Identifying Unsafe Driving in Connected Vehicle Environments and its Implications for Infrastructure Providers.

By:
Robert Moses Kluger

University of Virginia
Department of Civil and Environmental Engineering

Advisor:
Brian L. Smith

Presented to the
Dissertation Committee:

Michael D. Fontaine
Matthew Gerber
Byungkyu (Brian) Park
Gene McHale
John S. Miller

Date: October 4th 2016

TABLE OF CONTENTS

ABSTRACT

The goal of this research is to establish how data made available in a connected vehicle (CV) environments can benefit infrastructure providers in performing safety analysis. Specifically, the interest is to identify safety hot spots, or locations that are experiencing unexpectedly high numbers of crashes, traditionally found through methods using police-reported crash data. This process is used to evaluate transportation network safety and plan for safety-related improvements. It was expected that CV technology would provide two major improvements over the current methods, outlined by the Highway Safety Manual (HSM):

- CV has the potential to detect near-crashes (or conflicts);
- As a result, CV also has the potential to identify hot spots more proactively than current crash prediction models that rely on crash reporting, due to the availability of a much larger set of samples

The first step was to evaluate the feasibility of using existing CV datasets to identify hot spots. It was found that for this to process be successful in a connected vehicle environment, the standards must provide vehicles with a mechanism to alert infrastructure of an event that occurs and that vehicles themselves need to be able to recognize crash and near-crash situations using their on-board equipment. The focus then shifted to identifying safety-critical events, defined as crashes and near-crashes in this context, using data native to the CV standard Basic Safety Message (BSM). Three algorithms, trained using naturalistic driving study data, were proposed in three separate papers. The first was a pattern matching approach that calculated Euclidean distance between observed vehicle acceleration time series and those of some known, pre-defined actions. The algorithm saw success on a limited data set. Similarly using the same dataset, a speed prediction model was used to identify discrepancies between expected speeds and observed speeds, flagging groups of observations that were too far off from the expected speed. The third and final algorithm was trained on a much larger dataset, utilizing a discrete fourier transform and a k-means clustering algorithm to group events into clusters. This was successful on a robust dataset.

This compendium of work first, provides a comprehensive discussion of the findings related to how connected vehicle technology can benefit highway safety analyses. These findings provide a vision and a foundation for future methods and ideas as CV technology and implementation matures. Second it explores how crashes and near-crashes can be detected in connected vehicle environments. All of the hypothesized benefits of using connected vehicles for hot spot identification hinge on the ability to successfully detect crash and crash-surrogate events. As a result, a major focus of this research was modeling crashes and near-crashes in order to describe them in terms of connected vehicle data elements. Three creative methods were proposed as possible approaches to identifying these types of events, including a pattern matching approach, a speed prediction time series based approach, and a discrete fourier transform approach. Each of them have benefits and drawbacks in terms of both complexity and accuracy, but serve as excellent starting points for further research and the lessons learned are applicable and should be considered as additional models are proposed.

INTRODUCTION

RESEARCH MOTIVATION AND PROBLEM DESCRIPTION

The goal of this research is to establish how connected vehicle (CV) environments can benefit infrastructure providers in performing safety analysis. Specifically, the interest is to identify safety hot spots, traditionally found through police-reported crash data and used to evaluate transportation network safety. It was expected that CV technology would provide two major improvements over the current methods, outlined by the Highway Safety Manual (HSM):

- CV has the ability to detect surrogate safety measures such as near-crashes;
- As a result, CV also has the ability to identify hot spots more proactively than current crash prediction models, such as those outlined in the HSM.

The first step was to evaluate the feasibility of using existing CV datasets to accomplish this goal. It was found that some preliminary research needs to be done to effectively answer the original research question though findings indicate it is likely possible to accomplish this goal. The focus then shifted to identifying safety-critical events, defined as crashes and near-crashes in this context, using data native to the Basic Safety Message (BSM). Current event flags native to BSMs and similar applications use simple acceleration thresholds to flag potential events. Like all decision boundaries, these thresholds come with a trade-off between recall, the percentage of true events that get flagged, and precision, the percentage of flagged events that are true. Namely, lower thresholds have high recall and low precision and vice versa for high thresholds. As a result, a simple threshold was deemed unusable in a CV application and more robust models were required that considered how the vehicle dynamics changed over short periods of time. Successive papers were written exploring this issue in-depth.

DETECTION OF SURROGATE SAFETY MEASURES

Typically, safety is quantified using crashes as the primary metric. This is because data is readily available and there is no debate that these are events that are dangerous and should be reduced. However, a challenge with crashes is that numerous factors frequently go into these events and they often have an element of randomness or (bad) luck. This makes studying causal factors and trends difficult, both in a mathematical modeling sense, and in the sense that it takes time to build confidence in the numbers that are observed. It is also highly reactive since people need to get into crashes before they can be studied.

Crash surrogates are measurable events that can be collected and are theoretically, proportional to the number of crashes at a site. Commonly proposed crash surrogates include near-crashes and traffic conflicts. Historically, the trouble with these is that actually collecting data on these events as they occur is exceedingly difficult or expensive. But, connected vehicle technology perhaps offers a means to reliably collect data on surrogate events in the form of data projected in the BSM. The objectives of this research were first, to raise awareness that this is a research need, and second to begin modeling crashes and near-crashes that have been observed in other studies in terms of data elements native to the BSM. The modeling process was carried out using data sets from the Naturalistic Driving Study.

USING CONNECTED VEHICLES TO IDENTIFY HOT SPOTS

Hot Spot Identification is a key responsibility of infrastructure providers (like state DOTs). The process involves collecting crash data and building predictive models that estimate expected crash counts based on a mixture of roadway features and the number of crashes observed historically over a recent time period. The expected crash count is treated as a measurement of crash risk and sites with exceedingly high crash risk relative to an expected value are flagged as hot spots. This is a process that traditionally uses crash data as an input and requires three to five years' of data to have statistical confidence in the trends that are being observed.

Since connected vehicle technology is likely to be able to pick up crash surrogates on a network scale, there is reason to believe that data can be utilized to carry out the hot spot identification process once the technology is widely deployed. This should provide infrastructure providers with a more complete picture of what is occurring at specific locations. For example, if a site experiences exceedingly high numbers of near-crashes over a six-month period but only a modest number of crashes, that may still indicate that the site has a high risk for crashes in the long term. The expectation is that connected vehicle systems will allow for infrastructure providers to more proactively identify hot spots, rather than waiting for people to get into a critical number of crashes before realizing there is an issue.

OVERVIEW OF CONTENTS

Before proceeding to the research papers included in this compendium, it is necessary to discuss the importance of this research and the approaches that were taken. The papers are centered specifically on the approach and result of the individual method chosen but doesn't include some of the reasons those approaches were necessary. This will provide an overall discussion of the problem as well as how each research paper falls into the overall framework of the compendium.

The first paper in this compendium, titled "A Case for using Connected Vehicle Data to Support Improved Infrastructure Safety Hot Spot Identification", discusses why there is a need for improving the hot spot identification process and how this emerging technology can address the present issues with existing methods. It also highlights additional research needs before a method using connected vehicles can be fully implemented. This paper is under review for presentation and publication at Transportation Research Board for the annual meeting in 2017.

One of the conclusions from the first paper was that methods to identify safety-critical events in connected vehicle environments were critical to successfully implementing a hot spot identification process and many other CV safety applications. The following three papers explored approaches to modeling crash and near-crash events by using longitudinal acceleration, a data element native to the BSM.

The second paper, "Pattern Matching Longitudinal Acceleration in Time Series Data to Identify Crashes in Naturalistic Driving Data," employs a time series filter that identifies portions of the longitudinal acceleration time series that do not fit into a set of expected actions. This approach is novel and was successful on a limited data sample but perhaps the most impactful finding from this was the approach: to classify known or expected actions or behavior and search for sections that did not align with expected actions to flag potential crashes. A variety of potential further direction will be discussed in the paper along with a complete description of the approach.

The third paper, "Identification of Safety-Critical Events in a Connected Vehicle Environment," uses a physical model to predict a vehicle's speed over a short time period using an observed speed and acceleration. If the predicted speed deviated too much from the observed speed, the location was flagged for further analysis. Then, using characteristics of the deviations, including magnitude of deviation and length of deviation, a logistic regression model predicted the probability that a deviation was a crash. Since additional data arrived after this document was submitted for publication, there is a supplemental attachment, briefly describing this algorithm's performance on a more robust data set is provided immediately after the document.

The final paper, "Identification of Safety-Critical Events in Connected Vehicle Environments using a Discrete Fourier Transform (DFT)," takes a similar approach as the pattern recognition methodology with a few key improvements. The first improvement is that a Discrete Fourier Transform is executed on the subsequences of longitudinal accelerations, which is a common approach in pattern recognition. It also applies an unsupervised K-means clustering algorithm to flag events. This resulted in fewer engineering judgment decisions and required less background work into establishing normally expected patterns as the algorithms were able to decide what was normal and expected.

Finally, a discussion of some lessons learned throughout the extent of the research will be provided along with a discussion of the contributions to general knowledge made by the research. Suggestions for further direction in both the event detection work, and the hot spot identification work are described in detail. Both topics can be researched extensively and there is potential that methods could replace current safety analyses in the future once connected vehicle technology becomes more widely deployed and additional findings are made. Finally, some concluding remarks will be made.

CLASSIFICATION OF CRASH EVENTS FROM KINEMATIC VEHICLE DATA

In connected vehicle environments kinematic vehicle data will be collected in high volumes at high frequencies. To maximize the usefulness of this data, it is critical to be able to understand what is happening as data is observed. In other words, translating what is being seen in the data into something that practitioners can use to understand what is actually happening is one of the things that could make this data so valuable. One of the more interesting problems is having the ability to identify unsafe actions including crashes and near-crashes. In the case of crashes, these events are rare and frequently underreported while for near-crashes the events are not even recorded.

Both crashes and near-crashes can likely be used in varying capacities to evaluate safety at sites and connected vehicles offer an opportunity to address some of the issues regarding the lack of data. However, for the technology to be successfully used to evaluate safety, it is important to know when a crash actually occurred through the technology. Three algorithms were developed, utilizing kinematic data native to the Basic Safety Message, to identify crashes and near-crashes. This chapter will outline the approach and the lessons learned when developing those algorithms. Included will be a discussion of characteristics of the data and the impacts they had on algorithm design, data processing techniques that were used to counteract issues that were faced, and the classification techniques tested. The algorithms themselves and how this work falls into the larger overall framework is discussed in the following chapters.

DATA CHARACTERISTICS

The purpose of this section is to discuss the characteristics of the crash data received from the Naturalistic Driving Study. NDS data was in two groups. Both times data was acquired, an "event" set was acquired and a test set was acquired. The event set was where the crashes and near-crashes were, while the test set was simply driving that had no events occur. This was used for false positive testing to determine how frequently the model or algorithm was triggered.

The first set of data was received in complete trips. That is, the thirteen crash events received, were trips that culminated in a crash after some period of time, ranging from 2 minutes to 45 minutes. This set was used to train the data, with the baseline negative values being the time the subject vehicle spent driving before it entered the crash situation. Similarly, the normal driving received consisted of trips ranging from 35 to 75 minutes during which no events occurred. This was used for false positive testing after the algorithm was trained. Crashes were not held back in the test set due to a lack of data. There were only 13 crash events and many of them were different in nature.

The second set of data was slightly more robust featuring 92 events, this time consisting of both crashes and near-crashes. It was also requested that a variety of crash types and speeds were used to best capture as many situations that may occur as possible. These events were delivered in 30-second epochs as were 50 30-second baseline epochs. These were used as a training set. 50 additional hours' worth of normal driving was held back for the false positive testing.

The data was delivered in a .csv format with a file for each trip. Each entry was an observation of kinematic data elements from the DAS device. The frequency of observations was 10 Hz (ten times per second) though there were some fields that were collected at slower frequencies. For example, one of the speed fields was acquired from the vehicle network which may have only be providing data to the DAS at a frequency of 5 Hz.

DEFINITION OF SUCCESS IN CLASSIFICATION OF EVENTS

For typical classification models, metrics like precision and recall are commonly presented to show quality of classification. For this application, there are some issues with those metrics though. This requires a further look at the four possible ways to classify an observation or window – those would be, true positive, true negative, false positive (Type I Error), and false negative (Type II Error).

A positive event in this case is a crash or near-crash. A true positive is a model-classification of a crash or near-crash when a crash occurs. A false negative is a model classification of no-crash when indeed there was a crash occurring during the observation. The issue here is that a crash isn't always going to occur in a single observation or window. A crash could easily span two windows but the model may only classify a single one of them correctly as a crash. While normally the window that it didn't flag would be considered a false negative, the model was successful at finding the event and therefore should not be penalized for not flagging the entire event. This issue is illustrated in Figure 1 where the blue arrows represent a window the longitudinal acceleration time-series could be broken into and the red portion represents the crash event.

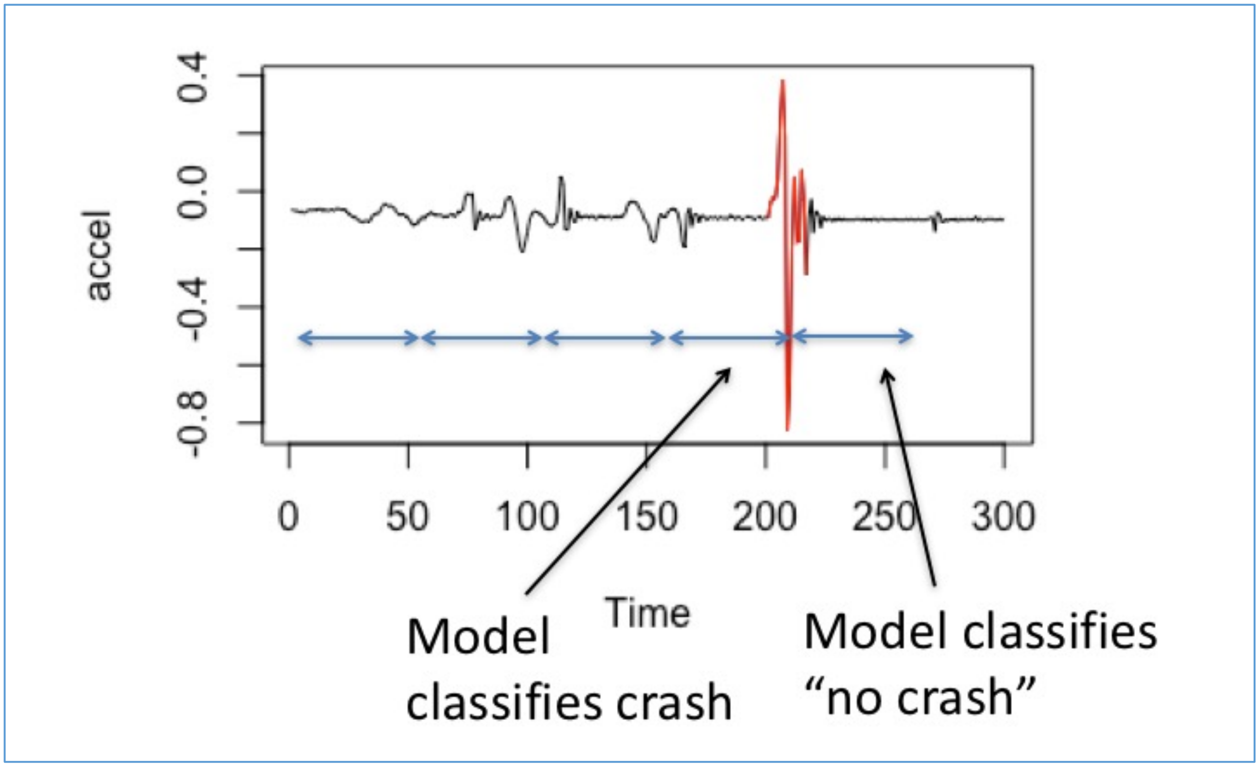

Figure 1. Crash Classification

Changing the definition of a crash event to the point of impact, reverses the problem where perhaps a crash is classified into two windows but the model sees one as a false positive. At the very least this disparity needs to be kept in mind when evaluating a method's performance, but in reality, this can be solved through some post-processing and heuristic indicators. For example, instead of defining classification success in terms of total windows, just define it in terms of crashes. So if there were 13 crash events and 12 were correctly identified in some capacity, – i.e. at least one of the windows spanning the crash were flagged by the model, then the recall is 12/13. Measures taken for each model will be described in detail where appropriate.

The negative values had their own issues. Negatives could either be correctly classified as true negatives or incorrectly classified as false positives. First of all, it is unclear what the number of true negatives even is for the continuous data, which is included in the denominator of metrics like specificity. So for a 20-minute trip with no events, how many true negatives are in the time series? Is it the 1200 observations that occur?... The 120 ten-second windows?... Or something else? This decision impacts the presented error rates and doesn't really make sense for the continuous data anyways. It was decided that false positives should be classified per unit of time (i.e. 1 false positive per 10 hours of driving).

