5-2020

Real-time Traffic State Assessment using Multi-source Data

Yuandong Liu
University of Tennessee, yliu140@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Yuandong Liu entitled "Real-time Traffic State Assessment using Multi-source Data." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Civil Engineering.

Lee Han, Major Professor

We have read this dissertation and recommend its acceptance:

Candace Brakewood, Christopher Cherry, Russell Zaretzki

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Real-time Traffic State Assessment using Multi-source Data

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Yuandong Liu
May 2020

DEDICATION

To my parents, Jiakun Liu, Yuping Xu, for all their endless love and support.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. Lee D. Han for his support and guidance throughout my graduate studies. I was very fortunate to have him as my advisor who continuously provides keen insight and wise judgement to my research. He sparks my research passionate and inspires me to think critically and creatively. Without him, I wouldn't be who I am now. Also, his aesthetics sensitivity and devotion towards it have profound influence on me. I would also like to thank my statistics advisor, Dr. Russell Zaretzki, and my committee members, Dr. Candace Brakewood and Dr. Christopher Cherry for serving on my committee. Special thanks to Dr. Candace Brakewood for her support and valuable advices on my research.

I will always be grateful to all my teammates: Hyeonsup Lim, Bumjoon Bae, Yang Zhang, Pankaj Dahal, Zhihua Zhang, Nima Hesenzadeh, Yangsong Gu and Harry Zhang. We work on a variety of projects and research ideas together, share our experience and life. It's been a great experience to work with all of them.

I always want to express my love to my families. My father, Jiakun Liu, who loves me unconditionally and encourages me to pursue a PhD degree aboard; My mother Yuping Xu, who teaches me to be responsible and honest, the traits I need not only in my PhD studies but also in my life; My brother, Yuanquan Liu, who is both a family member and a friend of me. We share memories and loves with each other and support each other through difficulties.

Last, but not least, I would like to thank all my lab mates: Ali Boggs, Mojdeh AzadDisfany, Meng Zhang, Ziwen Ling, Xiaobing Li, Mohsen Kamrani, Ramin Arvin, Jun Liu, Nirbesh Dhakal, Kwaku Boakye and many others.

ABSTRACT

The normal flow of traffic is impeded by abnormal events and the impacts of the events extend over time and space. In recent years, with the rapid growth of multi-source data, traffic researchers seek to leverage those data to identify the spatial-temporal dynamics of traffic flow and proactively manage abnormal traffic conditions. However, the characteristics of data collected by different techniques have not been fully understood. To this end, this study presents a series of studies to provide insight to data from different sources and to dynamically detect real-time traffic states utilizing those data.

Speed is one of the three traffic fundamental parameters in traffic flow theory that describe traffic flow states. While the speed collection techniques evolve over the past decades, the average speed calculation method has not been updated. The first section of this study pointed out the traditional harmonic mean-based average speed calculation method can produce erroneous results for probe-based data. A new speed calculation method based on the fundamental definition was proposed instead. The second section evaluated the spatial-temporal accuracy of a different type of crowdsourced data - crowdsourced user reports and revealed Waze user behavior. Based on the evaluation results, a traffic detection system was developed to support the dynamic detection of incidents and traffic queues. A critical problem with current automatic incident detection algorithms (AIDs) which limits their application in practice is their heavy calibration requirements. The third section solved this problem by proposing a self-evaluation module that determines the occurrence of traffic incidents and serves as an auto-calibration procedure. Following the incident detection, the fourth section proposed a clustering algorithm to detect the spatial-temporal movements of congestion by clustering crowdsource reports.

This study contributes to the understanding of fundamental parameters and expands the knowledge of multi-source data. It has implications for future speed, flow, and density calculation with data collection technique advancements. Additionally, the proposed dynamic algorithms allow the system to run automatically with minimum human intervention thus promote the intelligence of the traffic operation system. The algorithms not only apply to incident and queue detection but also apply to a variety of detection systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

The smooth movement of traffic is disturbed by abnormal events frequently. The impacts of the disturbance expand and lead to traffic congestion**.** Dynamic evaluation of traffic states and the identification of abnormal events as well as subsequent congestion can reduce the impacts and improve transport network efficiency and safety. This study aims to better describe traffic states and identify the change in the traffic state with multi-source data.

Traffic state is represented by speed, flow, and density, the three fundamental macroscopic traffic parameters. It is always of traffic researcher and practitioners' interests to obtain more complete and accurate information of the three parameters. The techniques to collect the three traffic features evolve over the past several decades. Traditionally, traffic researchers mainly rely on roadside sensors to obtain average speed, flow, and density, along with CCTV and highway patrol program to visually identify congestion or abnormal events on roads. In recent years, with the advancement in GPS-based mobile devices and sensing technologies, traffic features can be collected in multiples ways, including average speed calculated from continuous location information, road user reports about congestion or abnormal events, and potentially microscopic features from connected and autonomous vehicles in the future. One of the concerns with multi-sourced data is their consistency and reliability. The data collected by GPS devices are inherently different from data collected by traditional roadside sensors. How to correctly obtain fundamental parameters from collected information is critical to ensure data consistencies and accuracy. Also, data reported from road users could have inaccurate or even incorrect information and need to be validated before being incorporated into any real-time management and operation system.

The movement of traffic streams is represented by the change of traffic state over time and space. Incident and queue detection identify the change in traffic flow and recognize abnormal traffic behavior. It is a fundamental step to monitor the current traffic states and provide knowledge for road users and traffic practitioners. In recent years, a series of data-driven techniques and models have been developed to detect the change traffic states over time and space. The existing incident and queue detection system has limitations in two aspects. First, the limited transferability of detection models restricts their application in practice. Second, the established detection system applies to the urban area but not the rural area where limited monitoring devices were installed.

The purpose of this dissertation is essentially twofold: first, evaluate emerging data sources and understand how they contribute to traffic state knowledge; Second, harness the available data sources and develop a system that can detect abnormal traffic flow patterns dynamically. Four research papers were therefore compiled to address the purpose, each of which is presented in a single chapter. **Figure 1** demonstrates the framework of this dissertation. This dissertation combines both traditional data and emerging data and presents a framework composed of both data evaluation section and traffic assessment and detection section. The first chapter and the second chapter assess emerging crowdsources data. Chapter I aims to clarify how speed shall be accurately calculated with probe-based data. Chapter II then evaluates the reliability of

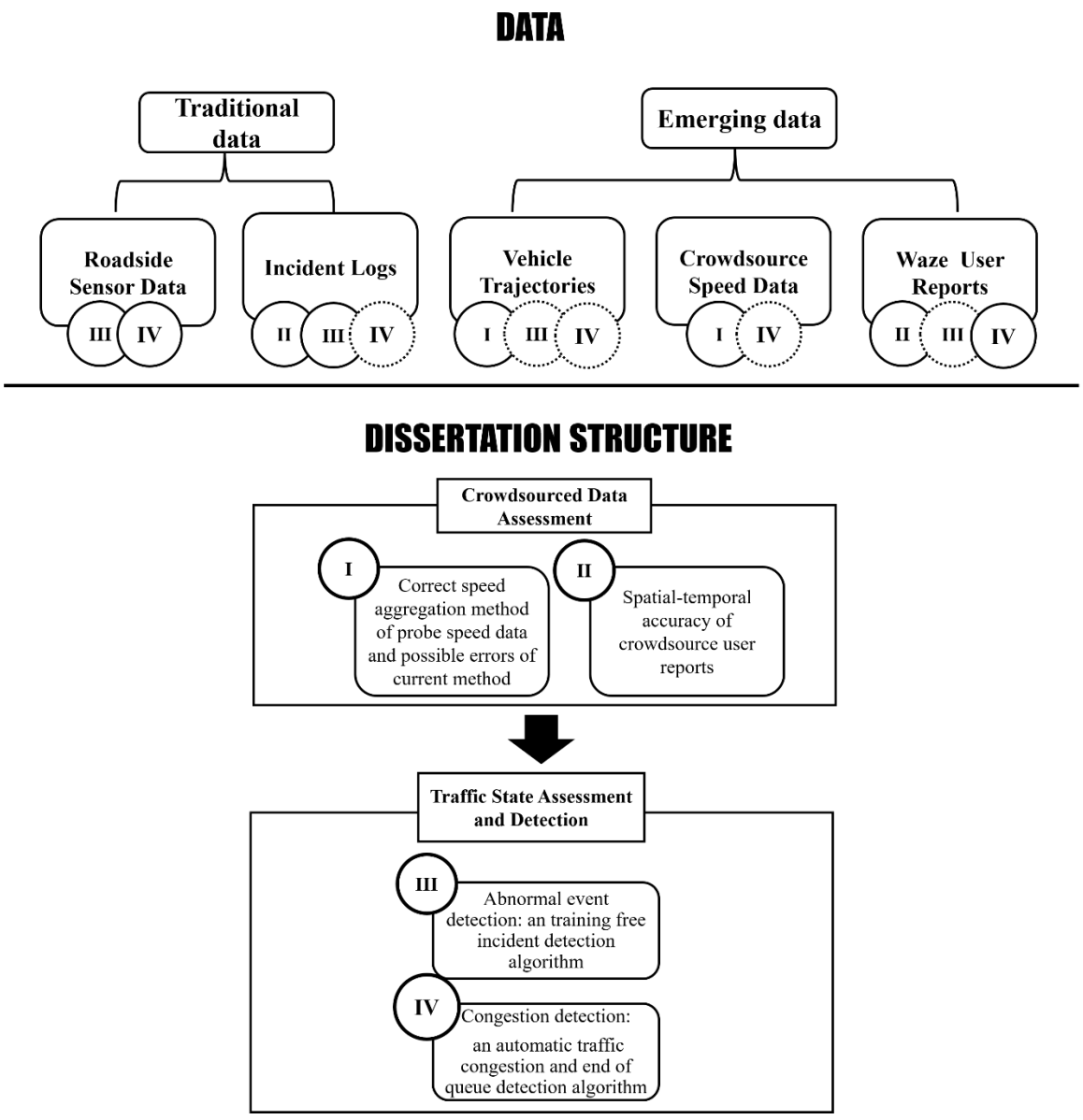


Figure 1. Data sources and dissertation framework.

crowdsourced user reports with a focus on spatial and temporal accuracy. The third chapter and fourth chapter propose dynamic detection algorithms that separately detect the start of abnormal congestion and its movement using both traditional data and emerging crowdsourced user reports. The data sources and their applications in each chapter are identified in **Figure 1**. In the data section, solid circles with chapter numbers inside represent that the data were used in the corresponding chapters; dotted circles indicate that the data sources can be used to address the problem to be solved in each chapter. The data sources are further detailed in **Table 1** with a summary of their strengths and limitations. The four chapters are organized in a journal article format:

- Chapter I tackles average speed, one of the three fundamental characteristics in traffic flow theory. In this chapter, it is pointed out that the speed data collected by probe vehicles are inherently different from speed collected by traditional roadside sensors and shall be aggregated differently. Instead of employing a well-recognized method, this chapter proposes a calculation method follows the definition of speed for probe-based data. A comparison between different aggregation approaches shows that the proposed method can produce unbiased results.
- Chapter II evaluates the temporal and spatial accuracy of crowdsourced reports (Waze reports) on interstate highways. A matching criterion is proposed to pair Waze abnormal traffic event reports with official reports collected by TDOT, which serves as the ground truth. Then the location and temporal accuracy are evaluated.
- Chapter III proposes a self-learning abnormal traffic event detection algorithm. The proposed algorithm incorporated a self-evaluation module that can assess the detection results to support the adaptive selection of thresholds at different locations and different times of day. The algorithm is tested with real incident data and the detection results are compared to the detection results of a benchmark model.
- Chapter IV proposes a congestion detection and end of queue identification algorithm that dynamically cluster crowdsourced reports based on the road geometry to track the movement of congestion. The algorithm is tested with a case study and the detection results are compared to the detection results based on road-side sensor data.

Table 1. **A summary of multi-source data.**

Data	Description	Chapter	Strengths & Limitation
Public Sector Data (Traditional data sources)			
RTMS (Remote Traffic Microwave Sensor) data	Roadside sensor data	Chapter IIIChapter IV	*Strength* • Covers urban area. • Provide flow and density information *Limitation* • High installation and maintenance fee • Limited spatial coverage
LocateIM	Official incident records	Chapter II Chapter III	*Strength* • High location accuracy • Validated incident report *Limitation* • Report timeliness • Does not cover all incident records
Private Sector Data (Emerging data sources)			
NGSIM (Next Generation Simulation)	Vehicle trajectories	Chapter I	*Strength* • Provide most detailed information (*1*) *Limitation* • Not available at other locations.
WAZE speed	Crowdsource speed	Chapter I & IV	*Strength* • High spatial coverage *Limitation* • Accuracy varies with penetration rate and reporting frequency (*2-4*) • Does not provide flow and density
WAZE reports	Crowdsource user reports	Chapter II & IV	*Strength* • Timeliness. • Covers incidents that not reported in official record.(*5-7*) *Limitation* • Duplicate reports • Reliability (*6*)

CHAPTER I. AN UNBIASED METHOD FOR PROBE VEHICLE AVERAGE SPEED CALCULATIONS – METHODOLOGY AND DEMONSTRATIONS WITH SIMULATED PROBE-BASED DATA

A version of this chapter was originally presented at TRB 2019 by Yuandong Liu, Yang Zhang and Lee D. Han. The paper has been submitted to *Journal of Applied Statistics*.

Abstract

Traffic engineers are often interested in measuring speed along a stretch of roadway for a given period of time. Typically, in the past, speed values are measured at a given location over some duration. After some initial confusions, traffic engineers correctly determined that harmonic means, instead of arithmetic means, should be used to calculate the average speed of the traffic stream. In the modern age of ubiquitous devices of mobile phones, vehicle speed data can be collected along a stretch of roadway frequently. The speed calculations, however, have not always been performed correctly with these data. Many users are under the impression that as long as individual speed data were aggregated using the harmonic mean method, the result would be correct, or at least "close enough." This, as is shown in this paper, is far from the truth. This paper examines calculation methods for the mean speed of probe vehicle data using different sampling strategies. It is demonstrated that average speed can be accurately obtained by taking the arithmetic mean of vehicle spot speeds if the sampling is done by time. Real-world vehicle trajectory data from the NGSIM database were used to verify and demonstrate that the traditional harmonic mean-based calculation can be quite erroneous and the average speed should be computed using simple arithmetic means if time-based sampling strategy is used. Aggregating the vehicle speeds by taking harmonic means usually leads to an underestimate of the mean speed, compared to the arithmetic mean approach.

Introduction and Literature Review

Mean speed and travel time for a road section over a time period are critical information at all levels of traffic operation, travel planning, and transportation policy. Techniques to measure speed and travel time have changed over the past 60 years. The basic approach to measure speed is to use roadside speed detection instrumentations (e.g., in-pavement loop detectors and pole-mounted detectors). Roadside sensors are widely deployed and have been the most commonly used traffic data collection methods for decades in the US. Loop detectors capture the time when the vehicle passed the loop and compute the speed as the ratio of loop width and vehicle effective length to crossing time. Other speed measurement techniques include aerial photographs, Bluetooth sensors, and License Plate Recognition (LPR) sensors. Aerial photographs provide valuable spatial data along a road segment. Density can be directly measured, if several continuous frames are available, allowing for vehicle tracking; speed can be obtained as well. In practice, cameras are used instead of aerial photographs, and due to the immaturity of video processing techniques, cameras are mostly used as surveillance devices instead of speed collection methods. Bluetooth sensor is a special kind of roadside sensor; it is placed both at the start and end of a road segment and collects travel time information of passing vehicles.

Speed can then be derived using distance divided by travel time(*8*). The advantage is that Bluetooth sensors directly measure the vehicle travel time along a segment, the speed of that vehicle traverses the segment can be obtained as well, as opposed to spot speed that ordinary roadside sensor measures. Therefore, travel time collected by Bluetooth is always regarded as ground truth. Similar to Bluetooth sensors, LPR devices record the time and plate number of passing vehicles, and the travel time of the vehicle on a specific road segment can be obtained by matching the license plate number(*9*). The concern with Bluetooth and LPR technologies is that they require costly installation of devices as well as maintenance fees. Taking both the advantages and limitations into account, Bluetooth and LPR devices are mostly adopted as supplemental measurements to validate speed collected by other devices, and so are not widely deployed by government agencies on roadways.

It is crucial to review all the data collection techniques because mean speed aggregation methods rely on how speed is measured. In this paper, the authors discuss various studies on how to correctly aggregate vehicle speed data. Most of these studies are based on two concepts brought up by Wardrop (*10*), who describe two types of speed aggregation in his research; One is based on the distribution of speed in time and the other is based on the distribution of speed in space. These two mean speed concepts, space mean speed and time mean speed, have been widely adopted by subsequent researchers.

Both the time- and the space-mean speed can be obtained if it is known how speeds were collected. For example, space mean speed measured by a roadside detector is calculated by taking the harmonic mean of vehicle speeds, and time-mean speed is computed by taking the arithmetic mean of vehicle speeds. When using aerial photographs, space mean speed is calculated as the arithmetic mean of vehicle speeds. However, because speeds are mostly measured by roadside sensors, some users and researchers are under the impression that as long as individual speed data were aggregated using the harmonic mean method, the result would be correct, or at least "close enough." This, as is shown in this paper, is far from the truth.

Researchers and practitioners agree that space mean speed, but not time mean speed, should always be used in speed, flow, and density relationships to describe speed along a road segment instead of at a certain location. Despite the basic definition, space mean speed for a road segment during a time period has been defined in multiple ways in the literature, but two main groups of definitions have emerged (*11*). The first group defines space mean speed as the arithmetic mean of speeds of all vehicles in a road section in a short time interval (*12*). The second group of authors defines space mean speed as total distances traversed inside a road segment divided by total time each vehicle spent inside the segment (*13*). In this study, to avoid confusion, we will use mean speed, instead of 'space mean speed,' in subsequent discussions. This term represents the average speed when the correct aggregation method is adopted regardless of the data collection techniques.

Probe Vehicle Data and Mean Speed

In recent years, with the ubiquitous use of mobile phones and GPS devices, probe vehicle data has become one of the main data sources in transportation. Currently, probe vehicle

data are mostly provided by commercial private vendors, such as INRIX, TomTom, and HERE. Each data provider uses a driver network, comprising vehicles, smartphones, and other GPS-enabled devices, to monitor basic location and speed attributes of the vehicle. There are multiple advantages of using vehicle trajectory data. Unlike stationary point sensors, it avoids high installation and maintenance costs. Moreover, probe vehicle data has better spatial coverage. The probe vehicle vendors now can generate link speeds and travel times for almost all the primary roadways, and update them at certain time rates (*2*). For instance, Google map and WAZE map are now able to update traffic conditions every one or two minutes, thus have the potential to be used for real-time traffic management and control (*14*). Because of these advantages, especially the improved spatial coverage, NPMRDS data becomes the official probe vehicle data set that was designated by FHWA for the computation of multiple travel time reliability measures for long-range planning purposes; it provides a five minute aggregated historical speed data at the beginning of every month (*15*).

Probe vehicle data has been one of the most prevalent data sources for traffic speed collection. However, the speed accuracy provided by these vendors is doubtful. Many previous studies evaluated the speed collected by probe vehicles and compared it with speed collected by either Bluetooth devices or roadside sensors (*3; 9; 16*). FHWA also requests NPMRDS data vendors to submit quarterly data quality reports that compare probe vehicle data to Bluetooth data at selected locations to control the quality of probe vehicle data. A prerequisite of this comparison is that the same mean should be used throughout any investigation so that all comparisons are fair. However, how the GPS-based vehicle speeds are aggregated by these private companies is still unclear. According to the description of NPMRDS data v2 (https://npmrds.ritis.org/analytics/), the average speed for a segment during a time interval is computed in the following manner. First, find the first and last observations of a certain vehicle during the time period of interest, then determine the distance traveled and time spent between the two observations. The average travel speed for each vehicle on the segment is expressed as the ratio of the traveled distance by the time of travel. By taking the harmonic mean of all the vehicle speeds on the segment within a specific time period, the average speed for the segment is available. This aggregation methodology is supported and adopted in several other papers (*17-19*). Wenjing claimed that harmonic mean speed should be used when averaging probe vehicle speeds (*17*). Hongyan, similarly, calculated the average travel speed of a segment during the time period by taking the harmonic mean of all individual vehicles (*18*).

As mentioned, the aggregation methods rely on data collection techniques. In the case of probe-based traffic data, it is important to clarify how the speed data are collected. Typically, there are two main sampling strategies for probe vehicle data: time-based sampling, where the vehicle trajectory is sampled at certain time intervals, and space-based sampling, where the vehicle trajectory is sampled at certain distance intervals(*20; 21*). Sometimes the sampling strategy is a mixture of those two basic protocols (*22*). Several papers discuss how to estimate the average speed with respect to different sampling strategies. Westgate recommends using the harmonic mean of the GPS speeds to estimate the travel time and proved it is an unbiased estimator for the true mean

travel time if GPS points are sampled by distance. However, if GPS points are sampled by time, the harmonic mean method overestimates the mean travel time (*22*). Juan proposes that if all vehicles sampled traverse the entire length of the arterial segment of interest, which is similar to the case that samples by distance, the average speed can be expressed as the harmonic mean of vehicle speeds (*7*). According to these researchers, ignoring the data collection techniques either by taking the arithmetic mean of travel time or taking the harmonic mean of vehicle speeds may lead to an overestimation or underestimation of mean speed.

Objective

The purpose of this paper is to investigate mean speed aggregation methods for probe vehicle-based traffic data. This paper is organized as follows: the second section briefly introduces the definition of generalized mean speed and proposes aggregation methods of mean speed for probe vehicle data according to different sampling schemes. The third section employs NGSIM data and investigates the errors introduced if the incorrect speed aggregation method is used. The final section concludes this paper and suggests directions for future research.

Mean Speed Calculation Methods

Little research has been done to address the speed aggregation method for probe vehicle data although it has become one of the most prevalent data sources for speed on road. As outlined earlier, the speed aggregation techniques are determined by the data collection methods, therefore in this section, we adopt the generalized definitions of flow, speed, and density that introduced by Edie(*13*), distinguish two types of data collection techniques for probe vehicle data and provide the correct speed aggregation methods.

Background on Generalized Fundamental Variables

Considering an arbitrarily shaped region A within a time-space diagram as shown in **Figure 2**. Part of the trajectory of vehicles ($i = 1,2, \dots n$) is enclosed by area A, thus the total distance traveled and time spent by vehicle i in this region are separately the projection of trajectory to distance and time axes. We use d_i , t_i to denote them. Then, the generalized flow and density of region A is given by,

$$q_A = \frac{\sum_{i=1}^{n} d_i}{|A|} \quad (1)$$

$$k_A = \frac{\sum_{i=1}^{n} t_i}{|A|} \quad (2)$$

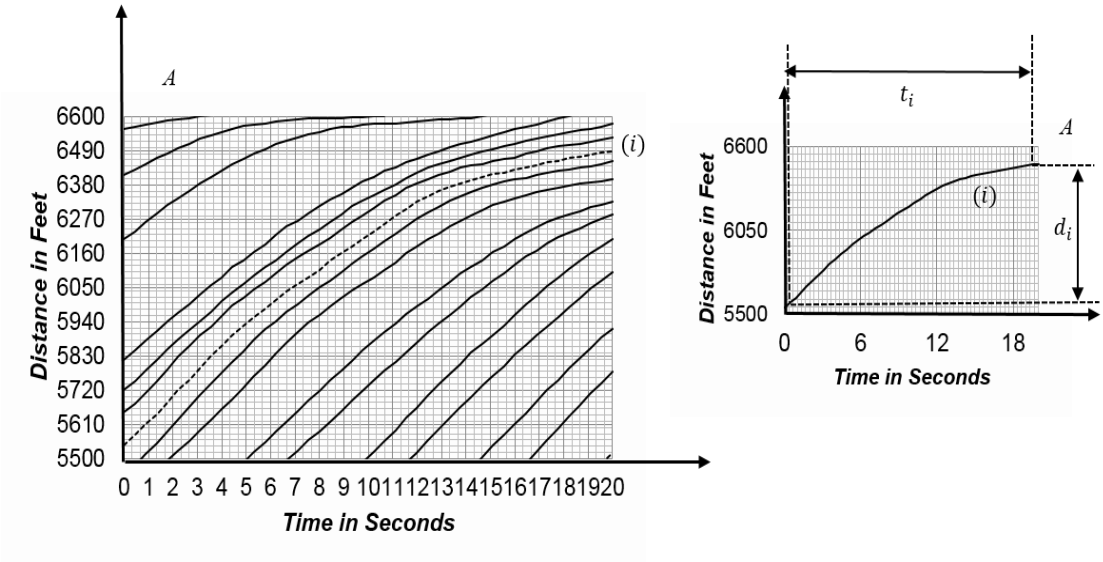


Figure 2. Vehicle trajectories enclosed in region A.

Where n is the number of vehicle trajectories enclosed in region A. $|A|$ is area of time-space domain. According to relations among three fundamental variables: $\bar{v} = q/k$, the average speed in area A is given by,

$$v_A = \frac{\sum_{i=1}^{n} d_i}{\sum_{i=1}^{n} t_i} \tag{3}$$

This is the generalized definition of speed. The formula can be understood intuitively; it means the average speed in a region is the ratio of aggregated distance traveled by all vehicles to aggregated time spent in traversing the corresponding distance. It can be demonstrated that if all the vehicles crossed the full section, D, the average speed in region A is calculated by taking the harmonic mean of all vehicle speeds,

$$v_A = \frac{n * D}{\sum_{i=1}^{n} t_i} = \frac{1}{\frac{1}{n}\sum_{i=1}^{n} \frac{t_i}{D}} = \frac{1}{\frac{1}{n}\sum_{i=1}^{n} \frac{1}{v_i}} \tag{4}$$

This is the special case of roadside sensors. However, for a time-space region, there will always be some vehicles that have not completed the crossing, therefore directly taking harmonic mean of vehicle speeds may lead to inaccurate results.

Mean Speed for Probe Vehicles

Probe vehicle speed data is collected by sampling vehicle trajectories. Vehicle location, time and sometimes the instantaneous speed are recorded for subsequent speed calculation and other analysis. Thus, we don't have the complete vehicle trajectory, different methods used to aggregate sampled vehicle trajectory data are discussed in this section.

