}} + 1, M]$
- **for** each candidate destination point $m_{\text{new}}$ between $m^{\text{steps}} + 1$ and $M$ **do:**
  - compute $\tilde{p}[m_{\text{new}}]$ 
- **end for**
- **select** the first point $m_{\text{new}}$ such that $\tilde{p}[m_{\text{new}}] > \text{rand}(0,1)$
- **estimate** a new position of $m_{\text{new}}$ as the final candidate destination for the next movement

$t = t + a_{ACO} \times \text{rand}(0,1)$

$s = s + b_{ACO} \times \text{rand}(0,1)$

**save** current $(t, s)$ as the final position of point $m^{\text{steps+1}}$

**end for**
update the PM matrix

find the minimum utility function $U'_{n.a} = \hat{\varepsilon}'_{RT}$ of all ants

$\varepsilon' = |U'_{n.a} - U'^{-1}_{n.a}|$

$r = r + 1$

end while

5.4 AUTOMATED IDENTIFICATION OF SECONDARY INCIDENTS

Given the estimated boundary of the impact area, the final task is to check whether a later incident is a SI. Visually, if the later one located within or at the boundary, it will be classified as a SI. One can manually review and check the relative position of the incident with respect to the boundary. However, this time-consuming procedure is not practical for a large-scale analysis. Without plotting the boundary and verifying by an analyst, a more efficient method is needed.

With the estimated boundary, determining whether an incident is inside the impact area of the PI is naturally equivalent to the classical Point-In-Polygon (PIP) problem. Thus, we can treat the incident as a point and check whether it is inside a complex polygon (the boundary of the impact area estimated in the previous section). Many existing algorithms can solve the PIP problem. In this dissertation, we propose to use the simple ray casting algorithm as it is simple and efficient. The ray casting algorithm is based on the even-odd rule. For any test point, an imaginary ray crossing the point is established. If the ray intersects with an edge of a polygon, then the count of crossing node increases one. To make it simple, we assume the ray is horizontal. If there are odd numbers of nodes on each side of the test point, then it is inside the polygon. If there are even numbers of nodes on each side of the test point, then it is outside the polygon.
Fig. 17 illustrates how the ray casting algorithm works. The horizontal ray crosses point 
\( B \) in Fig. 17(a) and intersects with two edges of the polygon. Obviously, there is only one (or odd 
#) crossing node on each side of \( B \). This suggests that \( B \) is inside the polygon. In contrast, points 
\( C \) and \( D \) are outside of the polygon as there were even numbers of crossing nodes on each side 
of them. The same algorithm can be applied in Fig. 17(b) and determine that incident \( B \) is a SI 
located inside the estimated impact boundary of the primary incident \( A \), whereas \( C \) is not.

---

**Fig. 17. Identifying Secondary Incidents by the Ray Casting Algorithm.**

Although a ray can be in any direction, for the sake of convenience, we assume it is 
horizontal as mentioned earlier. Now the key becomes the determination of how many crossing 
nodes are on each side of the test point (e.g., incident \( B \) in Fig. 17). Let \( \mathcal{V} \) be the vertex set of the 
boundary and \( \mathcal{V} = \{v_1, v_2, ..., v_H\} \) and \( E_{ab} = (v_a, v_b) \) be one of the edges determined by two vertices 
\( v_a \) and \( v_b \). \( E_{ab} \in \mathcal{E} \), where \( \mathcal{E} \) is the edge set. Each vertex represents the coordinate of a critical 
point at the boundary of the impact area, with \( v_a = (t_a, s_a) \) and \( v_b = (t_b, s_b) \). If the edge \( E_{ab} \) can 
intersect with the ray, the following conditions need to be met:
\begin{align}
(s_a < s_b \text{ and } s_b \geq s_a) \text{ or } (s_a \geq s_b \text{ and } s_b < s_a)
\end{align}

Based on the conditions in the above equation, the coordinate of the crossing node \((t_{node}, s_B)\) is mathematically determined by the following equation:

\[ t_{node} = t_a - \frac{s_a - s_B}{s_a - s_b} \times (t_a - t_b) \]

(28)

If \(t_{node} > t_B\) then the crossing node is on the right side of the incident \(B\). Otherwise, the node is on the left side of the incident \(B\). Thus, by checking all the edges that meet the requirements, we can obtain the total number of nodes on each side of the incident \(B\) and then determine whether it is inside the boundary based on the even-odd rule. The pseudo code for the ray casting algorithm is given in Algorithm 3.