Additionally, false positives needed to be grouped if they were related to one another. Two windows in a row getting flagged should be treated as a single false positive since they are likely related and could just be an artifact of how the window was divided. Again, some sort of heuristic or correlation analysis needs to be applied in order to determine if two flagged windows are related to one another.

It was found that limiting the rate at which false positives is extremely important. This is a simple matter of exposure. Crashes and near-crashes are highly infrequent relative to the amount of time people spend driving normally. For example, if, hypothetically, an algorithm was to have 1 false positive for every 10 hours of driving and crashes and near-crashes were to occur every 100 hours of driving, there would still be 10 false positives for every true positive the algorithm flags.

CLASSIFICATION

Many current methods used to detect crashes and near-crashes from kinematic data use a single observation to flag potential events. Current NDS flags as well as the BSM part II vehicle safety extension use an acceleration threshold as one of the flags for the purposes of flagging potential events. The trade-off between sensitivity and severity for a single variable threshold, like longitudinal acceleration is easy to see – as the threshold goes up, the recall of events goes down while the severity of the events found go up. For a threshold to be sufficiently high to limit the number of false positives, the added benefit of using this method over crash data for network screening is diminished. Alternatively, if the threshold is sufficiently low, false positives would dominate the flagged events since the amount of time drivers are exposed to normal driving is substantially larger than the amount of time drivers spend engaging in crashes, near-crashes, and other unsafe behaviors.

Additionally, an issue that may be solvable but nonetheless is present is the fact that it can be difficult to determine if multiple flags are part of the same event overarching event. So, if 3 observations are classified as an event over a 1 second period they are probably all from the same occurrence and should be grouped into a single event flag. Rules need to be developed to determine the acceptable length of time for multiple flagged observations to be grouped and classified as a single event.

Evaluating observations individually ignores the fact that the data in Basic Safety Messages are time-series in nature. In other words, the context in which an acceleration is observed should be taken into account. A jump in acceleration from -0.3 g to 0.3 g is not the same as a jump from 0.2 g to 0.3 g over the same period of time. This suggested that pattern recognition methodologies or time-series regression models, such as an ARIMA model, may be more appropriate.

Pattern recognition approaches, demonstrated in the proposed pattern matching algorithm and the Discrete Fourier Transform algorithm follow similar processes. First, each time series is broken into windows of pre-defined lengths. For the pattern matching approach, a filtering technique was applied to remove any window that appeared normal, which was defined using accelerations during known actions. For the DFT method, a clustering technique was used to sort between normal driving and the windows with events between them.

Other models were also tested after breaking the time series into windows. Features were extracted, such as maximum/minimum acceleration in the window, and predictive modeling approaches were taken including a model using Classification and Regression Trees (CART) and logistic regression. Neither model performed well enough to use such a simple technique which indicated that the additional processing conducted was warranted. These modeling approaches are clearly better from an implementation perspective so any improvements to these would constitute a substantial benefit to the field.

The speed Prediction followed an approach more similar to forecasting looking at the time-series while considering autoregressive factors. Specifically, an ARIMA model was tested and for the resulting coefficient on the acceleration at time *t-1* was close enough to 0.1 s (the time between observations) that a physical model using 0.1 s was used instead due to transparency and general benefits to conceptual understanding.

The following paper was submitted to Transportation Research Board's 2017 Annual Meeting for presentation and publication. Review is pending.

A CASE FOR USING CONNECTED VEHICLE DATA TO SUPPORT IMPROVED INFRASTRUCTURE SAFETY HOT SPOT IDENTIFICATION

Robert Kluger
University of Virginia
351 McCormick Road
Charlottesville, VA, 22903
rmk7su@virginia.edu

Brian Smith
University of Virginia
351 McCormick Road
Charlottesville, VA, 22903
bls2z@virginia.edu

Hyungjun Park
University of Virginia
351 McCormick Road
Charlottesville, VA, 22903
hpark@email.virginia.edu

Word Count: 6012 + 2 Tables +2 Figures =7012 Words
Submission Date: July 26, 2016

A CASE FOR USING CONNECTED VEHICLE DATA TO SUPPORT IMPROVED INFRASTRUCTURE SAFETY HOT SPOT IDENTIFICATION

ABSTRACT

Transportation infrastructure providers are responsible for supporting the safety of road users. Part of this responsibility is a process called network screening, in which potentially unsafe locations are flagged for additional analysis. Current network screening methodologies are very reactive and rely on crash databases that are incomplete and contain invalid entries. Connected vehicle technology is an emerging technology that is likely to change the way the network screening process is carried out. With vehicles broadcasting second-by-second kinematic data to roadside equipment, infrastructure providers can identify crashes quickly and more accurately, while also providing the ability to identify less severe events such as near-crashes and evasive maneuvers. Discussed is a high-level overview of how connected vehicle technology will impact and benefit the network screening process, as well as additional research needs arising from this topic. While this technology will be beneficial to safety screening processes, it will introduce new challenges and is likely completely change the way the way the process is carried out. With the vast amount of data being generated on a second-by-second basis, mining the data for crashes and near-crashes becomes a relevant topic to explore while there are evident trade-offs between how to collect the data and whether or not to store the data.

INTRODUCTION

Infrastructure providers, such as state and local Departments of Transportation (DOTs), are responsible for ensuring the safety of their constituents who use their facilities. A key component of this responsibility is network screening, a process outlined by the Highway Safety Manual *(1)* that identifies hot spots, or sites that are experiencing an unusually high number of crashes, often pointing to an infrastructure deficiency at the site.

Connected Vehicle (CV) technology has the ability to significantly improve the network screening process to evaluate the safety of road networks. In a CV environment, vehicles will be broadcasting enormous amounts of data to other vehicles as well as to roadside infrastructure controlled by the Department of Transportation. This data describing the experience of each individual vehicle could give infrastructure providers the ability to identify hot spots much faster than is possible using current network screening methods.

The purpose of this paper is to first, outline drawbacks of current methods, second, describe how a CV environment will provide a richer set of data that will be able to address some of these issues, and third, discuss some of the barriers and new issues that arise when implementing and managing a system like this in a CV environment. Finally data from the Safety Pilot Model Deployment (SPMD) CV testbed in Ann Arbor, MI *(2)* is used to provide an example case of network screening using vehicular data under a CV environment.

NETWORK SCREENING

Network screening is a procedure outlined by the Highway Safety Manual (HSM) as part of the Roadway Safety Management Process (RSMP), to identify potential hot spots. The purpose is simply to identify sites that are experiencing more crashes than expected for further analysis. Later parts of the RSMP are meant to diagnose the issues, and develop alternatives that are likely to reduce the number of crashes a site experiences *(1)*.

Data Input

There are two sets of data used to conduct network screening: crash data and roadway inventory data. Crash data is a database of crashes populated by police crash reports. The data schema will vary by agency, but will generally include crash location, type, and severity. Roadway inventory data is a database of roadway features, including elements such as lane width and Annual Average Daily Traffic (AADT).

The chain of custody of crash data varies greatly between states and jurisdictions. When the police are called to respond to a crash, they are responsible for filing a crash report either on paper or digitally through a laptop or tablet. These reports will consist of data describing crash time, location, type, and severity as well as a verbal account from the parties involved, when available. Crash location will sometimes be provided in GPS coordinates and sometimes be provided using reference points (e.g. "Route 29 Southbound: Milepost 43"). For crash type and severity fields, the officer is responsible for selecting an option from a predefined set of choices listed in the report. There is also usually a box where the officer can write notes and additional descriptions that may be necessary. If the crash reports are digital, they are often uploaded directly to a database system, but if the reports are paper, they must be manually entered into a database.

Network Screening In Practice

Methods

The HSM outlines a variety of methods for network screening, but considers the most robust method to be the Excess Expected Average Crash Frequency method with the Empirical Bayes (EB) Adjustment, referred to from here on as the EB Method, because it accounts for regression-to-the-mean (RTM) and identifies a threshold indicating sites experiencing more crashes than expected.

The listed data needs from the HSM for this method are:

- Historical Crash Data by Location by year
- Traffic Volume by year
- Basic Site Characteristics
- Safety Performance Functions (SPFs)

The first three items listed are included in the Crash and Roadway Inventory Databases. The SPFs are predictive models used to predict the number of crashes at a site based on specific characteristics. Traditionally these are calculated using negative binomial regression, however a variety of additional methods to predict crash frequency at sites have been tested including artificial neural networks, support vector machines, and Bayesian networks, among others *(3–7)*. An example of an SPF is shown in equation 1. The SPF is applicable for estimating all multi-vehicle collisions at a 3-leg, urban, stop-controlled intersection with a major AADT of less than 45,700 vehicles, and minor AADT of 9,300 vehicles. Each NB regression model has an overdispersion parameter, k.

$$N_{Pred} = e^{-13.36+1.11\times AADT_{Major}+0.14\times AADT_{Minor}} \quad (1)$$

SPFs are very specific to specific regions and are rarely transferable to other regions. Many states make their own SPFs *(8–10)* while others have the option to develop a calibration factor to apply to the existing HSM SPFs.

Using the SPF, and traffic volume data, the predicted crash frequency, N_{Pred} at the site is calculated. These predictions can be modified using Crash Modification Factors (CMFs) to account for additional roadway features not captured by the SPF's prediction (e.g. red light cameras at an intersection) Next, a correction factor, C_n, is calculated for each year of crash data being used, which is the ratio of crashes predicted by the SPF in year 1 to the prediction in year n.

The EB weight factor, w, is then calculated in equation 2.

$$w = \frac{1}{k \sum_{n=1}^{N} N_{Pred\_n}} \tag{2}$$

The excess expected crash frequency at the site is then calculated by taking the difference between the expected and the observed number of crashes at a specific site. Sites are then prioritized for further evaluation by using their expected excess crash frequencies *(1, 11)*.

Disadvantages

In its current state, there are variety of drawbacks to the network screening process. Many of them stem from the data source, but some are also inherent to the methodologies used. This section will describe these issues in more detail.

The chain of custody of crash data is the source of a substantial number of errors in both the quality and completeness of the crash database. For a crash to enter the database it must first be reported to the police. Drivers may have numerous reasons for not reporting crashes. If they were doing something illegal (driving on a suspended license, impaired driving, etc.) they may choose to leave the scene out of fear of consequences. If it is a single-vehicle collision or a minor multi-vehicle collision, drivers will often handle the incident through insurance (or not at all) rather than involving the police. Additionally, in the event that the police are called, entry to the database may be withheld due to a reporting threshold. The threshold varies by jurisdiction, but the responding officer is responsible for estimating the dollar value of damage and deciding whether or not it exceeds the state or locality's threshold. All of these factors combine to make crash databases biased toward more severe events. While there may be an argument to be made for a heightened interest in severe crashes, it is likely that some of the crashes that are not getting reported to the database may have similar causal factors to the severe crashes.

For the crashes that do get reported, there are numerous points in the reporting process that can lead to erroneous values in the data. Most of the information the officer collects in the report is based on accounts of the events from those who were involved, which can easily be distorted. Whether the account is purposely altered or inadvertently altered because of lapses in memory or differences in perspective, the officer won't be able to tell what happened unless he or she was a witness.

Other errors on the form can usually be attributed "human error", but these errors can be significant. Locations can be erroneous if the officer were to use unclear or incorrect reference markers. Even for digital forms where the GPS location can be pulled directly from the device, the location can be incorrect. For example, if parts of the report were not filled out at the scene of the crash, the GPS coordinates may not be accurate. If an officer is unclear on what a field means but fills it out anyways, all of his or her crash reports can be reasonably subject to doubt. For paper reports, mistakes during manual entry into the database could be a significant source of inaccuracy.

It can be difficult, if-not impossible, for a user of the data to tell if the report is valid or not without dedicating resources and effort to case-by-case investigations. Statistical language processing has been used to search for inconsistencies in crash reports between written account and the values recorded in the boxes *(12)* . Additionally, comparisons of the crash reports to roadway GIS databases *(13, 14)* and hospital records *(15, 16)* have been used to validate crash databases. While these methods may help find some inconsistencies, they do not necessarily help find hot spots, because they just remove data from a database that is already relatively sparse, rather than being able to fix and use the entries.

Aside from data inaccuracies, there are also a few deficiencies inherent to the methods themselves. The most significant drawback is that they are reactive, requiring enough time and crashes to occur before being able to confidently locate a hot spot. Most of this is due to the RTM effect observed in yearly crash counts with few observations relative to the exposure (traffic volume). Regardless, they require people to drive in unsafe conditions for 3-5 years before the problem can even start to be addressed. While it may be difficult to get a truly preemptive method to identify these locations, it would clearly be beneficial to reduce this time if possible.

Another issue inherent to the network screening process that it is calculating expected crashes and using the excess of the expected crash frequency to identify sites. However, expected crashes are the result of features present on the road. So if a feature of a site is increasing the number of expected crashes, that doesn't necessarily mean it will be flagged. Locations experiencing more crashes that cannot be explained by the model using existing features of the road are the ones being prioritized, but it does not identify locations with high expected crashes that are not exceeding the expectation in observed crashes.

Crash Surrogates

Safety evaluation using crash surrogates is also considered to be a promising research area. Crash surrogates are measurements that are correlated to the number of observed crashes and can therefore be used in place of crash events when performing safety assessments. Since a crash is considered the most severe event type, it is also the rarest event type, however many more events, such as near-crashes, occur that could result in a crash but do not result in a crash for a variety of reasons, such as the driver performing an evasive maneuver. A perfect crash surrogate is directly proportional to the number of crashes observed at a site and occurs more frequently than a crash *(17)*. With additional observations, yearly counts are less impacted by RTM.

Near-crashes and traffic conflicts are two commonly used surrogates for crashes in the safety community. Near-crashes are events with similar causal factors to crashes, meaning that they occur due to similar precipitating events but do not necessarily result in a crash due to a driver's ability to react in time or some element of luck *(18)*. Meanwhile, traffic conflicts are defined as "an observable situation in which two or more road users approach each other in space and time to such an extent that there is risk of collision if their movements remain unchanged" *(19)*.

Both of near-crashes *(17, 18)* and traffic conflicts *(20, 21)* have been shown to be useful for safety applications including network screening but collecting data on these on a large scale remains a challenge. In addition to simple human observation, computer vision techniques *(22)* and Naturalistic Driving Studies *(23, 24)* have been used to identify traffic conflicts and near-crashes but the bottom line is that data on surrogate measures cannot currently be collected on a network-wide scale. This is a major barrier to using crash surrogates to identify hot spots. The authors believe that connected vehicles provide DOTs with the best opportunity to access surrogate safety measures on a large scale.

CONNECTED VEHICLES

Overview of Technology and Deployments

Connected vehicles (CVs) are an emerging technology centered around providing vehicles with the ability to communicate with other vehicles as well as roadside infrastructure. Each vehicle will be equipped with on-board equipment (OBEs) which collects data from the vehicle and broadcasts that data to other vehicles and roadside equipment (RSEs). The OBE is also responsible for receiving and interpreting messages for CV applications.

The primary data broadcasted is called the Basic Safety Message (BSM). This message consists of two parts and is broadcast by vehicles 10 times per second according to the current standard. Table 1 shows the data elements in Part 1 of the Basic Safety Message.

Table 1 - Basic Safety Message Part 1

Message ID	Speed
Heading	Acceleration
Latitude	Longitude
Yaw Rate	Steering Wheel Angle

The BSM Part II, sometimes called the Vehicle Safety Extension, is not included in every BSM. Part II consists of a set of event flags, and when one of these event flags is triggered, the vehicle immediately broadcasts a BSM with the Part II extension. The event flags listed in the standard are presented in Table 2.

Table 2 - Basic Safety Message Part 2

Active hazard lights	Stop line violation
Anti-lock brake activation	Air bag deployment

Vehicle is an active emergency response vehicle	Vehicle is known to be carrying hazardous material
Stability control activation	Traction control activation
Vehicle lights change	Wipers change
Flat tire	Vehicle becomes disabled
Vehicle deceleration exceeds *0.4g*	

There are also other non-event related reasons for sending a BSM Part II, however these will not be elaborated on as they are less relevant to the presented work *(25)*.

There have been a variety of global efforts to research and implement this technology *(26)*. In the United States, there are, have been, or will be connected vehicle test deployments in numerous locations focusing on different applications but maintaining interoperability among the environments *(27, 28)*.

The first large-scale deployment of this technology in the US was called the Safety Pilot Model Deployment (SPMD) and took place in Ann Arbor, Michigan. The 1-year pilot study consisted of more than 2000 subject vehicles equipped with some level of CV capabilities *(2)*. There are other testbeds across the United States focusing specifically on different applications for connected vehicles and this shows a commitment by the USDOT to move forward with the technology.

CV For Network Screening

Concept and Purpose

CV technology provides a unique opportunity to reevaluate the historically used methods to evaluate road safety. The primary benefit will be its ability to collect large-scale network-wide, second-by-second data on crash surrogates.