Harmonic Mean Approach

As explained in the introduction section, the harmonic mean speed aggregation method has been adopted by some researchers, practitioners and data vendors, including INRIX company who supply NPMRDS data to FHWA (Federal Highway Administrative). According to the description of NPMRDS data v2(Figure 3), the average speed for a segment during a time interval is computed in the following manner:

$$v_i = \frac{d_i}{t_i} \tag{5}$$

$$\bar{v}_A = \frac{n}{\sum_{i=1}^{n} \frac{1}{v_i}} \tag{6}$$

i represents each vehicle, d_i and t_i separately represents the distance traveled and time spent in the time-space region for vehicle i , v_i is the average speed for vehicle i. Then, the average speed $\bar{v}_A$ in region A is computed as the harmonic mean of vehicle speeds for all sampled vehicles traveling a specific section during the examined period.

Proposed Approach

There are typically two sampling strategies for probe vehicle data, temporal sampling and spatial sampling. The speed aggregation approach for each sampling strategy is provided below.

Temporal Sampling (Sampled by time)

The vehicle trajectory is sampled at a fixed time interval from the previous sample time or at a certain time. In this case, equipped vehicles report their information at fixed time intervals T, regardless of their positions. This is the simplest and most commonly adopted sampling method. **Figure 4** illustrates an idealized time-based sampling procedure. **Figure 4**(a) is a set of vehicle trajectories traveling a road segment of 1100 feet for a duration of 20 seconds. A total of 16 trajectories are enclosed in the time space region. Assuming that all vehicles in this region report their position and time at a fixed time interval T(0.44 seconds in this example), as shown in **Figure 4**(b), each small circle on the trajectory indicates a sampled point of the corresponding vehicle. **Table 2** summarizes the information of each vehicle: the total distance traversed, total time used, and average speed of an individual vehicle in this region. The vehicle is numbered from upper left corner to lower right corner.

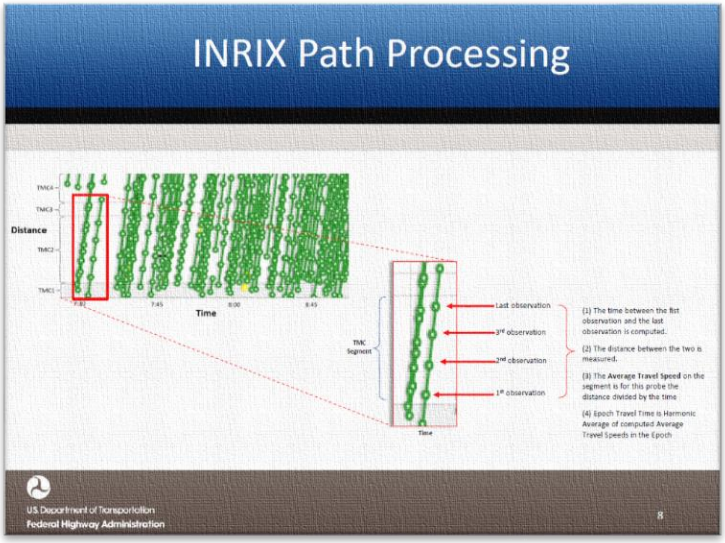


Figure 3. INRIX path processing method (NPMRDS data v2. (https://npmrds.ritis.org/analytics/)).

Then, according to the generalized definition of mean speed of a region, the ground truth mean speed within this time-space region can be calculated as,

$$\bar{v}_A = \frac{\sum_{i=1}^{n} d_i}{\sum_{i=1}^{n} t_i} = \frac{\sum_{i=1}^{16} d_i}{\sum_{i=1}^{16} t_i} = 32.96mph \tag{7}$$

Where i represents each vehicle, n is the total number of vehicles in the time-space region. In the above example, n equals 16. The mean speed of the time-space region is 32.96mph. However, if using the harmonic mean approach and take the harmonic mean of each vehicle speed, the mean speed for this region is,

$$\bar{v}_A = \frac{n}{\sum_{i=1}^{n} \frac{1}{v_i}} = 29.00mph \tag{8}$$

Where i represents each vehicle, v_i represent the speed of each vehicle traveling this segment during the time period, that is the last column of **Table 2**. This is lower than the ground truth speed.

As shown in **Figure 4(b)**, the vehicle reports its position at fixed time interval T. Each vehicle i traveled distance d_i during T. Mean speed of this region can also be computed as,

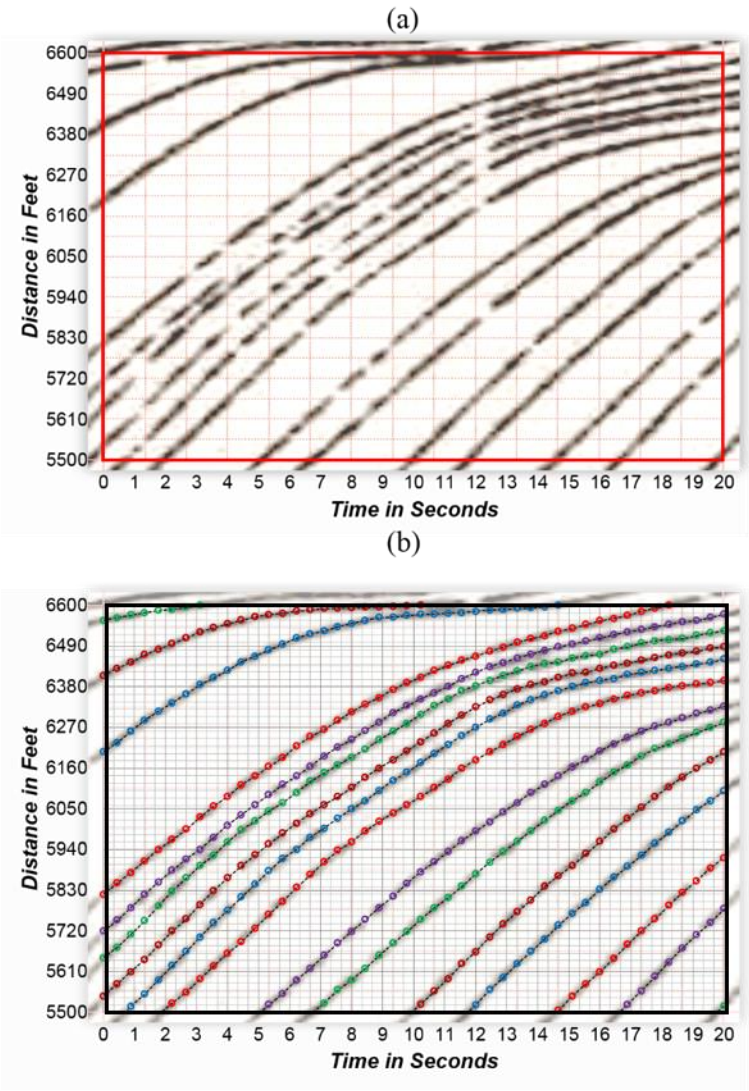


Figure 4. Digitized vehicle trajectories and time-based sampling.

Table 2.Vehicle traveling information.

Vehicle No.	**Total distance d (feet)**	**Total time t (second)**	**Speed v (mph)**
1	189.20	10.22	12.62
2	396.00	14.67	18.40
3	781.00	18.22	29.23
4	855.80	20.00	29.18
5	883.30	20.00	30.11
6	943.80	20.00	32.18
7	976.80	19.56	34.05
8	902.00	18.22	33.75
9	838.20	15.11	37.82
10	816.20	13.78	40.38
11	715.00	10.22	47.70
12	611.60	8.44	49.41
13	451.00	5.78	53.20
14	297.00	3.56	56.88
15	35.20	0.44	54.55
16	189.20	10.22	12.62
Total	9732.80	201.33	32.96

$$\bar{v}_A = \frac{\sum_{i=1}^{n} d_i}{\sum_{i=1}^{n} t_i} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{N_i} d_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{N_i} T} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{N_i} \frac{d_{ij}}{T}}{\sum_{i=1}^{n}\sum_{j=1}^{N_i} 1} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{N_i} v_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{N_i} 1} = 32.96mph \quad (9)$$

v_{ij} denotes the speed of vehicle i when making report j. n is the number of vehicles traveling in this region, N_i is the number of reports that vehicle i made within this region. $\sum_{i=1}^{n}\sum_{j=1}^{N_i} 1$ is the total number of reports received for all vehicles in this region. Therefore, the mean speed in region A can be expressed as the arithmetic means of all spot speeds in that region.

One limitation in many previous studies is that probe data is assumed to have homogeneous sample rates. In practice, probe vehicles may provide irregularly spaced or discontinued data (*23*). Experimental data collected from multiple providers feature heterogeneous characteristics, such as having a mixture of high- and low-frequency probe data (*2*). For instance, INRIX uses a broad collection of commercial and consumer GPS probe data as the source data, which are composed of data with different sampling intervals. Some data are provided to INRIX at one-second update rates while some sources can have as much as three minutes between data reports. Most data are provided to INRIX by sources with temporal granularity between 15 seconds and one minute. Estimating speed with a heterogeneous sampling rate is more complicated because it lacks detailed information about the irregularity of the sample. In this case, the most suitable method to obtain mean speed is according to the basic definition, total distance traveled by vehicles inside the spatiotemporal region divided by total time of travel.

Spatial Sampling (Sampled by distance)

Vehicle trajectory is sampled at a certain distance from a previous sample point or at certain locations. In this case, equipped vehicles report their information as they cross some spatially defined sampling locations, which is similar to a roadside sensor (*24*). This strategy has the advantage that the phone is forced to send data from a given location of interest. This type of sampling strategy is rarely implemented in practice. Two examples are found in the literature. The first one is the Mobile Century field experiment (*24*). Taking privacy issues into account, the experiment implemented a virtual trip lines sampling strategy, which acts as a spatial trigger for phones to collect measurements and send updates. Another example is recorded ambulance GPS information (*22*). In this case, the GPS readings are stored every 200 meters or 240 seconds, whichever comes first. This is a combination of temporal sampling and spatial sampling.

Assuming the 16 vehicle trajectories in **Figure 4** are sampled by distance. Each vehicle reports its location and time at fixed distance D. According to the generalized definition of speed, the mean speed is the distance traversed by all vehicles divided by the total time spent. Because the distance traversed for each report is the same, that is D, the mean speed calculation method can be formulated as below,

$$\bar{v}_A = \frac{\sum_{i=1}^{n} d_i}{\sum_{i=1}^{n} t_i} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{N_i} D}{\sum_{i=1}^{n}\sum_{j=1}^{N_i} t_{ij}} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{N_i} 1}{\sum_{i=1}^{n}\sum_{j=1}^{N_i} \frac{t_{ij}}{\mathrm{D}}} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{N_i} 1}{\sum_{i=1}^{n}\sum_{j=1}^{N_i} \frac{1}{v_{ij}}} = 32.96mph \quad (10)$$

The time needed to traverse the distance D is t_{ij}, where $j = 1,2,\ldots,N_i$ represents the j^{th} report made within this region for each vehicle i. v_{ij} denotes the speed of vehicle i when making the j^{th} report. n is the number of vehicles traveling in the time-space region, N_i is the number of reports that vehicle i made within this region. $\sum_{i=1}^{n}\sum_{j=1}^{N_i} 1$ represents the total number of reports received for all vehicles in this region. Therefore, if GPS points are sampled at fixed distance intervals, the mean speed of a time-space region is expressed as the harmonic mean of all vehicle spot speeds within that region.

Evaluation of Accuracy

This section presents two case studies that apply the harmonic mean method and proposed methodology to compute mean probe vehicle speeds to demonstrate the accuracy of different methods. The two case studies represent two different traffic flow conditions - interrupted flow and uninterrupted flow, respectively. Because most probe data vendors collect GPS data at a constant time interval, we focus on examining time-based sampling strategy in this section. The analysis procedure can be easily applied to spatial sampling in future research.

Evaluation Method

To fully understand the errors caused by different speed aggregation methods, the trajectory data from the Next Generation Simulation (NGSIM) project (*1*) were adopted to simulate the probe vehicle sampling mechanism. The dataset provides complete vehicle trajectories traversing a road segment (**Figure 5**), thus ground truth speed is available for validation purposes.

Before the simulation, two concepts are adopted, which were introduced in this paper(*4*). Because not all vehicles on the road are equipped with GPS devices and act as probe vehicles, we assume that a proportion of the vehicles are probes. *Penetration rate* is used to measure the proportion, which describes the percent of vehicles that are probes. For example, 1% penetration rate indicates 1 of 100 vehicles acts as probes on average.

In addition, reporting frequency also affects the data amount received and thus have an impact on the mean speed accuracy. We introduce another concept to describe reporting frequency: *sampling frequency*. It is how often the selected probe vehicle reports its positions. We use sampling interval to measure sampling frequency. Specifically, under time-based sampling strategy, sampling interval denotes sampling time interval. A sampling interval of 30 seconds indicates that the probe vehicle reports their location every 30 seconds.

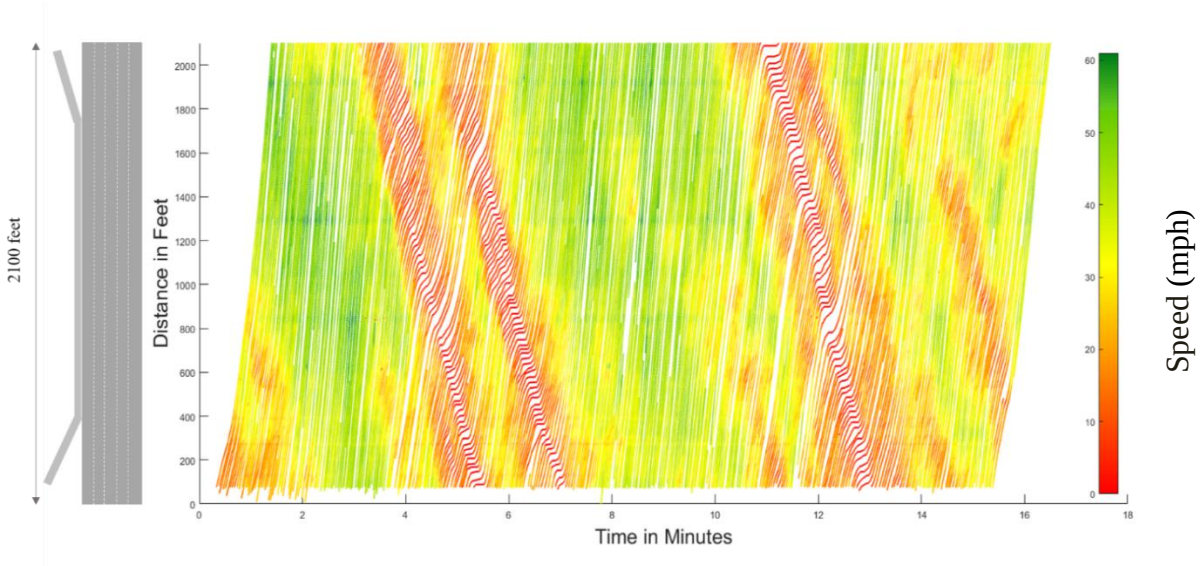


Figure 5. Vehicle trajectories of US 101 Highway, third lane.

Sampling Procedure

Step1: divide the entire time-space region into smaller regions. For each region, repeat step 2 to step 5.

Step 2: compute the number of vehicles to be sampled in each time-space region.

Step 3: randomly sample vehicles from the population.

Step 4: randomly generate timestamp that each sampled vehicle starts to report its location.

Step 5: collect the total sampled data points and compute average speed.

To assess the accuracy of different approaches, we adopted mean percentage error (MPE), and 10th percentile error:

$$MPE_A = \frac{\sum_{i=1}^{N} \frac{\overline{v}_i - v_A}{v_A}}{N} * 100\% \quad (11)$$

$$MPE = \frac{\sum_{i=1}^{M} MPE_i}{M} * 100\% \quad (12)$$

For each time-space region A, the error varies if a different set of vehicles are sampled, thus N is used to represent the number of sampling cases for a specific time-space region. v_A represents the ground truth speed for sampling case i for region A, $\overline{v}_i$ is the computed mean speed using either the arithmetic mean method or harmonic mean method. MPE_A is the mean percentage error for region A. Next, we average the mean

percentage error and obtain the mean percentage error for all time-space regions. In equation (12), M represents the number of time-space regions.

$$10^{th}\ percentile\ error_A = \left\{\frac{\overline{v_i} - v_A}{v_A}\right\}_{10^{th}\ percentile} * 100\% \quad (13)$$

$$10^{th}\ percentile\ error = \frac{\sum_{i=1}^{M} 110^{th}\ percentile\ error_i}{M} * 100\% \quad (14)$$

MPE demonstrates on average, the expected percentage error, 10th percentile error demonstrates the possible error range. Variance was not used because the results suggest the error is generally skewed towards the lower part, a 10th percentile error shows the possible variance more clearly.

Freeway

Eight video cameras were deployed on US Highway 101 to collect vehicle trajectories, covering a 2100 feet highway section during the congestions (7:50 a.m. to 8:35 a.m.). This road section has five lanes as demonstrated in **Figure 5**. The NGSIM data have been extracted from video recordings, consisting of trajectories of 6101 vehicles that traveled this section during the monitored period. Speed and position of each vehicle are available every 0.1 s. **Figure 5** is an illustration of the road segment being monitored as well as the vehicle trajectories traveling the segment from 7:05 to 8:20 in the middle lane. The color indicates different vehicle speeds, ranges from 0 mph to 60 mph.

Taking both the penetration rate and the sampling frequency into account, 96 scenarios were designed. The penetration rate ranges from 5% to 100%, 100% of the vehicles indicate that all vehicles are probes, 5% of the vehicles ensure that at least two vehicles are sampled during the time period. The sampling interval ranges from 0.1 seconds to 40 seconds. 0.1 second is the finest sampling interval because it is the resolution of the trajectory data. 40 seconds is the largest sampling interval because some vehicles traverse the segment within 40 seconds, a longer sampling interval results in zero reports and makes no sense. Next, we analyzed the errors caused by the harmonic mean aggregation approach under different combinations of penetration rate and sampling interval.

Because the best data resolution that many probe vendors can provide is 1 minute. We divided the time-space diagram into 45 regions, each region is composed of a 1-minute time extent and a 2000 feet spatial extent. The mean speed was computed with respect to each region. For each combination of a penetration rate, a sampling interval, and a time-space region, the sampling procedure was performed 100 times and the error was computed for each run. Next, the errors were averaged to obtain the mean error for the specific time-space region.

Computation Results and Some Discussions

The mean percentage errors of different approaches are shown in **Figure 6**. In the subfigures, each column represents a different sampling interval, ranges from 40s to 0.1s, each row represents a probe vehicle penetration rate, ranges from 5% to 100%. (a) and (b) separately demonstrate the MPE of the arithmetic mean approach and the harmonic

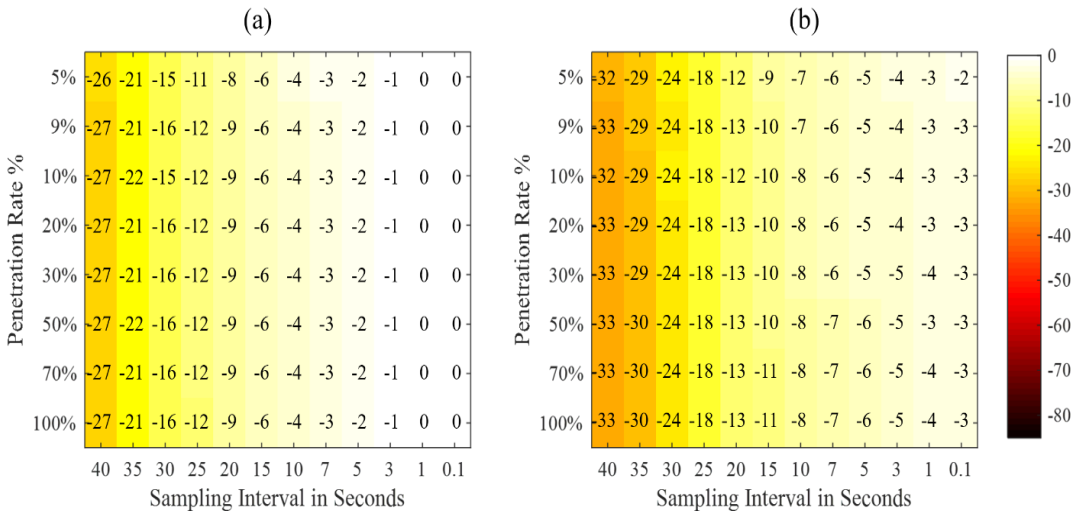


Figure 6. Mean percentage error of two methods. (a), arithmetic mean approach; (b), harmonic mean approach.

mean approach. On average, the MPE of the arithmetic mean and harmonic mean does not demonstrate a signification change with the increase in the penetration rate but improves with finer sampling interval. From both (a) and (b), it appears that the mean speed collected based on temporal tends to be consistently lower than the ground truth. This is partly caused by the time-based sampling strategy. Vehicles with lower speed are more likely to be sampled compared to the high-speed vehicles if time-based sampling strategy is implemented. Especially in the case of low-frequency probe vehicle data, regardless of the penetration rate. An extreme example is, considering several vehicles that traverse a short segment, if the traverse time for high-speed vehicles is shorter than the sampling interval, then the speed of these vehicles will not be recorded, thus only those vehicles with traverse times that shorter than sampling interval (that is lower speed vehicles) are sampled. Therefore, without considering the GPS measurement errors, a time-based sampling strategy is likely to underestimate the mean speed for a segment in a certain time period. The impacts of the underestimation on mean speed depends on the length of the segment as well as the report time interval.

By comparing (a) with (b), the errors of the harmonic mean method tend to be higher than that of the arithmetic mean method. It is demonstrated that, as expected, the average speed estimated by harmonic mean is less accurate than that estimated by the arithmetic mean approach, the harmonic mean approach constantly underestimates the mean speed from 0% percent to 8% compared to the arithmetic mean method. In **Figure 6**, The lower right corner represents the perfect case, when the penetration rate is 100% and the sampling interval is the finest, 0.1 seconds, indicating we have complete information for each vehicle within the time-space region. In this case, the arithmetic mean estimation error is zero and the harmonic mean estimation error is -4.1%. The

estimation errors are largest when the sampling interval is around 30~40 seconds. In reality, the estimation error differences are influenced by many factors, such as the ratio of the road segment length to the aggregation interval, a long segment length and a short aggregation interval might introduce more errors because fewer vehicles completely traverse the entire road segment. Moreover, the speed range and variation could also affect the accuracy as well.

Figure 7 demonstrate the 10^{th} percentile accuracy for both speed aggregation methods. Again, the arithmetic mean approach constantly outperforms harmonic mean approach. Indicating the arithmetic mean approach is more reliable compared to the harmonic mean approach.

Signalized Intersections

Data used for signalized intersections represent travel on Peachtree Street, an arterial running primarily north-south in Atlanta, Georgia. The speed limit on Peachtree Street is 35mph. Eight video cameras were deployed to collect vehicle trajectories, covering a 2100 feet road segment (4:00 p.m. to 4:15 p.m.). The NGSIM data consist of trajectories of 325 vehicles that traveled this section during the monitored period. Speed and position of each vehicle are available every 0.1 s. **Figure 8** is an illustration of the vehicle trajectories extracted from the video. The color indicates different vehicle speeds, ranges from 0 mph to 50 mph. There are five intersections along this road segment, four can be easily identified on the trajectory figure as vehicles slow down and queue up.

Computation Results and Some Discussions

The time-space diagram is divided into 15 regions, each region is composed of a 1-minute time extent and a 2000 feet spatial extent. The time-space region ranges from 4:00pm to 4:01pm, 4:01pm to 4:02pm, till 4:14pm to 4:15pm for the entire segment. The mean speed was computed with respect to each region.

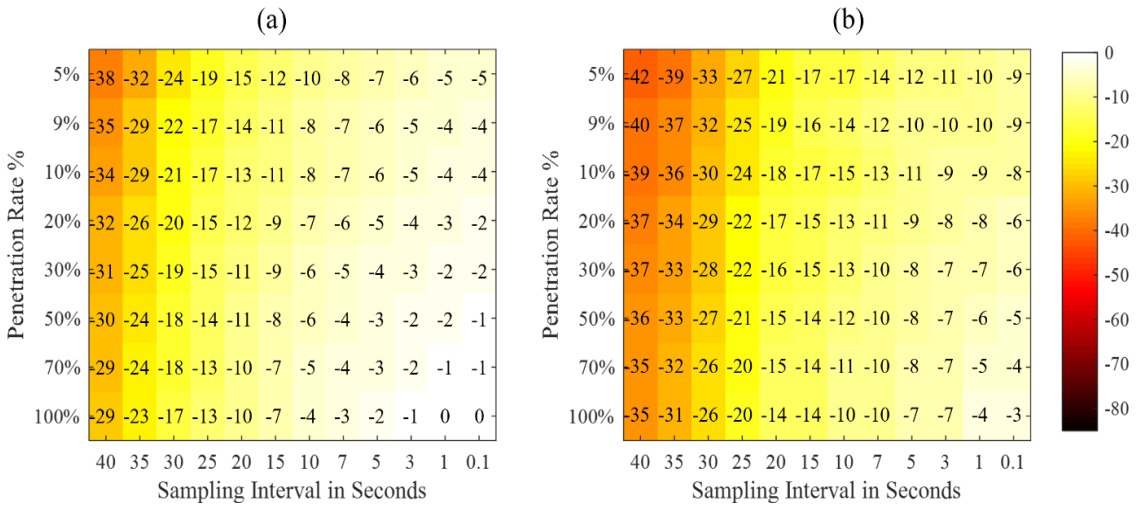


Figure 7. 10th percentile percentage error of two methods. (a), arithmetic mean approach; (b), harmonic mean approach.