**Algorithm 3. Ray Casting Algorithm for Identifying Secondary Incidents**

**Input:** edge set \(E\) and the vertex set \(\mathcal{V} = \{v_1, v_2, \ldots, v_H\}\) that determine the boundary, and the coordinate \((t_{\text{incident}}, s_{\text{incident}})\) of each incident

**for** each edge \(E_{ab} = (v_a, v_b) \in E\) **do:**

**Initialization:** temporal counters \(P_L = 0\) and \(P_R = 0\)

Compare \(s_{\text{incident}}\) of the incident and \(s_a, s_b\) of the two vertices

**if** \((s_a < s_{\text{incident}} \& \ s_b \geq s_{\text{incident}})\) || \((s_a \geq s_{\text{incident}} \& s_b < s_{\text{incident}})\) **do:**

Calculate the coordinate \(t_{node}\) for the crossing node and check \(t_{node} \text{ vs. } t_{\text{incident}}\)

**if** \(t_{node} < t_{\text{incident}}\) **do:**

Node is on the left side of incident: \(P_L = P_L + 1\)

**else do:**

Node is on the right side of incident: \(P_R = P_R + 1\)
end if
end if

update index for \( a \) and/or \( b \)
end for

if \( P_L \) and \( P_R \) are odd numbers do:

return this incident is a secondary incident

else do:

return this incident is \( NOT \) a secondary incident

end if

5.5 SIMULATION TEST

The proposed approaches were tested based on probe data collected from simulation models developed in the Paramics traffic simulator. The models simulate a hypothetical two-lane highway with a speed limit of 65 mph. The highway is a 10-mile straight section with a demand of 2,400 veh/h. A number of parameters such as reaction time and headway will affect the behavior of simulated vehicles. As a testbed, we assume that both headway and reaction time have a mean value of 1.0 second. To simulate the impact of an incident, incidents (assumed to be incidents) were created using the built-in incident module of Paramics. Each incident was scheduled for their occurrence time, location, and duration. A specific lane will be closed until the clearance of the incident. All simulated scenarios are illustrated in Fig. 18. In total, nine scenarios were simulated based on the duration of the initial incident: \( T_C = \{5\text{ min}, 10\text{ min}, 20\text{ min} \} \). These durations reflect different impacts (e.g., a minor incident removed quickly or a severe one requiring more clearance
Fig. 18. Examples of Simulation Test Scenarios.
time). Note that the scenario in Fig. 18(c) created two independent incidents: the incident $B$ occurred 5 minutes later at the downstream of the incident $A$. This scenario is designed to test a more complicate case where two isolated incidents independently occur, but their impact areas may overlap.

The simulation models enable us to collect the trajectory (coordinates) of each vehicle every 0.1 second. All trajectories together provide the ground truth of the simulated traffic states. Considering the fact that not all vehicles will be probe vehicles in reality, a fraction of the simulated vehicles was randomly sampled as the probe vehicles. Different market penetration rates were considered under each simulated scenario: $MPR = \{5\%, 10\%, 15\%, 20\%, 25\%\}$. The performances of the proposed method can be tested based on each MPR.

### 5.6 RESULTS AND DISCUSSION

The simulation scenarios described in the previous section were implemented in Paramics. Each scenario is designed to be run for 75 minutes, including 15-min warm-up period. The simulated trajectories of all the vehicles passing the segment with the simulated incidents were extracted for further analysis. The proposed methodologies were implemented in Matlab programs.

#### 5.6.1 ESTIMATION OF THE IMPACT AREA

To compare the performance of different algorithms, the scenario with two incidents in Fig. 18(c) is used as the demonstration example. Assume the first incident $A$ occurred at simulation time $t_A = 20\text{ min}$ and it lasted for 20 minutes before the lane was reopened. There was another incident $B$ occurred downstream at simulation time $t_B = 25\text{ min}$ but removed 5 minutes later. For the 1-hour simulation time there were 2,451 vehicles entered the section. In total, the
trajectories of these vehicles consist of 4,448,371 points, which establish the space-time diagram shown in Fig. 19(a). We can clearly see the congested area induced by the incidents. However, the shockwaves that define the boundary of the area are not necessarily straight lines. Fig. 19(b) shows the trajectories of the probe vehicles that were randomly sampled with a hypothetical MPR of 20%. A total of 490 vehicles were randomly selected and their trajectories are composed of 871,563 probe points \( TR_{ij} = (s_{ij}, t_{ij}, v_{ij}) \), where \( i \) and \( j \) are time step and vehicle ID, respectively. Clearly, we can see that the back-of-queue shockwave starts at around 1,200s and the front-of-queue shockwave begins at around 2,400s in this scenario. We can also see intersecting trajectories on the space-time diagram in Fig. 19(b). This is because that vehicles can change to another lane if only one lane was blocked by the incident, with a lower speed passing the incident scene. This suggests that the shockwave speed calculated based on the assumption of the fully jammed condition as did in the early studies will not be reliable.