It has also been shown that data elements native to the BSM can be leveraged to identify crashes and near-crashes *(24, 29)*. This means that using BSMs collected by RSEs in connected environments, an infrastructure provider could reasonably be able to detect crash surrogates, like near-crashes. The purpose of an identification algorithm should not be to replicate the crash databases that states one, already have access to, and two, can be extremely flawed as discussed earlier. The purpose is to identify all events that occur and could be considered safety-critical. This includes crashes, near-crashes, evasive maneuvers, and traffic conflicts.

CV technology can be used for identifying both short-term (e.g. debris in the road, overgrown tree blocking sight distance, etc.) and long-term issues (e.g. mistimed signal, poorly aligned roadway, etc.) with infrastructure by searching for anomalies in the BSMs received at specific locations, depending on how those anomalies are distributed across time and location.

The recent push to study connected vehicle environments in a variety of capacities by the USDOT and some states shows a commitment to implementing this technology in the near-future. CV technology has such a wide-range of applications that the financial burden of implementing this technology would not fall directly on any one division of the infrastructure provider.

How CV Technology Addresses The Disadvantages Of Current Network Screening Techniques

CV Technology addresses many of the disadvantages of current network screening methods previously discussed, while not adding additional disadvantages other than perhaps a required expertise in handling big data to conduct analysis. For reference, the key disadvantages of current techniques that were discussed were:

- Reactive method the requires a long period of data collection before hot spots can be identified. This is due to the tendency of observed crash frequencies to regress-to-the-mean.
- Quality of input data in the method is subject to human error in multiple points throughout the collection process. It is also difficult to validate.
- Methods are biased toward more severe events, such as crashes where damage exceeds a reporting threshold or crashes with injuries/fatalities. These are unlikely to be the only indicators of unsafe conditions and tend to be the most infrequent events observed at sites.

Identification of surrogate safety events through machine learning models and other creative algorithms will shift the errors in data collected from human error to model error and sensor error. While there is still error present, as the modeling techniques and sensors equipped to the vehicles are both improved the error rate will decrease. The error rates and model tendencies can also be more easily quantified and accounted for than currently possible with police-reported crash data. Additionally, improved models can be retroactively applied to older data to correct findings from previous studies. Conversely, crash data in its current capacity is unlikely to be adjusted, and even less likely to be accurately adjusted, once a short period of time has elapsed after the crash.

Using CV technology also will make network screening a less reactive process. Current methods using police-reported crash data require a certain number of severe accidents to occur before a hot spot can even be identified. One key argument for the use of surrogates is that more events occur, which means the frequency of those observations, whether it be near-crashes, traffic conflicts, or another surrogate will vary less from year-to-year. By increasing the sample size of yearly events, the variations that do occur in observed frequency will have less impact on the findings and allow for more confidence in the short-term results. CV technology provides a resource to collect crash surrogates on a large-scale which has never been readily available in the past. While this still requires some events to occur before the problem can be identified and action can be taken, the hope is that the amount of time that process takes is significantly reduced.

Crashes have historically been used because they are all that has been available, but that doesn't necessarily mean that they are the best or only indicators of locations that are most likely to experience an abundance of crashes in the future. Using surrogates collected by CV technology will provide a different perspective on safety than crash data alone. While the primary goal of safety analysis is to reduce the number or injuries and fatalities on the road, there are also economic and operational benefits to preventing all crashes. The goal of this is not to simply replicate the network screening results that could already be achieved using traditional methods. By collecting surrogates, seemingly less severe issues can also be identified.

Additional Research Needs

Since CV technology is relatively new, there are a variety of research needs that must be addressed before it can be applied to safety screening. Some can be addressed now, while others may require more widespread deployment for assessment to become feasible.

First, it is necessary to define crashes, near-crashes, and other surrogates in terms of the Basic Safety Message data elements. Understanding and modeling the patterns present in the events using the observed speeds, accelerations, and other kinematic data is critical to apply this technology for many safety applications, not just screening. Additionally differences between sites, vehicle types and drivers could impact how these algorithms perform and that should be quantified. This research is feasible to do now, using either Naturalistic Driving Studies *(24, 29)*or Crash tests, in fact, doing the research in these settings would likely be preferable. This is because ground truth data is available and can be used to verify findings and model predictions, while CV environments don't have the same luxury.

Second, it is important to determine how to apply algorithms that identify surrogate events in a live CV environment. CV environments will be producing huge amounts of data every second and infrastructure providers looking to carry out machine learning and other applications will need the appropriate computing capabilities and expertise. One alternative could be to have a vehicle's OBE search for events as it collects data from the vehicle network and project flags as part of the BSM Part II safety extension. This would offer two benefits. First each individual vehicle's OBE would serve as a way to carry out parallel computing in real-time rather than relying on the DOT to do that computing post-hoc. Second, it would help avoid some potential issues with the BSM standard designed to keep users' privacy. Specifically, the Message ID field changes frequently so vehicles cannot be tracked across the network, but this could get in the way of models that rely on BSMs collected in a series.

Another alternative is for the DOT to carry out this work on BSMs that get stored in a database. This would require an abundance of computing capabilities and storage space but would also provide benefits of its own. The biggest benefit is that saved BSMs can be analyzed using new models and algorithms as they get developed, rather than being stuck with the events provided by event flags in older versions of the standards. A system-level analysis of both of these alternatives must be carried out, because it is not clear which is the preferred alternative, especially once additional factors such as cost implications to a range of different stakeholders are considered. This research is critical to start before wide-spread implementation because it will be very difficult to change approaches after implementation. CV testbeds have the opportunity to explore this in their system designs and developers of CV standards should be involved as well.

Third, it should be determined how to use the detected events to find hot spots, and prioritize them for further diagnosis and countermeasure development. The safety community needs to first determine if current methods can, or even should, be modified to using events found in CV technology. With this new information, the definition of a hot spot may be redefined to something more applicable or relevant, and the methods to locate them could be completely altered as a result. This research will likely need to wait until widespread deployments and data from connected vehicles is available, and will be ongoing once deployment occurs.

Finally, how CVs can be used in the rest of the RSMP described by the HSM could be investigated. For example, if detection models are able to differentiate between crash types and predict severity, it is likely that the diagnosis phase, where the cause of the hot spot is determined, could also be augmented with these technological improvements.

Sample Application

An illustrative example is performed to explore the feasibility of this and explore how this topic can be approached in early deployments. The purpose is simply to provide some lessons learned when using the data and demonstrate that this data can address some of the issues discussed, particularly the volume of data issue.

Two months (April and October) of BSM data was acquired from the Safety Pilot Model Deployment CV testbed in Ann Arbor, Michigan. Data was collected from vehicles through a variety of different sensor systems. Some vehicles were fully equipped to send and receive BSMs while other simply had aftermarket devices which only broadcast and saved the BSMs. The equipment type impacted how the kinematic data was obtained, as some of it was derived from GPS locations, while others were collected directly from the vehicle network. Roughly 2,000 vehicles, or an estimated 2% of vehicles in Ann Arbor were equipped with some level of connected technology equipped *(2)*. The data was collected and stored in a flat data file and had over half a billion BSM observations.

When searching for clear outliers in kinematic data, such as acceleration during a safety-critical event, the type of sensor collecting the data matters. Specifically, speeds and accelerations derived from GPS position can be high due to GPS errors in addition to extreme events. When CV technology is manufactured in vehicles, this will likely no longer be an issue as this data will be acquired from the vehicle network. For the purposes of this exploration, BSMs observed off the road were removed as well as accelerations that were equal to the artificial maximum enforced by the OBEs. There may still be erroneous BSMs due to smaller sensor errors that happened to flag locations on the roadway. While this introduces a new type of data quality issue, technology improvements and post-processing methods could be developed to account for this.

Another issue with analyzing the SPMD data in a spatial manner was a spatial bias. Much of the recruiting took place in two locations, University of Michigan's medical center, and the University of Michigan Transportation Research Institute. This was done to help with recruiting and to maximize interactions of equipped vehicles for the purpose of testing applications. As a result, most locations with a high-density of events were in those areas. The spatial distribution of the BSMs collected are clearly not representative of the spatial distribution of traffic volumes in Ann Arbor's entire network, which restricts a true, network-wide analysis from being carried out. To counteract this issue, a single route, Washtenaw Avenue, was analyzed.

Washtenaw Avenue, running diagonally in figure 1, was selected due to its relatively high volume of BSMs and because it is a direct route to and from the medical center, one of the primary points of recruitment for the study. The selected segment was six miles long and ran between the medical center in the northwest and ends around Hamilton street in the southeast. There is one interchange around mile 3 with Route 23.

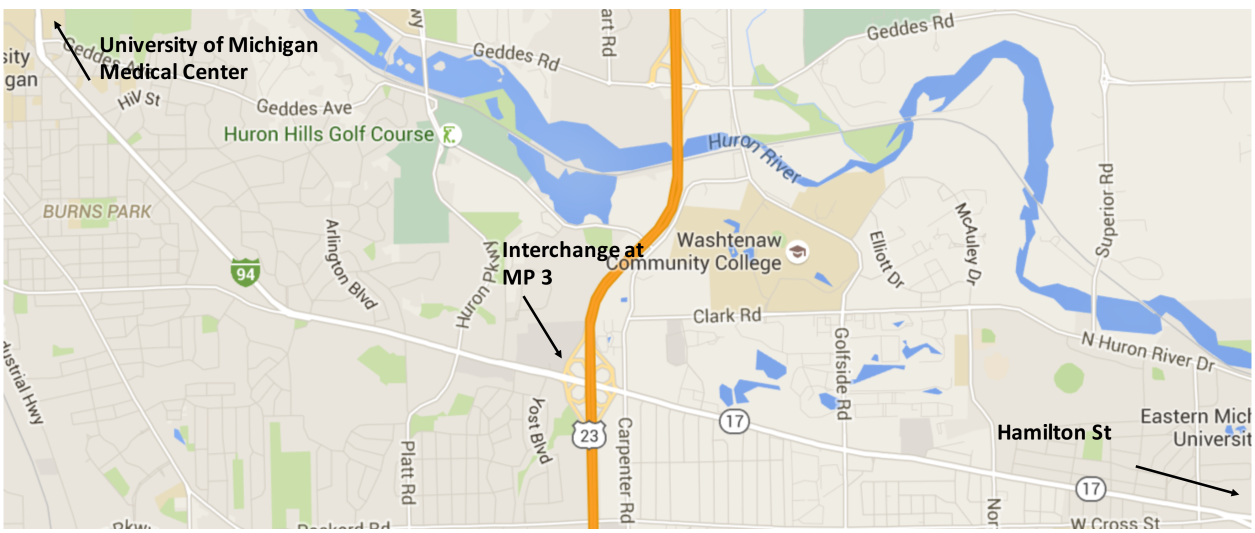

Figure 2 - Washentaw Avenue. *Base map Source: Google Maps®*

Events were found along Washentaw Avenue by using a longitudinal acceleration threshold of 0.6 g'. This is one of the event flags published and used in the 100-Car NDS to flag potential events *(23)* . The validity of using a simple threshold is certainly debatable, as discussed in the research needs, however, for the purposes of this simple case that is merely meant to demonstrate the type of data that CV technology provides and a high-level overview of a possible use case, the authors believe it is acceptable.

In total 178 BSMs were flagged as possible events in just the two months with limited market penetration of CV technology on a restricted study area. The events were placed into 0.2 mile-long bins along the route, and the number of events, and density of events, in each bin was then calculated. Figure 2, shows the density of events, calculated using a kernel density estimation technique, along Washentaw Avenue, with "Mile 0" being the Northwest corner in Figure 1 and "Mile 6" being the Southeast corner.

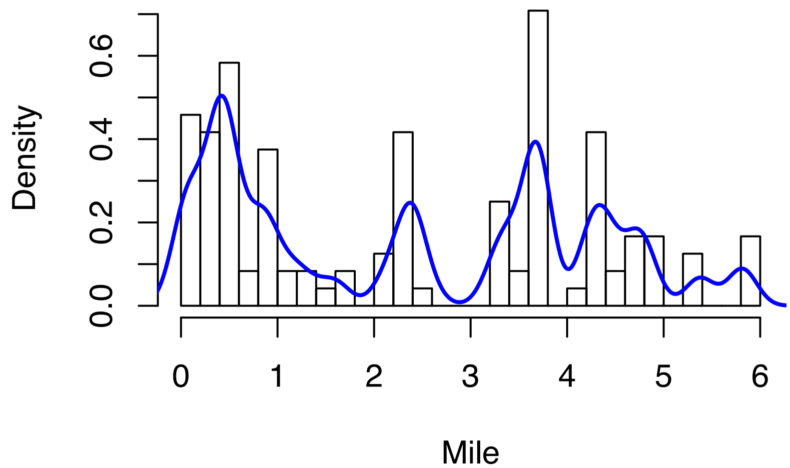

Figure 3 - Location and Density of Events

It can be seen that in just two months and with an estimated 2% of vehicles, peaks are starting to form with clusters of Event Flags. The cluster around 0.5 miles is likely due to medical center and University of Michigan as it is close to campus and has emergency vehicles frequently. The dip around 3 miles illustrates another challenge as it aligns perfectly with Route 23, which likely means all of the BSMs from that area were assigned to Route 23 in the database. The other peak around 3.6 is a segment with an abundance of driveways which may be a source of the problem.

If those specific peaks persist remains to be seen as do their meaning in the context of roadway safety. This will become clearer as additional connected vehicle data is collected and further research is conducted in this area. The key takeaway with this is that with the expected vast amounts of data being received, it is quite conceivable that safety issues can be identified quickly.

CONCLUSION

Network screening is a process that many transportation infrastructure providers are responsible for carrying out. Since CV technology will most likely be implemented system-wide due to the benefits it offers in all aspects of transportation, it makes sense to explore using the technology to carry out network screening.

Established network screening practices are the way they are in order to counteract historic data limitations. CV technology will provide DOTs with the ability to access a more holistic data set, rather than relying on crash reports and the disadvantages that come along with them in terms of error and bias. As a result, a more complete view of network safety can be attained faster than is currently possible.

Challenges remain before network screening can be carried out using CVs, as demonstrated by the Safety Pilot Model Deployment case study presented. Additional research needs to be done to establish details into carrying out the actual network screening methodology, and the technology itself needs to progress to the point that abundant, high-quality data is collected on a network-wide basis. Nonetheless, there is widespread commitment to CV technology in both the public and private sector, and research is only raising interest in implementation as further applications are developed and benefits are identified.

ACKNOWLEDGMENTS

The authors would like to thank the Connected Vehicle/Infrastructure University Transportation Center and the Federal Highway Administration's Office of Operations Research and Development for funding different phases of this work.

1. AASHTO. *Highway Safety Manual*. Washington, D.C., 2010.
2. Harding, J., G. Powel, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons, and J. Wang. *Vehicle-to-vehicle communications: Readiness of V2V technology for application. (Report No. DOT HS 812 014)*. Washington, 2014.
3. Li, X., D. Lord, Y. Zhang, and Y. Xie. Predicting motor vehicle crashes using Support Vector Machine models. *Accident Analysis and Prevention*, Vol. 40, No. 4, 2008, pp. 1611–1618.
4. Ma, J., K. M. Kockelman, and P. Damien. A multivariate Poisson-lognormal regression model for prediction of crash counts by severity, using Bayesian methods. *Accident Analysis and Prevention*, Vol. 40, No. 3, 2008, pp. 964–975.
5. Park, E. S., and D. Lord. Multivariate Poisson-Lognormal Models for Jointly Modeling Crash Frequency by Severity. *Transportation Research Record*, Vol. 2019, No. -1, 2008, pp. 1–6.
6. Xie, Y., D. Lord, and Y. Zhang. Predicting motor vehicle collisions using Bayesian neural network models: An empirical analysis. *Accident Analysis and Prevention*, Vol. 39, No. 5, 2007, pp. 922–933.
7. Yu, R., and M. Abdel-Aty. Analyzing crash injury severity for a mountainous freeway incorporating real-time traffic and weather data. *Safety Science*, Vol. 63, 2014, pp. 50–56.
8. Garber, N. J., and G. Rivera. Safety Performance Functions for Intersections on Highways Maintained by the Virginia Department of Transportation. *Fhwa/Vtrc 11-Cr1*, 2010.
9. Savolainen, P. T., T. Gates, D. Lord, S. Geedipally, E. Rista, T. Barrette, B. J. Russo, and R. Hamzeie. *Michigan Urban Trunkline Intersections Safety Performance Functions (SPFs) Development and Support*. Lansing, MI, 2015.
10. Srinivasan, R., D. Carter, and K. Bauer. *Safety Performance Function Decision Guide: SPF Calibration versus SPF Development*. Washington, D.C., 2013.
11. Hauer, E. *Observational Before-After Studies in Road Safety*. Pergamon Press, Oxford, UK, 1997.
12. Gerber, M. S., and L. Tang. Automatic quality control of transportation reports using statistical language processing. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14, No. 4, 2013, pp. 1681–1689.
13. Austin, K. The identification of mistakes in road accident records: Part 1, locational variables. *Accident Analysis and Prevention*, Vol. 27, No. 2, 1995, pp. 261–276.
14. Loo, B. P. Y. Validating crash locations for quantitative spatial analysis: A GIS-based approach. *Accident Analysis and Prevention*, Vol. 38, No. 5, 2006, pp. 879–886.
15. Austin, K. The identification of mistakes in road accident records: Part 2, casualty variables. *Accident Analysis and Prevention*, Vol. 27, No. 2, 1995, pp. 277–282.
16. Aptel, I., L. R. Salmi, F. Masson, A. Bourdé, G. Henrion, and P. Erny. Road accident statistics: Discrepancies between police and hospital data in a French island. *Accident Analysis and Prevention*, Vol. 31, No. 1-2, 1998, pp. 101–108.
17. Wu, K. F., and P. P. Jovanis. Defining and screening crash surrogate events using naturalistic driving data. *Accident Analysis and Prevention*, Vol. 61, 2013, pp. 10–22.
18. Guo, F., S. G. Klauer, M. T. McGill, and T. A. Dingus. *Evaluating the Relationship Between Near-Crashes and Crashes: Can Near-Crashes Serve as a Surrogate Safety*

Metric for Crashes? 2010.