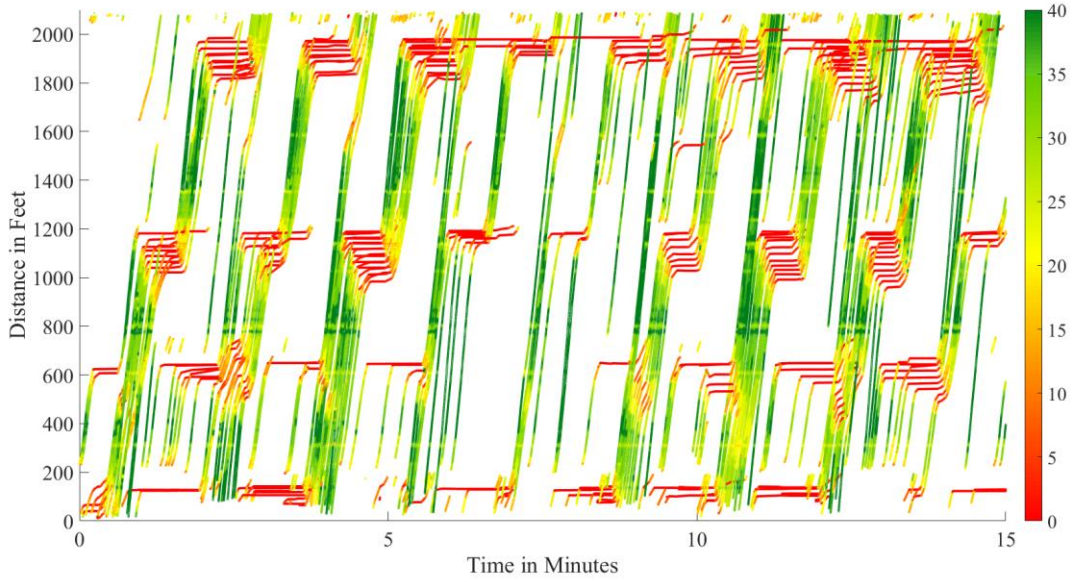


Figure 8. Vehicle trajectories on Peachtree Street, northbound.

Similar to the previous case study, 96 scenarios were designed based on the penetration rate and sampling interval. The penetration rate ranges from 5% to 100%, 5% is the smallest penetration rate which ensures that at least one vehicle is sampled during the time period. The sampling interval ranges from 0.1 seconds to 40 seconds. The sampling procedure was performed 25 times for each time-space region to ensure every vehicle is at least sampled once. Then, the error was computed for each time-space region and each run. The errors were then averaged to obtain the mean error.

Figure 9 demonstrates the mean percentage error for different combinations of penetration rate and sampling interval. Comparing **Figure 9(a)** with **Figure 9(b)**, it is apparent that in general, the arithmetic mean method is more accurate than the harmonic mean methods. Both methods tend to underestimate vehicle speed, especially for large sampling intervals. For the arithmetic mean method, the accuracy decreases with the increase of the sampling interval. For the harmonic mean method, the accuracy varies and does not have an obvious pattern. In general, the higher the penetration rate, the lower the accuracy. This is because the harmonic mean estimation is biased, the higher the penetration rate, the calculated average speed approaches the true harmonic mean speed but deviates from the true mean speed. As demonstrated, the error can be as high as 70% lower than the ground truth if the harmonic mean method is adopted (when the penetration rate is 100% and the sampling interval is 5 to 10 seconds. In the meanwhile, the arithmetic mean method can achieve less than 10% errors.

Figure 10 demonstrates a 10th percentile percentage error for different combinations of penetration rate and sampling interval to illustrate the error range introduced by different methods. Comparing Figure 10**(a)** with **Figure 9(a)**, similar

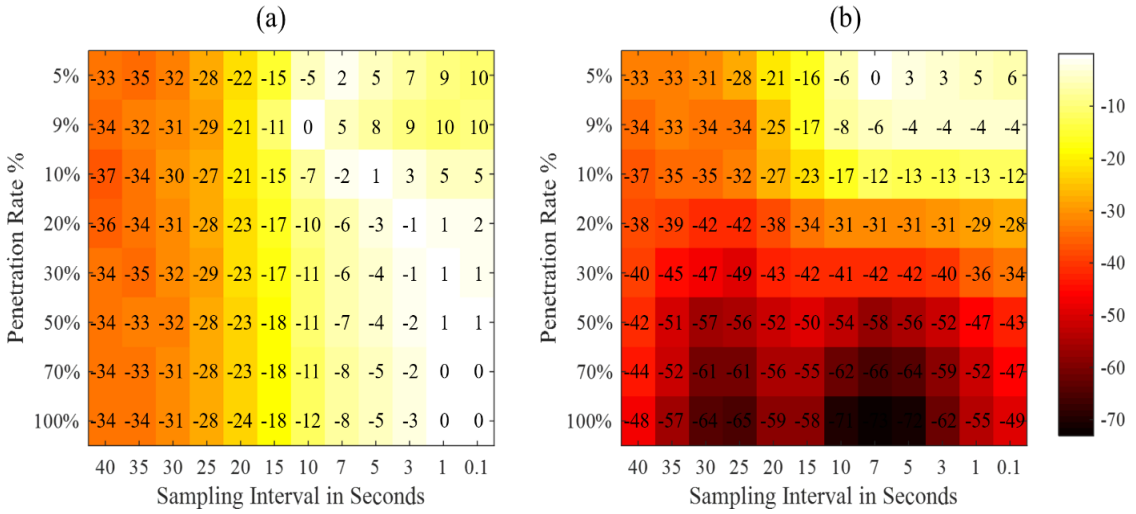


Figure 9. Mean percentage error of two methods. (a), arithmetic mean approach; (b), harmonic mean method.

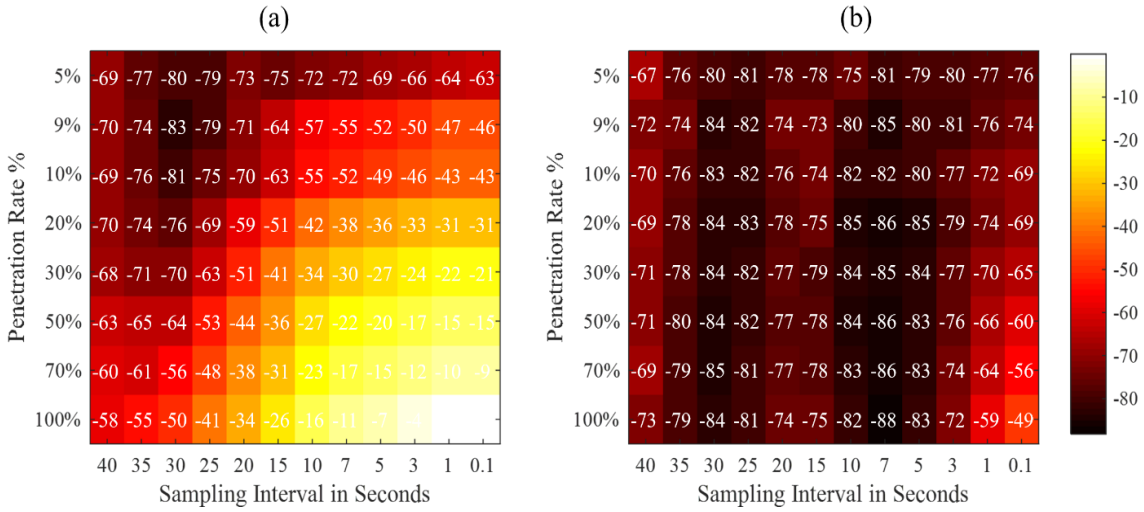


Figure 10. 10th percentile error of two methods. (a), arithmetic mean approach; (b), harmonic mean method.

patterns can be observed that with the increase of sampling interval, the less accurate the speed average results. Also, with the decrease of penetration rate, while the mean error (**Figure 9(a)**) does not change much, the 10th percentile accuracy decreases obviously. For example, for the 15s sampling interval, the accuracy decreases from -26% to -75%. Comparing Figure 10**(a)** with Figure 10**(b)**, the 10th percentile error for the arithmetic mean method is still smaller than that of a harmonic mean method, demonstrating that the proposed method is not only more accurate on average but also more reliable.

Both case studies demonstrate the harmonic mean approach is a biased estimation of the true speed, and the proposed arithmetic mean method is more accurate and more reliable than the harmonic mean method. Interestingly, Comparing the error of signalized intersections to that of a freeway, while the error for the arithmetic mean method does not deviate much, the error for the harmonic mean method are quite different from each other. The errors introduced by the harmonic mean method for signalized intersections is higher than that for a freeway. This can be partially explained that vehicle trajectories for a freeway are more homogeneous than that for signalized intersections in the case study.

Conclusions

So far, the mean speed computation method for probe vehicle data has received little attention. The practice that harmonic mean is typically used for roadside sensor speed aggregation sometimes leads to a misunderstanding that associates the mean speed with harmonic mean of traffic streams. Because of this, some commercial probe data

providers developed speed aggregation method by taking the harmonic mean of individual vehicle speeds in the aggregation period. We demonstrate in this study that; the harmonic mean approach is a biased estimator and may cause erroneous results. It is later shown that, for data obtained from probe vehicles, the mean speed aggregation method depends on the data sampling strategy. If the sampling is done by distance, harmonic mean should be used to aggregate spot vehicle speeds. Conversely, if the sampling is done by time, the arithmetic mean approach should be used to aggregate spot vehicle speeds. Moreover, regardless of the sampling schemes, total distances traveled within the time-space region divided by total times of travel is always the correct way to compute mean vehicle speeds, this is particularly preferred when the GPS data received have irregular reporting frequencies.

The case study adopted NGSIM data to demonstrate the possible errors if different aggregation method is used. The results show that the time-based sampling strategy prone to sample low-speed vehicles than high-speed vehicles and leads to an underestimation of the mean speed. Additionally, compared to the arithmetic mean method, the harmonic mean approach under a time-based sampling strategy is likely to result in a lower estimation of the mean speed. In the case study, the harmonic mean approach results in a lower estimation of 0% to 8% compared to the arithmetic mean approach for the freeway case and as high as 76% for the signalized intersection case. The results have implications for both practitioners and researchers. Since the average speed calculation methods for many crowdsource-based speeds data remain unclear, the case study section offers practitioners and researchers an understanding of the error range if the traditional calculation method were adopted. In addition, this study also has implications for future speed, flow, and density calculation. Regardless of the data collection technologies, the key point is to derive the calculation method based on the definition, instead of directly employing an existing method.

This study focused on aggregating average speed based on individual data points. One of the future research directions is to investigate how to correctly aggregate the ‘average’ speed. For example, aggregate the average speed for continuous road segments to obtain the average speed for a road facility. Another future research direction is to estimate speed variance from average speed over time. In this study, it is demonstrated that while the mean speeds are close, the variances could be quite different. The variance provides further insights into the traffic states.

CHAPTER II. EVALUATION OF CROWDSOURCED EVENT REPORTS FOR REAL-TIME IMPLEMENTATION – SPATIAL AND TEMPORAL ACCURACY ANALYSIS

A version of this chapter was presented at TRB 2020 by Yuandong Liu, Nima Hoseinzadeh, Lee D. Han, and Candace Brakewood.

Abstract

With the ubiquitous use of mobile devices, road users can contribute to traffic knowledge via multiple ways, such as sharing location information and posting traffic conditions on social media. Traffic monitoring agencies are increasingly aware of the importance of crowdsourced data and consider them as a possible complementary data source for real-time operations. However, the data might have redundant, inaccurate, and incorrect information that needs to be validated before being incorporated into real-time applications.

The validity of crowdsourced data involves multiple aspects, and this study focused on report timeliness and location accuracy of Waze data. We retrieved the data from five months of Waze reports on the interstate highway in Tennessee and compared them with existing official records. The results indicate that 67% of the crash reports (85% of the stopped-vehicle report) in official records can be matched in Waze. Among all official records, 40% crash reports (57% stopped-vehicle reports) are reported sooner on Waze than in official records. On average, Waze reports a crash 2.2 minutes earlier and report a stopped vehicle 8 minutes earlier, which highlights the potential application of Waze reports in real-time event detection. Also, the results reveal the high location accuracy of Waze reports. On average, the distance difference between Waze reports and official records is less than 0.001-mi for all crash reports. The finding suggests that users tend to make a report at the exact location, which has implications for several other types of reports, whose location is hard to be evaluated directly.

Introduction

The vast number of mobile devices in use has allowed people to contribute to the collective knowledge by sharing locations or directly posting traffic information on social media. The crowdsource-based applications can now effectively collect a variety of traffic information with a high degree of accuracy.

In general, there are two types of crowdsourced data. First, users allow their location information to be collected while do not share any information actively. Some navigation applications are able to integrate those passive data collected from users who have activated the application and provide traffic speed. Second, users actively report the traffic conditions they observed on social media such as Twitter. The first type of crowdsourced data is collected on a routine basis and forms the foundation of probe-vehicle based speed data, the accuracy and reliability of which has been comprehensively evaluated. The second type, posting traffic condition related information on social media relies on users' active participation, and the accuracy and reliability depend on user behavior, which has not been thoroughly evaluated.

Traffic managers strive for the most accurate information about road hazards (crashes, roadside vehicles, etc.) to actively respond to these incidences. Timely response to those events can improve the efficiency, safety, and reliability of the transportation system. In general, hazard information comes from public reports, law enforcement, traffic monitoring devices such as CCTV, and automated event detection algorithms based on real-time data. The high cost, sparsity, and limited spatial coverages of physical roadside sensors and devices raise considerable challenges to effectively detect traffic events in a timely manner, especially in rural areas. In contrast, users' prompt reporting is low cost and plays an important role.

Traditionally, users can make reports to 511 or the police, but with the prevalence of crowdsourced data, users can contribute in multiple ways, not only making a phone call but also posting information on social media. The Waze app was developed to allow users to post traffic conditions on its platform and is then disseminated to Waze users. In recent years, Waze has established partnerships with numerous transportation agencies and provides them rich information, including traffic speed and user reports. Reports are made by Waze users once they observe something on the road. Waze supports multiple reports categories including traffic jams, crashes, road hazards. Traffic agencies are eager to incorporate this information into their traffic management and operation systems, especially the users' reports. Compared to the operation and maintenance cost of other traveler information and hazard detection systems, Waze is free. Several states have attempted to use Waze reports as a source of incident detection (*6; 25*). However, transportation agencies are hesitant to incorporate the data into their system before the reliability and accuracy of Waze reports are evaluated. Therefore, validating the accuracy of Waze data to ensure its credibility and quality is crucial.

Literature Review

Social media data have demonstrated potential in detecting special event occurrences such as earthquakes promptly (*26*). In recent years, a variety of research has been conducted to mine and analyze twitter data to extract traffic event information that will aid in real-time event detection. Most of these studies focused on developing a model to extract the traffic related information (*27-30*). The paper proposed a general approach to mining twitter data and concluded that mining tweets hold great potentials to complement existing traffic incident data in a very inexpensive way (*27*). Some interesting facts related to social media traffic event data have been revealed in these studies. Overall, individuals tend to report incidents more frequently during the daytime than at night, especially during traffic peak hours. Within a given week, social media data are posted more often on weekends than on weekdays. Also, in terms of spatial distribution, arterials receive more extensive coverage on social media. Previous studies provide a general discussion of the extracted information, but did not provide much insight into the traffic information, such as its timeliness, its reliability etc.

Whereas extracting social media data has been a hot topic in recent years, limited research has been found to evaluate Waze data. Jussara provides a general understanding

of Waze data and identifies its spatial coverage as well as its limitations (*31*). Three recent papers evaluate Waze crash reports by comparing them with official data.

One of the papers estimates traffic crash counts using Waze crash counts. It links Waze events to official reports and develops a random forest model to estimate the number of police-reported crashes. The estimated crashes from the models have similar spatial and temporal patterns compared to the observed police reported crashes. The models appear to capture unreported crashes, including minor crashes that might not require a police presence but can seriously impact congestion (*32*). The paper is an application of Waze data and does not directly address the reliability issues.

The second paper focuses on the comprehensive assessment of the event reports (incident reports and traffic jam reports), including reliability and coverage, and developed a methodology to find the added value of Waze reports (*6*). This paper comprehensively evaluates the Waze reports; it states that the reports are reasonably accurate geographically, but do not provide detailed information about the accuracy of the location. Also, the paper does not distinguish urban roads and interstate highways. Analysis of Waze reports might show urban roads and interstates with different patterns.

The third paper (*5*) compares Waze crash and disabled vehicle records with video ground truth; it is the first paper to evaluate Waze data with ground truth data, although the conclusion is limited by the sample size, which may be because extracting ground truth from video is strenuous. It is interesting to see that the false alarm rates for incidents are much lower than those for disabled vehicles. The study also found that the Traffic Operation Center was generally aware of crashes and disabled vehicles before they were reported in Waze, which contradicts the findings of the earlier paper.

In summary, previous research found that Waze reports are usually a reliable source for monitoring road conditions during the daytime. Waze could capture unreported events and is reasonably accurate in terms of time and location. This suggests that Waze is a promising data source for real-time event detection.

Research Question

The reliability of Waze has two implications. First, if there is an event, what is the possibility it will trigger a report? Analyzing the percentage of total events reported by Waze can provide insights into this. As shown in this paper (*6*), 43% of ATMS crash and congestion reports are covered by Waze reports, in other words, the likelihood of having a Waze report is 43%. Second, if a report is made, does it represent an actual event? We discuss the second aspect in detail below.

1. If a report is made, does it represent an actual event? There are several cases where a report does not indicate an actual event.

- **False alarm.** Some reports do not denote an actual event, an error that a user could make either intentionally or unintentionally. Verification of the event reports will need information from other sources such as roadside sensors and video cameras. In his study, Mostafa (*6*) uses cameras and finds that 0.3% of the reports were false alarms. Noah, in his study, found 5% of 40 crashes were false alarms. This false alarm rate is lower than expected and suggests that false alarms might not be a predominant problem for Waze.

- **Duplicate reports.** One characteristic of Waze data is that users might make multiple reports about the same event at different times and locations. For example, for a crash lasts for hours and causes long delay, users keep making reports until the crash is cleared. The reports could be made at the location where the crash occurred or made within the traffic queue caused by the crash. It is an arduous task for traffic practitioners to determine whether the report is a duplicate of an earlier report or it indicates a new accident.
- **Cleared event and retained reports.** Waze may continue to show events in locations after events have been cleared. This is not an issue for practitioners since the report is validated after it is made, but more of an issue to Waze users: If reports are not cleared in time, they can be regarded as false information to the users and erode user trust in Waze reports.

2. How accurate are Waze event reports in terms of time and space? This means how well an event report matches the time-space existence of an actual event.

- **Temporal accuracy.** Temporal accuracy means the start time and end time of a report match the start and end times of the corresponding event. Transportation agencies are generally more interested in the consistency of start times because end times are known to transportation agencies, and they can provide that information to Waze. However, the exact occurrence time is usually unknown.
- **Spatial accuracy.** Because the reports are made by users, several factors might affect location accuracy. A user could make reports before they reach the location or after they pass the location. The location accuracy shall be examined to provide traffic practitioners a sense of where the event is when getting a report.

This study examined the spatial and temporal accuracy of Waze reports. In Waze, a variety of report types are provided, including crash, weather hazard, traffic congestion, stopped vehicle, police, and closure, etc. Among the nine types of reports, only reports concerning crashes and stopped vehicle occurrences at a fixed location are maintained as official records. Therefore, these two types of reports were selected and examined for this paper. The results could have implications for other types of reports, such as traffic jam reports. This study evaluates Waze reports on interstate highways, which has not been looked at in many previous studies. Tennessee Department of Transportation (TDOT) maintains official documentation for crashes and abandoned vehicles on freeways. A matching criterion is then developed to link Waze reports to official records to evaluate the temporal and spatial accuracy. Researchers can use the same process to analyze the reports on urban road networks. In addition, the study discussed several challenges to incorporate Waze data into real-time operations.

Methodology

Data

The data used in this study are Waze report data and LocateIM incident data.

Waze data: Waze (https://www.Waze.com/about) is a navigation application that leverages crowdsourced user reports to provide service to its users. Users can report traffic crashes, congestion, and road hazards. In 2016, TDOT established a partnership with Waze to share traffic information. The data provided to TDOT is user reports feed. Each report provides type, location, time, and measurements of reliability. (https://support.google.com/Waze/partners/answer/6324421?hl=en).

LocateIM data: TDOT records traffic events on the interstate highway, including location, time, and duration. The dataset is validated by TDOT operators and thus serves as a reference to evaluate Waze data. However, the events recorded in this dataset may not cover all cases. For example, some crashes do not involve the police or transportation agencies but still exist and have an impact on traffic.

The Waze data used in this study covers five months of crash and stopped vehicle reports on all of I-40 in Tennessee from August 1st to December 27th,2017. So does the LocateIM. The number of reports retrieved for each data type are shown in **Table 3**. In LocateIM, the location of a report is represented by milepost, while in Waze, the location of a report is represented by longitude and latitude. The location of Waze reports is first converted to milepost for future analysis.

Matching Methodology

This study matches Waze reports to LocateIM records and evaluates the temporal and spatial accuracy of Waze reports. If two records are close enough in both time and space, a match is established. The proximity of the two data sources is determined in terms of both temporal and spatial differences.

LocateIM and Waze use different terminologies for traffic events; for example, Waze provides two types of stopped vehicle reports, *vehicle stopped on road* and *vehicle stopped on roadside*. Correspondingly, LocateIM has two types of records that can be matched to the Waze stopped vehicle reports, separately *disabled vehicle* and *abandoned vehicle*. Waze denotes crashes as *accident* reports while LocateIM uses *single vehicle crash, multi-vehicle crash,* etc. We first match Waze terminologies to LocateIM terminologies. It is expected that stopped vehicle on road has a similar pattern to accident reports while stopped vehicle on roadside might manifest different patterns. However, in LocateIM, the two types of reports cannot be differentiated, therefore we combined them for future analysis. **Table 4** shows the terminology match used in this study.

Table 3. Number of reports for each data type.

	Waze	*LocateIM*
Accident	8,068	2,052
Stopped vehicle	93,707	5,459

Table 4. **Terminology match.**

Waze	*LocateIM*
Accident	Crash, Overturned vehicle, Vehicle on fire
Stopped vehicle	Disabled vehicle, abandoned vehicle

The report time and location for LocateIM records are separately denoted as $T_{LocateIM}, D_{LocateIM}$. The report time and location for Waze reports are denoted as T_{WAZE}, D_{WAZE}. The time difference between LocateIM and Waze report is $\Delta T = T_{WAZE} - T_{LocateIM}$. The distance difference between LocateIM and Waze report is $\Delta D = D_{WAZE} - D_{LocateIM}$.If the time different ΔT and spatial difference ΔD are within a threshold, T_{thre}, D_{thre}, Then a potential match is established.

Road name is a possible matching criterion, but because all the events happened on I-40, road name is not one of the matching criteria in this study. Also, it happens sometimes that a Waze user observes an event (especially a crash) traveling in the opposite direction and reports it. The proposed matching criteria take this into account and match reports traveling in both directions. The matching algorithm pseudocode is shown in **Table 5.**

Each LocateIM report keeps a matching list to store matched Waze reports. Following the previous steps, one Waze report could be matched to more than one LocateIM report while one LocateIM could have multiple Waze matches. To eliminate some unlikely matches and improve the accuracy of the matching results, we propose additional matching criteria. **Table 6** demonstrates two different scenarios and corresponding matching criteria.

In summary, if the Waze report is downstream from LocateIM records, that is $\Delta D > 0$, then, abs(ΔD /ΔT) < max vehicle speed, if abs(ΔD /ΔT) >max vehicle speed, means that Waze user need to travel at a speed that is higher than the max vehicle speed to make the reports at the current location, which is unlikely. Similarly, If the Waze report is

Table 5. **Pseudocode.**

```
For each report in LocateIM:
   For each report in Waze:
      Compute ΔT = T_WAZE − T_LocateIM
      If abs(ΔT) < T_thre:
         Compute ΔD = D_WAZE − D_LocateIM
         If abs(ΔD) < D_thre:
            Add the Waze reports to the matching list of LocateIM reports
```

Table 6. **Matching criteria for different scenarios**.

Scenario1	*Scenario 2*
$*$ ΔD >**0**	ΔD <**0**
Abs (ΔD /ΔT) < max vehicle speed	Abs (ΔD /ΔT) < max shockwave speed

Note*:

- $\Delta D = D_{WAZE} - D_{LocateIM}$. $\Delta D > 0$, indicates Waze report is downstream to LocateIM report, otherwise, Waze report is upstream to LocateIM report.

upstream from LocateIM records, that is $\Delta D < 0$, then abs (ΔD /ΔT) < max shockwave speed. If abs (ΔD /ΔT)> max shockwave speed, means Waze user makes report about the event even the queue hasn't backed up to his/her current location. Which is also unlikely. The maximum vehicle speed and maximum shockwave speed are set to 90mph and 15mph based on our experience.

Results

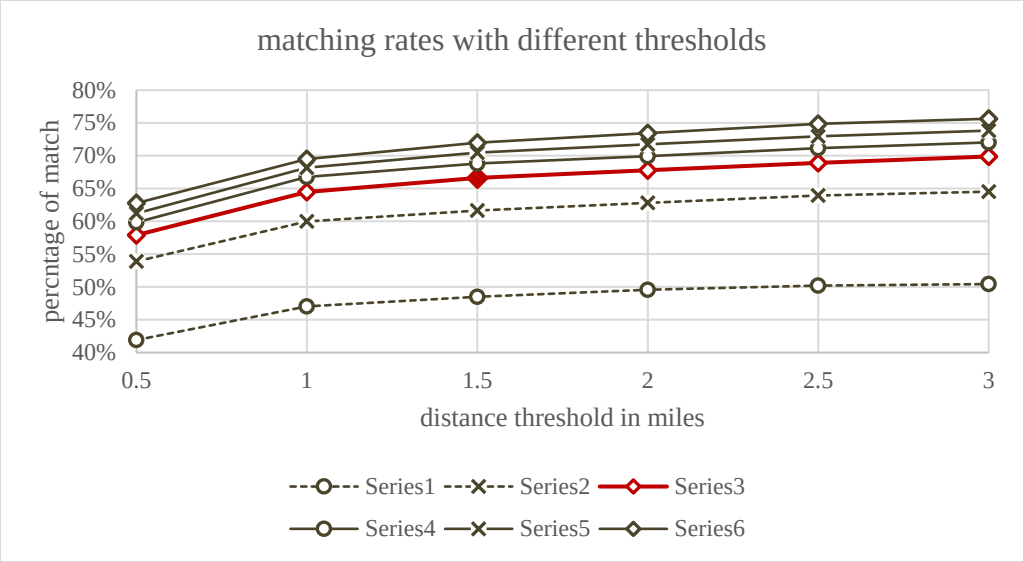
This section includes three subsections. First, we discuss the process to establish the temporal-spatial threshold. Then, the matching results, as well as temporal-spatial accuracy of crash reports and stopped vehicle reports, are presented.