Instead of calculating the shockwave speeds to define the impact area, we implemented the clustering approaches described in the methodology section. Fig. 20 shows an example of the clustering results. The \( K - \text{means} \) based approach classified the probe data points \( TR_{ij} \) into two clusters according to the speed information of each point. There are 416,330 probe points highlighted in red as the impact area in Fig. 20(a). This cluster has a mean speed of 9.26 mph. The other non-congested cluster in blue color has 455,233 probe points and a mean speed of 65.45 mph. Unlike the discrete classifications of the \( K - \text{means} \) clustering, the FCM-based approach estimates the degree of membership for each point and those points with similar membership values can be grouped together. Users can customize the membership values to partition the points into multiple groups. Fig. 20(b) shows the FCM clustering results. The color code represents the membership values. Since we consider two clusters in our case, we only need a single membership
value (e.g., 0.5) to separate the probe points into two groups: 415,550 points with membership values greater than or equal to 0.5 and other points with membership values less than 0.5. A larger membership threshold means a more conservative estimate of the impact area (see the deep-red area in Fig. 20(b)). The mean speed of the congested area is 8.24 mph. A smaller membership threshold means a more liberal estimate of the impact area (see the deep-red area plus the green/yellow area in Fig. 20(b)). The mean speed of the blue area is 65.75 mph. Due to its flexibility, the FCM-based method is used for further analysis.

Fig. 19. Simulated Vehicle Trajectories.
According to the procedure presented in Section 5.3, the upper and lower boundary points of the congested area (in Fig. 20(b)) at each time step were extracted. In total, 30,690 points were obtained as shown in Fig. 21(a). Since the time step is as short as 0.1s, for certain periods there might be only points continuously selected from an individual vehicle as the boundary points. These points together look like short lines in Fig. 21(a). In order to get a smooth boundary, two steps were performed: (a) if boundary points identified belong to one single vehicle in a period, the mean values of its positions and time stamps were used as the coordinate of a new boundary point; and (b) the moving average technique was applied to smooth out the noisy spikes. After this process, the updated boundary points are presented in Fig. 21(b). The singular points (denotes as
the blue circles) in Fig. 21(b) were estimated based on the rotation angle as described in Section 5.3. These points all have rotation angles that are greater than seventy percent of the maximum rotation angle observed among all boundary points.

As mentioned in Section 2.2, directly assuming that the shockwaves enclose a standard triangle shape is not appropriate. As shown in Fig. 22(a), if the lower vertex of the triangle is the point presented at the maximum time step, many points will be ignored by the triangle formed by the thinner black lines. Likewise, if the lower vertex of the triangle is the point presented at the maximum distance upstream, many (red) points will be excluded by the triangle with sold black lines. Given the fact that the arrivals of vehicles are often non-uniform (e.g., Poisson distributed), the shockwaves speed will vary significantly. This obviously makes the traditional shockwave-based approach insufficient to identify the irregular impact area of an PI.

Fig. 22(b) shows the estimation based on the multi-stage linear approximation method described in Section 5.3.1. It approximates the boundary of the impact area by a polygon. As discussed earlier, the supplementary edges can lead to a few sharp jumps as shown in the Fig. 22
Meanwhile, the performance will be affected by the accuracy of the estimated singular points used to train and build the boundaries. Compared to this linear approximation method, the GA-based approach described in Section 5.3.2 only requires the number of segments. The estimated boundary based on the GA approach is shown in Fig. 22(c). The optimal chromosome can be automatically obtained with the preset of the maximum iteration of 100, crossover probability of 0.8, and mutation probability of 0.2.