19. Amundsen, F., and C. Hyden. Proceeding of First Workshop on Traffic Conflicts. 1977.
20. Chin, H. C., and S. T. Quek. Measurement of traffic conflicts. *Safety Science*, Vol. 26, No. 3, 1997, pp. 169–185.
21. El-Basyouny, K., and T. Sayed. Safety performance functions using traffic conflicts. *Safety Science*, Vol. 51, No. 1, 2013, pp. 160–164.
22. Saunier, N., T. Sayed, and K. Ismail. Large-Scale Automated Analysis of Vehicle Interactions and Collisions. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2147, 2010, pp. 42–50.
23. Dingus, T., S. G. Klauer, V. L. Neale, A. D. Petersen, S. E. Lee, J. Sudweeks, M. A. Perez, J. Hankey, D. Ramsey, S. Gupta, C. Bucher, Z. R. Doerzaph, J. Jermeland, and R. R. Knipling. *The 100-Car Naturalistic Driving Study Phase II – Results of the 100-Car Field Experiment*. 2006.
24. Hankey, J., M. Perez, and J. McClafferty. *Description of the SHRP 2 Naturalistic Database and the crash, near-crash, and baseline data sets.* Blacksburg, VA.
25. SAE International. *Surface Vehicle Standard: J2735 Dedicated Short Range Communication Message Set Dictionary*. 2009.
26. Bishop, R. *Global V2X Deployment: Contrasts with U. S. Approach*. 2015.
27. US Department of Transportation. US DOT Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) Technology Test Bed and Affiliated Interoperable Test Beds. http://www.its.dot.gov/presentations/pdf/USDOTV2VV2ITesBed_ITSJPOSlides.pdf.
28. US Department of Transportation. Connected Vehicle Test Bed. http://www.its.dot.gov/connected_vehicle/dot_cvbrochure.htm.
29. Kluger, R., and B. L. Smith. Pattern Matching Longitudinal Time Series Data To Identify Crashes in Naturalistic Driving Data. 2014.

The following paper was submitted and accepted to ITS World Congress in Detroit, Michigan 2014. It was presented in a panel session and included in the proceedings

PATTERN MATCHING LONGITUDINAL ACCELERATION TIME SERIES DATA TO IDENTIFY CRASHES IN NATURALISTIC DRIVING DATA

Robert Kluger
University of Virginia
351 McCormick Road,
Charlottesville, Virginia, 22904

Brian L. Smith
University of Virginia
351 McCormick Road,
Charlottesville, Virginia, 22904

ABSTRACT

This paper uses kinematic vehicle data collected from the SHRP2 naturalistic driving study to develop a technique for identifying crashes and safety critical events using longitudinal acceleration. The method treats the acceleration data as a time series, and applies pattern matching to classify readings into different driving activities. Groups of unclassified readings were considered potential crashes – or safety critical events. Thirteen crash events were acquired and twelve were successfully identified as crashes by using this method. Additionally there were three false positives, though one could be defined as a safety-critical event that did not result in a crash. Applications of this include data mining large datasets for abnormal driving actions and detecting events in real-time in a connected vehicle environment.

KEYWORDS

Naturalistic Driving Data, Pattern Matching, Crash, Near-Crash, Safety-Critical Event Time Series Acceleration Data, Data Mining

INTRODUCTION

The purpose of this paper is to show that by applying a pattern matching technique to longitudinal vehicle acceleration data, one can detect crashes and near-crashes. The primary application of this algorithm would be to detect crashes and near-crashes in a connected vehicle environment, where in-vehicle kinematic data would be passively collected by government-owned roadside infrastructure.

In-vehicle kinematic data (3-direction acceleration, speed, position, etc), has been limited mostly to a single vehicle until recently. Wireless technology has made the availability of this data a possibility on a larger scale, and it is expected to become even more common in the future with the introduction of connected vehicle technology (1). One benefit of these vehicle-based datasets is that we will have more insight than ever into crashes and pre-crash scenarios. This type of data can also provide insight into near-crashes and other safety-critical events, both of which have been studied as surrogate safety measures with similar causal factors as crashes (2, 3), but have not been well defined or detectable.

Detection of crashes and near-crashes using kinematic vehicle data is a relatively new problem. It becomes much more relevant with both the introduction of connected vehicle technology, and the availability naturalistic driving data. Given that it is new, it is unclear the best approach to take in detecting them.

While the long-term application of detecting crashes and near-crashes is in connected vehicle environments, Naturalistic Driving Data is more available and is a viable substitute with a few applications that are also relevant to detecting crashes. The SHRP2 Naturalistic Driving Study conducted at Virginia Tech Transportation Institute (VTTI), is a large scale study where the subjects have been involved in numerous crash and near-crash events. In this study a set of 6 different triggers were developed to flag potential crashes, two of which were related to the vehicles' accelerations and one of which was yaw rate, both available in connected vehicle data. The other triggers were possible due to other equipment being used in the study, such as radar and an event button, neither of which will be available in connected vehicles. Additionally, this study had camera in the vehicle, for an analyst to verify that the trigger was truly a crash. The three triggers that could be feasible in early connected vehicle environments where some, but not all, vehicles are equipped with the capabilities are longitudinal acceleration, lateral acceleration, and yaw rate, which simply had thresholds to flag the events (2, 4). In those cases, either, there is a low threshold and the manpower required to process flagged events is very high incurring a large cost, or the threshold is too high and many events won't be identified. Thus, there is a tradeoff between the desire to detect every crash or safety critical event and the desire to minimize false alarms. Additionally the use of a single threshold fails to account for the fact that acceleration readings should be treated as a time-series since there is autocorrelation present in the data. Other modeling and data mining techniques, such as logistic regression or classification trees may have potential, but cannot address the fact that this data may need to be treated as a time series.

Other potentially relevant literature would be the development of automated crash detection systems, but when considering the purpose of both systems the fit is not so great. The purpose of automated detection systems is to alert the proper authorities upon the occurrence of an event without sending false alarms and calling the police and other first responders for nothing. The majority of the crashes that get reported to police and input into the crash database would also cross the thresholds used by VTTI, however the goal is to detect those in addition to near-crashes and the ones that do not get reported or do not cross the reporting threshold made by state police (high property damage or personal injury). False alarms do not have the same drawbacks for hotspot detection as they do for automated detection systems.

Subsequence matching is the pattern matching technique used in time series data to detect occurrences of a specific subsequence within a new time series or another portion of the same time series. Pattern matching has very rarely been applied to vehicle accelerations, and only to determine vehicle location and route choices (5). In this study, longitudinal acceleration readings for naturalistic driving trips were matched against five different normal driving patterns that were observed while subjects were driving. Once normal driving patterns were assigned to different subsections of the time series, the unassigned values were examined for crashes.

Applications of detecting crashes using this data set will vary. More accurate data mining methods can benefit future studies where this type of data is collected, since this procedure could minimize the time and effort required by analysts to identify safety-critical events that occur in similar connected vehicle or naturalistic driving studies while only slightly increasing the computational cost. Additionally, emerging connected vehicle technology means this data will soon be available from the majority of vehicles. This means that comprehensive network screening methods can use locations that frequently have vehicles flagged by this technique to identify hot spots along the network, without relying on police crash reports.

This study should be treated as a proof of concept, showing the viability of this method and that it is potentially better than employing a simple threshold in order to detect crashes and near-crashes. Throughout the methodology, it will become evident that numerous decisions had to be made, that can be optimized in further studies. Given the data limitations when designing this detection algorithm, it did not make sense to spend too much time on that problem, since it was not likely to hold up as the optimal solution in an expanded dataset. This paper shows the algorithm can work, however it can also likely be improved upon with additional analysis and more data.

DATA SET

The Naturalistic Driving Study (NDS) is part of the Strategic Highway Research Program (SHRP II) and is being managed by Virginia Tech Transportation Institute (VTTI). The study consists of 2,300 vehicles in six locations across the United States (6). Thirteen trips with crashes were acquired from VTTI for use in this study. NDS data provided included two types of files. The first consisted of two video files, a front camera view (Figure 1a) and rear camera view (Figure 1b). The second part of the data was an event log with kinematic data collected at a frequency of 10 Hz. These were connected by a timestamp on the front camera view where anything that is seen in the video could be matched to a specific reading.

Figure 1a - Front Camera View

Figure 1b - Rear Camera View

Trip lengths varied with some being less than two minute long trips while the longest was over thirty minutes. Each trip concluded in the subject vehicle being involved in some type of crash, most frequently a rear-end collision. Crash types and distribution of conditions are shown in Table 1.

Table 1 – Crash Characteristics

Crash Type	Count	Conditions	Count
Rear End	9	**Day**	11
Sideswipe	1	**Night**	2
Lane Departure	1	**Clear**	11
Angle	2	**Precipitation**	2

Kinematic data was collected using a data acquisition system installed in the vehicle. This data included:

- Speed
- x, y, z – direction acceleration
- Latitude and Longitude
- Heading
- Lane position
- Pitch, Roll, Yaw
- Timestamp

Video was watched for each of the crash events acquired and the type of crash and time of crash was recorded and added to the event file.
Clearly with only 13 crashes with which to design the algorithm, it is not possible to find the optimal solution. For this reason this should be considered a proof of concept and pending further testing to determine how much better the algorithm truly is over the threshold. This will be discussed further in the results and conclusions section.

METHODOLOGY

It makes intuitive sense that a vehicle's current acceleration will depend on what it was in the immediate past, especially for normal driving tasks where actions are deliberate and repeated throughout a trip. Seeing an acceleration drop from -0.2g's to -0.3g's is very different than an acceleration dropping from 0.3g's to -0.3g's over the same time period. Using pattern recognition, this difference can be captured, while employing a reasonable threshold will not identify that.

Figure 2 shows the longitudinal acceleration (g's) for a trip that took place primarily on the highway. In this trip the vehicle had to stop suddenly on the freeway around time 125 seconds, which can be seen by the rougher acceleration pattern at that point. The vehicle then rear-ended the vehicle it was following at a high speed at time 170 seconds, where it can be seen that the acceleration dropped to -3 g's upon impact.

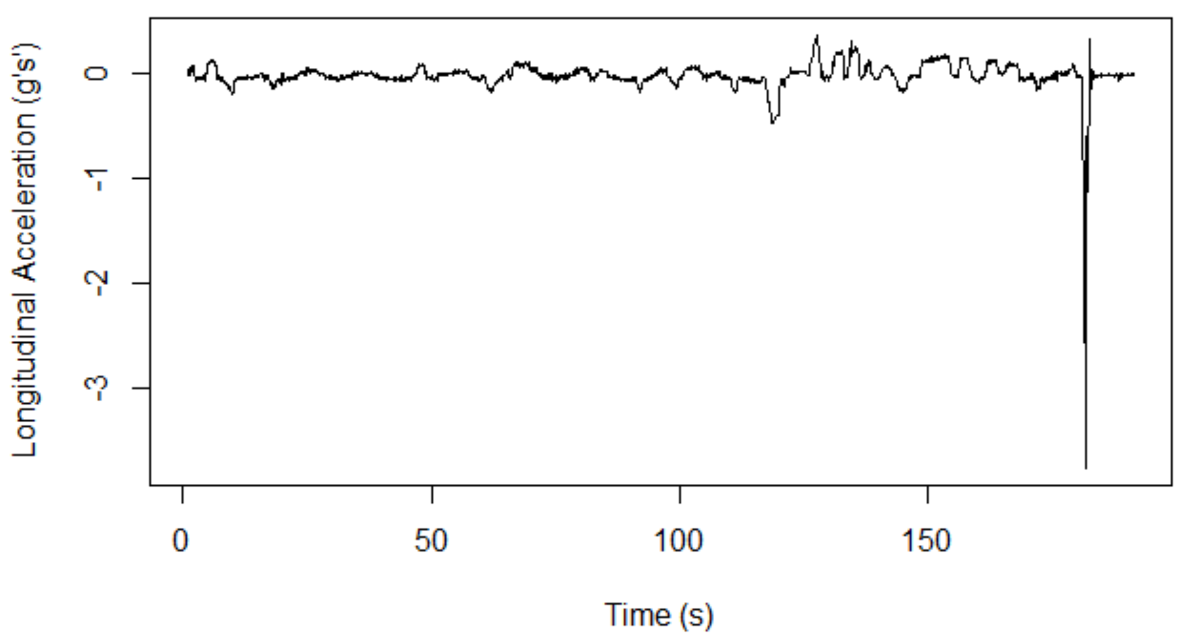

Figure 2 - Sample Time Series of Longitudinal Accelerations, Rear End Crash at Time 170 Seconds

Figure 3 shows a panel of 4 other crashes events. It can be seen that the y-axis values vary from crash to crash, and will depend on a combination of crash type, type of vehicle involved, and speed at impact. Thus, simply implementing threshold can lead to issues when trying to detect crashes in this dataset and similar datasets, forcing the analyst into a tradeoff of selecting a high threshold and missing lower severity crashes, or selecting a low threshold and having false positives. In the NDS setting, false crash readings can be screened by video data, but in other environments without corresponding video data, this will not be possible.

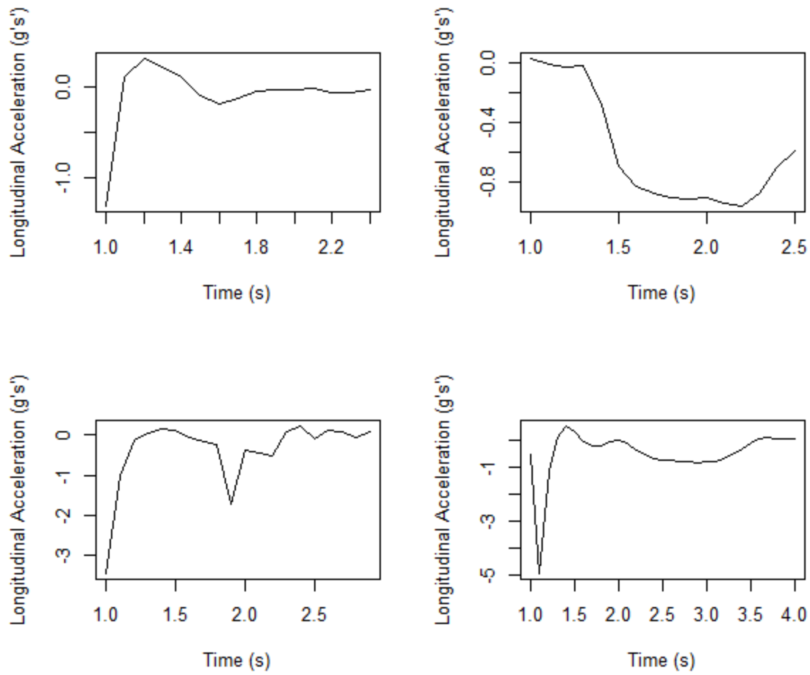

Figure 3- Some Acceleration Profiles during Crashes

By examining figure 2 again, it can be seen that certain patterns appear to repeat themselves throughout the trip. Based on that observation, it was hypothesized that if one could develop an algorithm to identify or filter out the normal driving actions, one would be left with less common driving activities such as crashes and near-crashes, which will not follow a consistent pattern.

Using the video data from three different trips, five baseline time series were selected to represent five different common driving actions. The primary selection criterion for the baselines was watching different videos to find different examples of vehicles performing these actions and extracting that subsequence from the time series. Those actions include, accelerating from a stop, accelerating to adjust speed, constant speed, braking to adjust speed, and braking with the intent of stopping. The selected baselines are shown in figure 4. Sensitivity analysis was done, with different series selected as baselines, and with different numbers of baselines. Using too few baselines will result in numerous stretches being unable to be identified, while too many baselines can lead to confusion about what type of action is happening at a specific point.