Establish a Threshold

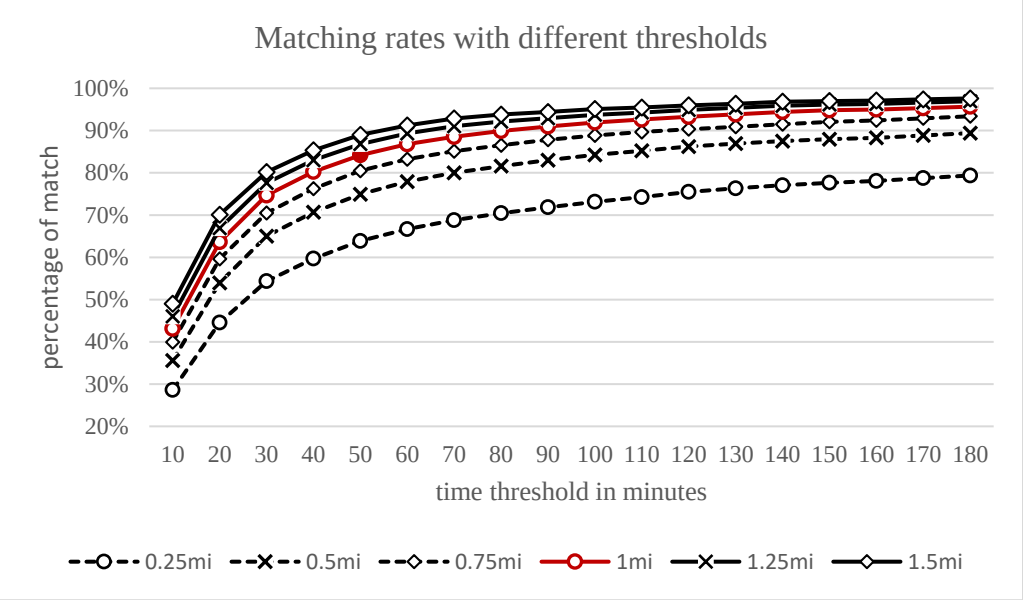
The matching criteria require a reasonable threshold for time and distance differences. A group of threshold combinations is evaluated, as shown in **Figure 11**, which demonstrates the matching rate with different time and distance thresholds.

According to the matching methodology and thresholds determined in the previous section, the Waze crash reports are linked to LocateIM crash records reports. Each crash record could have more than one Waze match. Among all the matched Waze reports, only the earliest report is taken into consideration in the time and distance differences analysis because it is the first Waze report made for the crash; it is called the first match.

Figure 11 (a) shows that the matching rate increases with the increase of time threshold, but the increment in matching rate decreases with the increase of time threshold. When the threshold exceeds 30 minutes, the increased matching rate is not obvious. Similarly, the increase in matching rate levels off when the distance threshold exceeds 1.5 miles. Therefore, 30 minutes and 1.5 miles are selected as the threshold for crash matches; any reports within this range will be matched to LocateIM records. The threshold results in 67% matching rates, meaning that 67% of the crashes in LocateIM have at least one match in Waze. Figure 11 (b) illustrates a slightly different pattern. The distance threshold is chosen as 1.0 mile, and the matching rate does change much with the increase of distance.



(a)



(b)

Figure 11. Matching rate with different combinations of thresholds. (a) crashes (b) stopped vehicles.

The time threshold is also different. The pattern shows that when the time threshold reaches 50 minutes, the matching rate begins to stabilize. Therefore, 50 minutes is chosen as the threshold. The longer time threshold can be intuitively explained. Unlike crash, it takes a longer time for traffic agencies to respond to disabled vehicles or abandoned vehicles on roadside. Spatial and Temporal Accuracy for Crash Reports.

Spatial and Temporal Accuracy for Accident Reports

This section presents the spatial-temporal accuracy for accident reports. **Figure 12** shows the characteristics of the first match. The time difference is the time between Waze reports and LocateIM reports. It is a heatmap plot that shows the distribution of time and distance differences between the two sources. Negative time suggests that a Waze report occurred earlier than a LocateIM report. Distance difference is the distance between Waze reports and LocateIM reports. Negative distance suggests that a Waze report is upstream of a LocateIM report.

As shown in **Figure 12**, Waze report time is, on average, 2.2 minutes earlier than that of LocateIM. This does not suggest an obvious gain. **Table 7** displays in detail the number and percentage of reports that Wazers made sooner than LocateIM users. It can be observed that, among all LocateIM reports that can be matched, over 60% of them were reported sooner on Waze. It is often the case that traffic engineers are aware of the crash several minutes before a report is verified and log it into the system. In that sense, the percentage of reports that were made 0~ 20 minutes sooner than LocateIM is presented in **Table 7** as well. For example, Waze reports that were made five minutes earlier than LocateIM reports constitute 35% of all the LocateIM records being matched. This suggests that Waze has the potential to be used as one of the real-time sources for crash detection and may help to decrease crash identification time. However, a high percentage of the early reports are within -10 ~ 0 minutes, a mechanism that can quickly verify Waze reports is needed to take advantage of the short time difference.

Figure 13(b) shows the distance differences between the two sources. On average, the distance between Waze and LocateIM reports is -0.001miles (less than 6 feet), which suggests a high location accuracy of Waze crash reports. A normality test was conducted, and the results show that the distance between Waze and LocateIM reports does not follow a normal distribution; the distance is more concentrated compared to normal distribution**. Figure 13(a)** demonstrates the cumulative distribution of absolute distance difference. Forty percent of the crash reports from Waze is within 0.1 miles of the exact location. Sixty percent of those are within 0.2 miles. While on average, users tend to make reports at the exact location, some variance exists. It is worth noticing that the LocateIM reports location precision is 0.1 miles, which might have impact on the final accuracy.

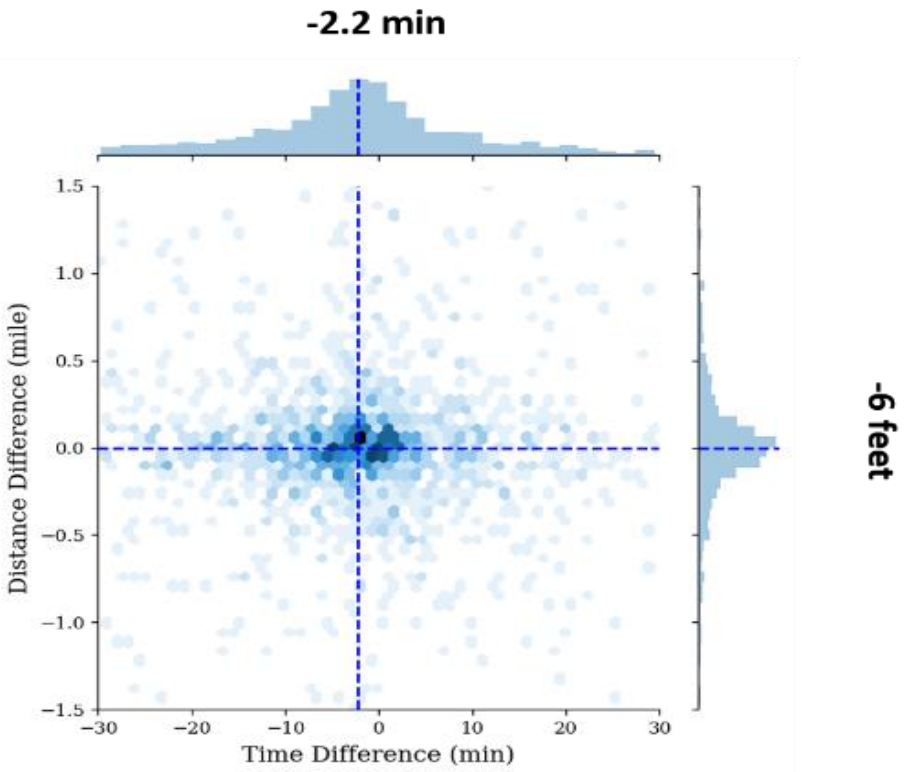
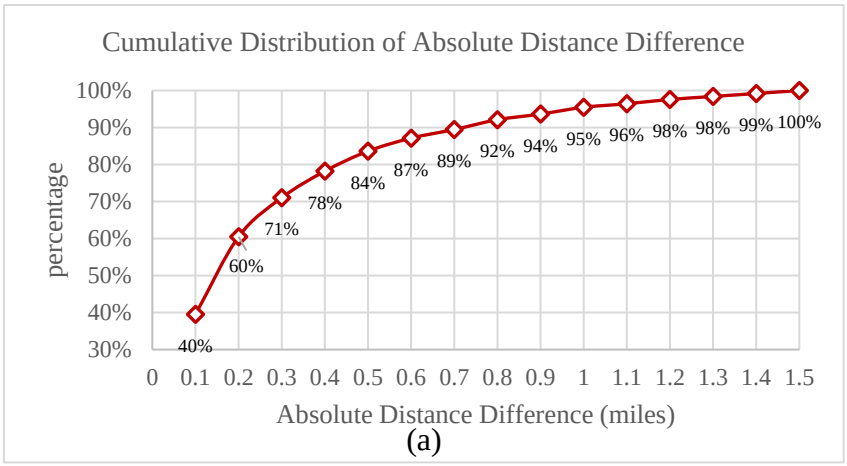


Figure 12. Heatmap of the spatial and temporal differences between Waze reports and LocateIM records.

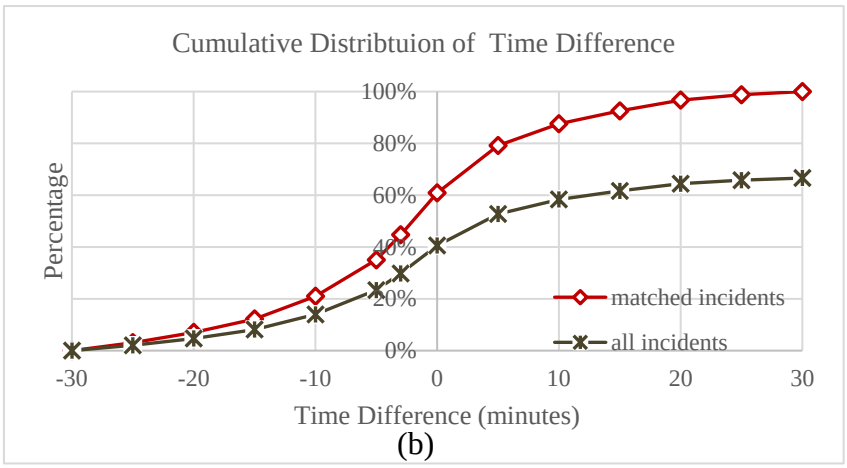
Table 7. Number and percentage of Waze report made earlier than LocateIM.

Time difference (min)	*Number*	*Percentage*
*-0**	832	60.9%
-3	611	44.7%
-5	479	35.0%
-10	287	21.0%
-15	167	12.2%
-20	96	7.0%

Note: *, *-3 indicate Waze reports 3 minutes earlier than LocateIM.*



(a)



(b)

Figure 13. (a) Cumulative distribution of absolute distance differences. (b) Cumulative distribution of time differences.

Location accuracy has some implications. As mentioned, for a certain type of reports, such as traffic jam reports and police reports, it is usually hard to determine the exact location (e.g., a police car is moving, traffic jam is growing) and no official record is maintained. This study reveals user behavior and gives some implications of the location accuracy for those types of reports. The location accuracy analysis also helps Waze to determine the user alert distance. Waze usually alerts its users 0.5 miles in advance of the reports. However, some Waze users complain that the alert distance is too short to take any action(https://www.waze.com/forum/viewforum.php?f=657&sid=458b46071de8cb72d1d7dc13fb613286). Our results suggest that around 53% of the crash reports are downstream from the exact location, which should be considered by Waze when setting up user alert distances.

Spatial and Temporal Accuracy for Stopped Vehicle Reports

In a previous section, we examined the spatial and temporal accuracy of crash reports. In this section, the same analysis was performed to evaluate the stopped vehicle reports. Stopped vehicle reports are evaluated independently because traffic agencies take prompt reacts to crashes but not stopped vehicles unless it is reported especially for vehicle stopped on roadside, thus the time and location difference may demonstrate different patterns. According to matching methodology and threshold determined, the Waze stopped reports are linked to LocateIM records. Each crash record could have more than one Waze match. Using the selected threshold, 86% of crashes in LocateIM have at least one corresponding report in Waze. The coverage for stopped vehicle reports is higher than that of accident reports.

Similarly, to examine the spatial and temporal accuracy, only the earliest match in Waze is taken into consideration. All remaining matches are regarded as duplicate reports. **Figure 14** shows the characteristics of the matched reports. **Figure 14** is a heatmap plot that shows the distribution of time difference and distance difference between the two sources**.** on average, Waze report time is eight minutes earlier than that of LocateIM for stopped vehicles. **Table 8** shows in detail the number and percentage of reports that Wazers made sooner than LocateIM users. It can be observed that, among all the matched reports, over 66% of the time Waze generates reports in advance of LocateIM among all LocateIM records. The percentage of reports that made 5~45 minutes earlier than LocateIM is presented in **Table 8** as well. For example, Waze reports that is 5 minutes earlier than LocateIM reports constitute 53% of all the LocateIM records.

The distance differences between the two sources are shown in **Figure 14(b)**. On average, the distance between Waze and LocateIM reports is -0.025miles (less than 140 feet); while this difference is higher than that of crash reports, this still suggests comparatively high accuracy. **Figure 15** displays the cumulative distribution of absolute distance difference. Sixty-five percent of the stopped vehicle reports from Waze is within 0.1 miles of the exact location. Seventy-one percent of those are within 0.2miles.

Table 8. **Number and percentage of Waze report earlier than LocateIM.**

Time difference (min)	*Number*	*Percentage (matched reports)*
*-0**	3090	66%
-5	2498	53%
-15	1618	35%
-20	1003	21%
-35	533	11%
-45	172	4%

* Note: *-5 indicate Waze reports 5 minutes earlier than LocateIM.*

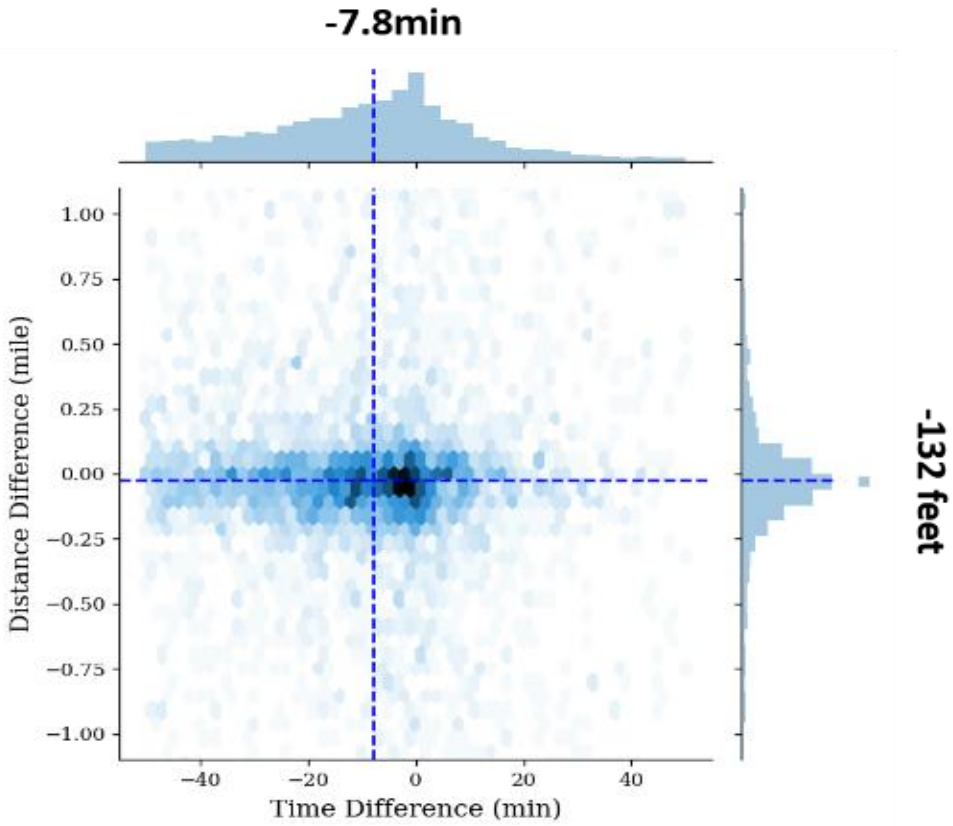


Figure 14. **Heatmap of the spatial and temporal differences between Waze reports and LocateIM records.**

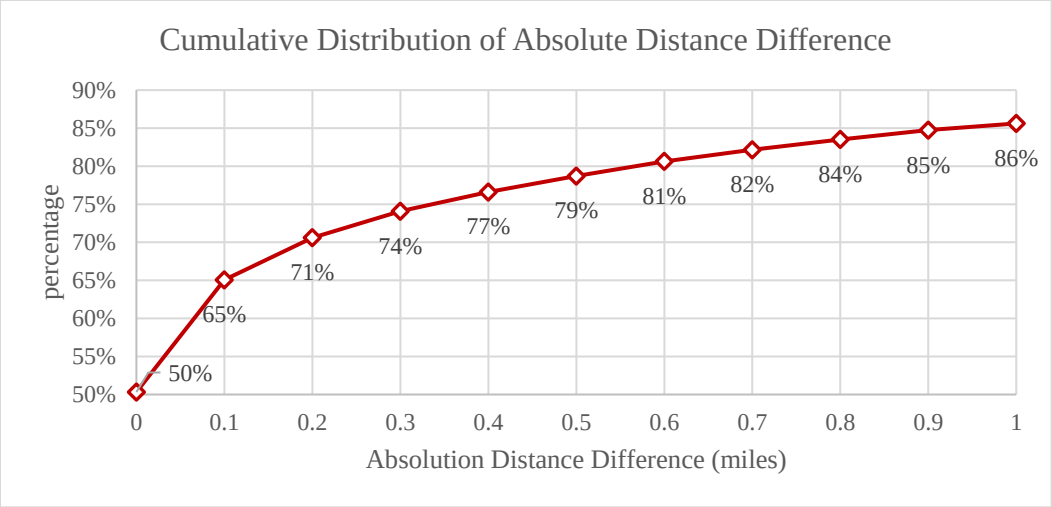


Figure 15. **Cumulative distribution of absolute distance differences.**

Real-Time Implementation

This section discusses some applications and challenges of real-time implementation of Waze event reports. Previous sections thoroughly analyze the timeliness of the Waze reports. The analysis shows that incorporating Waze into real-time operation and management has the potential to shorten the event detection time and response time. Two possible applications of Waze data are:

- **Early Indicator.** Traffic agencies take the Waze event report as an early indicator of a possible event.
- **Feeds for AID (automatic incident detection).** Many automatic incident detection algorithms have been proposed throughout the years, and most of them require off-line training. If the training is incorrect or not thorough, the performance of the algorithm might be harmed. With the availability of Waze incident reports, on-line training algorithms can be proposed to take advantage of the real-time information.

Also, there exist many same events that being reported at different times and locations, which brings in redundant information and causes extra work for practitioners to differentiate those reports from an actual event in real-time incident detection. **Table 9** shows the number of matched reports and the percentage of the matched reports among total reports. The matched reports account for only 26% of the total crash reports and 14% of the total stopped vehicle reports. **Figure 16** further shows a Venn diagram of all accidents and accidents covered by WAZE and LocateIM. This study focused on identifying the intersection of Waze reports and LocateIM records. For those reports reported by WAZE that are not covered by LocateIM, a high percentage of which could

Table 9. Total records vs. matched records.

	WAZE		*LocateIM*	
	Crash	Stopped vehicle	Crash	Stopped vehicle
Total Records	8068	93707	2052	5459
Matched	2066	13203	1374	4674
Percentage	26%	14%	67%	85%

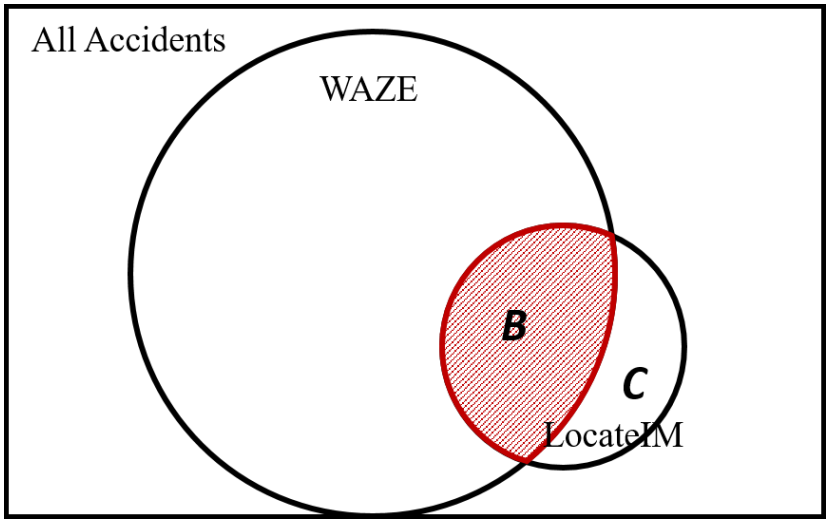


Figure 16. Venn diagram.

be duplicate reports. Once the duplicate reports are removed, the events that not captured by LocateIM can be estimated.
Suggestions are given to handle those two problems.

- **False alarm.** For operation agencies, once an event report is received, according to its location and time, the incident can be verified or confirmed using additional data sources such as traffic speed, traffic monitoring cameras, etc.
- **Duplicate report.** For the same event, the reports related to it should be within some time and distance range. It will be interesting to develop a clustering algorithm that can cluster the event reports related to the same event and the same group.

To address these two challenges, Waze may adopt some strategies to preprocess the data and make it easier for transportation agencies to use. Waze can offer the reports along with the vehicle speed as an indicator of the vehicle status when it makes the report.

Conclusion

Crowdsourced data are typically provided at low cost to traffic management administrators. While other data sources (sensors, third-party probe data, or even law enforcement reports) require high installation and maintenance costs, Waze data are available for free. Incorporating Waze data into real-time implementation has been a hot topic in recent years.

However, there are challenges in working with Waze data. One of the biggest concerns is its reliability. The reliability contains two aspects, the existence of the event and the accuracy of reports. This research evaluated the temporal and spatial accuracy of crowdsourced reports on interstate highways. Results suggest that 67% of the official crash records were reported by Waze, as were 85% of the stopped vehicle records. On average, Waze reports are made 2.2 minutes sooner than LocateIM reports (7.8 minutes for stopped vehicle). Forty percent of the crash reports (57% of stopped vehicle reports) in LocateIM are reported earlier by Waze than LocateIM, although different states may have different incident detection systems and the exact percentage may vary.
In addition, the analysis demonstrates the high location accuracy of the crash reports. On average, the distance between LocateIM reports and Waze reports is -0.001 miles for crashes and -0.025 miles for stopped vehicles. For many report types, such as police reports and traffic jam reports, the location accuracy is hard to validate, the analysis of crash reports and stopped vehicle reports reveals people's behavior and implies the reports for those events shall be pretty accurate as well. The high location accuracy and its timeliness suggest Waze reports can be a supplemental resource for real-time highway operation and management, especially incident detection.

Waze users can generate multiple reports for the same road hazards at different locations and times, especially for road hazards that exist for a long time. For example, a crash that is not cleared for a while and holds up traffic causes people in the traffic queue to report the crash though they are still some distances from the crash location. Result suggests that a high percentage of Waze reports cannot be matched to official incident

records. Among those reports, the percentage of redundancies and false alarms remains unclear. The future direction of this study is to propose a mechanism to atomically remove duplicate reports and false alarms. If the duplicate reports and false reports can be identified and removed, the remaining reports shall be incidents covered by WAZE that not reported officially. This helps us to investigate the contribution of Waze data and build a more complete incident dataset.

CHAPTER III. A CALIBRATION-FREE FREEWAY INCIDENT DETECTION ALGORITHM BASED ON REAL-TIME TRAFFIC STATUS SELF-ASSESSMENT

The chapter presents a modified version of a research paper by Yuandong Liu, Bumjoon Bae, and Lee D. Han.

Abstract

Existing automatic incident detection algorithms achieve good performance if calibrated properly. However, those algorithms may yield unacceptable false alarm rates without proper calibration, which restricts their implementations in practice. This study proposed a self-learning incident detection algorithm that is free of calibration. The proposed algorithm can train a detection model dynamically by evaluating the detection results on a regular basis without labeled incident data. The training is performed by estimating typical traffic conditions using both historical data and real-time data. If the real-time traffic pattern deviates from the estimated normal traffic pattern, an incident alarm is raised. Multilevel of deviations are maintained to detect the abnormal traffic at different levels. A self-evaluation module is then proposed to assess the traffic states over a period and determines the occurrence of an abnormal event. The self-evaluation module serves as a calibration procedure to support the selection of an appropriate deviation level at different times of the day and different locations. The performance of the proposed algorithm is compared to the performance of a benchmark model with 31 incident cases from Knoxville, Tennessee. Results suggest that the proposed algorithm outperforms the benchmark model, which has been proved to perform better than existing models in the literature.

Introduction

Incident detection is an essential component of traffic operation and management systems. Quick incident detection reduces the response time and thus reduces the impacts of an accident. In general, Traffic Management Centers (TMCs) rely on the following methods to detect an incident on highway: 1) CCTV monitoring; 2) highway emergency local patrol (HELP); 3) witness report, and 4) automatic incident detection algorithm (AID). The first three methods depend on visual identification and human report which suffers from unreliable detection time and detection rate, while the fourth one, AID, ideally only depends on field data and has the potential to detect incidents in real-time with high reliability. Much effort has been made in the past decades to develop automatic incident detection algorithms, many of those are reported to have good performance.

Despite substantial research, the implementation of incident detection algorithms has been limited due to restricted performance reliability, considerable calibration needs, and strong data requirements(*33*). A survey conducted by interviewing 32 Traffic Management Center throughout the US in 2007 shows that most TMC remains hesitant to rely on AID algorithms as key components of TMC operations(*34*), 87.5% of the centers claimed to have not been using a fully functional AID algorithm, among the TMCs that equipped with AID unit, More than 60% of the centers disabled their AID algorithm. The

primary reason for disabling the AID algorithms is that they yield unacceptable false alarm rates. Besides, the initial and ongoing calibrations for those methods are usually complicated and time-consuming. A thorough investigation into existing AID algorithms and how they are implemented reveals that most of the algorithms. while perform adequately theoretically, need to be properly calibrated site by site. This requires a level of understanding of the algorithm details, which is not attainable for local TMC personnel.