The ACO-based approach described in Section 5.3.3 is a totally unsupervised method which does not need any information about the singular points and the estimated boundary is shown in Fig. 22(d). It automatically learns how many segments would best describe the edges and perform well with the maximum iteration of 100. It should be noticed that computationally the ACO-based approach is faster than the GA-based approach during the simulation. This is because the pheromone in the ACO scheme can efficiently eliminate those meaningless searches that are prone to occur in the GA scheme.

In summary, the shockwave-based approaches using the triangle shape is not suitable in practice whereas the multi-stage linear approximation can roughly identify the impact area of the PI. Both GA- and ACO-based approaches are intelligent algorithms that can better approximate the irregular impact area but require little or no additional information.
5.6.2 EVALUATION OF PERFORMANCE

The raw trajectory points are labeled based on the FCM clustering technique with a membership value of 0.5. We randomly selected 10 percent of all labeled points in each scenario as the simulated incident events to test the performance of the proposed approaches in identifying secondary incidents. Each point is either labeled as a congested or non-congested. We evaluated the performance of the proposed approach by checking the confusion matrix and four relevant metrics including accuracy rate, precision rate, true positive rate, and false positive rate were calculated (Note: Positive means a point is in the congested area). The accuracy rate is calculated.
based on the number of points that have been correctly re-identified by the estimated boundary of the impact area. The following equation is used for calculating the accuracy rate $Acc$:

$$Acc = \frac{|SI, NSI \cap SI', NSI'|}{|SI, NSI|} \times 100\%$$

(29)

where $\{SI, NSI\}$ is the set of the congested points and the non-congested points (denoted as non-secondary incidents: NSI ) classified in the sampled trajectory points. Given an estimated boundary of the impact area, $\{SI', NSI'\}$ is the set of the congested points (denoted as secondary incidents: $SI'$) and the non-congested points (denoted as non-secondary incidents: NSI') re-identified from the sample set based on the identification algorithm presented in Section 5.4. In total, ten random seeds were considered to sample the points and the average of the calculated accuracy based on each set of sample points is used to approximate the final accuracy.

The precision rate was calculated based on the number of correctly identified SI divided by the number of total identified SI points. The true positive rate was calculated by the number of correctly identified SI divided by the number of originally labeled SI points. The false positive rate was calculated by the number of original NSI points being classified as SIs divided by the number of originally labeled NSI points.

Fig. 23(a), (b), and (c) illustrate the results for scenarios shown in Fig. 18 (a), (b), and (c) with a primary incident duration of 20 minutes, respectively. The height of each bar in the figure represents the mean value of the accuracy rate while the solid line on each bar represents the standard deviation. Overall, we can see that the GA- and ACO-based approaches outperform the multi-stage linear approximation in re-identifying the trajectory points. Depending on the market penetration rate, the approach based on the multi-stage linear approximation can only correctly re-identify about 75 to 85 percent of the sampled points. A higher market penetration rate helps
improve the accuracy. In contrast, both GA- and ACO-based approaches can achieve an accuracy rate of about 80 to 95 percent. If MPR was 15 percent, both approaches achieved an accuracy rate of about 90 percent. The accuracy of the ACO-based approach is slightly higher than that of the GA-based approach when the MPR is 15 percent or more. These results suggest that an MPR of 15 percent is sufficient to identify the secondary incident with a reasonable accuracy.

Since the multi-stage linear approximation-based approach cannot achieve high accuracy, it was excluded from further comparisons. Table 6 summarizes the results when applying GA- and ACO-based approaches in all scenarios. Note that “L-5” refer to the primary incident occurred in the left lane with a duration of five minutes (Fig. 18(b)). “R-L-5” represents scenario of Fig. 18(c) with a primary incident presenting five minutes in the right lane. The same rules apply to other scenarios. As observed in Fig. 23, the increase in market penetration rate leads to higher accuracy. This is independent from the location of the primary incident as well as its duration. The results also suggest that a MPR of 15 percent or more enables a relatively high accuracy in terms of re-identifying secondary and non-secondary incidents correctly.