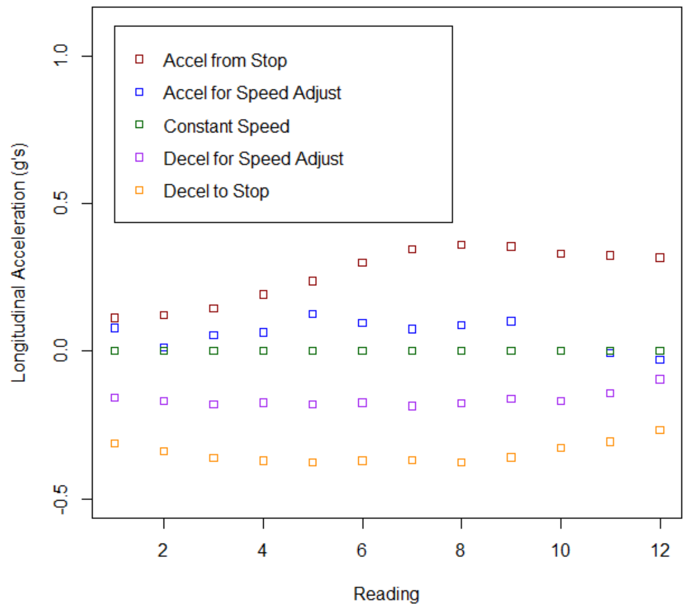

Figure 4 - Selected Baselines

The euclidean distance between the baseline and a portion of the time series was calculated for every stretch of 12 readings (~1.2 seconds) using a sliding window performing an exhaustive search for each subsequence along the time series. The decision to use 1.2 seconds was a somewhat arbitrary one that dictated many of the subsequent decisions. However that length was chosen because it was short enough to capture the majority of drivers' actions, and not so long that it could capture many additional actions over one time series.

$$d = \sqrt{\sum (b_i - y_i)^2}$$

Where b = baseline vector
y = test vector

Each window was matched to a baseline that had the minimum euclidean distance, provided that distance was no larger than $d = 0.5$. If no baseline was matched with the window, the stretch was marked as unidentified, to be reviewed later. The baseline was settled upon after testing multiple candidate baselines. After no apparent difference in the results between different baselines tested, the candidate baseline with the closest to the chosen length of 12 readings was selected.

Since a sliding window method was used, every individual point was pattern matched 12 times and thus, may have been associated with multiple patterns if it occurred during a transition between actions. The solution is to assign each individual point as part of one of the five actions or as unidentified, based on the results of the twelve sliding windows. If six or more of the windows the reading was a part of identified it as a specific action, the point was assigned to that action. Otherwise, the action was listed as unidentified. Six was settled on as a threshold through sensitivity analysis and because it ensured that the majority of the windows indicated the point was a part of the action.

Figure 5 shows a crash (black dots) compared to the baselines from figure 4. Just by inspection, it can be seen that the pattern quickly deviates from all of the baselines. Any set of points that had more than eight unidentified readings in a row were examined further to see what had occurred at those times.

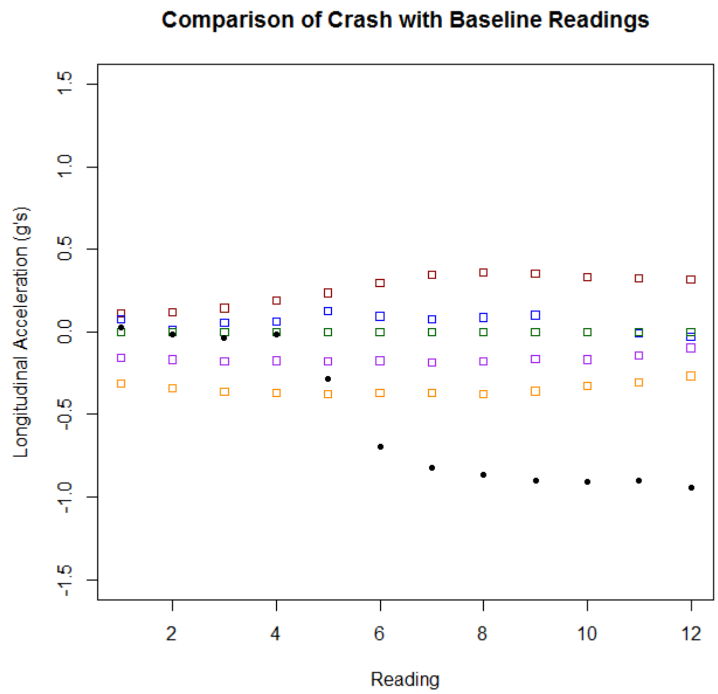

Figure 5 - Crash comparison (Black Dots) with baseline readings

RESULTS AND CONCLUSIONS

Pattern matching had very positive preliminary results identifying 12/13 crashes while producing only three false positives. Upon further inspection, the unidentified crash was a very low-speed, rear-end collision that did not exceed a deceleration of -0.3 g's at the point of impact. Additionally, the false positives occurred at explainable points upon reviewing the video. The first occurred when a vehicle went over a speed bump and the second occurred when the vehicle (acceleration series shown in figure 2) was forced to stop suddenly on the freeway which could be defined as a safety-critical event and arguably should be detected to inspect if it is a frequent occurrence on that segment. The last occurred when a vehicle began accelerating from a stop and had to quickly stop to avoid rear-ending the lead vehicle.

The pattern matching results were then compared to the results of using two different thresholds as identifiers of events. The results were favorable for the pattern matching technique. Table 2 shows the number of events correctly identified, in addition to the number of false positives detected, by the pattern matching technique, in addition to thresholds of 0.6g's and 0.4g's.

Table 2 – Method Comparison

	Pattern Matching	Threshold 0.6g's	Threshold 0.4g's
Detection Rate	12/13	10/13	11/13
False Positives	3	3	5

It should be noted that pattern matching and thresholds did not always detect the same false positives. In general the pattern matching technique's false positives tended to be either near-crashes or safety conflicts, while for the thresholds, this was not necessarily the case, especially with the lower threshold of 0.4 g's. Additionally, a challenge with using the thresholds was that since a single value had to be crossed, it was sometimes difficult to tell if two violations were related, while the algorithm was developed with a logical way to account for successive indications of crashes.

At numerous points throughout the methodology sensitivity analysis was brought up as the justification for decisions made. A complete sensitivity analysis is necessary but it is important to keep in mind that the algorithm was only designed on 13 crashes, so finding the optimal values to use in the algorithms is not going to be possible on a general basis. That being said, decisions had to be made on what distance metric to use, number of baselines to use and their value, length of windows, and criteria for the classification of unidentified points, among a few other things. Sensitivity analysis was done and it was found that many of the choices made at different points of the algorithm have a range of acceptable decisions, but are for the most part, related. In other words, the selected length of the window is going to impact later steps in the algorithm such as the value of the Euclidean distance.

In this study, only longitudinal acceleration was examined to identify crashes. However it is likely that other data categories collected may be able to improve the capabilities of this method jointly, such as lateral acceleration, vertical acceleration, or yaw rate. For example, in the case of the vehicle traveling over a speed bump, it is possible that an algorithm for the vertical acceleration could potentially detect that and prevent the false positive.

While this on the surface appears to be highly successful, it is necessary to test the entire algorithm and process on more trips to see if the method holds. While fine-tuning of the algorithm is clearly necessary, this paper does show that this method can be used to detect crashes in an environment where acceleration data is available on a large scale. This means safety analysis may no longer require police crash reports, and by detecting lower severity, near-crashes and safety-critical events, it could be possible to identify hot spots – unsafe portions of roadway due to infrastructure or operational issues – more quickly than waiting for crashes to occur and for police reports to get processed.

It is currently not possible to say if there are certain situations where this algorithm will fail or certain situations where the algorithm majorly outperforms another method. For example, there were only two trips where the subject vehicle ever entered any sort of heavy traffic congestion and those were both for short periods of time during which the subjects both crashed. Basically, the algorithm may not be possible to completely fine tune until connected vehicle is available on a large scale.

Overall, the results of this indicate that pattern matching longitudinal acceleration data can potentially be an improvement over simple acceleration thresholds when detecting crashes, near-crashes, and other safety critical events. Further research must be done to locate patterns within the lateral and vertical accelerations readings to determine if the combined results could improve the accuracy of this technique. Additionally more computationally efficient pattern matching techniques must be explored if this were to be implemented on a large scale and further sensitivity analysis needs to be done.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Michael Fontaine for his advice in the development of this problem and its solution.

REFERENCES

(1) "U.S. Department of Transportation Announces Decision to Move Forward with Vehicle-to-Vehicle Communication Technology for Light Vehicles" Available at www.nhtsa.gov
(2) Feng Guo, Sheila G. Klauer, Jonathan M. Hankey, and Thomas A. Dingus, "Near Crashes as a Crash Surrogate for Naturalistic Driving Studies," Transportation Research Record No. 2147. 2010. p 66-74.
(3) G.A. Fitch, S.A. Soccolich, F. Guo, J. McClafferty, Y. Fang, R.L. Olson, M.A. Perez, R.J. Hanowski, J.M. Hankey, and T.A. Dingus. "The Impact of Hand-Held and Hands-Free Cell Phone Use on Driving Performance and Safety-Critical Event Risk," Report No. DOT HS 811 757. April 2013.
(4) T.A. Dingus, S.G. Klauer, V.L. Neal, A. Petersen, S.E. Lee, J. Sudweeks, M.A. Perez, J. Hankey, D. Ramsey, S. Gupta, C. Bucher, Z.R. Doerzaph, J. Jermeland, and R.R. Knipling."The 100-Car Naturalistic Driving Study: Phase II – Results of the 100-Car Field Experiment." Report No. DOT HS 810 593. April 2006.
(5) Pramod K. Vemulapalli, Gregory S. Ledva, Sean N. Brennan, and Karl M. Reichard, "Pattern Matching of In-Vehicle Acceleration Time Series Data," Presented at ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference.
(6) Kenneth L. Campbell, "The SHRP 2 Naturalistic Driving Study: Addressing Driver Performance and Behavior in Traffic Safety," TR News 282. September-October 2012. p 30-35.

The following paper was presented in a poster session at the Transportation Research Board's Annual Meeting in January 2016 in Washington, DC and the TransInfo UTC Conference in Arlington, Virginia in August 2016.

IDENTIFICATION OF SAFETY-CRITICAL EVENTS IN CONNECTED VEHICLE ENVIRONMENTS

Robert M Kluger
University of Virginia, Department of Civil and Environmental Engineering
351 McCormick Rd. , PO BOX 400742
Charlottesville, VA, 22904-4742
e-mail: rmk7su@virginia.edu
Phone: 513-884-6757

Brian L Smith, PE, PhD
University of Virginia, Department of Civil and Environmental Engineering
351 McCormick Rd., PO BOX 400742
Charlottesville, VA, 22904-4742
e-mail: bls2z@virginia.edu
Phone: 434-243-8585

Hyungjun Park, Ph.D.
University of Virginia Center for Transportation Studies
351 McCormick Rd., PO BOX 400742
Charlottesville, VA22904-4742
e-mail: hpark@email.virginia.edu
Phone: 434-924-1651

Keywords: Crashes, Near-Crashes, Safety-Critical Events, Connected Vehicles, Kinematic Data Elements, Basic Safety Message

ABSTRACT

Presented in this paper is a 5-step heuristic algorithm that can identify safety-critical events, such as crashes, near-crashes, and other safety related events, using kinematic data of a single vehicle. This kinematic data contains only data elements outlined in the connected vehicle standards. Naturalistic driving study data was used as a surrogate for connected vehicle data to design the algorithm. The algorithm first estimates speed at a future time using speed and acceleration at a previous time. If this is done across a very short time-span, major discrepancies between the predicted speed and the actual speed could indicate a safety-critical event occurred. Events with such discrepancies are flagged. A logistic regression model was then constructed to predict the probability of a flagged event being a crash or near-crash. The algorithm showed promising results on a limited data set and should be treated as a proof-of-concept until further validation can take place.

INTRODUCTION

High resolution, kinematic vehicle data (second-by-second speed, acceleration, yaw, etc.) is becoming more available than ever to transportation researchers, engineers and policy makers. With this influx of data, there are a considerable number of potential benefits to a wide range of safety applications, such as real-time emergency response or screening the network for safety hot spots. However, for these benefits to be realized, there is a need to first be able to mine the kinematic data to identify when and where crashes and near-crashes have occurred.

The motivation behind this paper was to develop ways to identify safety-critical events, defined in this context as crashes, near-crashes, and other unsafe vehicle maneuvers, in a partially-saturated connected vehicle environment. In a partially saturated environment, only a small subset of the entire active vehicle fleet will be broadcasting and receiving kinematic data, in the form of a standardized, Basic Safety Message (BSM) [1]. The primary effect of the constraint imposed by a partially saturated environment (which is expected to exist for many years following a potential mandate of vehicular on-board equipment by NHTSA), in this application, is that time-to-collision (TTC) and other metrics describing the interaction between a lead and a following vehicle cannot be applied.

A heuristic algorithm has been designed to detect safety-critical events in a simple and efficient manner. The algorithm focuses purely on speed and acceleration values of a single vehicle, collected in time-series, both elements included in the BSM broadcast by connected vehicles [1]. The benefit to focusing on a single vehicle is that this algorithm can be deployed without relying on a wide deployment of connected vehicle technology. The output provides a set of potential safety-critical events with an associated probability that the identified event is safety-critical.

It should be emphasized that this paper is a proof-of-concept meant to show that *a.)* there is merit to being able to identify safety-critical events using kinematic vehicle data of a single vehicle, and *b.)* it can be done by creatively processing the time series information projected based on the BSM of an individual vehicle. Given that there is only a limited sample size of such data currently available, the authors do not intend to find the optimal solution. Rather, the goal is to show that there is potential even with a limited data set without any capabilities for validation. Thorough validation with a larger data set and expansion of this methodology will be designated to a future paper when more data becomes available.

In this paper, first, background information relevant to connected vehicles and the identification of crashes and near-crashes, will be outlined. Next the algorithm steps will be presented and their intended purpose will be detailed. The algorithm's performance will then be reported using an example, safety-related application. Finally, limitations and future direction will be discussed.

BACKGROUND

Kinematic Data Sources

Connected vehicle technology is an emerging vehicle technology that will allow equipped vehicles to communicate with other equipped vehicles and roadside infrastructure. This communication comes in the form of standardized messages, one of which is called the Basic Safety Message (BSM), which is projected at 10 Hz and contains the kinematic data necessary to detect the occurrence of a safety-critical event [1]. Testbeds for connected vehicles exist in multiple locations both within the US [2,3] and around the rest of the world [4]. This technology is currently in a prototype development and implementation phase, but is likely to penetrate the commercial market in the future. Early estimates have projected that as many as 75% of new cars could be equipped with connected vehicle technology by 2020 [6].

Another source of kinematic data are naturalistic driving studies (NDS)[7,8], in which subjects' vehicles were retrofitted with data-acquisition systems, radar, and video cameras. While these studies are not connected vehicle studies, they collect similar data at the same frequency, and they still consist of real data, collected in a naturalistic setting, rather than a controlled experiment. In naturalistic settings, crashes, near-crashes, and other safety-critical events can (and do) occur. The presence of a large number of test subjects and video data make these studies advantageous for training a classification model, because there are numerous events that occurred, and they could be validated using the videos.

Identification of Crashes and Near-Crashes

The goal of this algorithm is to identify safety-critical events that occur in partially saturated connected vehicle environments where only a single vehicle data would be available rather than a data set for a pair of adjacent vehicles. So, for example, in the connected vehicle testbed in Ann Arbor, MI, only an estimated 2% of the vehicle fleet has the technology equipped [2]. Additionally, it is not anticipated that radar will be a standard piece of equipment in a connected vehicle. Consequently, relying on the communication capabilities or radar to derive TTC is not going to be feasible. This has not been reflected in previously published methods for identifying crashes and/or near-crashes.

The 100-Car Naturalistic driving study was one of the first settings where a method to identify crashes and near-crashes was necessary. Table 1 shows the triggers Virginia Tech Transportation Institute (VTTI) used to identify potential events that occurred in the 100-Car Study, as well as the expanded follow-up study, the SHRP II Naturalistic Driving Study. The percent of valid events identified is the percentage of true positives that the trigger was able to account for, while the percent of identified events that were invalid was the percent of events identified by the trigger that were not crashes or near-crashes. The unit "g" stands for gravitational acceleration.

While these triggers are very suitable for the NDS purposes, they have limitations that prevent their application in a connected vehicle setting. First the TTC thresholds, as well as the event button (which was a button that allowed users to manually alert VTTI to the occurrence of a crash or near-crash) cannot be used in a partially saturated connected vehicle setting. Second, these triggers are very conservative because this specific study had video recordings of its subjects. This means that any event that the trigger specifies can be validated, or invalidated, by reviewing a corresponding video. As stated in Table 1, 66.4% of the time when the Longitudinal Acceleration threshold was exceeded, it was a false positive. Similarly there was no crash or near-crash 91.3% of the time when the Lateral Acceleration threshold was exceeded. Meanwhile, those two thresholds combined detected less than 50% of all crashes experienced in the 100-Car study. The prediction accuracy experienced here would be similar when applying those two thresholds to a connected vehicle study, but the high false alarm rate seen in the NDS implementations was only acceptable because there is video to validate any events flagged by the triggers.