Some TMCs develop their algorithms for local traffic conditions, which has little connection with the algorithms described in the literature. The developed incident detection systems are effective principally for major incidents or for incidents that occur near a sensor. Generally, the incident is reported from other (non-automatic) sources first(*35*). Therefore, CCTV and witness reports remain to be the primary methods of incident detection and verification.

Literature Review

Incident detection is a classification problem with two classes: normal traffic conditions and abnormal traffic conditions. Most incident detection or abnormal traffic event detection algorithms can be categorized into two groups, the first group defines the normal traffic conditions and set up thresholds, any pattern that exceeds the threshold are labeled as incidents. The second group adopts advanced machine learning techniques and categorizes traffic conditions based on fully trained models.

The first type of traffic incident detection algorithm attempts to estimate the behavior of normal traffic and classify the current traffic status as normal or abnormal based on preset thresholds. The developed algorithms fall into three categories: comparative algorithm, statistical algorithm, and macroscopic algorithm:

Comparative algorithm

The comparative algorithm is one of the earliest developed algorithms. Most of the algorithms are variations of the California algorithm(*36*). The algorithm uses the lane occupancy values (either raw or smoothed) at a single station or between two adjacent stations as input, a set of thresholds values are set to characterize the state of the traffic flow(*36; 37*). The comparative algorithm has the advantage that it is straightforward but its use is limited since it generally requires laborious calculations of a threshold for each location where it is installed.

Statistical algorithms

Filtering algorithm

The representative algorithms in this category include a double exponential smoothing algorithm, a low-pass filter algorithm, etc. The Minnesota algorithm uses a low-pass filter to smooth data series, the 5-min occupancies are stored, large differences between the consecutive occupancy values indicate an incident(*38; 39*). This logic gets rid of the laborious calibration process from site to site. The drawback, however, is the considerable amount of time it takes to detect incidents due to the time needed to smooth the data(*40*).

Time series analysis

Time-series techniques were adopted to predict short term traffic patterns if the present traffic values deviate significantly from the prediction results, an alarm is triggered. The simplest method in this category is the standard normal deviate algorithm. The algorithm computes the mean u and standard deviations σ of some measurements such as occupancy based on historical data and predicts the normal traffic pattern to be $u + k * \sigma$, k is a parameter that needs to be set in advance. The algorithm is simple and produces acceptable results and is widely adopted for comparison in other studies. Several variations for this algorithm exists(*41*), Ying(*41*) proposed a spatial-temporal mining algorithm based on SND method, it updates the threshold with incoming traffic data and is adaptable to environmental changes. However, the detection rate, false alarm rates and mean time to detection of the algorithm remains to be further analyzed. Other models include autoregressive integrated moving average (ARIMA) model(*42*), double exponential smoothing(DES) algorithm(*43*), those algorithms are similar to SND but use more complicated forecasting methods. The absolute error between the predicted and observed value is used as an incident indicator.

Macroscopic algorithm (theoretical algorithm)

This type of approach uses macroscopic traffic-flow modeling to describe the evolution of traffic variables and identify incidents. McMaster is an example (*44; 45*). It is composed of two stages; the first stage uses the fundamental diagram to detect congestion and the second stage distinguishes incident-related congestion from recurrent congestion based on upstream sensors. It defines congestion to be caused by incident if the downstream sensor is in uncongested status while the upstream detector is in congested status. The thresholds in both stages need to be calibrated for specific stations.

Comparative algorithms, statistical algorithms, and macroscopic algorithms are classified into the same group because all of them contain a critical part: establishing a threshold. Because the incident conditions vary with road geometry, weather condition, and traffic conditions, the thresholds fluctuate at different locations and different times of the day. Setting the thresholds is usually time-consuming. The complexity of setting the threshold varies with the algorithm, but in general, it demands a good understanding of the algorithm as well as local traffic patterns.

The second group of algorithms utilize advanced machine learning techniques and categorizes traffic conditions based on fully trained models.

Machine learning algorithms

Machine learning techniques are suitable for classifying data points into different categories and are widely adopted in incident detection algorithms. Many advanced machine learning techniques have been adopted and proved to have good performance, such as Artificial Neural network (ANN), Support Vector Machine (SVM), decision trees and naïve Bayes.

Most the proposed machine learning models are supervised learning algorithms and require correct labeled training dataset to properly train the model, which involves a substantial amount of manual work to provide complete and accurate information for the incidents.

Balke(*46*) reviewed some of the algorithms and noticed that improper calibration appears to be the most prevalent reason for high false alarm rates. The performance of the same algorithm can differ considerably in different environments and that the algorithm could not be properly calibrated unless an incident affects every detection zone. Therefore, although many of the algorithms demonstrate high performance with well-performed calibrating procedures and comprehensive dataset, the transferability of those algorithms is greatly limited. In this study, we aim to propose a calibration-free algorithm that can adaptively adjust the threshold from site to site and from time to time. Thus, an additional step is proposed to assist the algorithm to evaluate the detection results and select a proper threshold at different locations and different time.

Methodology

To detect an incident, proper traffic features shall be selected. A previous study(*47*) shows that the occupancy difference between upstream and downstream detectors is one of the best traffic features to differentiate abnormal traffic from normal traffic. In this study, the occupancy difference Δo_t between upstream and downstream detectors is adopted as the input parameter: $\Delta o_t = \Delta o_{up} - \Delta o_{down}$. If an incident occurred between two detectors, the traffic flow is disturbed. The downstream density(occupancy) decreases because of less flow while the upstream density increases because of the queue building up, thus Δo_t will decrease. We assume during any 5 minutes from time t to $t + 5min$, The 30-s occupancy difference Δo_t follows normal distribution $\Delta o_t \sim N(\mu_t, \sigma_t^2)$, this assumption is tested in an earlier study(*48*). The basic idea behind the proposed algorithm is to estimate normal traffic condition parameters μ_t, σ_t^2 use historical data for each 5-min period. Then if the current traffic state deviates from the normal condition significantly, an incident alarm will be triggered. Multi-thresholds are kept to represent the 'significant level' of the deviation. The adaptive selection of threshold is based on an evaluation module. The proposed model is composed of three modules: Incident Detection, Self-evaluation and Training, and Threshold Selection. The three modules are explained in detail below.

Incident Detection

The goal of this step is to detect incident based on the estimated parameters μ_t, σ_t^2.

- *Step1: initialize/update parameters*
  This step initializes the μ_t^0, σ_t^0. If historical data is available, then

$$\mu_t^0 = median(\Delta v_t)$$
$$\sigma_t^0 = absolute\ deviation(\Delta v_t)$$

  Instead of using the mean value to estimate μ_t^0 and standard deviation to estimate σ_t^0. We use median and absolute deviation to estimate the two parameters since the mean and standard deviation values are sensitivity to outliers as suggested by Leys(*49*).

- *Step2: detection*

With the estimated μ_t and σ_t^2, a threshold h_t from t to $t + 5min$ can be established: $th_t = \mu_t + k\sigma_t^2$(the selection of k will be discussed in *Threshold Selection Module*), If the incoming data point Δo_t exceeds the threshold $th_t(\Delta o_t < th_t)$, an incident alarm is triggered.

Self-evaluation and Training

This step independently evaluates the traffic status every 15-min after detection is performed. We assume the 15-min average speed $\overline{\Delta o_t}$ also follows a normal distribution $\overline{\Delta o_t} \sim N(\overline{\mu_t}, \overline{\sigma_t^2})$. $\overline{\Delta o_t} = {}^1/_{15} \sum_{i=1}^{15} \Delta o_{t+i}$.

- *Step1: Evaluate detection results*

The occurrence of an incident is confirmed if the average occupancy difference $\overline{\Delta o_t}$ during a 15-min period differs from normal traffic condition: $\overline{\Delta o_t} > \overline{\mu_t} + 2 * \overline{\sigma_t^2}$. Then, the detection results from the incident detection step can be classified into two categories by comparing it with the evaluation results: true detection and false detection. The definition of the category is shown in **Table 10**.

True detection means the self-evaluation module confirms an abnormal event while the event is detected by the incident detection module. False detection means the self-evaluation step cannot confirm the detection results. For a threshold th_t, the number of true detections and false detections can be accumulated over time.

- *Step2: Update parameters*

Based on the classification results in step 1, μ_t and σ_t^2 are updated with the following equations. for all data points during the t to t + 5min period, if the data point does not exceed the threshold th_t, it is classified into the normal traffic condition group. Then the μ_t and σ_t^2 is updated with the normal data set.

$$u_t^{new} = \frac{u_t^{old} * n + \sum_{i=1}^{m} \Delta o_{t+i}}{n + m} \tag{15}$$

$$(\sigma_t^{new})^2 = \frac{(\sigma_t^{old})^2 + \sum_{i=n}^{n+m} \left(\Delta o_t^i\right)^2 - \frac{1}{n+m}\left((n+m)u_t^{old} + \sum_{i=n}^{n+m} \Delta o_t^i\right)^2}{n + m} \tag{16}$$

Table 10. Correctness label.

		Self-evaluation and Training	
		Positive	**Negative**
Incident Detection	**Positive**	True Detection	False Detection
	Negative	False Detection	/

n denotes the number of historical data points during the corresponding 5min period that is classified as normal; m demotes the number of data points during the 5-min period that is classified as normal. The sum of square values $(\sigma_t^{old})^2$ and the u_t^{old} shall be stored to update the two parameters. Equation (15) updates the mean value. It allows us to update the parameter by storing the counts and the historical mean value instead of storing all the normal traffic data, which saves memory space. Equation (16) updates the standard deviation. To update the standard deviation, only one extra measurement needs to be stored: $(\sigma_t^{old})^2$, which is the variance of historical normal data. The derivation of equation (15) and equation (16) can be found in the Appendix.

With the self-evaluate and training step, the proposed algorithm not only detects crashes on road but also detects abnormal events that have obvious impacts on traffic which might not be recorded because of no property damage or injuries.

Adaptive Threshold Selection

Choosing a proper threshold is a crucial step in the automatic incident algorithm. The setup of thresholds generally involves burdensome calibration processes and is performed manually. This study proposes an automatic threshold selection process. For each 5-min period, a group of thresholds is maintained: $th_t^i = \mu_t + k^i\sigma_t^2, i = \{1,2,3 \dots n\}$. At the self-evaluation and training step, the number of true detections and false detections is calculated. With the true detection counts and false detection counts, the detection rate, false-alarm rates for each threshold th_t^i can be calculated. The one produces the highest true alarm rate or lowest false alarm rate is selected as the threshold to detect incident in the next step. The threshold selection criteria vary with the needs for the detection. For example, if high detection rates are desired, then, the threshold produces a higher detection rate shall be selected.

A flow chart of the proposed model is presented in **Figure 17** to demonstrate how the three modules are connected. A detailed explanation of each step of the model is provided below.

Case Study

The proposed algorithm was tested with accident cases from Oct. 1st, 2017 to December 31st, 2017 on westbound I-40 from milepost 374(west end) to milepost 394(east end) in the Knoxville area, Tennessee. This site was selected as a testbed because it has high traffic volumes and encounters traffic accidents frequently. Totally 115 accidents happened on Tuesday, Wednesday and Thursday were extracted in this area. Among the 115 accidents, 84 were removed either because of detector failure or cannot be visually identified. The algorithm was evaluated on the remaining 31 accident cases. The performance of the algorithm is compared with a benchmark model (*48*). **Figure 18** shows a map that demonstrates the study area and functional detector locations. Totally 44 detectors are installed on this stretch of highway with an average spacing of 0.5miles.

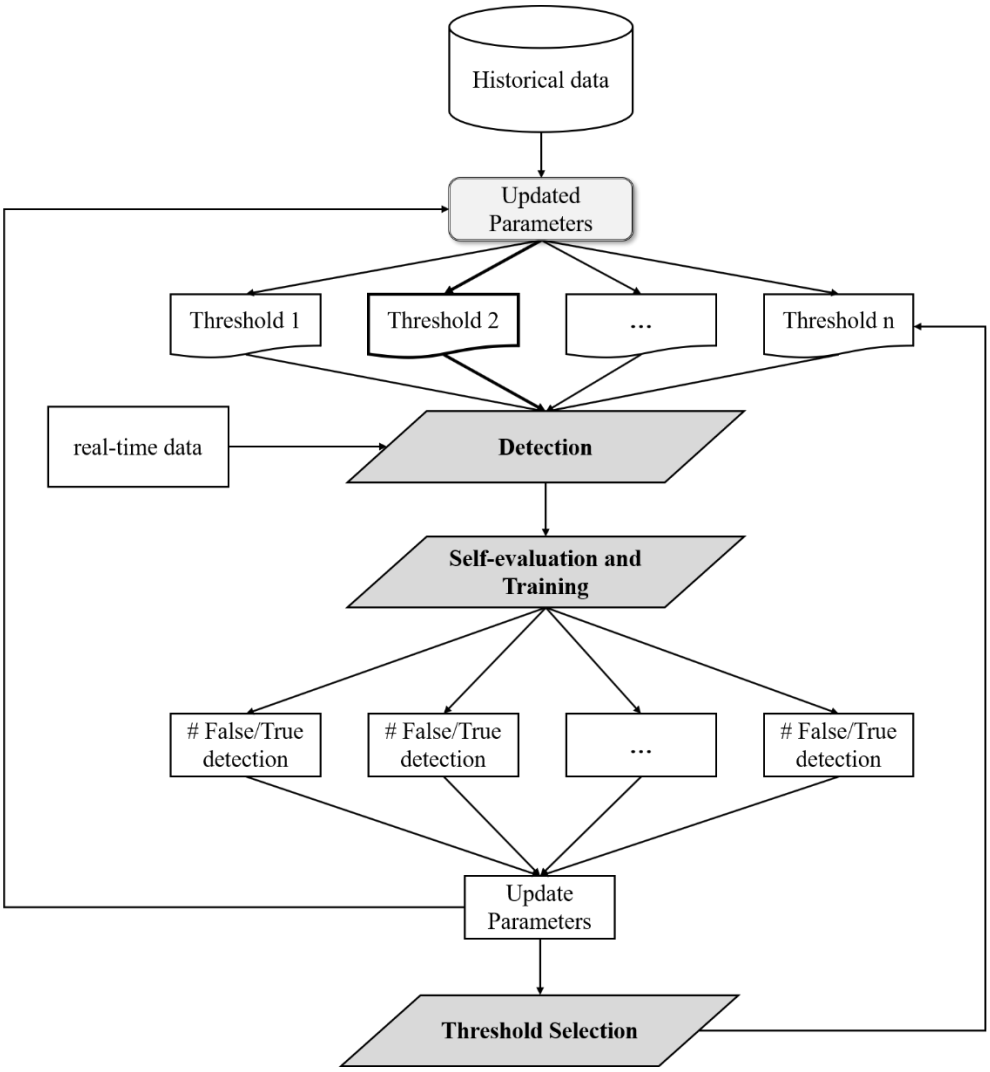


Figure 17. Flow chart of the proposed model.

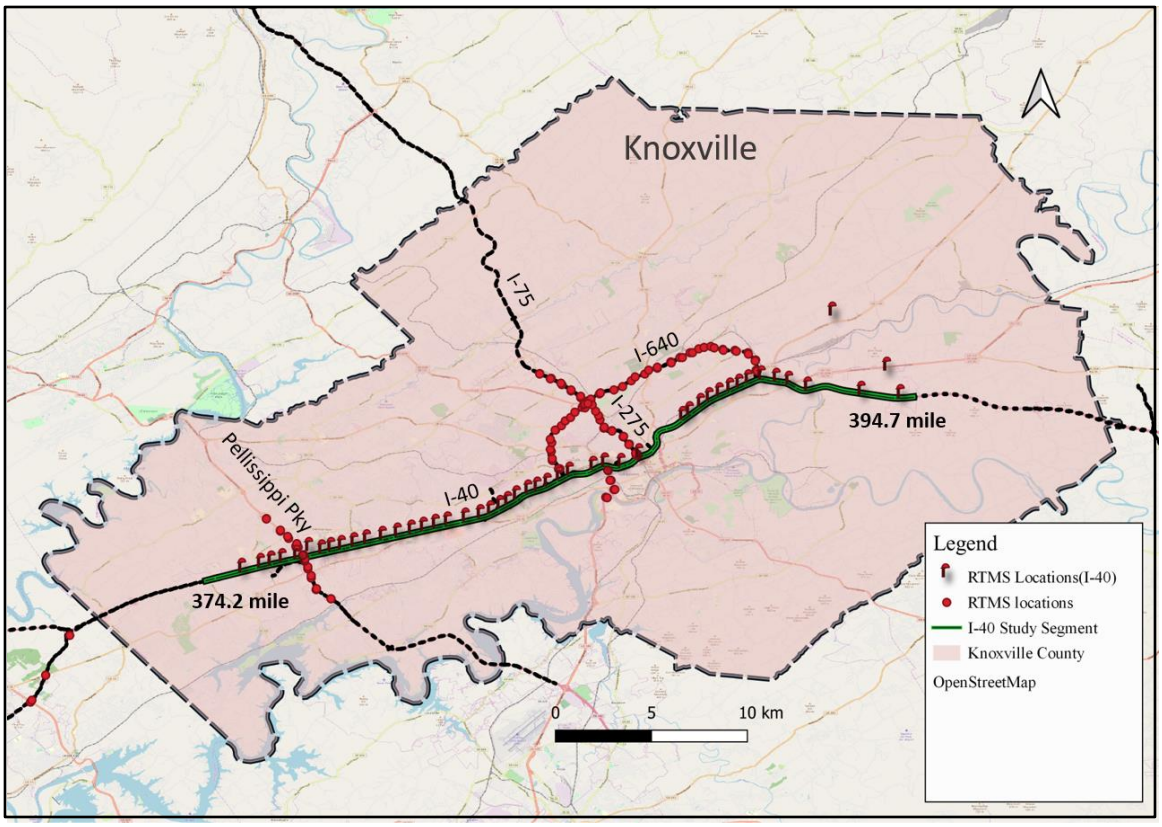


Figure 18. Case study location in Knoxville, Tennessee.

Data Preparation

Sensor Data: The data used in this study is Remote Traffic Microwave Sensors (RTMS) data collected by the Tennessee Department of Transportation (TDOT). RTMS reports average speed, flow, and occupancy every 30-second. Speed relative differences between upstream and downstream detectors were adopted to perform the detection. One year of RTMS data was used in this study for training purposes.

Incident Data: TDOT maintains official incident records in the LocateIM system, which details the location and the report time of each incident. The logs are used to validate the detection results. Because the start time recorded in the LocateIM system is the time that the incident is reported instead of the true occurrence time, we visually inspect the 30-s roadside sensor data and choose the start time as one interval(30-seconds) before the traffic disturbance started.

Evaluation Metrics

Generally, three measurements are used to evaluate the performance of automatic incident detection algorithm:

- Detection Rate (DR)

$$DR = \frac{Number\ of\ true\ detections}{Total\ number\ of\ incidents} * 100\% \tag{17}$$

- False Alarm Rate (FAR)

$$FAR = \frac{Number\ of\ false\ detections}{Total\ \#\ of\ detections} * 100\% \tag{18}$$

- Mean Time to Detection (MTTD)

$$MTTD = \frac{1}{n}\sum_{i=1}^{n} T_{detection} - T_{incident} \tag{19}$$

$T_{detection}$ is the time the incident is detected. $T_{incident}$ is the time when the incident happens. n denotes the total number of true detections. In general, high detection rate, low false alarm rate and short mean time to detection are desired. However, the three measurements cannot achieve the best performance at the same time. The final criterion is usually a trade-off among the three measurements.

Model Evaluation

In this section, the detection algorithm is evaluated with the four criteria. A self-evaluation module and threshold selection module were proposed in the methodology section to support the adaptive selection of the thresholds. To demonstrate the advantage of the inclusion of the self-evaluation and threshold selection modules, the detection results of the proposed model are comparing to the detection results of the model without

the adaptive threshold selection process. The model to be compared to is described in detail in this paper(*48*). A brief introduction of the algorithm is presented below.

Training-free-algorithm

Similar to the proposed algorithm, the Training-free-algorithm assumes the 5-min occupancy difference between upstream and downstream detectors follows the normal distribution. Based on the assumption, the normal traffic pattern every five minutes is estimated with historical data. Then, a threshold can be established every 5-min to detect the incident. For comparison purposes, different threshold levels for the Training-free-algorithm were used. The detection results from multiple thresholds are compared to the detection results of the proposed calibration-free algorithm.

Threshold Selection

Figure 19 demonstrates how the threshold is selected to avoid false alarms. The figure shows the occupancy difference for two detectors separately locate at I-40 milepost 384.1 and I-40 milepost 384.5. Six threshold levels are maintained for demonstration purposes, separately level 1(T1, k = 2), level 2(T2, k = 2.5), level 3(T3, k = 3), level 4(T4, k = 3.5), level 5(T5, k = 4), and level 6(T6, k = 4.5), represented by six parallel lines in the figure. The blue line that fluctuates represents the occupancy difference between two neighboring detectors.

As can be seen in the figure, the threshold changes every five minutes to adapt to the traffic conditions at different times of the day. The occupancy difference between the two dotted lines exceeds threshold level 1 to level 3 and an incident alarm is raised. In the meanwhile, the thresholds above level 3 do not detect any abnormal patterns. The evaluation results with 15-min average speed data show that the 'detected incident' is just a disturbance instead of an actual accident. Thus, for level 1, level 2 and level 3, the false detection counts are increased by one, while for level 4, level 5 and level 6, the false detection counts remain the same. On the contrary, if the detection is proved to be an accident based on the evaluation process, the true detection counts for level 1 to level 3 will increase by 1while remain the same for level 4 to level 6. Over time, the true detection counts and false detection counts of multiple threshold levels are accumulated to determine the threshold with the best performance. The best performance can either be a high detection rate, or a low false alarm rate or a combination of both.

Model Comparison

The previous section demonstrates how the evaluation module and threshold selection module work. In this section, the comparison results between the proposed algorithm and Training-free-algorithm are demonstrated. For the Training-free-algorithm, six levels of threshold were used, separately k = 2, 2.5, 3, 3.5, 4 and 4.5(recalling that k is the value multiplies standard deviation). The proposed algorithm selects an appropriate threshold after a detection among multiple thresholds, while the Training-free-algorithm does not support

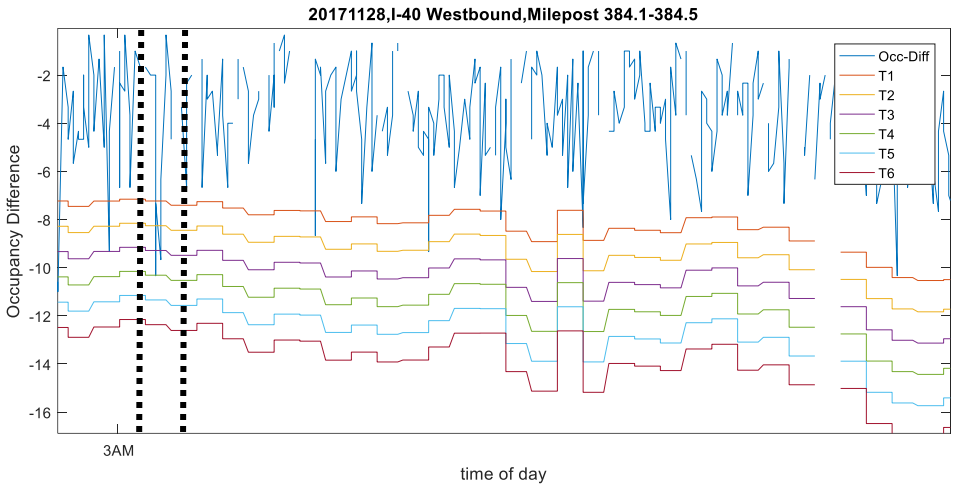


Figure 19. Demonstration of the threshold selection process.

threshold selection. Therefore, the performance of the Training-free-algorithm with six different threshold levels is compared to the performance of the proposed algorithm.

In **Table 11 & Figure 20,** T1 to T6 represents different threshold levels for the Training-free-algorithm, k values are demonstrated in the parentheses. The theoretical false alarm rate is determined based on the confidence interval. For example, if k = 2, then based on the normal distribution table, the chances that the normal traffic conditions exceed the threshold is 5%. Therefore, the theoretical false alarm rate is 5%. The true-false alarm rate is demonstrated as well, which is calculated based on equation (18). As illustrated in the table, the detection rate of the Training-free-algorithm decreases with the decrease of the false alarm rate. The mean time to detection increases at the same time. When k = 2.5 and 3, while the detection rate is high, the Training-free-algorithm produces high false alarm rates (notice that while 1.1% false alarm rate seems not to be high, but it denotes 30 false alarms every day for each site). When k = 3, the threshold achieves an acceptable false alarm rate as well as a comparatively high detection rate. The proposed model, on the other hand, produces a higher detection rate (84% as compared to 71%) with a lower mean time to detection and similar false alarm rate thus outperforms the Training-free-algorithm.

In addition to DR, FAR and MTTD, the transferability of the algorithm shall be considered as a measurement of performance.

- Ease of implementation/transferability (EI)

  The ease of implementation evaluates the effort needed to generalize or transfer the incident detection algorithm to other contexts of settings. *EI* is evaluated in terms of both data requirements and training complexity.

The proposed model is transferable to any traffic situation because it only requires traffic data and adaptively learns the typical traffic pattern over time at different locations. The basic logic behind the proposed algorithm is a two-level comparison: First,

Table 11. Comparison of proposed algorithm and Training-free algorithm.