Additional tables on precision rate, true positive rate, and false positive rate are presented in our paper [21]. The relatively high precision rates, true positive rates, and low false positive rate all suggest that the proposed approach can achieve relatively good performance even under low market penetration rate.
Fig. 23. Estimated Accuracy Based on Each Approach (scenarios: PI with 20-min duration.)
<table>
<thead>
<tr>
<th>Scenario</th>
<th>Method</th>
<th>MPR=0.05</th>
<th>MPR=0.1</th>
<th>MPR=0.15</th>
<th>MPR=0.2</th>
<th>MPR=0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>L-5</td>
<td>ACO</td>
<td>0.503±0.126</td>
<td>0.724±0.085</td>
<td>0.805±0.031</td>
<td>0.849±0.025</td>
<td>0.865±0.022</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.605±0.105</td>
<td>0.740±0.061</td>
<td>0.818±0.039</td>
<td>0.837±0.049</td>
<td>0.841±0.022</td>
</tr>
<tr>
<td>L-10</td>
<td>ACO</td>
<td>0.646±0.096</td>
<td>0.815±0.035</td>
<td>0.869±0.018</td>
<td>0.912±0.021</td>
<td>0.911±0.014</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.717±0.069</td>
<td>0.836±0.026</td>
<td>0.867±0.022</td>
<td>0.910±0.019</td>
<td>0.914±0.020</td>
</tr>
<tr>
<td>L-20</td>
<td>ACO</td>
<td>0.816±0.045</td>
<td>0.912±0.017</td>
<td>0.928±0.009</td>
<td>0.949±0.013</td>
<td>0.954±0.005</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.837±0.035</td>
<td>0.897±0.019</td>
<td>0.918±0.016</td>
<td>0.933±0.019</td>
<td>0.934±0.018</td>
</tr>
<tr>
<td>R-5</td>
<td>ACO</td>
<td>0.514±0.148</td>
<td>0.715±0.067</td>
<td>0.804±0.035</td>
<td>0.856±0.018</td>
<td>0.878±0.021</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.574±0.112</td>
<td>0.779±0.057</td>
<td>0.841±0.024</td>
<td>0.865±0.024</td>
<td>0.878±0.016</td>
</tr>
<tr>
<td>R-10</td>
<td>ACO</td>
<td>0.647±0.049</td>
<td>0.825±0.031</td>
<td>0.879±0.023</td>
<td>0.914±0.018</td>
<td>0.923±0.013</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.717±0.038</td>
<td>0.845±0.041</td>
<td>0.879±0.023</td>
<td>0.906±0.017</td>
<td>0.910±0.012</td>
</tr>
<tr>
<td>R-20</td>
<td>ACO</td>
<td>0.806±0.043</td>
<td>0.886±0.019</td>
<td>0.921±0.013</td>
<td>0.946±0.010</td>
<td>0.947±0.005</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.831±0.042</td>
<td>0.902±0.015</td>
<td>0.912±0.013</td>
<td>0.940±0.014</td>
<td>0.937±0.018</td>
</tr>
<tr>
<td>R-L-5</td>
<td>ACO</td>
<td>0.592±0.067</td>
<td>0.761±0.049</td>
<td>0.847±0.029</td>
<td>0.884±0.015</td>
<td>0.904±0.016</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.630±0.063</td>
<td>0.808±0.073</td>
<td>0.874±0.023</td>
<td>0.894±0.018</td>
<td>0.902±0.032</td>
</tr>
<tr>
<td>R-L-10</td>
<td>ACO</td>
<td>0.678±0.084</td>
<td>0.811±0.041</td>
<td>0.870±0.024</td>
<td>0.903±0.013</td>
<td>0.918±0.010</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.716±0.089</td>
<td>0.848±0.028</td>
<td>0.883±0.048</td>
<td>0.898±0.032</td>
<td>0.917±0.013</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>R-L-20</td>
<td>ACO</td>
<td>0.798±0.076</td>
<td>0.903±0.021</td>
<td>0.938±0.013</td>
<td>0.948±0.009</td>
<td>0.954±0.006</td>
</tr>
<tr>
<td></td>
<td>GA</td>
<td>0.831±0.038</td>
<td>0.901±0.016</td>
<td>0.931±0.021</td>
<td>0.941±0.009</td>
<td>0.940±0.011</td>
</tr>
</tbody>
</table>

*Note: The numbers in each cell represent mean ± standard deviation.*
5.7 CONCLUSIONS