Table 1 - VTTI Event Triggers and Corresponding Performance, (6)

Trigger	Description	Percent of Valid Events Identified (Recall)	Percent of Identified Events that were Invalid (1-Precision)
Lateral Acceleration	Lateral Motion equal to or greater than 0.7 g.	3.5	91.3
Longitudinal Acceleration	Acceleration or Deceleration equal to or greater than 0.6 g. Acceleration or deceleration equal to or greater than 0.5 g coupled with a forward TTC of 4 s or less. All longitudinal decelerations between 0.4 g and 0.5 g coupled with a forward TTC ≤ 4 s, and with a corresponding forward range value at the minimum TTC not greater than 100 ft.	44.7	66.4
Event Button	Activated by the driver pressing a button located by the rearview mirror when an event occurred that the driver deemed critical.	8.4	69.9
Forward TTC	Acceleration or deceleration ≥ 0.5 g coupled with a forward TTC of 4 seconds or less. All longitudinal decelerations between 0.4 g and 0.5 g coupled with a forward TTC ≤ 4 s, and with a corresponding forward range value at the minimum TTC not greater than 100 ft.	56.4	86.4
Rear TTC	Any rear TTC trigger value of 2 s or less that also had a corresponding rear range distance of ≤ 50 ft AND any rear TTC trigger value in which the absolute acceleration of the following vehicle is greater than 0.3 g.	4.6	59.9
Yaw Rate	Any value greater than or Equal to a plus AND minus 4-degree change in heading (i.e., vehicle must return to the same general direction of travel) within a 3-s window of time.	21.7	91.1

Two additional studies had similar goals and were able to achieve reasonable classification rates for crash and near-crash events but they relied on radar data, or other information on the interaction between lead and following vehicles. The first was done using radar data from the 100 Car NDS study and showed that the relationship between Range Rate (speed moving toward radar target) and Range, could achieve a decision boundary that was accurate 74% of the time in identifying crashes or near-crashes, with only 20% false alarms [6]. Talebpour et al. [8] also devised an algorithm using data collected in the NGSim study [9] to identify near-crashes using the distribution of individual drivers' longitudinal accelerations and their interactions with lead and following vehicles. Again, the reliance on information from a lead and following vehicle make this a less feasible approach until there is a high market penetration of the connected vehicle technology.

Wu and Thor [10] suggested that a safety frontier of a vehicle that experiences a high braking event could be used to identify whether or not that hard braking resulted in a safety-related event. The safety frontier is calculated by determining required deceleration to stop before hitting the lead vehicle and the deceleration actually experienced. Again, this requires information on lead and following vehicles which is not something that can be accurately assessed in a partially saturated connected vehicle environment.

Wu and Jovanis [11] presented a 4-step process to identify surrogate events, or events that have a positive correlation to a crash, using NDS data. The proposed process involved a preliminary screen of kinematic data, a chow test to classify crashes by intersection and non-intersection, and then a secondary screening. A model was then developed to predict the conditional probability of an observed surrogate event being a crash, under certain circumstances. It isn't clear what the false positive rate is when applied to non-crash or non-near-crash scenarios (i.e. how many times will this trigger over 10 hours of driving with no safety-critical events), and it's unknown what the scalability of this process into a big data environment is, but this could be another potential way to identify crashes observed in BSMs because the methodology is limited to kinematic data.

An event similar to the near-crash is the traffic conflict, an incident where a crash would occur barring one or more vehicles involved performing an evasive maneuver, such as hard braking or a sudden change of trajectory. Traffic conflicts can vary by perceived severity, from minor conflicts to serious conflicts depending on how close the event is to a collision, often measure in the form of TTC. Serious conflicts and near-crashes are synonymous, so near-crashes are a specific type of traffic conflict. These events have been used in numerous studies as surrogates for crashes since they are much more common than crashes but often provide the same insight if they can be captured. The problem is actually capturing these events, as they require observation, which can be time consuming and involves subjective judgment to identify [12]. Computer vision analysis techniques can be utilized to intake video data from deployed cameras and automatically track vehicles. They can be used to identify traffic conflicts by predicting future vehicle positions and estimating when two vehicles would collide. This requires the deployment of cameras and software to track vehicles, but is a good way to achieve consistency in defining conflicts and avoids having to designate time to watch video and manual record conflicts. It also can successfully measure vehicle speeds, which can in turn be used to measure TTC, acceleration, and jerk [13,14].

In summary, numerous substantial work had been done to automatically identify when and where different events that have safety implications are occurring using emerging technology. Many of the previous studies have had interesting findings and solid methodologies, but cannot necessarily be used as stand-alone methods to identify when and where safety-critical events are occurring without relying on information on vehicle interactions.

METHODOLOGY

Data

In this study, time series and video data was acquired from the 100-Car Study [6] on 11 trips with crashes and 14 trips without crashes. Each trip with a crash in it had a time-series data file with more than 50 data elements as well as two video files, one from a camera facing out the front windshield (Figure 1), and one from a camera facing out the rear windshield. The timestamp in the bottom left corner of the front video is how observations in the data file were matched to points in the videos.

Figure 1 - Front Video of a Trip in the 100 Car Naturalistic Driving Study

The data that was used was the vehicle's acceleration, derived from a GPS unit equipped on the vehicle, and vehicle speed, which was collected from the vehicle's data bus. The speed field was collected at varying frequencies ranging from 10 Hz, to 1 Hz. For the speeds collected at less than 10 Hz, the values of speed were interpolated, and the collection frequency was recorded. Other strings missing values were handled in a similar manner as long as they were not for more than 1 second. If they were longer than 1 second, the time series was broken and the resulting pieces were analyzed individually.

The distribution of crash types is shown in Table 2. The sample size is very small and the majority of the crashes are rear end crashes. As a result, the algorithm is focused on speed and longitudinal acceleration, though future work can modify the algorithm to include lateral acceleration, yaw rate, braking, and numerous other variables. For this reason, this paper must be treated as a proof-of-concept and not something that is ready for use in a real setting until more data can be acquired. However if the results hold on a valid data set, this is a simple process with high accuracy that can easily be expanded upon.

Table 2 - Crash Type Distribution

Crash Type	***Count***
Rear End	7
Sideswipe – Same Direction	1
Lane Departure	1
Angle	2

For the 14 normal driving trips, these trips had no crashes or near-crashes, and were used to establish a false positive rate. This data consisted of over 10 hours of driving in total across all of the trips. For the trips with crashes, the time spent driving before the crash was also used to test for false positives. This time varied between the acquired data and ranged from less than 2 minutes, to more than 40 minutes.

Algorithm Purpose and Development

In general, this algorithm estimates what a vehicle's future speed should be, given a current speed and acceleration. Then, if the estimate and the future speed are not within a certain margin of error, the observation is flagged as a potentially safety-critical event. Finally, a logistic regression model was built to predict the probability of a flagged observation being safety-critical, using the length of the flagged event and the extent at which it exceeded the allowable margin of error. This section will outline the steps in the proposed algorithm, while discussing the purpose and functionality of each step. The algorithm detects safety-critical events using the following process:

1. Predict speed at a future time, given current speed and acceleration in a time series. The speed and acceleration time series data must be collected at a high frequency.

$$S_{Predicted\_t+1} = S_{Actual\_t} + \Delta t \, a_t$$

S = Speed (m/s),
a = Acceleration (m/s^2),
t = Time

This step implements a well-known physical model to describe future speed based on current speed and average acceleration over a given time period. This calculation can be made over an entire vehicle's time series. The key assumption is that, if Δt is small enough, the observed instantaneous speed collected from the sensors can be treated as an average speed, as long as nothing major occurs between the observations (like a crash).

2. Calculate the prediction discrepancy using the difference in predicted speed and actual speed.

$$\boldsymbol{\delta}_t = S_{Predicted\_t} - S_{Actual\_t}$$

S = Speed (m/s),
δ = Speed Prediction Discrepancy
t = Time

The second step is to quantify the difference between the prediction of the physical model described in the first step and the actual observed speed. As the vehicle collects new data, the difference between the predicted speed, calculated from the previous time period's observed speed and acceleration, and the observed speed can be determined. In a connected vehicle system, speed and acceleration are being collected at a high frequency, so this calculation can be done in real time, or this can be done post-hoc once the time series data has already been collected.

3. Find all observations in the time series where the error, δ_t, exceeds a certain threshold, $\delta_{threshold}$. Flag:

$$\boldsymbol{\delta}_{threshold} \leq \boldsymbol{\delta}_t$$

$$-\boldsymbol{\delta}_{threshold} \geq \boldsymbol{\delta}_t$$

Once all of the $\boldsymbol{\delta}$'s have been calculated, the next step is to screen them for the points where the assumption in the first step is violated. If a safety-critical event does happen, the actual speed observed will be affected, while the predicted speed will not, since the prediction was calculated before the event occurred. Any major discrepancies between the predicted and observed speeds will be flagged for further investigation in this step.

4. Identify groups of observations with a prediction discrepancy exceeding the threshold, and for each group, record the number of observations, *n*, and the maximum or minimum observed error, δ_{Max}. Since direction is no longer important, the absolute value should be taken before getting the value of δ_{Max}.

The fourth step is to define events and then determine the extent to which they violate the assumptions. The purpose of this step is to aggregate observations that are close together into a single event, and use this to describe the likelihood of the group of observations being an event, considering the extent and magnitude of the prediction discrepancy. Not all flagged groups of observations are likely to be safety-critical events, but the events that are will likely deviate from the prediction by a larger amount, for a longer period of time, or both.

Multiple observations that exceed the threshold but occur within a certain time window will be grouped into a single event, rather than being defined as multiple single short-lived events. The value of *n* will need to be corrected to account for the collection frequency of speed data from the data bus. By virtue of the method for interpolation, the data that was collected at 1 Hz is going to have a higher number of observations flagged in the event of a high discrepancy, than the same event would if the data had been collected at 10 Hz. Because of this, a correction was made to *n*, using the following formula.

$$n_{corrected} = \frac{n}{10/frequency}$$

The effect of the above formula is that a set flagged at a frequency of 10 Hz is only going to count 1/10 of the observations, while a collection frequency of 1 Hz will count all of them. It should be noted that this is only applicable if the collection rate of the dataset is 10 Hz and values were interpolated.

Additionally, the maximum observed error in each group quantifies the severity of the event. The result of this step is a data table that has a measurement of length, and maximum difference in discrepancy for each event that is flagged by the algorithm through this point.

5. Determine the probability of a group of observations occurring during a safety-critical event using a classification model.

The final step in this algorithm is to build, or apply, a classification model, using the $n_{corrected}$ and δ_{Max} values as the independent variables. The dependent variable can either be a probability or binary variable to classify an event as "safety-critical" or "not safety-critical." The model only needs to be built once, but should be application-specific due to the trade off between detection and false positives. For the same modeling technique and data sets, as the number of detected events increases, the number of false alarms will also increase. An example will be shown in the results section that will demonstrate what can be done once a model is established.

RESULTS

Example Application

The specific application these events are being used for is important to remember when applying the algorithm. Namely, the cost of having a false alarm or the cost of missing a true positive are going to drive the decisions made when applying the algorithm. For example, in an automated detection system to alert emergency services to a crash, the cost of a false positive is going to be very expensive as it wastes the emergency response team's time and potentially prevents them from responding to another incident. Conversely, if this is used in a screening application to identify potentially unsafe sites, then any false positives will likely be revealed in later in the process of site evaluation, so having some false positives to identify all of the major safety events acceptable. The latter is more relevant to the authors, as this algorithm will be applied in larger study where the goal is to use kinematic data to replace crash reports in traffic engineering safety analyses.

For the first step, the value of Δt in this application was 0.1 seconds, the time between observations collected in the 100-Car study. The tenth of a second frequency of observations is small enough to reasonably treat instantaneous observations as an average. It is unclear at what frequency the assumption becomes invalid due to the time interval between observations, however there were a few trips that collected speed data at a lower frequency.

The value of $\delta_{threshold}$, used in step three, was 2 m/s in this application. For reference, a change in 2 m/s is a change in 4.4 mph, so an observation's detected speed would need to change by 4.4 mph in 0.1 seconds for the algorithm to flag it as a potential event for further investigation. Values of 1.5 and 2.5 m/s were also tested but were less successful, with the 1.5 m/s threshold providing too many false positives and the 2.5 m/s threshold failing to identify enough actual events, which was the result of the limited sample data set. The selected threshold should reflect the goals of the application while maintaining the ability to correctly group sets of events. If the threshold is too low, there will be a large number of observations that get flagged which may lead to more false observations as well as multiple different events being conglomerated into a single group by the algorithm, while if the threshold is too high some safety-critical events won't be flagged. This is where the application is important, because most applications have an associated cost with false positives and false negatives. In a real-time application the cost of a false positive is sending emergency services to a location and disrupting traffic for no apparent reason. So, in that application you want to minimize false positives. However if the application is to identify safety-critical events that occurred, regardless of their severity, to safety at a location, a false positive may still be interesting or provide information. Additional testing to find the optimal threshold can be carried out using sensitivity analysis, however given the sample size, finding an optimal value is unlikely to hold to a further application.

After the implementation of steps one through three, Table 3 provides an example of what the results could look like on a single trip. The trip presented in Table 3 was a trip during which a vehicle was driving on the highway in heavy traffic. The subject vehicle approached the end of the queue and had to hit the brakes quickly on multiple occasions. Eventually the subject vehicle was unable to slow down in time struck the lead vehicle. In a real application the "Crash" column would not be present, but for algorithm design purposes, the video was used as a way to validate the findings. It should also be noted that this trip was selected since it had a lot of false positives so it can be used to clearly provide examples for subsequent steps. Most trips flagged very few, if any, observations outside of the crash event.

Table 3 – Observations Exceeding the Error Threshold in a Sample Trip

Time (s)	***Speed Discrepancy***	***Crash***	***Group***
631.2	5.348964	0	1
644.3	-4.615082	0	2
1777.4	4.45491	0	3
1785.9	-4.398706	0	3
1786.2	4.941294	0	3
1786.3	-4.648707	0	3
1828.1	4.517498	0	4
1846.7	-4.424654	0	5
1846.8	4.609572	0	5
1846.9	-4.85905	0	5
2204.3	11.873522	1	6
2204.4	3.57367	1	6
2204.5	3.089111	1	6
2205.7	2.509898	0	6
2205.9	2.12455	0	6
2206.0	2.751791	0	6
2206.1	2.571878	0	6
2206.4	2.043342	0	6
2206.5	4.42576	0	6
2210.3	-5.105168	0	6

In the fourth step, for each group of the flagged observations, obtain the number of flagged observations in the group, n, as well as the maximum discrepancy in speed prediction, δ_{max} over that timespan. Two observations are considered part of the same group if they occurred within 10 seconds of each other. This was usually long enough to cover brief stop-and-go situations on the freeway where a vehicle approaches the end of a queue without generating two separate groups, one for when the vehicle comes to a stop and then restarts motion. Lowering the value will increase the number of events identified, and may lead to the identification of multiple events that are results of the same action performed by a driver, such as a brief stop on the freeway. Increasing the value of the time window can potentially lead to a grouping of flagged observations that stem from multiple actions the driver performs.

Table 4 shows the results of applying step 4 to the trip outlined in Table 3. In this case, the speed data was collected at 10 Hz so no correction was applied to *n*. This process was done for every trip and resulted in a large data frame with which to build a model. Again, the crash column was for algorithm training purposes and is not present when applying algorithm to new data. After steps one through four were applied to all of the trips in the training data set, 39 groups were identified among all of the trips. This included ten actual crash events (all but one of the 11 crash events) and 29 other potential events.

Table 4 - Identified Groups after Step 4

Group	$N_{corrected}$	***|Speed Discrepancy|***	***Crash***
1	1	5.348964	0
2	1	4.615082	0
3	4	4.941294	0
4	1	4.517498	0
5	3	4.85905	0
6	10	11.873522	1

Modeling and Performance

In steps 1-4, a preliminary filtering was carried out to identify segments of trips that could potentially be safety-critical. The error rate is still fairly high; especially considering the sample of trips being used to build the algorithm has a very high crash rate, relative to what would normally be experienced in a real setting. So, there is a need to further distinguish between the potential events using characteristics of those events.