Algorithm	Threshold level	FAR	False Detection/day
Proposed Algorithm	Adaptive Selection	0.4%	12/day
Training-free Algorithm with Multiple Threshold	T1(k = 2)	4.3%	124/day
	T2(k = 2.5)	1.1%	32/day
	T3(k=3)	0.4%	11/day
	T4(k=3.5)	0.2%	6/day
	T5(k = 4)	0%	0/day
	T6(k = 4.5)	0%	0/day

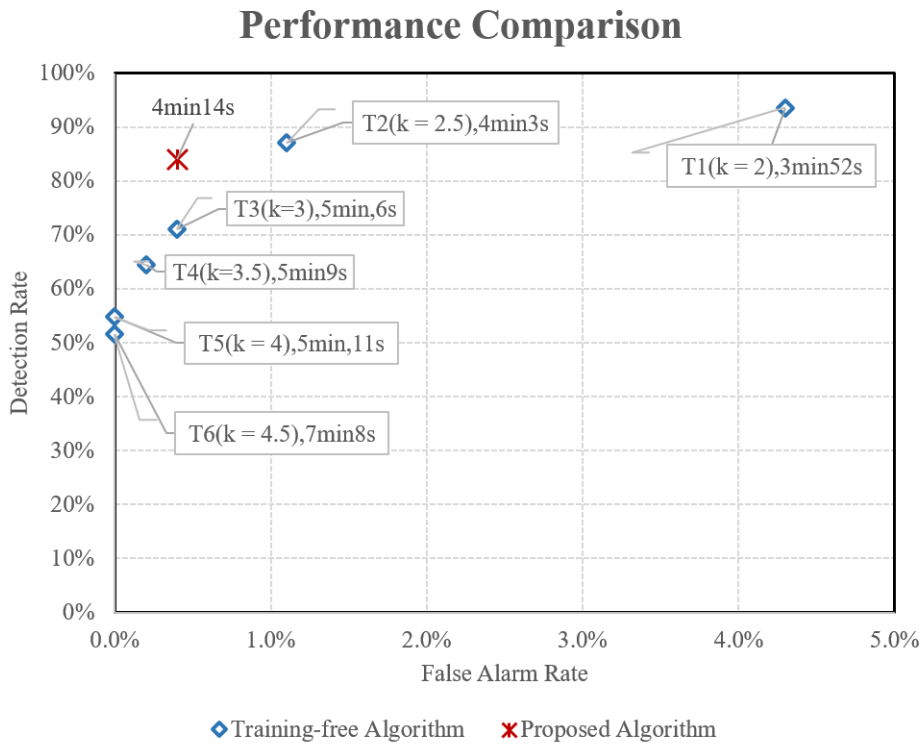


Figure 20. Comparison of proposed algorithm and Training-free algorithm.

the algorithm compares the 30-s data with the 30-s threshold to detect an incident. Then 15-min average data is compared to the 15-min threshold to validate the detection results. The model is not as complex as machine learning models such as ANN (artificial neural network) or SVM (support vector machine) model which is hard to understand by personnel at traffic operation agencies and limits the application of those algorithms.

Incident not Detected

Several incident cases were not detected by the proposed algorithm. By looking into the incident data and the traffic sensor data, it can be observed that the incidents that cannot be detected are generally incidents hidden in peak hours. The traffic conditions demonstrate high variances during the peak period and the threshold level is usually high. A lower threshold will generate high false alarm rates and is not selected by the algorithm. **Figure 21** demonstrates a typical incident case that cannot be detected by the algorithm. The traffic features between the two dotted black lines dropped because of an accident. However, the traffic typically demonstrates high variance during peak hours at this location and result in a high threshold level. Solely by comparing the traffic features with historical patterns is not enough to detect the accident. In future research, the occupancy difference between nearby period will be taken into consideration to detect the abnormal traffic changes over time at the same location.

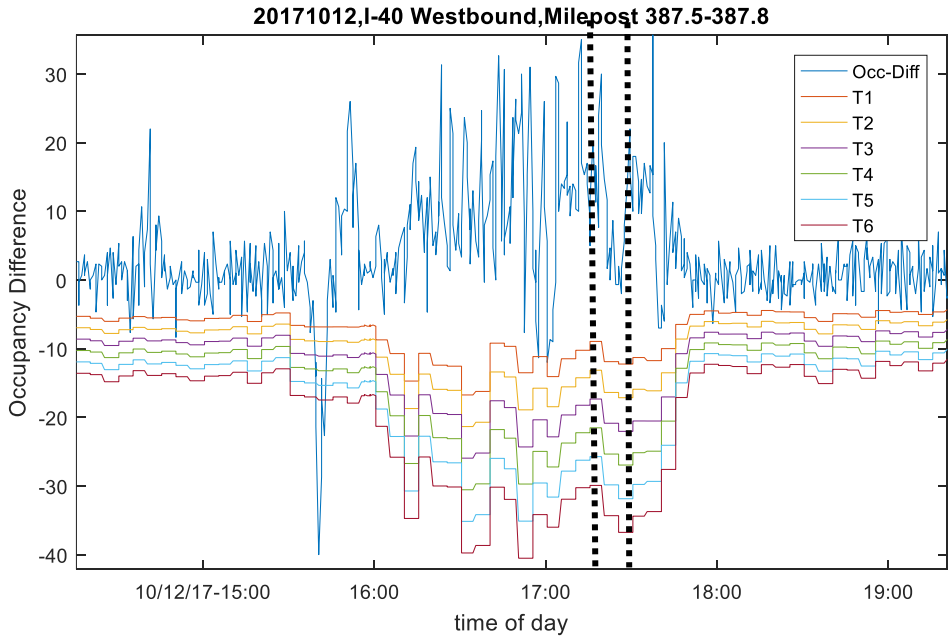


Figure 21. An example of an incident that cannot be detected by the proposed algorithm.

Conclusion and Future Work

Incident detection is an essential component of traffic operation and management systems. Quick incident detection reduces the response time and also the impacts of an accident. Numerous automatic incident detection algorithms have been established with excellent performance. But the implementations of those algorithm have been limited because of considerable calibration requirements. In this study, a self-evaluation module is proposed which compares the average speed data with historical data to justify the occurrence of an abnormal event thus eliminate the needs of calibration. While this method does not guarantee to find all the abnormal events-some minor events that do not have an obvious impact on the traffic thus cannot be detected by the evaluation process, results suggest on average it has good performance and improves the detection rate.

The proposed algorithm is composed of three main modules: incident detection, self-evaluation and training, and threshold selection. The incident detection module performs incident detection based on an established method. Then, the detection results are reevaluated under the self-evaluation and training module. Finally, based on the accumulated evaluation results over time, the threshold selection module picks a threshold that produces the best result. The best result can be either a high detection rate or a low false alarm rate or a combination of both.

The case study tested the proposed algorithm with 31 incident cases in the Knoxville area. The algorithm was compared with a benchmark model that performs better than many existing algorithms. The proposed algorithm outperforms the benchmark model in terms of detection rate, and mean time to detection. The false alarm rate is at the same level. Compared to other existing algorithms, the proposed self-learning algorithm requires no additional dataset and can be easily transferred to different sites.

In future research, other traffic parameters such as speed, flow, and traffic flow fundamentals will be tested and incorporated to enhance the performance of the algorithm during peak hours.

CHAPTER IV. DYNAMIC TRAFFIC QUEUE-END DETECTION USING WAZE JAM REPORTS

The chapter presents a modified version of a research paper by Yuandong Liu, Zhihua Zhang, Lee D. Han and Candace Brakewood. The paper has been submitted to *International Journal of Geographical Information Science.*

Abstract

Traffic queues, especially queues caused by non-recurrent events such as incidents, are unexpected to high-speed drivers approaching the end of queue (EOQ) and become safety concerns. Though the topic has been extensively studied, the identification of EOQ has been limited by the spatial-temporal resolution of traditional data sources. This study explores the potential of location-based crowdsourced data, specifically Waze user reports. It presents a dynamic clustering algorithm that can group the location-based reports in real-time and identify the spatial-temporal extent of congestion as well as the end of queue. The algorithm is a spatial-temporal extension of DBSCAN (density-based spatial clustering of applications with noise) algorithm for real-time streaming data. A dynamic spatial-temporal threshold selection approach is proposed to automatically determine the threshold for the algorithm. The proposed algorithm was tested with 34 traffic congestion cases in the Knoxville, Tennessee area. It is demonstrated that the algorithm can effectively group Waze reports and identify traffic congestion. The EOQ identification results are compared to the detection results from roadside sensor data. The results are promising. The EOQ identification time of Waze is similar to the EOQ detection time of traffic sensor data, with only 1.1 minutes difference on average. In addition, Waze generates 1.9 EOQ detection points every mile, compared to 1.8 detection points generated by traffic sensor data, suggesting the two data sources are comparable in terms of reporting frequency. The results demonstrate that Waze is a valuable complementary source for end of queue detection where no traffic sensors are installed.

Introduction and Literature Review

Introduction

Traffic congestion, especially nonrecurring congestion such as congestion that caused incidents, is a major source of uncertainty in freeway operations. It leads to much longer delays as well as increases the likelihood of rear-end crashes. To protect the drivers approach the congestion, an end of queue warning system is generally deployed. The system alerts drivers of slowdown traffic ahead using devices such as variable message signs. Proper deployment of an end of queue (EOQ) warning system, can reduce crashes by up to 45%(*50*). The warning system requires accurate information about the spatial-temporal movement of the slowdown traffic on a real-time basis.

Prior research has been conducted to identify or predict queue movement in real-time (*51-55*) with a focus on signalized intersections and freeway work zones. Limited studies have been dedicated to the more general analysis of freeway traffic queues(*56; 57*). Most of the developed methodologies rely on fixed traffic sensors data to estimate the temporal and spatial extent of congestion and identify queue locations (*57-59*). The detection accuracy of those methodologies is restricted by the spatial and temporal resolutions of traffic sensors.

Recently, a number of studies have explored the potential of new location-based data sources in estimating queue movement and detecting traffic events and congestion(*29; 52; 53; 60-66*). The results have shown that incorporating traffic information from location-based data, especially probe vehicle data has great potential for improving the estimation accuracy of traffic situations, especially where no traffic detectors are installed. However, many probe data are averaged every 5-min, 10-min, which is insufficient for real-time traffic queue detection. Higher time resolution probe data are still not available to most traffic agencies.

Waze, a GPS-based application, is one of the most popular navigation applications used by drivers in the United States. It collects users' speed and GPS location information to detect the current traffic status and provide route guidance to users. A distinctive feature of Waze is it allows users to report traffic information by adding geometry points on a Waze map to indicate hazards, accidents, traffic jams, or police appearances. Each individual report provides the exact location and time, and thus can be viewed as a piece of location-based probe vehicle information. Waze has established partnerships with local government agencies through the Connected Citizen Program and provides real-time feeds for those reports to government agencies.

Compared to the traditional data sources, an advantage of Waze reports is that they provide real-time data and have high road network coverage. Waze reports provide data that covers interstate and urban roads where no traffic surveillance system is installed and thus can be a valuable data source for traffic operation. However, before incorporating the Waze reports into any real-time application system, it is important to evaluate the data quality. Previous research has been conducted to evaluate the quality and accuracy of Waze reports(*5; 6*). It was found that 0.3% of the reports were false

reports(*6*). The low false alarm rates, as well as high spatial-temporal accuracy(*6*), proves that Waze reports are reliable sources for real-time operation systems.

This study proposes a clustering algorithm to identify congestion and end of queue using Waze reports. Space-time interaction arises when nearby reports occur at about the same time. Two reports that are spatially and temporally close to each other may indicate that two drivers are within the same queue caused by a particular incident and thus can be associated with one another. By recursively associating and clustering the Waze reports in both spatial and temporal dimensions, we are able to identify the spatial-temporal extent of congestion and track the movement of traffic queues. It is expected that the ability to accurately identify shock wave locations and speeds can serve as the foundation of queue movement prediction.

The rest of the paper is organized as follows. The next section presents a review of relevant literature. This is followed by a comparison of Waze jam reports with traffic sensor data. Then, a real-time spatial-temporal DBSCAN algorithm is proposed to cluster geometry point data with its timestamp. The next section presents a case study that utilizes the proposed methodology to cluster Waze jam reports and identify the end of queue in real-time. And then the detection results are compared with detections from traditional traffic sensor data. Finally, the last section concludes this article and provides directions for future work.

Literature Review

Spatial-temporal (ST) clustering has been a major research field of spatial-temporal data mining and knowledge discovery. Spatial-temporal studies aim to find spatial-temporal patterns and identify spatial-temporal clusters. Compared to a conventional one-dimension cluster, clusters with an additional time dimension can be used to track the evolution of clusters over time and reveal both spatial and temporal trend patterns of data. Traffic on roads exhibits obvious spatial-temporal patterns; traffic queue formation and dissipation is a typical spatial-temporal motion that contains interesting patterns to be mined (*67*).

In general, spatial-temporal data are classified into five different types(*68*): spatial-temporal event, geo-referenced variables, geo-referenced time series, moving objects and trajectories. The Waze report belongs to the basic type: spatial-temporal event (ST event). Each event is static and associated with the location where it was recorded and a corresponding timestamp. Finding a cluster among ST events is to discover groups of elements that lie close both in time and in space, and possibly share other non-spatial properties (*68*).

Existing ST event spatial-temporal clustering methods can be classified into three different types: spatial scan methods(*69*), distance-based methods(*70*), and density-based methods(*71; 72*). Spatial scan statistics search spatial-temporal cylinders (radius determined by spatial distance and height determined by time interval) where the density of events of the same type is higher than the density of events outside the cylinders(*73*). The results of this type of method are highly affected by the choice of scanning windows. Distance-based methods usually define a single distance measure that combines both spatial and temporal distances between spatial-temporal objects and uses traditional clustering methods to detect spatial-temporal clusters. However, the single distance

measure on many occasions is hard to define(*74*). Another type of distance-based method defines a spatial-temporal proximity relationship from spatial and temporal aspects respectively by pre-defined parameters. It begins with assuming the times of occurrence of the ST events are distributed randomly across the case location and assumes the time-space event follows a certain distribution(*75*). This is not applicable in Waze case since Waze reports are not randomly distributed; it is highly correlated with the traffic status. The third type, density-based method, intends to find densely clustered objects. It is usually derived from classical density-based spatial clustering of applications with noise (DBSCAN) algorithm. For instance, ST-DBSCAN(spatial-temporal DBSCAN), an extension of DBSCAN to handle spatial-temporal events, was proposed in multiple different studies for analysis of spatial-temporal events (*71; 76*).
This paper presented a real-time application of ST-DBSCAN. Among different clustering algorithms, ST-DBSCAN is selected because it has the ability to discover clusters with arbitrary shapes, which is suitable for identifying congestion since the time-space region of congestion may exhibit various shapes. In this study, we improved the conventional ST-DBSCAN clustering algorithms in three important aspects. First of all, the current clustering algorithm is static and does not meet real-time clustering requirements; therefore, a real-time implementation of ST-DBSCAN algorithms is proposed to cluster streaming data. Second, the algorithm parameters are generally determined based on domain knowledge; in this study, an approach is developed to discover Waze users report spatial-temporal patterns and automatically select threshold parameters. Third, instead of using Euclidean distance that is adopted by most DBSCAN-based clustering algorithms, the road network connection is considered and the realistic road network distance is adopted to measure the distance among reports.

Methodology

Data Description

Waze establishes partnerships with government agencies and provides partners with real-time, anonymous, Waze-generated incident and slowdown traffic information feeds through the Connected Citizen Program(https://www.waze.com/ccp). The information contained in each incident or jam report includes the location (longitude and latitude coordinates), a timestamp (report time to the nearest seconds), type (incident, and jam), and multiple other variables that are not used in this study.

In this study, both Waze accident reports and jam reports are used to identify the end of queue. For simplicity, jam report will be used in the later on context to represent both accident and jam reports. The accident reports are used because they often indicate the start of a queue. Typically, Waze jam reports are composed of three levels: moderate traffic, heavy traffic and standstill traffic. All three types of jam reports are used in this study.

Algorithm

In order to support two-dimensional spatial data clustering, Derya(2010) proposed a spatial-temporal DBSCAN (ST-DBSCAN) algorithm that extends the conventional DBSCAN algorithm by adding a temporal dimension to take into account the temporal correlations among objects (*71; 76*). Detailed descriptions of the algorithm can be found in Derya(*71*).

The main difference between spatial-temporal DBSCAN and DBSCAN is that the neighborhood radius ε in DBSCAN is separated into two radii: the spatial neighborhood radius ε_s and temporal neighborhood radius ε_t in ST-DBSCAN. Therefore, three parameters will be used in ST-DBSCAN algorithms. ε_s, ε_t , and *minPts*. ε_s, ε_t specify the temporal and spatial thresholds. A point p is the eps-neighborhood of point q if and only if the point p is within the ε_s-neighborhood and ε_t-neighborhood of point q. *minPts* specifies the minimum number of eps-neighborhoods needed for a point q to be a core point. If q has more than *minPts* eps-neighborhood, q is called core point. Similarly, the other concepts in spatial-temporal DBSCAN should also be extended accordingly based on DBSCAN.

ST-DBSCAN starts with obtaining the eps (ε_t, ε_s) neighbors of each data point and identifying the core points with more than *minPts* neighbors. Then, it finds the connected components of core points on the neighbor graph, ignoring all non-core points. Last, it assigns each non-core point to a nearby cluster if the cluster is an eps (ε_t, ε_s) neighbor; otherwise assign it to noise.

Application of ST-DBSCAN to Waze Data

In this study, the proximity of two reports is defined at both the spatial and temporal levels. The construction of distance function and real-time implementation of ST-DBSCAN using Waze data is demonstrated as follows.

Temporal distances (Δt)

The temporal distance is computed as the report time differences between every two Waze reports i and j in seconds.

$$\Delta t_{i,j} = t_i - t_j \tag{20}$$

If $\Delta t_{i,j}$is greater than zero, meaning report j occurred after event i, and vice versa.

Spatial distances (Δs)

The spatial distance is mostly measured by Manhattan distance, Euclidean distance, or Minkowski distance given coordinates in spatial clustering studies(*71*). However, directly measuring the spatial distances between two geometry points may result in clusters that have small Euclidean distances but do not have road connections among elements(*77*). To this end, we use the actual road network distance to measure the spatial distance in this study. The conventional Dijkstra shortest path algorithm is implemented to obtain the shortest path between two reports, and the distance of the shortest path represents the spatial distances.

$$\Delta s_{i,j} = Dijkstra(i,j) \quad (21)$$

$\Delta s_{i,j}$ represents the spatial distance between two reports. If $\Delta s_{i,j}$ is greater than zero, meaning report j is located downstream of report i, and vice versa.

Dynamic ST-DBSCAN Algorithm

While ST-DBSCAN extends DBSCAN by adding a temporal dimension, it is still a static clustering procedure. Our problem demands an algorithm that can be implemented in real-time. In addition, the ST-DBSCAN algorithm has difficulties in distinguishing two different clusters that start at different times and locations but propagate and merge into each other over time. In order to differentiate clusters from one another in this situation, we propose a real-time implementation of ST-DBSCAN. The algorithm forms clusters dynamically and can differentiate clusters that start at different locations and times.

In real-time ST-DBSCAN, there are two distance parameters, spatial neighborhood radius ε_s and temporal neighborhood radius ε_t . A point p is the ε-neighborhood of point q if and only if the point p is within the ε_s-neighborhood and ε_t-neighborhood of point q.

The pseudocode of the algorithm is described in detail in **Figure 22**. D is a streaming dataset composed of jam reports that iarecontinuously updated by Waze. The algorithm starts with retrieving the eps (ε_t, ε_s) neighbors of each new coming point (the point that has not been labeled with any cluster). If the point has more than *minPts* neighbors, each neighbor is assigned to either a labeled dataset (N_label) or an unlabeled dataset ($N\_unlabel$) based on its current label status. If all neighbors of this point are not labeled earlier, then a new cluster starts and the neighbors are assigned to the new cluster. If all the labeled neighbors belong to the same cluster A, all the unlabeled neighbors are assigned to cluster A as well. If the neighbors belong to different clusters, each point in the unlabeled dataset is assigned to a specific cluster according to the *Assign_cluster* function. *Assign_cluster* is a function used if the neighbors of the subjected point are associate with two or more clusters.

Automatic Threshold Selection

One of the critical problems in the clustering algorithm is selecting reasonable thresholds to form meaningful clusters. In ST-DBSCAN, two thresholds, temporal distance ε_t and spatial distance ε_s are to be determined. The two parameters are generally determined based on domain knowledge or based on k-distance plot(*78*), where k represents *minPts*. It is expected that the core points and border points k-distance are within a certain range, while noise points can have much greater k-distance, thus an elbow pattern can be observed in the k-distance plot. However, in this study, the k-distance plot is rather smooth and does not show an obvious elbow pattern. In this section, an automatic threshold selection approach is proposed. Before showing the automatic threshold selection procedure, two concepts are introduced:

```
Real Time DBSCAN (D, ε_t, ε_s, minPts)
    Cluster_id = 0
    for each point P in dataset D
        mark P as visited
        ε_s-neighborhood = spatialNeighbors(G, P, ε_s)
        ε_t-neighborhood = timeNeighbors(D, P, ε_t)
        N = ε_s-neighborhood ∩ ε_t-neighborhood
        if sizeof (N) < minPts:
            mark P as Noise
        else
            N = N ∪ P
            Split N into N_unlabel and N_label
            nCluster = unique(N_label) /*nCluster store the number of unique
                                         clusters of each point in N_label*/
            If nCluster ≡ 0:
                Cluster_id = Cluster_id + 1
                Assign all points to N with Cluster_id
            Else if nCluster ≡1:
                Each point p_u in N_unlabel is labeled with the only
                cluster
            Else if nCluster >1:
                For each point p_u in N_unlabel:
                    Assign_cluster (p_u,N_label)

Assign_cluster(p_u,N_label)
    For each point p_l in N_label:
        Compute the Euclidean distance d_ul between p_u and p_l.
    Assign p_u to the cluster that p_l is in with smallest Euclidean distance
d_ul.
```

Figure 22. Real-time DBSCAN implementation.

Closer: for a report A, and two reports B and C, if $\text{abs}(\Delta s_{A,C}) < \text{abs}(\Delta s_{A,B})$ and $\text{abs}(\Delta t_{A,C}) < abs(\Delta t_{A,B})$, then report C is closer to report A than report B in space and in distance.

Nearest report: for reports A and C, if there exists no other report that is closer to C than A, then report C is one of the nearest reports of A.

Figure 23 demonstrates the concept of closer and nearest report. The temporal and spatial absolute distance for each report to report A is identified in the figure. In **Figure 23(a)**, report C is closer to report A than B, since $\Delta s_{A,C} < \Delta s_{A,B}$ and $\Delta t_{A,C} < \Delta t_{A,B}$. Similarly, report C is closer to report A than D. Therefore, report A has only one nearest report, which is C. However, in **Figure 23(b)**, there does not exist a report that is closer to report A than B and D. Therefore, report A has two nearest reports: B and D. Based on the nearest report concept, the proposed automatic threshold selection is demonstrated below.

In the automatic ST-threshold selection module (**Figure 24)**, the Waze user report patterns are explored. The dynamic clustering is performed with a large temporal-spatial threshold initially. Then, based on the clustering results, for each report A in the cluster, the nearest report B and its temporal and spatial distance to A is identified and stored in a list named ST-Distance-List.

ST-Distance-List reveals Waze user behavior of their report frequency and distance as well as the correlation between report distance and report frequency. In this paper, we do not consider the correlation of report frequency and distance when setting the threshold. That is, the report frequency and distance are considered as independent from each other. For all $\Delta t_{i,j}$ stored in ST-Distance-List, we compute the $(1-\alpha)$% values and set it as time threshold. That is, $(1-\alpha)$ % percent of Waze users will make reports within the

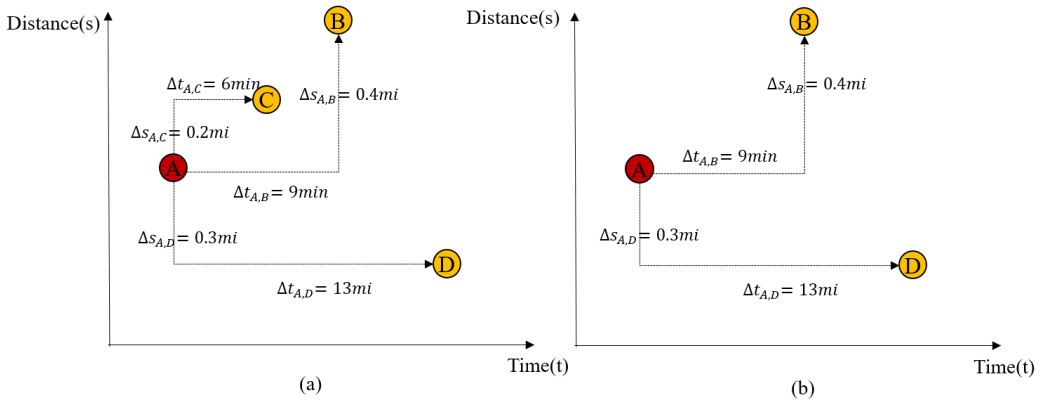


Figure 23. Closer and Nearest report.

Automatic ST-threshold Selection
Initialize a large threshold ε_s, ε_t
Call Real-Time ST-DBSCAN
Return clustering results
Create a ST-Distance-List list
For each cluster C:
    For each report i in C:
        Identify Nearest Report set NR.
            For each report j in NR
                Compute $\Delta t_{i,j}, \Delta s_{i,j}$ and append $(\Delta t_{i,j}, \Delta s_{i,j})$ to ST-Distance-List
Update the large threshold ε_s, ε_t:
ε_s is calculated as $(1-\alpha)\%$ values for all temporal distances stored in ST-Distance-List
ε_t is calculated as $(1-\beta)\%$ values for all spatial distances stored in ST-Distance-List

Figure 24. **Automatic thresholds election pseudocode.**

time threshold if congestion occurred. For all $\Delta s_{i,j}$ stored in ST-Distance-List, we compute the $(1-\beta)\%$ values and set it as the distance threshold. that is, $(1-\beta)$ % percent of Waze users will make reports within the distance threshold if congestion occurred.

End of Queue Identification

While clustering the jam reports, the following procedure is implemented to identify the end of queue dynamically. The 'term' queue has been defined in various ways in the literature. The most commonly adopted is vehicle speed less than some predefined threshold. For instance, a vehicle speed lower than 60km/h(38mph) is regarded as the end of queue in this paper(*64*). Since no speed information is associated with each Waze jam report, each report is regarded as within queue status. Identifying the end of queue is, therefore, identifying the jam reports that comprise the boundary of each cluster.

In this section, we focus on the queues that propagate upstream and define a backward forming shock wave front. As shown in

Figure 25, $\Delta d_{0,i}$, and $\Delta t_{0,i}$ separately represent the spatial distance and temporal distance from the first report to the i^{th} report (red point represents the first report in this cluster and green point represent the i^{th} report). Note that $\Delta d_{0,i}$ is less than zero, and $\Delta t_{0,i}$ is greater than zero in the example.