A data-driven analysis framework for identifying secondary incidents is proposed. The proposed approach intends to leverage the untapped potential of ubiquitous probe vehicle data for secondary incident identification. The developed framework consists of three major components: detecting the impact area of a primary incident, estimating the boundary for the impact area, and identifying secondary incidents within the boundary. The first component uses the clustering algorithms to highlight the congested area induced by the primary incident based on the probe data. This helps visually check the spatiotemporal impact of the incident. The second component introduces three estimation algorithms to capture the spatiotemporal impact area of the incident. These algorithms analyze the trajectory points in the congested area and approximate the boundary of the area based on the multi-stage linear approximation, genetic algorithm, and ant colony optimization, respectively. With the estimated boundary, the third component automates the identification process based on the ray-casting algorithm. The proposed framework has been tested based on probe data collected from different simulation models. The results show that the impact area induced by a primary incident can be well represented by the estimated boundary, especially by the GA- and ACO-based methods. With the MPR increasing from 5 percent to 25 percent, the accuracy of re-identifying secondary incidents can reach up to 95 percent. An MPR of 15 percent already yields a relatively high accuracy.
CHAPTER 6
CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

The overall goal of this dissertation is to investigate incident analysis based on advanced machine learning algorithms and a variety of sensor data. Three related research topics are proposed in this dissertation.

The first topic proposed an AI-based framework for incident detection. The integration of memory unit enables the framework to learn from historical information. The framework using either lane-based data or station-level average data performed relatively better than the benchmark approaches, including CA#7 and ANN. The augmented performance of the proposed approach has been primarily demonstrated in terms of the shorter detection time, lower false alarm rate, and higher detection rate. The presented results show the improved performance of the proposed AI-based framework regardless of the sensor spacing.

The second topic proposed an unsupervised learning approach, FCM, to relabel incident occurrence times and further examined its impact on three different incident detection approaches (CA#7, ANN, and SVM). In order to better automatically relabel three types of inaccurate mapping between reported incident occurrence times and loop detector measurements, BIC values and additional restriction rules are applied. The proposed relabeling strategy is evaluated given 360 simulated incident scenarios. The results showed that the relabeling strategy can help improve the performance of three TID approaches in terms of a higher DR and a lower FAR. Comparisons of TID performances under different scenarios also imply that TID approaches need more efforts to
predict incidents with a longer loop detector gap, the incident occurrence in the middle between two loop detectors, and under a lower traffic volume condition.

The third topic proposed a data-driven analysis framework for identifying secondary incidents. The proposed approach intends to leverage the untapped potential of ubiquitous probe vehicle data for secondary incident identification. The developed framework consists of three major components: detecting the impact area of a primary incident, estimating the boundary for the impact area, and identifying secondary incidents within the boundary. The proposed framework has been tested based on probe data collected from different simulation models. The results show that the impact area induced by a primary incident can be well represented by the estimated boundary, especially by the GA- and ACO-based methods.

6.2 FUTURE WORK

Even though our proposed work and results have reached promising performance, there still exists room for improvements for three topics. First, field test data can be further applied to verify the incident detection performance. This dissertation used simulated data and it should be noted that field test data will differ from field test data, and will suffer from issues such as imbalance, data accuracy, and data missing. The data cleaning and outlier analysis should be applied before tuning the incident analysis framework.

Second, with the emerging connected vehicle technologies, the investigation on the prevention of secondary incident would be interesting. Previously, we have simulated connected vehicles and investigated its impact on mitigating secondary incident risks [91]. Connected vehicles can get information of occurred incidents, and therefore informed drivers will make optimal driving behaviors to reduce secondary incident risks based on simulated scenarios. It
should be noted that the communication delay and potential data losing issues during the vehicular communication are not considered. Hence, our future plan is to apply field test data of connected vehicles and investigate their impact and potential countermeasures to prevent secondary incidents.

Meanwhile, automated vehicles (AVs) serve as the promising technology around the corner and raise diverse hot topics such as environment perception and anomaly detection. If AVs are introduced, their impact on incident detection and incident prevention should be investigated. In addition, AVs heavily rely on AI algorithms such as deep learning approaches (DNN, CNN) to percept surrounding environment and make decisions. We had summarized relevant literature on AI application in the development of AVs [104], and plan to further explore AI algorithms such as SVM, CNN, and LSTM on detecting surrounding environments, identifying potential primary incidents, and preventing secondary incidents in the future.
REFERENCES


VITA

Zhenyu Wang

Department of Computational Modeling and Simulation Engineering

Old Dominion University

Norfolk, VA 23529

Zhenyu Wang received his B.S. in English and M.S. in Communication and Information System from Beijing University of Post and Telecommunications, China in 2011 and 2014, respectively. He started pursuing Ph.D. in Modeling and Simulation at Old Dominion University in 2015.
LIST OF PUBLICATIONS

Journals


Conferences