To further classify the flagged events, a logistic regression model was trained. The model used in an application of this algorithm should reflect the goals of the application. In this paper, an example application will be provided to demonstrate how the process should be carried out. That example will be to detect crashes and near-crashes to do a preliminary network screening for a regional safety analysis. In this setting, false alarms are less of an issue, because a subsequent site diagnosis phase should catch them after a more comprehensive analysis takes place. The form of the logistic regression model is:

$$Odds = e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_i X_i}$$

Odds ratio can then be used to calculate probability in the following manner:

$$Probability = \frac{Odds}{Odds + 1}$$

Twelve candidate models were analyzed before a model was selected. The candidate models tested the inclusion of difference in predicted speed and observed speed and length of event, with difference combinations, interactions, and transformations of the two variables. The best models were selected using minimum AIC (Akaike Information Criterion) and cross-validated error as the criteria, and the model's performance was quantified using a combination of area under the ROC (Receiver Operator Characteristic) curve, and model accuracy in terms of precision and recall on the training set. There was not enough data to build a robust test set, and the implications of this will be discussed in the discussion section. Table 5 presents the three best models.

Table 5 - Candidate classification models

Model 1	***Coefficient***	***SE***	***z-value***	***p-value***	
Intercept	-4.583	1.419	-3.23	0.00124	**
log(|Discrepancy|)	2.783	1.029	2.704	0.00685	**
Model 2	***Coefficient***	***SE***	***z-value***	***p-value***	
Intercept	-2.6602	0.7466	-3.563	0.000367	***
$\mathbf{N_{corrected}}$	0.7296	0.2937	2.484	0.012985	*
Model 3	***Coefficient***	***SE***	***z-value***	***p-value***	
Intercept	-4.554	1.5246	-2.987	0.00282	**
log(|Discrepancy|)	2.0096	1.191	1.687	0.09153	.
$\mathbf{N_{corrected}}$	0.4619	0.323	1.43	0.15267	

Significance: . = 90%, * = 95%, ** = 99%, *** = 99%

The first model included the log of the extreme value of the difference in predicted and observed speeds, with both coefficients being significant with 99% confidence. It had 14 false positives, which a total of five drivers were responsible for, and it classified all crash events correctly. The second model only included the time-length of the event, measured in number of consecutive observations. There were nine false positives, seven of which came from the same driver, however this model only correctly classified six of the ten crashes. The third model had both of the variables and there were six false positives across two drivers. There were also three false negatives. For comparison, applying the NDS acceleration thresholds from Table 1, four of the crashes were flagged, and one false positive was identified.

Table 6 presents the cross-validated error, AIC, and area under the ROC curve for each candidate model. The ROC curves are shown in Figure 2. Table 7 presents the third model's prediction of the event with 1 being an event occurred and 0 being no event occurred. It should be noted that this example was chosen because it best illustrated the way to carry out the algorithm and not because. This driver experienced some unusual circumstances and was somewhat aggressive which led to a larger number of initial flags. Most of the trips with a crash only had the crash flagged.

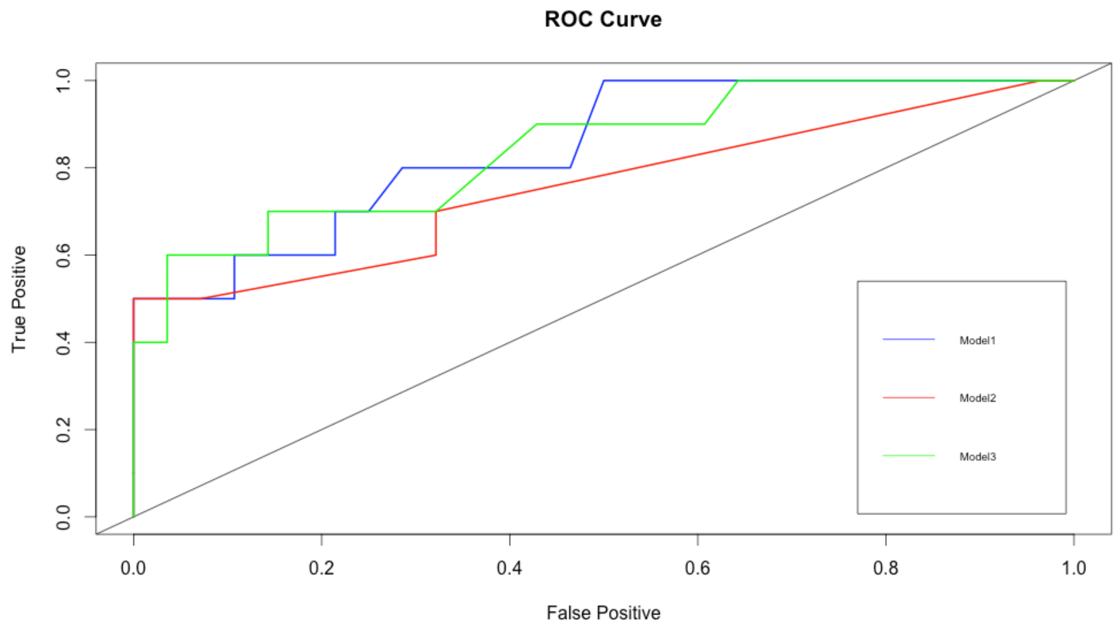

Figure. 2 - Receiver Operator Characteristic Curves of Each Model

Table 6 - Model performance metrics

	Cross-Validated Error	***AIC***	***Area under ROC Curve***
Model 1	0.1483	35.664	**.845**
Model 2	**0.1434**	36.392	.755
Model 3	0.1491	**35.075**	.841

Table 7 – Example model output

Group	***$N_{corrected}$***	***|Speed Discrepancy|***	***OR***	***Probability***	***Crash***
1	1	5.348964	0.485684551	0.326909606	0
2	1	4.615082	0.361042657	0.265269171	0
3	4	4.941294	1.655646723	0.62344389	0
4	1	4.517498	0.345864915	0.25698338	0
5	3	4.85905	1.008598742	0.502140483	0
6	10	11.873522	154.0650849	0.993551095	1

Of the three final candidate models, none stand out as much better than the other two. There is merit to selecting each of the models and it is difficult to make a definite suggestion without a more complete dataset. The cross-validated error suggests that model 2 would hold up best on a validation set, but on the training set, the other two models are clearly better. If having a specific probability is not enough for the application, a decision boundary threshold can be selected to place observations into a binary, event/no-event classification.

For an example, the probabilities from the sample events flagged in the previous section (Table 4) will be calculated using model 3. If the decision boundary were a probability of 0.5, the highlighted rows would be flagged as events by the algorithm. It should be noted that for this particular scenario, there were 2 false positives with a threshold of 50%. In most cases, only the crash was flagged, but this particular trip was chosen as the example since it is easier to see what is happening with the previous steps of the algorithm.

DISCUSSION

The presented material should be treated primarily as a proof-of-concept, to show that this methodology can work. In the authors' opinion, this work is promising in that it appears to outperform the simple acceleration thresholds at first glance, but is not supported with a robust data set yet. Additional data is required to confirm this, and validate the algorithm. It is also believed that additional data will increase the significance of coefficients in the models, and will allow for the testing of models with additional variables, such as change in acceleration over the span of an event.

Additionally, the authors say that this algorithm detects safety-critical events, but the only events to train the algorithm on were crashes. Again additional data that includes all safety-critical events would either validate or invalidate that this model can indeed detect near-crashes and other, less severe events. It should also be noted that many of the false positives could be considered near-crashes, depending on the observer.

When video was available, the false positives flagged by the algorithm were reviewed. There were a few common situations that tended to flag the algorithm in situations without a crash. Additionally, any missed crash events were reviewed to find out why the event may have been missed. For the first 4 steps, all but one of the crash events was flagged, but 29 other non-events were also flagged. That event was a rear-end crash where the subject driver released the brakes and rolled into the lead vehicle while in a queue at a traffic signal.

Certain occurrences other than crashes tended to flag the algorithm. Frequently a single driver would falsely flag the algorithm multiple times which could imply a few things: either the driver was generally more aggressive on stops or starts, the driver was in a certain set of conditions where false positives are more likely to happen (like approaching a queue on the freeway), or their in-vehicle equipment may have been defective or not have been well calibrated. Regardless, a few trips tended to comprise many of the original flags, indicating there is likely a vehicle-specific, or driver-specific calibration that can be done to filter out false predictions.

CONCLUSION

In this study, a 5-step, heuristic algorithm was presented as a way to identify safety-critical events using speed and acceleration time series data collected at a high frequency. The algorithm performed well on the limited data set, but further testing is required to validate and refine this algorithm, so the maximum precision and recall can be achieved. The algorithm should be used as a starting point to classify safety-critical events and can be expanded upon with further research. For example, if TTC metrics were to become available, they can be added to the algorithm in either the preliminary screening or the secondary model. It is also very simple, and can be easily integrated into connected vehicle technology in its current state. Further work must be done to validate this algorithm and sensitivity analysis needs to be completed to find optimal values for thresholds and other inputs, once a larger, representative data set is available.

Finally, including separate steps in the algorithm that can identify the difference between different types and severities of crashes would also provide a major benefit to applications. In its current state, the algorithm only implies that a safety-critical event likely happened at a certain point of time, but it does not provide any additional information about the severity, or even what type of interaction likely occurred. Examining each type of crash and evasive maneuver individually would create a more specific algorithm that may be able to increase the accuracy, as well as provide additional information about what occurred once an event is identified.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Michael Fontaine, Dr. Brian Kluger, and Congyu Wu for providing comments on the paper. This work was funded by the Connected Vehicle/Infrastructure University Transportation Center and the Eisenhower Graduate Research Fellowship Program.

NOTATION

The following symbols are used in this paper:

$S_{predicted}$ = Predicted Speed
S_{Actual} = Actual Speed
t = Time
a = Acceleration
$\delta = S_{predicted} - S_{Actual}$ = Speed Discrepancy
$\delta_{threshold}$ = Threshold of Discrepancy
n = Number of Consecutive Observation Flagged
f = Collection Frequency (Hz)
$n_{corrected}$ = n corrected for f
Crash = Variable Indicating Presence of a Crash (0/1)
OR = Odds Ratio
AIC = Aikaike Information Criterion

References

(1) Society of Automotive Engineers International (2009). *J2735 Surface Vehicle Standard: Dedicated Short Range Communications Message Set Dictionary*.

(2) Harding, J., G. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons, and J. Wang. *Vehicle-to-vehicle communications: Readiness of V2V technology for application.* Report No. DOT HS 812 014. Washington DC: National Highway Traffic Safety Administration. August 2014.

(3) Connected Vehicle/Infrastructure University Transportation Center. *Testbed Development.* www.cvi-utc.org. Accessed June 2015.

(4) Bishop, Richard. *Global V2X Deployment: Contrasts with U.S. Approach.* Bishop Consulting. January 2013.

(5) Greenough, J. *The 'connected car' is creating a massive new business opportunity for auto, tech, and telecom companies.* Business insider. http://www.businessinsider.com/connected-car-forecasts-top-manufacturers-2015-2. Accessed June 2015.

(6) Dingus, T.A., S.G. Klauer, V.L. Neal, A. Petersen, S.E. Lee, J. Sudweeks, M.A. Perez, J. Hankey, D. Ramsey, S. Gupta, C. Bucher, Z.R. Doerzaph, J. Jermeland, and R.R. Knipling. *The 100-Car Naturalistic Driving Study: Phase II – Results of the 100-Car Field Experiment.* Report No. DOT HS 810 593. April 2006.

(7) Committee for the Strategic Highway Research Program 2: Implementation. *Implementing the Results of the Second Strategic Highway Research Program: Saving Lives, Reducing Congestion, and Improving Quality of Life.* TRB Special Report 296. Transportation Research Board. Washington DC. 2009.

(8) Talebpour, A., H.S. Mahmassani, F. Mete, and H. H. Samer. *Near-Crash Identification in a Connected Vehicle Environment.* Transportation Research Record: Journal of the Transportation Research Board, No. 2424, Transportation Research Board of the National Academies, Washington, D.C., 2014, pp. 20–28.

(9) FHWA. *NGSim Insterstate 80 Data Set Fact Sheet. http://www.fhwa.dot.gov/publications/research/operations/06137/index.cfm.* Last Accessed August 2014.

(10) Kun-Feng Wu and Craig Thor 2015. Utilizing Naturalistic Driving Study Data to Study Freeway Rear-end Crash Sequences. Accepted for Publication by the 94th Annual Meeting of the Transportation Research Board of the National Academies, Washington D.C.

(11) Kun-Feng Wu and Paul P. Jovanis. 2013. Defining and Screening Crash Surrogate Events using Naturalistic Driving Data. Accident Analysis and Prevention. Volume 61, Elsevier, Amsterdam, pp 10-22.

(12) Hoong-Chor Chin and Ser-Tong Quek 1997. Measurement of Traffic Conflicts. Safety Science. Volume 26 No. 3, Elsevier, Amsterdam. pp 169-185.

(13) Saunier, N., T. Sayed, and K. Ismail. Large-Scale Automated Analysis of Vehicle Interactions and Collisions 2010. Transportation Research Record: Journal of the Transportation Research Board, No. 2147,Transportation Research Board of the National Academies, Washington, D.C., pp. 42-50.

(14) Zaki, M.H., T. Sayed, and K. Shabaan 2014. Use of Drivers' Jerk Profiles in Computer Vision-Based Traffic Safety Evaluations. Transportation Research Record: Journal of the Transportation Research Board, No. 2434, Transportation Research Board of the National Academies, Washington, D.C., pp. 103-112.

The following paper was submitted and accepted to Accident Analysis and Prevention in August 2016. It is available via ScienceDirect and is in press for Accident Analysis and Prevention 96 (2016) pp 162-168.

IDENTIFICATION OF SAFETY-CRITICAL EVENTS USING KINEMATIC VEHICLE DATA AND THE DISCRETE FOURIER TRANSFORM

Corresponding Author:
Robert Kluger
rmk7su@virginia.edu
University of Virginia
Department of Civil and Environmental Engineering
351 McCormick Road
Charlottesville, VA, 22903

Authors
Brian L. Smith, University of Virginia
Hyungjun Park, University of Virginia
Daniel J. Dailey, University of Washington

Abstract

Recent technological advances have made it both feasible and practical to identify unsafe driving behaviors using second-by-second trajectory data. Presented in this paper is a unique approach to detecting safety-critical events using vehicles' longitudinal accelerations. A Discrete Fourier Transform is used in combination with K-means clustering to flag patterns in the vehicles' accelerations in time-series that are likely to be crashes or near-crashes. The algorithm was able to detect roughly 78% of crashes and near-crashes (71 out of 91 validated events in the Naturalistic Driving Study data used), while generating about 1 false positive every 2.7 hours. In addition to presenting the promising results, an implementation strategy is discussed and further research topics that can improve this method are suggested in the paper.

Keywords: Crashes, Near-Crashes, Safety-Critical Event, Naturalistic Driving Study, Discrete Fourier Transform

Funding: This work was funded through the Connected Vehicle/Infrastructure University Transportation Center [DTRT12-G-UTC20]; Federal Highway Administration Office of Operations Research and Development [DTFH6414G00180]; and the FHWA Office of Safety R&D. The research project data set was purchased from Virginia Tech Transportation Institute (VTTI) using the FHWA SHRP2 Safety Implementation program budget.

1. Introduction

High resolution, kinematic vehicle data (second-by-second speed, acceleration, yaw, etc.) is becoming more available than ever in the transportation community. With this influx of data, there are a considerable number of potential benefits to a wide range of safety applications, including monitoring driver performance, identifying unsafe locations on the road (hot spots), or even providing real-time emergency response. However, before these benefits can be realized, there is a need to be able to identify unsafe driving activities, like crashes and near-crashes, amongst a vast amount of regular driving.

The goal this paper to is to develop ways to identify safety-critical events (SCEs), defined in this context as crashes, near-crashes, and other unsafe driving behaviors using kinematic data from single vehicles. Creating an algorithm that can detect SCEs using only the trajectories of single vehicles could have a variety of applications including:

- Allowing infrastructure providers to identify SCEs in connected vehicle environments and evaluate road network safety
- Allowing taxis and shared ride service providers to monitor their drivers and ensure they provide safe rides to customers
- Allowing insurance companies to monitor their customers' driving tendencies and better evaluate risk
- Allowing agencies to monitor fleet vehicles (e.g. buses, snow plows, etc.) for both driver performance and tort liability claims
- Allowing transportation management agencies to monitor traffic and provide emergency response when necessary
- Providing real-time alerts to emergency response services in connected vehicle environments
- Identifying events in large-scale naturalistic driving studies

Each of these applications is slightly different and will likely require varying inputs to a method or algorithm when identifying SCEs but there is a clear benefit to a variety of stakeholders by having the ability to identify them.

Many established methods for identifying unsafe driving, whether it be SCEs, or a specific subset of SCEs, rely on information to be available describing how one or more vehicles are interacting. One example of such information is Time-to-Collision (TTC), which is an estimate of how much time a vehicle has on its current trajectory before it would collide with a lead vehicle. This typically requires access to radar data, which can be expensive to equip on large fleets of vehicles. As a result the analysis was restricted to kinematic data that is native to connected vehicle standards (SAE International, 2009) and can be collected from smart phones or aftermarket devices.