Backward forming shock wavefront: a set of reports that for any report i, if there does not exist another report j in the same cluster that $\Delta t_{0,j} < \Delta t_{0,i}$ and $\Delta d_{0,j} < \Delta d_{0,i}$, then report i is defined as backward forming shock wavefront, or in other words, end of queue at the time report i is made.

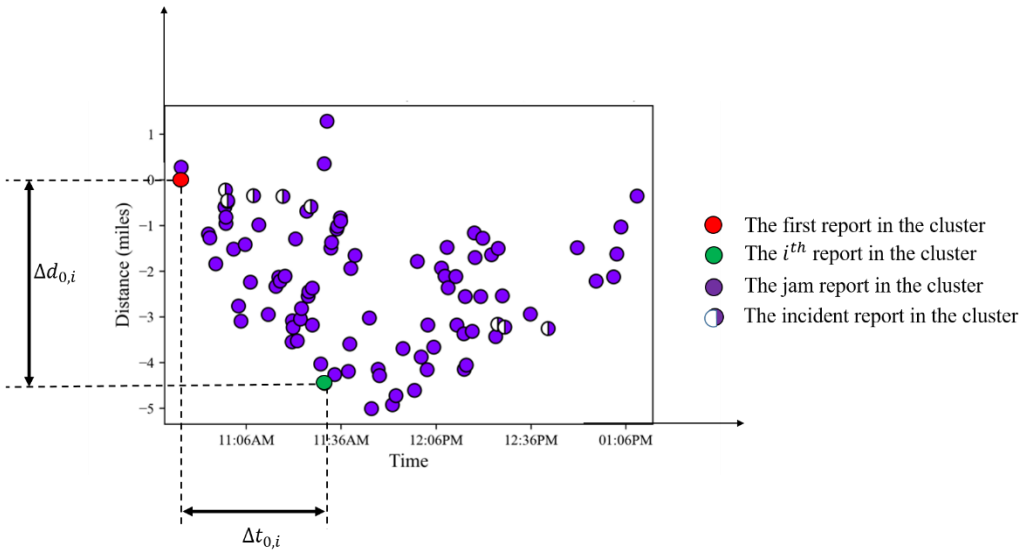


Figure 25. Backward forming shock wave.

Application: Case Study of Knoxville

The city of Knoxville is conveniently located just off I-40 and I-75, minutes from I-81, and is within a day's drive of half the continental U.S. Since the roads in Knoxville are always under excessive pressure at peak-hours and the likelihood of incidents and congestion is high, the accurate detection of traffic congestion is important to transportation agencies, engineers, researchers, and planners.

The Waze jam reports used in this study were collected from Aug. 2017 to Dec. 2017 in the Knoxville region. Figure 26 is an overview of the selected region as well as the spatial distribution of jam reports on the road network. In the figure, the color scale of road segment represents Waze reports density; Red represent highest density level; Yellow represents middle density level; Green represent low density level. a layer of heat-map is added to show the naturally formed clusters without considering the temporal dimension. In this section, we demonstrate an implementation of our algorithm using streaming Waze data and present the clustering results, as summarized:

(1) *Parameters selection.*
We implement the dynamic ST-DBSCAN algorithm as well as the dynamic parameter selection approach.

(2) *Comparison of static ST-DBSCAN and dynamic ST-DBSCAN.*
The detection results from dynamic ST-DBSCAN is compared to that of static ST-DBSCAN algorithm.

(3) *End of queue identification.*
The end of queue in each cluster is located in real-time based on the proposed EOQ identification method.

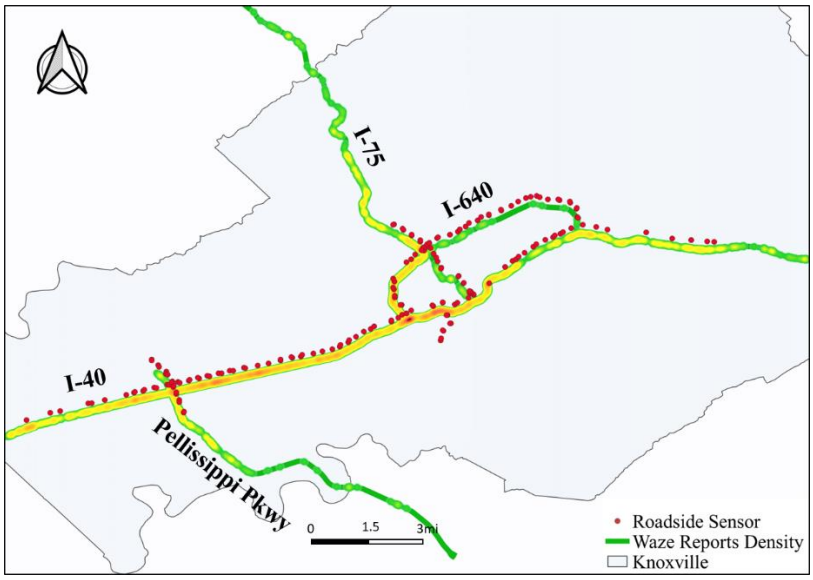


Figure 26. The spatial distribution of Waze jam reports, Knoxville.

(4) *Benchmarking.*
This section compares the EOQ identification results with detection results from road side sensor data.

Results

This section first presents the parameter selection of the proposed algorithm. Then, the proposed dynamic algorithm is compared with the static algorithm. Finally, the end of queue identification module is implemented and the detection results are compared to the results of a well-established method.

Parameter Selection

There are three parameters to be determined separately: *minPts*, a temporal threshold ε_t, and a spatial threshold ε_s. There is no general way of choosing *minPts*; it typically requires the knowledge of the dataset. In this study, because the reports are located on the roadway, sometimes clusters reveal linear patterns (on a straight line like on an Interstate Highway) in which case the core point is occasionally surrounded by only two points. Therefore, the *minPts* should not be larger than 3.

Using the automatic threshold selection procedure developed in the methodology section, temporal radius ε_t and spatial radius ε_s can be determined. **Figure 27** demonstrates the distribution of the spatial and temporal distance of two nearest reports. In the figure, it can be observed that the spatial-temporal distance can be as high one hour

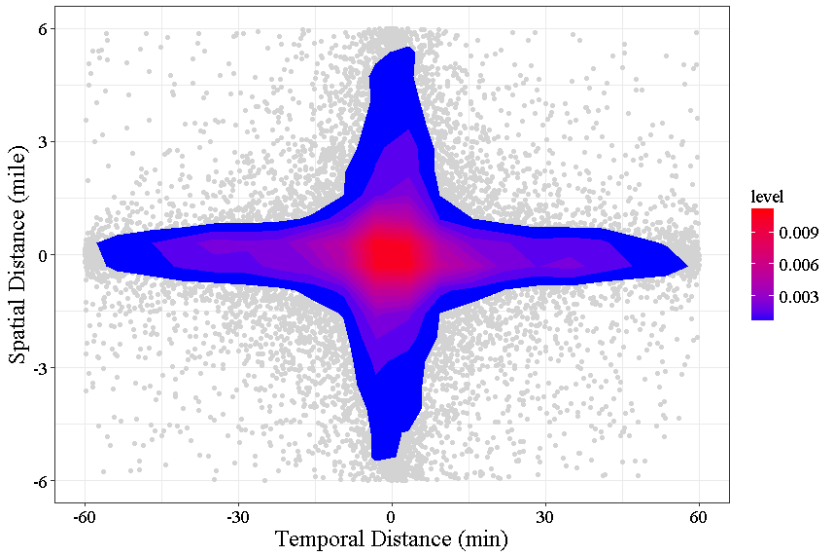
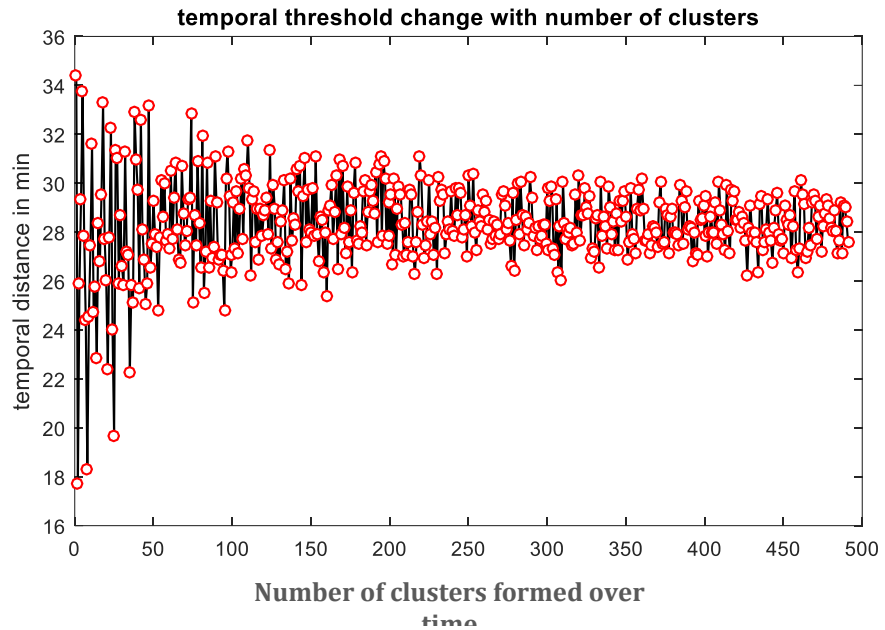


Figure 27. **The temporal and spatial distance of two nearest reports.**

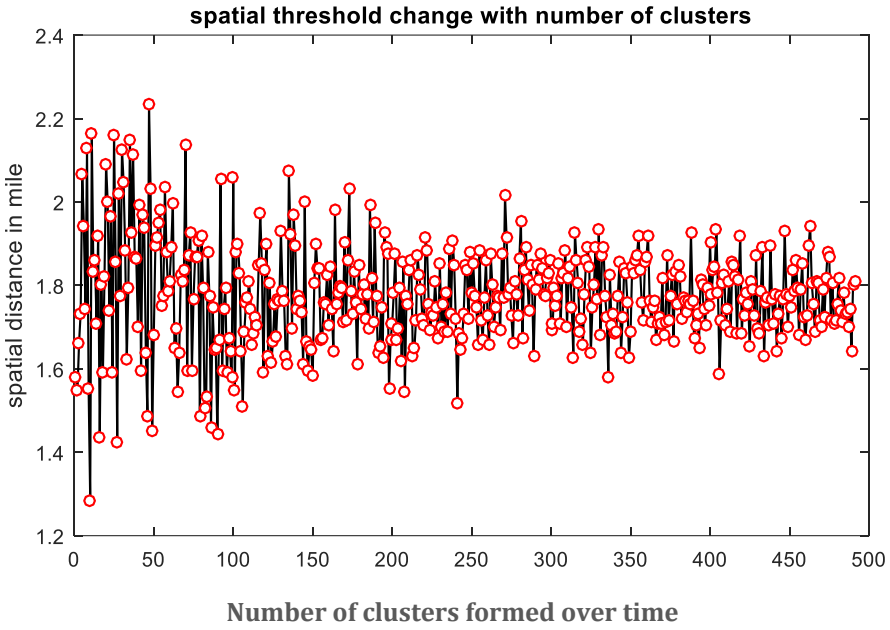
and 6 miles. Therefore, it is necessary to choose appropriate thresholds to produce meaningful clusters.

Figure 28(a) and **Figure 28(b)** show the change of temporal and spatial threshold over time. The thresholds are computed using the 90th percentile values. The x axis is the number of clusters identified over time. **Figure 28(a)** shows that the temporal threshold started at 34-min and changed with different number of clusters. After some initial fluctuations, the threshold got stable and was around 28-min. Similarly, the spatial threshold got stable at around 1.7 miles (**Figure 28(b)**).

The 28-min time threshold is reasonable since Waze keeps a report from a user for at least 30 minutes unless it is confirmed as a false alarm. Therefore, the majority of Waze reports remain in the system for at least 30 minutes. If there is a report already in place, people are less likely to make another report during the same period; instead, they are more likely to confirm the report with thumbs up or communicate under the existing report which further prolonged the existence duration of the report. The 1.6-mile distance threshold is reasonable as well because sometimes the queue propagates quickly, especially queue caused by crashes. It only takes several minutes for a queue to move 1.6 miles. Therefore, the distance between any two nearest reports belongs to the same cluster can be as long as 1.6 miles.



(a)



(b)

Figure 28. Threshold selection. (a) temporal threshold changes over time; (b) spatial threshold change over time.

Comparison of Static ST-DBSCAN Algorithm and Dynamic ST-DBSCAN Algorithm

Next, a comparison is made between the static clustering method and the dynamic clustering method. We run the static and dynamic algorithms respectively on the same dataset; the results are shown in the following figure. **Figure 29** is composed of three examples: Aug. 21th 2017, Sept 15th, 2017 and Oct 8th, 2017 on I-40 westbound. The left column plots consist of three clusters identified using static DBSCAN and the right column plot shows the clusters identified using dynamic DBSCAN.

In **Figure 29**, different colors/shapes represent different clusters. As shown in the figure, the data are grouped in the same cluster using a static algorithm (same color: purple), while they are classified into different clusters that feature different start time and location using a dynamic DBSCAN algorithm. For example, on September 15th, the static algorithm identifies one congestion that colored with purple; The dynamic algorithm identifies two clusters separately colored with red and green. The dynamic algorithm is more reasonable. The congestion featured with red color is different from the congestion featured with green color since they started at different locations and different times and merged over time.

End of Queue Identification and Benchmarking

The end of queue identification method is applied. Figure 30 demonstrates the identification results for a specific case. All the points in this figure belong to the same cluster. The congestion started at around 3:10 PM and milepost 377.5. The queue propagated upstream over time. Points colored with green represent the identified end of queue for this cluster. Each point represents the end of queue location when the report was generated.

For further assessment, the proposed algorithm is compared to the speed threshold algorithm developed in a previous study(*79*). The speed threshold method is often employed as a benchmarking method in traffic queue detection algorithms(*57*). The thresholds of this algorithm have been identified as a range between 30 mph and 40 mph for the freeways in Portland and San Diego, U.S, respectively(*79; 80*). In this study, 30mph is selected as the threshold to detect congestion based on roadside sensor data and the detection results based on two data sources are compared.

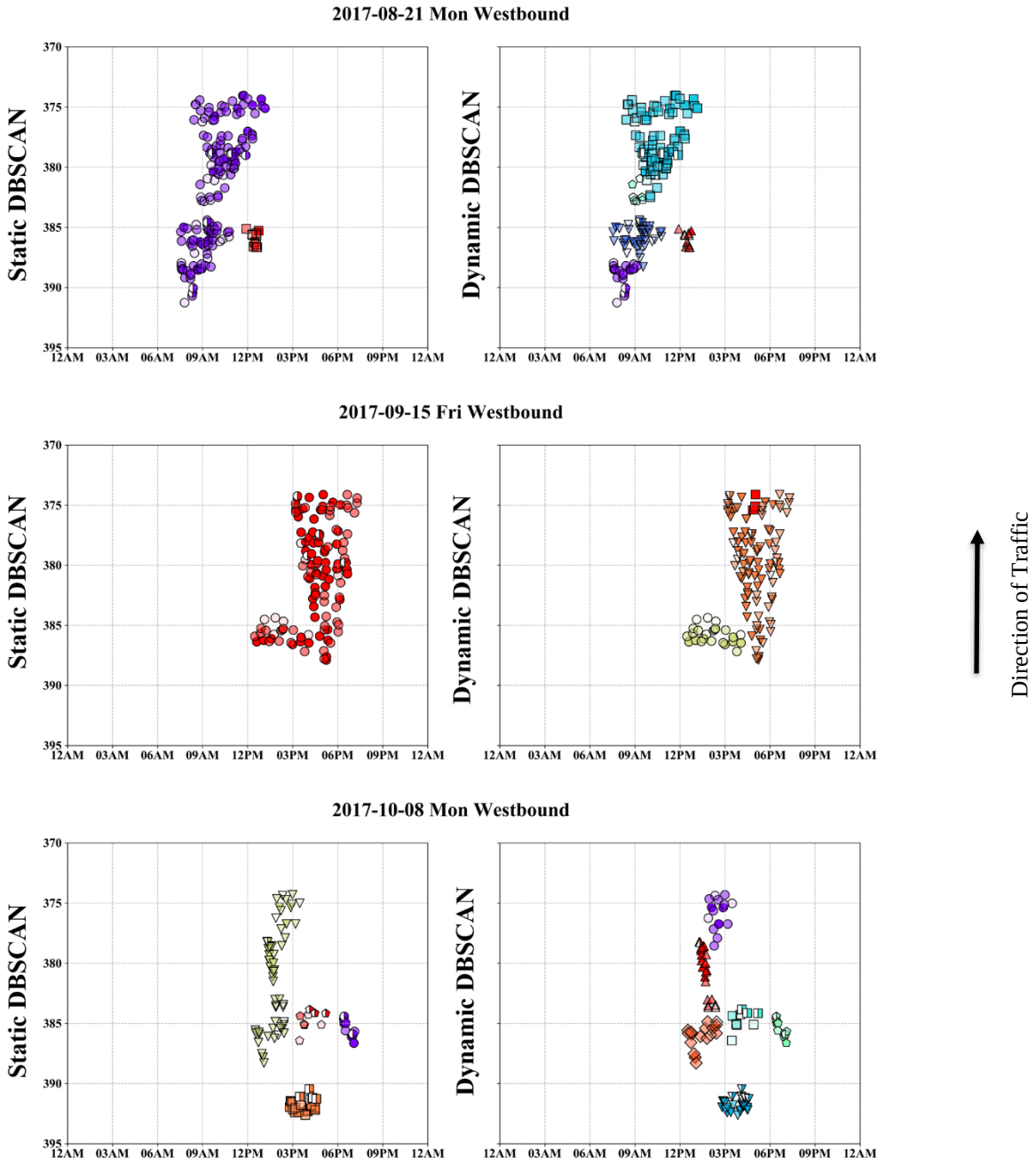


Figure 29. Clusters discovered using the static and dynamic algorithm.

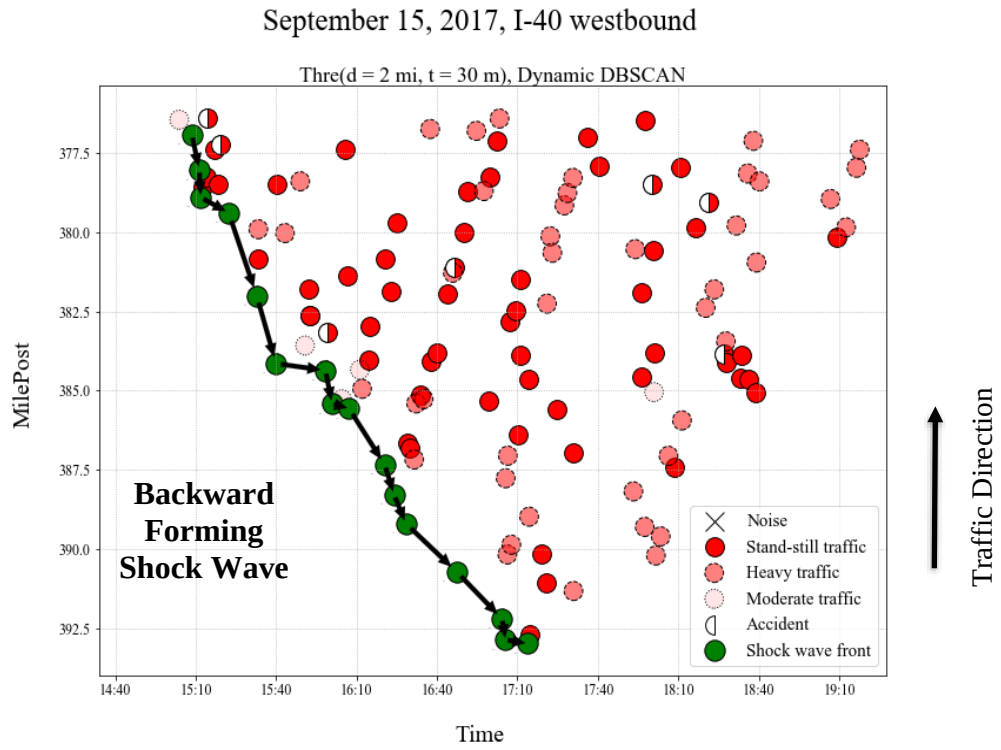


Figure 30. Example of backward forming shock wave detection.

Figure 31 is a demonstration of the comparison of the end of queue detection results using Waze and remote traffic microwave sensor (RTMS) data. Two remote traffic microwave detectors separately locate at d_1 and d_2. The purple points denote where and when the EOQ is detected by the detectors using the Speed Threshold method. The red point denotes where and when the EOQ is detected by the proposed method based on Waze reports. Assuming there is another detector located at d_A, where the red point is located. The detection results for this virtual detector are estimated using linear interpolation:

$$t_{A'} = \frac{t_2 - t_1}{d_2 - d_1} * (d_A - d_1) + t_1 \quad (22)$$

t_1, d_1 separately represent the end of queue detection time and location of the first detector. Similarly, t_2, d_2 separately represent the end of queue detection time and location of the second detector. t_A ,d_A represent the detection time and location of the proposed method based on Waze data. $t_{A'}$ represents the estimated detection time of speed threshold method assuming a detector locates at d_A. Then, the difference between the proposed method and speed threshold method is $\Delta t = t_A - t_{A'}$.

Severe congestion which has remarkable impacts on traffic from Aug. 1st to Dec. 27th in the Knoxville are identified with more than a certain number of Waze reports, resulting in 43 cases. 9 cases were then removed either because of no obvious backward queuing pattern or because of questionable roadside sensor data. The detection results of the speed threshold method and the proposed method for the 34 cases are compared.

Table 12 shows the end of queue detection frequency for both datasets. For RTMS data, on average, 1.8 points are reported every mile of congestion. For Waze data, on average, 1.9 points are reported every mile. The results suggest that Waze has comparable end of queue reporting frequency with roadside sensor data. Then, for each

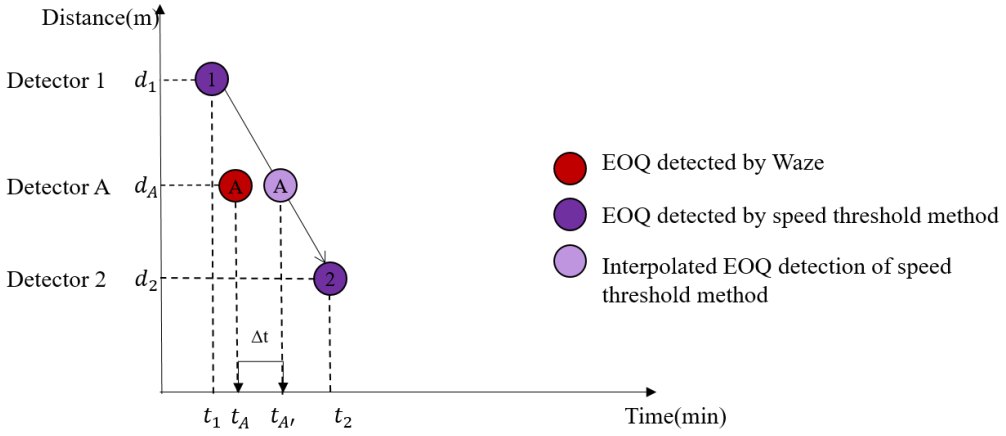


Figure 31.Comparison of end of queue detection results based on Waze data and roadside detector data.

Table 12. Detection statistics for RTMS data and WAZE data for the 43 cases.

	# of Detections	Total distance(mile)	Average detection per mile
RTMS data	276	152	1.8
Waze data	286	152	1.9

of the 385 Waze EOQ reporting points, the detection time difference between the proposed method and speed threshold method is computed based on Equation (3). The average detection time differences for the 385 Waze EOQ reporting points between the speed threshold method and the proposed method are 1.1 minutes, with a standard deviation of 7.2. That is, the speed threshold method based on traffic detector data reports the end of queue 1.1 minutes earlier than the proposed method based on Waze data. The small detection time difference between the two data sources suggest they are comparable in terms of timeliness.

Conclusion and Future Work

Knowledge of the location of traffic queues after non-recurring events such as incidents serves as the foundation of managing and protecting the queue in real-time to reduce delays as well as decrease the occurrence of rear-end collisions. The objective of this study is to take advantage of location-based crowdsourced data from Waze reports to develop an automatic clustering algorithm and tracking the movement of the queue.

In this study, we propose an algorithm that clusters the live Waze reports, specifically, incident reports and jam reports. The algorithm is a real-time extension of the traditional spatial-temporal DBSCAN algorithm. It dynamically forms clusters with incoming Waze reports and tracks the movement of congestion as well as end of queue over time. A dynamic threshold selection approach is developed by characterizing the Waze user reports spatial-temporal distributions under congestion.

The proposed algorithm was tested in the Knoxville area with 34 severe traffic congestion cases. Both the static and dynamic ST-DBSCAN algorithm was executed with the automatic threshold selection approach. Results demonstrate the proposed dynamic algorithm outperforms the static algorithm as it can distinguish clusters that start at different times and locations.

A comparison of the proposed algorithm with the speed threshold method based on roadside sensor data was performed. On average, the detection results of the proposed method and speed threshold method are quite close. The detection time difference is only 1.1 minutes. In addition, the reporting frequency of Waze and traffic sensor data is close as well. Waze reports 1.9 end of queue detection points every mile, while the traffic detector generates 1.8 detection points every mile. The comparison demonstrates that

Waze data is similar for congestion and end of queue detection. This is particularly valuable for the many stretches of roadway without traffic sensors.

This research mainly assessed the performance of the EOQ detection during severe congestion, under which the queue is propagating quickly and is most dangerous to drivers. In future research, the performance shall be evaluated for different traffic jam conditions. Moreover, despite the information directly provided in each record, all Waze reports have reliability scores and existence-interval (the time duration the Waze reports present on the application). In this study, we only consider the timestamp that the report is made but not the time duration that the report presents. In future research, the duration information, as well as reliability, shall be considered to improve the detection results.

CONCLUSION

This dissertation compiled a series of studies promote the understanding of data collected by various sources and abnormal traffic movements s in a spatial-temporal domain to support real-time traffic operations. These studies were conducted to propose multiple applications to propose data aggregation method for probe vehicle-based data, evaluate the reliability of crowdsourced user reports, detect abnormal events on freeway, and the evaluate the possibility of detecting the spatial-temporal impact of congestion with a complementary data source.

First, a sampling strategy-based aggregation method for probe vehicle speed data was proposed. Two sapling strategy: time-based sapling and distance-based sapling strategy discussed in this paper. The proposed aggregation method is evaluated with real-word NGSIM data. Results shows that the speed estimation accuracy of the proposed method is consistence higher comparing to the harmonic mean method under different traffic conditions.

Second, the crowdsourced user reports were evaluated by comparing them with official records in both spatial and temporal dimension. The comparison results suggest Waze users, on average, tend to make reports at the exact location. In addition, the Waze user reports are made sooner than the official records, and thus can be incorporated in the real time system to reduce the abnormal event detection time. This study can be furthered by modeling Waze user behavior in both temporal and spatial dimension to improve the estimation accuracy of the location and occurrence time of an event.

Third, a self-learning freeway incident detection algorithm is proposed which requires no calibration. The algorithm is composed of two modules: self-training and self-evaluation. In the training module, the proposed algorithm outperformed the benchmark algorithm is in terms of detection rate, false alarm rate, and mean time to detection. Further study is recommended to enhance the detection ability during peak period.

Finally, a queue-end detection algorithm was proposed to detect the spatial-temporal impacts of abnormal events to warn unaware approaching drivers. A dynamic clustering algorithm was developed to cluster Waze reports and determine the impact region. A case study was performed to demonstrate the ability of the proposed algorithm. The detection results were compared to a benchmark method that uses road slide sensor data. results suggest that Waze data is a reliable alternative for roadside sensor data in the application of queue-end detection.

Altogether, this dissertation provides a real-time traffic state assessment and detection framework that consists of data quality evaluation tools and algorithms for traffic operations of highway facilities.

This dissertation contributes to the understanding of fundamental traffic parameters and provides innovative algorithms to promote the dynamic detection of abnormal traffic status. Speed, flow and density are the three fundamental parameters in traffic flow theory and are involved in every aspect of traffic analysis. Incorrect speed, flow and density measurements may shake the foundation of the research based on those parameters. Therefore, one of the key contributions of this dissertation is that it clarifies

how the mean speed shall be correctly calculated for prevalent probe vehicle-based data and points out the direct employment of a traditional method leads to biased results. The study has implications for the calculation of speed, flow and density in the long run. Suitable calculation methods shall be derived from the basic definition of the fundamental parameters if innovative data collection techniques emerge, such as autonomous and connected vehicle technologies.

Building on the first chapter, the dissertation proposed a detection algorithm and a dynamic clustering algorithm. The proposed algorithms are readily usable to traffic practitioners. The algorithm proposed in Chapter III addresses the limited transferability issue of current incident detection algorithms and proposes an evaluation module. The module can be adopted by future researchers and serves as a calibration process to improve the transferability of their own incident detection algorithms. The concept of using long term performance to evaluate and verify the detection results applies to different detection systems as well. The algorithm proposed in Chapter IV cluster streaming data in real-time to find meaningful patterns. While the algorithm is developed specifically for WAZE data, a modification of it applies to other data sources. The two studies represent a step towards the 'fully intelligent' traffic operation system since both algorithms were designed to achieve good performance with minimum human interventions.

The dissertation focused on traffic detection utilizing various data sources. The detection system provides real-time traffic states for strategy making. In addition to the future research potentials addressed at the end of individual chapters, the future research direction of the entire dissertation is to include a prediction and decision-making system. **Figure 32** provides a diagram that incorporates future studies into the current dissertation structure. The prediction system takes both the fundamental parameters and the detection results as input and predicts the queue movement. The prediction results feed into a proactive management system to support practitioners' decision making. The operational strategies such as queue mitigation, traffic diverting further change the traffic flow movement, and reversely affect the prediction.

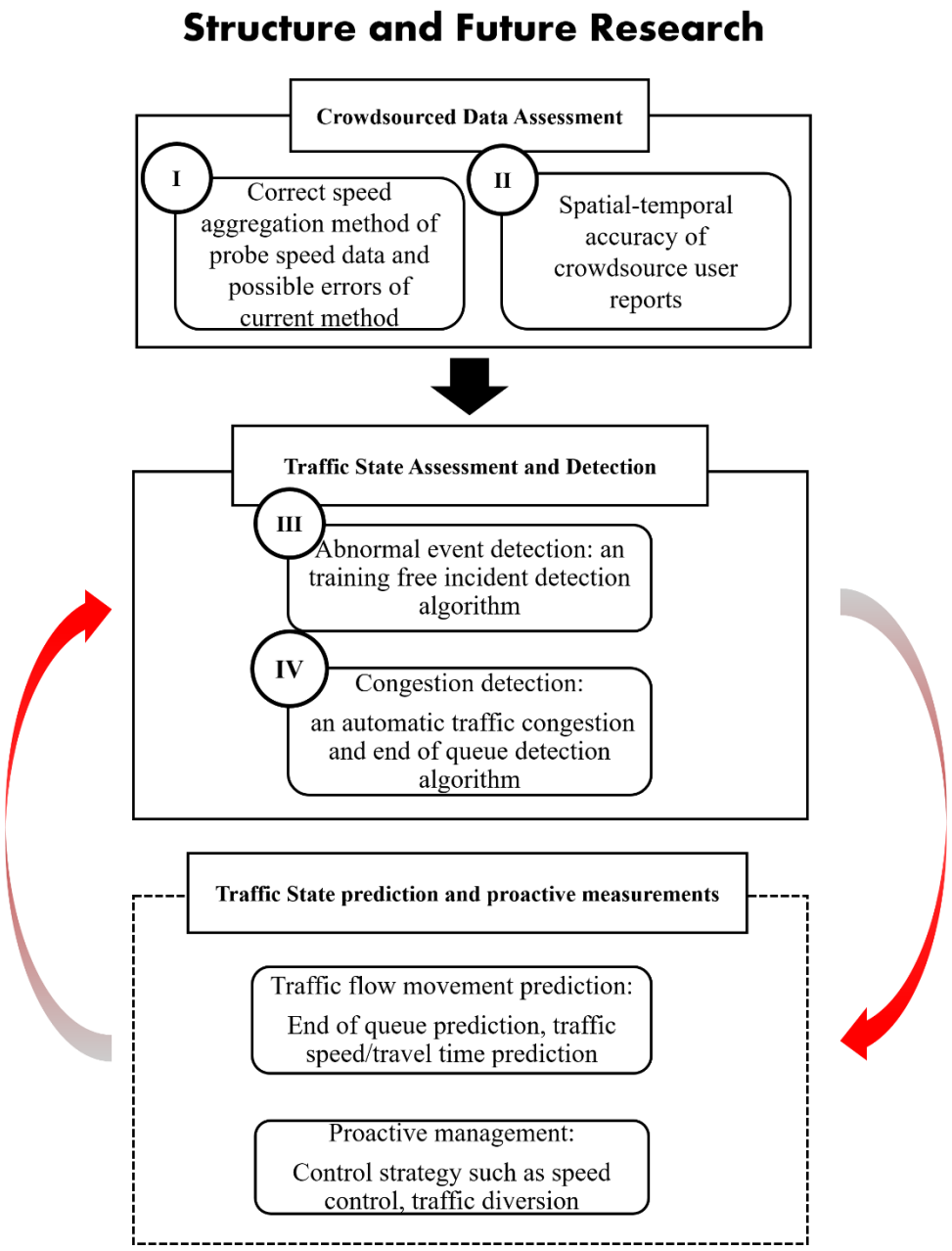


Figure 32. Future study.

REFERENCES

[1] Alexiadis, V., J. Colyar, J. Halkias, R. Hranac, and G. McHale. The next generation simulation program. *Institute of Transportation Engineers. ITE Journal,* Vol. 74, No. 8, 2004, p. 22.
[2] Patire, A. D., M. Wright, B. Prodhomme, and A. M. Bayen. How much GPS data do we need? *Transportation Research Part C: Emerging Technologies,* Vol. 58, 2015, pp. 325-342.
[3] Kim, S., and B. Coifman. Comparing INRIX speed data against concurrent loop detector stations over several months. *Transportation Research Part C: Emerging Technologies,* Vol. 49, 2014, pp. 59-72.
[4] Bucknell, C., and J. C. Herrera. A trade-off analysis between penetration rate and sampling frequency of mobile sensors in traffic state estimation. *Transportation Research Part C: Emerging Technologies,* Vol. 46, 2014, pp. 132-150.
[5] Goodall, N., and E. Lee. Comparison of Waze crash and disabled vehicle records with video ground truth. *Transportation Research Interdisciplinary Perspectives,* 2019, p. 100019.
[6] Amin-Naseri, M., P. Chakraborty, A. Sharma, S. B. Gilbert, and M. Hong. Evaluating the Reliability, Coverage, and Added Value of Crowdsourced Traffic Incident Reports from Waze. *Transportation Research Record,* Vol. 2672, No. 43, 2018, pp. 34-43.
[7] Argote-Cabañero, J., E. Christofa, and A. Skabardonis. Connected vehicle penetration rate for estimation of arterial measures of effectiveness. *Transportation Research Part C: Emerging Technologies,* Vol. 60, 2015, pp. 298-312.
[8] Haghani, A., M. Hamedi, K. Sadabadi, S. Young, and P. Tarnoff. Data collection of freeway travel time ground truth with bluetooth sensors. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2160, 2010, pp. 60-68.
[9] Hargrove, S. R., H. Lim, L. D. Han, and P. B. Freeze. Empirical Evaluation of the Accuracy of Technologies for Measuring Average Speed in Real Time. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2594, 2016, pp. 73-82.
[10] Wardrop, J. G. Some theoretical aspects of road traffic research.In *Inst Civil Engineers Proc London/UK/*, 1952.
[11] Hall, F. L. Traffic stream characteristics. *Traffic Flow Theory. US Federal Highway Administration*, 1996.
[12] Wardrop, J. G. ROAD PAPER. SOME THEORETICAL ASPECTS OF ROAD TRAFFIC RESEARCH. *Proceedings of the institution of civil engineers,* Vol. 1, No. 3, 1952, pp. 325-362.
[13] Edie, L. C. *Discussion of traffic stream measurements and definitions*. Port of New York Authority, 1963.
[14] Quiroga, C. A., and D. Bullock. Travel time studies with global positioning and geographic information systems: an integrated methodology. *Transportation research part C: emerging technologies,* Vol. 6, No. 1, 1998, pp. 101-127.
[15] Kaushik, K., E. Sharifi, and S. E. Young. Computing Performance Measures with National Performance Management Research Data Set. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2529, 2015, pp. 10-26.

[16] Lattimer, C., and G. Glotzbach. Evaluation of third party travel time data.In *ITS America 22nd Annual Meeting & ExpositionITS America*, 2012.
[17] Pu, W. Standardized Data Processing: When Is It Needed in the Mining of Private-Sector Probe-Based Traffic Data to Measure Highway Performance? *Transportation Research Record: Journal of the Transportation Research Board*, No. 2338, 2013, pp. 44-57.
[18] Gao, H., and F. Liu. Estimating freeway traffic measures from mobile phone location data. *European journal of operational research,* Vol. 229, No. 1, 2013, pp. 252-260.
[19] Ou, Q., J. Van Lint, and S. Hoogendoorn. Piecewise inverse speed correction by using individual travel times. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2049, 2008, pp. 92-102.
[20] Jenelius, E., and H. N. Koutsopoulos. Probe vehicle data sampled by time or space: consistent travel time allocation and estimation. *Transportation Research Part B: Methodological*, Vol. 71, 2015, pp. 120-137.
[21] Liu, K., T. Yamamoto, and T. Morikawa. Comparison of time/space polling schemes for a probe vehicle system.In *Proceedings of the 14th World Conference on Intelligent Transport Systems*, 2007.
[22] Westgate, B. S., D. B. Woodard, D. S. Matteson, and S. G. Henderson. Travel time estimation for ambulances using Bayesian data augmentation. *The Annals of Applied Statistics*, 2013, pp. 1139-1161.
[23] Ye, Q., W. Y. Szeto, and S. C. Wong. Short-term traffic speed forecasting based on data recorded at irregular intervals. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 13, No. 4, 2012, pp. 1727-1737.
[24] Herrera, J. C., D. B. Work, R. Herring, X. J. Ban, Q. Jacobson, and A. M. Bayen. Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment. *Transportation Research Part C: Emerging Technologies*, Vol. 18, No. 4, 2010, pp. 568-583.
[25] Adler, J., J. Horner, J. Dyer, A. Toppen, L. Burgess, and G. Hatcher. Estimate benefits of crowdsourced data from social media.In, United States. Department of Transportation. Intelligent Transportation …, 2014.
[26] Sakaki, T., M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors.In *Proceedings of the 19th international conference on World wide web*, ACM, 2010. pp. 851-860.
[27] Gu, Y., Z. S. Qian, and F. Chen. From Twitter to detector: Real-time traffic incident detection using social media data. *Transportation Research Part C: Emerging Technologies*, Vol. 67, 2016, pp. 321-342.
[28] Salas, A., P. Georgakis, and Y. Petalas. Incident detection using data from social media.In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2017. pp. 751-755.
[29] Xu, S., S. Li, and R. Wen. Sensing and detecting traffic events using geosocial media data: A review. *Computers, Environment and Urban Systems*, Vol. 72, 2018, pp. 146-160.

[30] Paule, J. D. G., Y. Sun, and Y. Moshfeghi. On fine-grained geolocalisation of tweets and real-time traffic incident detection. *Information Processing & Management*, Vol. 56, No. 3, 2019, pp. 1119-1132.
[31] Silva, T. H., P. O. V. De Melo, A. C. Viana, J. M. Almeida, J. Salles, and A. A. Loureiro. Traffic condition is more than colored lines on a map: characterization of waze alerts.In *International Conference on Social Informatics*, Springer, 2013. pp. 309-318.
[32] Flynn, D. B., M. M. Gilmore, and E. A. Sudderth. Estimating Traffic Crash Counts Using Crowdsourced Data: Pilot analysis of 2017 Waze data and Police Accident Reports in Maryland.In, John A. Volpe National Transportation Systems Center (US), 2018.
[33] Hourdakis, J. AIDDS: A System for Developing and Testing Incident Detection Algorithms. *IFAC Proceedings Volumes*, Vol. 30, No. 8, 1997, pp. 119-124.
[34] Williams, B. M., and A. Guin. Traffic management center use of incident detection algorithms: Findings of a nationwide survey. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 8, No. 2, 2007, pp. 351-358.
[35] Parkany, E., and C. Xie. A complete review of incident detection algorithms & their deployment: what works and what doesn't.In, 2005.
[36] Payne, H. J., and S. C. Tignor. Freeway incident-detection algorithms based on decision trees with states. *Transportation Research Record*, No. 682, 1978.
[37] Cohen, S., and Z. Ketselidou. A calibration process for automatic incident detection algorithms.In *Microcomputers in Transportation*, ASCE, 1993. pp. 506-515.
[38] Stephanedes, Y. J., and A. P. Chassiakos. Application of filtering techniques for incident detection. *Journal of Transportation Engineering*, Vol. 119, No. 1, 1993, pp. 13-26.
[39] Chassiakos, A. P., and Y. J. Stephanedes. Smoothing algorithms for incident detection. *Transportation Research Record*, Vol. 1394, 1993, pp. 8-16.
[40] Martin, P. T., J. Perrin, B. Hansen, R. Kump, and D. Moore. Incident detection algorithm evaluation. *Prepared for Utah Department of Transportation*, 2001.
[41] Jin, Y., J. Dai, and C.-T. Lu. Spatial-temporal data mining in traffic incident detection.In *Proc. SIAM DM 2006 Workshop on Spatial Data Mining, No. 5*, 2006.
[42] Ahmed, S. A., and A. R. Cook. Time series models for freeway incident detection. *Transportation engineering journal of the American Society of Civil Engineers*, Vol. 106, No. 6, 1980, pp. 731-745.
[43] Ahmed, S., and A. R. Cook. Application of time-series analysis techniques to freeway incident detection. *Transportation Research Record*, Vol. 841, 1982, pp. 19-21.
[44] Persaud, B. N., F. L. Hall, and L. M. Hall. Congestion identification aspects of the McMaster incident detection algorithm. *Transportation Research Record*, No. 1287, 1990.
[45] Gall, A. I., and F. L. Hall. Distinguishing between incident congestion and recurrent congestion: a proposed logic. *Transportation Research Record*, No. 1232, 1989.
[46] Balke, K., C. L. Dudek, and C. E. Mountain. Using probe-measured travel times to detect major freeway incidents in Houston, Texas. *Transportation Research Record*, Vol. 1554, No. 1, 1996, pp. 213-220.
[47] Levin, M., and G. M. Krause. Incident detection: a Bayesian approach. *Transportation Research Record*, Vol. 682, 1978, pp. 52-58.

[48] Castro-Neto, M. M., L. D. Han, Y.-S. Jeong, and M. K. Jeong. Toward Training-Free Automatic Detection of Freeway Incidents: Simple Algorithm with One Parameter. *Transportation Research Record,* Vol. 2278, No. 1, 2012, pp. 42-49.
[49] Leys, C., C. Ley, O. Klein, P. Bernard, and L. Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology,* Vol. 49, No. 4, 2013, pp. 764-766.
[50] Consortium, A. W. Z. S. Innovative End-of Queue Warning System Reduces Crashes Up to 45%. *ARTBA Work Zone Safety Consortium*, 2015.
[51] Sharma, A., D. M. Bullock, and J. A. Bonneson. Input–output and hybrid techniques for real-time prediction of delay and maximum queue length at signalized intersections. *Transportation Research Record,* Vol. 2035, No. 1, 2007, pp. 69-80.
[52] Li, J.-Q., K. Zhou, S. E. Shladover, and A. Skabardonis. Estimating Queue Length under Connected Vehicle Technology: Using Probe Vehicle, Loop Detector, Fused Data. *Transportation Research Record,* Vol. 2366, No. 1, 2013, pp. 17-22.
[53] Cheng, Y., X. Qin, J. Jin, and B. Ran. An exploratory shockwave approach to estimating queue length using probe trajectories. *Journal of Intelligent Transportation Systems,* Vol. 16, No. 1, 2012, pp. 12-23.
[54] Khan, A. M. Intelligent infrastructure-based queue-end warning system for avoiding rear impacts. *IET intelligent transport systems,* Vol. 1, No. 2, 2007, pp. 138-143.
[55] Jiang, X., and H. Adeli. Object-oriented model for freeway work zone capacity and queue delay estimation. *Computer-Aided Civil and Infrastructure Engineering,* Vol. 19, No. 2, 2004, pp. 144-156.
[56] Hourdos, J., Z. Liu, P. Dirks, H. X. Liu, S. Huang, W. Sun, and L. Xiao. Development of a queue warning system utilizing ATM infrastructure system development and field-testing. 2017.
[57] Bae, B., Y. Liu, L. D. Han, and H. Bozdogan. Spatio-temporal traffic queue detection for uninterrupted flows. *Transportation Research Part B: Methodological,* Vol. 129, 2019, pp. 20-34.
[58] Chung, Y., and W. W. Recker. A methodological approach for estimating temporal and spatial extent of delays caused by freeway accidents. *IEEE Transactions on Intelligent Transportation Systems,* Vol. 13, No. 3, 2012, pp. 1454-1461.
[59] Geroliminis, N., and A. Skabardonis. Identification and analysis of queue spillovers in city street networks. *Geroliminis N. and Skabardonis, A.,(2011). Identification and Analysis of Queue Spillovers in City Street Networks. IEEE Transactions on Intelligent Transportation Systems,* Vol. 12, No. EPFL-ARTICLE-175550, 2011, pp. 1107–1115.
[60] Elfar, A., C. Xavier, A. Talebpour, and H. S. Mahmassani. Traffic Shockwave Detection in a Connected Environment Using the Speed Distribution of Individual Vehicles.In, 2018.
[61] Ban, X. J., P. Hao, and Z. Sun. Real time queue length estimation for signalized intersections using travel times from mobile sensors. *Transportation Research Part C: Emerging Technologies,* Vol. 19, No. 6, 2011, pp. 1133-1156.
[62] Comert, G., and M. Cetin. Queue length estimation from probe vehicle location and the impacts of sample size. *European journal of operational research,* Vol. 197, No. 1, 2009, pp. 196-202.

[63] Izadpanah, P., B. Hellinga, and L. Fu. Automatic traffic shockwave identification using vehicles' trajectories.In *Proceedings of the 88th Annual Meeting of the Transportation Research Board (CD-ROM)*, 2009.
[64] Dinh, T.-U., R. Billot, E. Pillet, and N.-E. El Faouzi. Real-Time Queue-End Detection on Freeways with Floating Car Data: Practice-Ready Algorithm. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2470, 2014, pp. 46-56.
[65] Ramezani, M., and N. Geroliminis. Queue profile estimation in congested urban networks with probe data. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 30, No. 6, 2015, pp. 414-432.
[66] Kan, Z., L. Tang, M.-P. Kwan, C. Ren, D. Liu, and Q. Li. Traffic congestion analysis at the turn level using Taxis' GPS trajectory data. *Computers, Environment and Urban Systems*, Vol. 74, 2019, pp. 229-243.
[67] Li, L., L. Zhu, and D. Z. Sui. A GIS-based Bayesian approach for analyzing spatial–temporal patterns of intra-city motor vehicle crashes. *Journal of Transport Geography*, Vol. 15, No. 4, 2007, pp. 274-285.
[68] Kisilevich, S., F. Mansmann, M. Nanni, and S. Rinzivillo. Spatio-temporal clustering.In *Data mining and knowledge discovery handbook*, Springer, 2009. pp. 855-874.
[69] Pei, T., C. Zhou, A.-X. Zhu, B. Li, and C. Qin. Windowed nearest neighbour method for mining spatio-temporal clusters in the presence of noise. *International Journal of Geographical Information Science*, Vol. 24, No. 6, 2010, pp. 925-948.
[70] Deng, M., Q. Liu, J. Wang, and Y. Shi. A general method of spatio-temporal clustering analysis. *Science China Information Sciences*, Vol. 56, No. 10, 2013, pp. 1-14.
[71] Birant, D., and A. Kut. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, Vol. 60, No. 1, 2007, pp. 208-221.
[72] Amini, A., H. Saboohi, T. Herawan, and T. Y. Wah. MuDi-Stream: A multi density clustering algorithm for evolving data stream. *Journal of Network and Computer Applications*, Vol. 59, 2016, pp. 370-385.
[73] Kulldorff, M. A spatial scan statistic. *Communications in Statistics-Theory and methods*, Vol. 26, No. 6, 1997, pp. 1481-1496.
[74] Kulldorff, M., and U. Hjalmars. The Knox method and other tests for space-time interaction. *Biometrics*, Vol. 55, No. 2, 1999, pp. 544-552.
[75] Jacquez, G. M. A k nearest neighbour test for space–time interaction. *Statistics in medicine*, Vol. 15, No. 18, 1996, pp. 1935-1949.
[76] Wang, M., A. Wang, and A. Li. Mining spatial-temporal clusters from geo-databases.In *International Conference on Advanced Data Mining and Applications*, Springer, 2006. pp. 263-270.
[77] Zhang, Y., and L. D. Han. Dijkstra-DBSCAN: Fast, Accurate and Routable Density Based Clustering of Traffic Incidents on Large Road Network.In, 2018.
[78] Ester, M., H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise.In *Kdd, No. 96*, 1996. pp. 226-231.

[79] Chen, C., A. Skabardonis, and P. Varaiya. Systematic identification of freeway bottlenecks. *Transportation Research Record*, Vol. 1867, No. 1, 2004, pp. 46-52.
[80] Bertini, R. L., R. Fernandez, J. Wieczorek, and H. Li. Using archived ITS data to automatically identify freeway bottlenecks in Portland, Oregon.In *Proceedings 15th World Congress on ITS, No. 5*, 2008.

APPENDIX

Derivation of Equation (15):

$$u_t^{new} = \frac{u_t^{old} * n + \sum_{i=n}^{m} \Delta o_{t+i}}{n+m}$$
$$= \frac{\sum_{i=1}^{n} \Delta o_{t+i} + \sum_{i=n}^{n+m} \Delta o_{t+i}}{n+m}$$
$$= \frac{u_t^{old} * n + \sum_{i=1}^{m} \Delta o_{t+i}}{n+m}$$

Derivation of Equation (16):

$$(\sigma_t^{new})^2 = \frac{\sum_{i=0}^{n+m} (\Delta o_t^i - \overline{\Delta o})^2}{n+m}$$
$$= \frac{1}{n+m}\left(\sum_{i=0}^{n+m} \left(\Delta o_t^i\right)^2 - 2\overline{\Delta v} * \left(\sum_{i=0}^{n+m} \Delta o_t^i\right)\right) + (\overline{\Delta o})^2$$
$$= \frac{1}{n+m}\left(\sum_{i=0}^{n+m} \left(\Delta o_t^i\right)^2 - \overline{\Delta v} * \left(\sum_{i=0}^{n+m} \Delta o_t^i\right)\right)$$
$$= \frac{1}{n+m}\left(\sum_{i=0}^{n} \left(\Delta o_t^i\right)^2 + \sum_{i=n}^{n+m} \left(\Delta o_t^i\right)^2 - \frac{1}{n+m}\left(\sum_{i=0}^{n} \Delta o_t^i + \sum_{i=n}^{m} \Delta o_t^i\right)^2\right)$$

$$= \frac{(\sigma_t^{old})^2 + \sum_{i=n}^{n+m} \left(\Delta o_t^i\right)^2 - \frac{1}{n+m}\left((n+m)u_t^{old} + \sum_{i=n}^{n+m} \Delta o_t^i\right)^2}{n+m}$$

VITA

Yuandong Liu was born and raised in Lianyungang, China. In 2012, she graduated from Harbin Institute of Technology with a Bachelor's degree in Traffic Engineering with an honor. In 2015, Yuandong graduated from Dalian University of Technology with her Master's degree in Transportation Planning and Management. In 2020, She was granted a doctoral degree in Civil Engineering with concentration in Transportation Engineering and her Master's degree in Statistics at UT.

As a Ph.D. student, Yuandong worked on multiple research projects in traffic operations, including travel time reliability analysis, traffic queue detection, prediction and protection, evaluation of temporal-spatial changes of national performance measurements, Waze data assessment, Waze user behavior analysis and so on. Her research interests include traffic operations, advanced technologies in transportation, spatial-temporal traffic movement analysis, crowdsourced data and user behavior analysis.

During her graduate studies, Yuandong received several scholarships and awards, including TSITE student paper competition award, ITSTN scholarship, Traffic tournament finalist, Graduate Student Senate travel award and so on.