For this study, crash and near-crash data was acquired from the SHRP2 Naturalistic Driving Study (NDS) (Virgina Tech Transportation Institute, 2013). The methodology outlined performs subsequence-matching techniques on longitudinal accelerations observed in vehicles during a set of crashes and near-crashes. A Discrete Fourier Transform (DFT) is used to transform subsequences of the observed time-series and a K-means clustering algorithm is then used to classify those subsequences as events or baseline driving.

2. Research Goals

The primary goal of this study is to develop a methodology for identifying safety critical events when given a high-resolution time series of kinematic vehicle data, specifically longitudinal acceleration. With recent advancements in vehicle and roadside technology, learning how to identify unsafe driving behavior using high-resolution data streams has become a practical endeavor that can provide benefits in a variety of applications.

Time series data was acquired from the SHRP2 NDS during crash and near-crash events. The goal of the algorithm developed was to identify time series where a crash or near-crash occurred without flagging time series that did not contain any SCEs. Before proceeding, it is necessary to provide definitions relevant to this study.

- *Crash: "Any contact with an object, either moving or fixed, at any speed in which kinetic energy is measurably transferred or dissipated. Includes other vehicles, roadside barriers, objects on or off the roadway, pedestrians, cyclists, or animals."*
- *Near-Crash: "Any circumstance that requires a rapid evasive maneuver by the participant vehicle or any other vehicle, pedestrian, cyclist, or animal, to avoid a crash. A rapid evasive maneuver is defined as steering, braking, accelerating, or any combination of control inputs that approaches the limits of the vehicle capabilities."*
- *Baseline: Any time series without a crash or near-crash.*
- *Safety-Critical Event (will be used synonymously with the term "Event"): Any crash or near-crash event.*

The crash, near-crash and driving definitions were those used by VTTI for their naturalistic driving studies (Guo, Klauer, McGill, & Dingus, 2010), since that is the source of the data. The authors defined a safety-critical event as any crash or near-crash, though a case can certainly be made to include other situations and will also be discussed further at a later point.

The proposed algorithm takes the following steps:

- Break time-series into small subsequences or "windows" to examine specific sections in time
- Perform Discrete Fourier Transform to identify the strength of different frequencies present in each window
- Execute K-means clustering to group each window by the strength of different frequencies.

Relevant literature is examined, including additional context for the research motivation as well as some information on previous approaches to this problem. Then a description of the methods used and why they were applied is provided. While the range of applications is diverse, the specific inputs the presented methodology addresses is light vehicle crashes and near-crashes. The discussion section addresses how this algorithm may change based on specifics of each application.

3. Literature Review

Discussed in this section will be background information on two key topics relevant to this paper. The first will outline a few studies that collect high-resolution kinematic data on a large-scale. Second, there will be a review of literature that uses this type of data to classify events, or other patterns and behaviors, with a description of the methods being used.

Two studies that have successfully recorded kinematic data during crashes are the 100-Car Naturalistic Driving Study (Dingus et al., 2006) and the larger follow up, SHRP2 Naturalistic Driving Study (Virgina Tech Transportation Institute, 2013). In both of these studies, subjects were recruited to equip their vehicles with cameras, radar, and a Data Acquisition System (DAS) designed by Virginia Tech Transportation Institute (VTTI). High-resolution trip-level data was then generated for subjects over the life of each study. Other studies such as the Safety Pilot Model Deployment in Ann Arbor, Michigan (Harding et al., 2014), the NGSim study in California (Halkias & Colyar, 2006), and Integrated Vehicle-Based System Safety Field Operation Test in Ann Arbor, Michigan (Sayer et al., 2008) Also collected similar data on different scales and in different contexts. Since all of these studies are naturalistic, subjects could, and on occasion did, get into crashes and near-crashes.

In particular, the SHRP2 NDS is unique due to the scale of the study in terms of both network coverage and number of participating subjects, the presence of a system for documenting events, and the presence of a suite of cameras equipped to vehicles for establishing ground truth. While some of the other listed studies also had some of those qualities, they were unable to accomplish all of those at the level of the SHRP2 NDS.

In the SHRP2 NDS, VTTI and the field teams at each site were responsible for identifying when and where their subjects got into crashes. Their approach was to use a collection of criteria to flag potential events in the trip data collected. Those flags include a longitudinal acceleration threshold, a lateral acceleration threshold, some time-to-collision (TTC) thresholds, a yaw rate trigger, and an event button that subjects could press to signal a collision. Individual thresholds alone (e.g. 0.6 *g*'s of longitudinal acceleration) tended to have low recall (true positives/total events) and many of them also had poor precision (true positives/test positive) (Dingus et al., 2006). The SHRP2 Study has adjusted the criteria to flag events by removing most of the radar-based triggers, adding a time-component to the deceleration, adjusting the acceleration thresholds to 0.75 *g*'s of lateral acceleration and 0.65 *g*'s of longitudinal acceleration, and adding some vehicle-safety system activation triggers. The individual triggers often had recall in the single digits, with the best individual flag had around 20% recall. While VTTI was successful in locating crashes despite the low individual identification rates of individual flags, they were able to include some data elements that were not native to the BSM and they have video to verify if an event did or did not occur for trips that were flagged.

Vehicle trajectories from the SHRP2 NDS and similar studies have been analyzed to classify certain occurrences on the road in terms of kinematic data elements. Engström and Victor developed and patented at method using neural networks to classify driving patterns and demonstrated the method on vehicle trajectories in different roadway setting (Engstrom & Victor, 2005). McDonald et. al used a computationally efficient SAX-VOX method to transform time series data into character strings and perform natural language processing to identify commonly observed action and patterns (Mcdonald et al., 2013).

In terms of specifically detecting safety-critical events, Wu and Jovanis proposed a novel algorithm to classify crash types using the maximum differences over time in both lateral and longitudinal accelerations during crashes and near-crashes. They also outlined the sensitivity and specificities they were able to achieve for a variety of thresholds in those calculated kinematic elements, which were an improvement on the NDS event flags (K Wu & Jovanis, 2013a; K Wu & Jovanis, 2013b). Kluger and Smith used Euclidean distance to classify crashes with longitudinal acceleration time series data, assigning known patterns to a pre-defined action and flagging any subsequence that did not fit into one of those patterns. This analysis was performed on a limited sample size with promising results (Kluger & Smith, 2014).

A range of additional studies in crash and near-crash dynamics has occurred using TTC or other lead-vehicle-following-vehicle information as a metric. Wu and Thor proposed the idea of a safety frontier calculated by temporal headway and the difference in speeds between the lead and following vehicles. They showed that if the safety frontier was violated, a rear-end crash was likely to occur (K. Wu & Thor, 2015). Talebpour et al. developed an algorithm that specifically identifies near-crashes in connected vehicle environments using drivers' accelerations and behavior during car-following situations to identify near-crashes and specifically highlights the differences between drivers (Talebpour, Mahmassani, Mete, & Hamdar, 2014).

The last type of work in event detection methodologies relates to the concept of traffic conflicts, which has frequently been proposed as a surrogate event for crashes. A traffic conflict is defined as "an observable situation in which two or more road users approach each other in space and time to such an extent that there is risk of collision if their movements remain unchanged" (Amundsen & Hyden, 1977). This traffic conflict technique is used frequently in both simulation and application, however the prevailing concern with this method is that traffic conflicts can often be subjective. Additionally, identifying traffic conflicts in a large network requires an enormous amount of video data reduction (Chin & Quek, 1997). Computer vision techniques that identify vehicles, calculate frame-by-frame trajectories, and if the TTC is below a certain acceptable amount, the event is identified as a conflict. While this is going to consistently call certain types of actions conflicts, it requires a widespread deployment of cameras and software capable of performing this on a large scale in order to capture these (Saunier, Sayed, & Ismail, 2010). Our proposed method does not require additional equipment such as cameras, and is utilizing technology that has already been deployed in many fleets and will continue to be deployed as additional technologies are developed.

4. Data

Data for this study was acquired from the aforementioned SHRP2 Naturalistic Driving Study. The NDS data set contains the same trajectory data elements, collected at the same frequency, 10 Hz. Furthermore, with safety-critical events identified in the data set through analysis by VTTI staff using video data, the NDS provides validated events that are critical to the research.

The event data received consisted of 91 unique incidents, 49 near-crashes and 42 crashes. From here on, the unique incidents will be referred to as "safety-critical events", or sometimes just "events". The events followed a specific, predefined distribution in order to try and encompass most situations that could occur on the road. Table 1 shows the breakdown of the events received by type and speed at the time of event occurrence. The crashes had varying degrees of severity with some being police-reportable and others incurring little to no damage.

Obviously, the specific crash type had to be observed within the study, which did limit the sample size for some of the more severe crash types, however the authors are confident that those tend to be the easier crashes to detect as they experience the exceedingly high accelerations relative to what is normally observed. The "other" rows were requested to be filled out with the less commonly seen types in the NDS. For the near-crashes other referenced accelerating and no reaction near-crashes, while for crashes they included animal strikes and any other type of collision not listed in the predefined distribution. It should be noted that in some scenarios the subject driver was hit while in other cases the subject driver struck another vehicle.

Each measured event was a 30-second time series of kinematic data collected by the Data Acquisition System. The time series received consisted of about 300 observations collected around the time of the event. The series was made up of three portions, one pre-event, one during the event, and one after the event.

Table 1. Distribution of Crash and Near-Crash Events

		Speed				
		< 20 mph	***20-29 mph***	***30-39 mph***	***40-49 mph***	***> 50 mph***
Evasive Maneuv	**Braking**	3	5	4	3	4
	Steering	4	4	4	4	4
	Other	2	1	3	2	2
Crash Type	**Rear End**	3	2	1	0	1
	Sideswipe	1	2	2	0	1
	Angle	2	4	2	2	0
	Run-off the Road	1	2	0	0	1
	Curb Strike	1	2	3	2	1
	Other	0	0	4	1	0

The additional test data included the same data elements for 35 hours of driving during which no known safety-critical events occurred. This data will be referred to as the normal, or baseline, driving set and was used for false positive testing as a way to validate the proposed methodology. Video data was not acquired for either data set but analysts at VTTI watched it and verified there were no safety-critical events before providing it to the authors.

5. The Discrete Fourier Transform

Every time series, $x = \{x_1, x_2 \ldots x_t\}$, can be expressed as a combination of unique circular patterns of varying frequencies, amplitudes, and phases by using a Fourier Transformation. This will result in a function, $X = \{X_1, X_2 \ldots X_f\}$, which is dependent on frequency (f) values instead of time (t). In the case of discrete data, such as time series data sampled at a specific frequency, the Discrete Fourier Transform (DFT) is used to estimate the amplitude, phase, and frequencies. For a time series of length n, the DFT is equation 1 where $j = \sqrt{-1}$.

$$X_f \stackrel{\text{def}}{=} \sum_{t=0}^{n-1} x_t e^{-j2\pi ft/n} \; for\; f = 0,1,2,\ldots,n-1 \tag{1}$$

The "$\stackrel{\text{def}}{=}$" in equation 1 stand for "equals, by definition." The time series can also be reconstructed using equation 2, the inverse DFT.

$$x_t = \frac{1}{N}\sum_{t=0}^{n-1} X_f e^{-j2\pi ft/n} \; for\; f = 0,1,2,\ldots,n-1 \tag{2}$$

By considering Euler's formula in equation 2, equation 1 can be represented as a sum of real and imaginary waves of different frequencies, amplitudes, and phases in equation 3 (Smith, 2007).

$$e^{j\theta} = \cos\theta + j\sin\theta \tag{3}$$

$$X_f \stackrel{\text{def}}{=} \sum_{t=0}^{n-1} x_t \left(\cos\left(-\frac{2\pi ft}{n}\right) + j\sin\left(-\frac{2\pi ft}{n}\right)\right) \; for\; f = 0,1,\ldots,n-1 \tag{4}$$

The transform was used to obtain relationship between amplitude and frequency in subsequences of the time series. For reference, the time domain is used to describe the series of accelerations observed over time while the frequency domain will be used to describe the relationship after the DFT has been executed.

The software, R, was used to perform the DFT using built in functions. The functions use an expanded version of the Cooley-Tukey algorithm (Cooley & Tukey, 1965) presented by Singleton (Singleton, 1969) to quickly estimate the amplitude and phase at each frequency.

Agrawal et al. suggested a method to group similar subsequences by some observed characteristics of the DFT of those subsequences (Agrawal, Faloutsos, & Swami, 1993). A similar approach was used in the present application, but our interest is in the subsequences that do not match with what is normally expected from drivers. In essence, subsequences are being compared to baseline driving and the ones that do not fit will be flagged as possible events.

6. Methodology

To begin with, the longitudinal acceleration time series were broken into 2.5 second subsequences or "windows". This was done to examine local areas of the time series so the exact region of the time series that was a crash could be identified. Additionally, actions that happen far enough apart are unlikely to be related to the current action. The implications of selecting the specific value of 2.5 seconds for the window length will be addressed in the discussion section. If a data point was missing in the time series, it was interpolated linearly, as long multiple consecutive observations were not missing. If there were multiple observations missing, the window was removed from the analysis.

Each window of acceleration time series data was treated as a single observation and placed in a data frame. The DFT was then performed on each window, and the amplitudes at each frequency were then recorded and placed in a new data frame. For windows with no event, the transform generally had amplitudes very close to zero for all frequencies past the first ($f = 0$), but transforms on windows with crashes tended to have one or more frequencies with relatively high amplitudes. Figure 1a shows a selection of windows of longitudinal accelerations in the time domain and figure 1b shows the same windows in the frequency domain after undergoing a DFT. The red windows are windows with crashes and the black windows are baseline driving. The transformed windows appear to have a higher amount of separation in the frequency domain compared to the time domain.

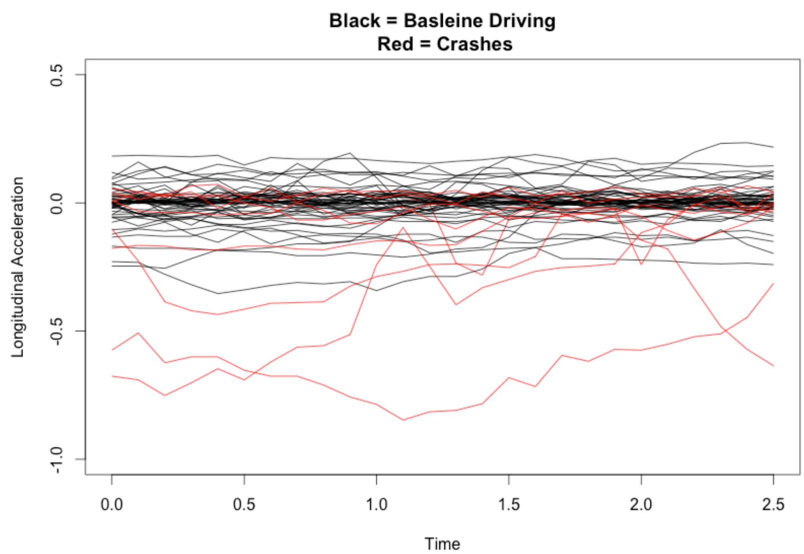

Figure 1a. Time series windows of accelerations during both baseline driving and during crashes

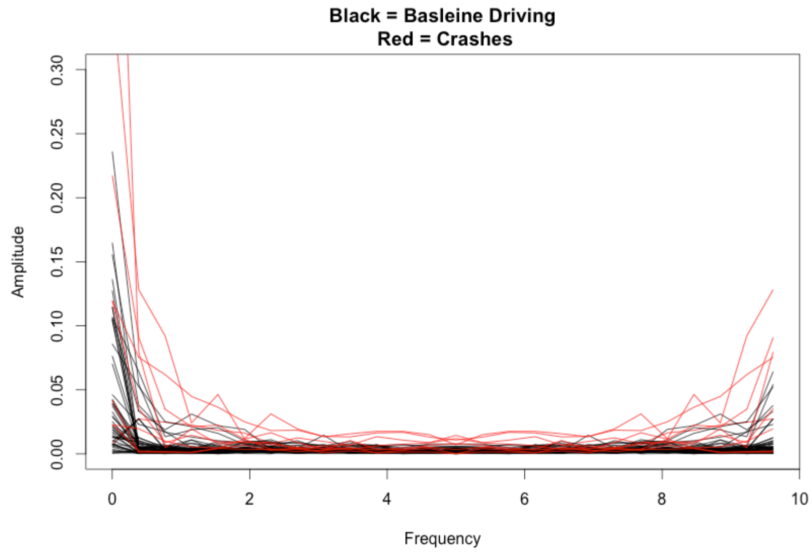

Figure 1b Windows from 1a converted to the frequency domain with the DFT

The final step in data preparation before classification was to characterize the location of observed peaks. To achieve this, the transform was segmented into P equal sections and the area under the windows in the frequency domain was calculated for each of those sections. The trapezoidal area was used to estimate the area using equation 5. Note that the the first value of the transform X, was omitted because amplitudes observed at frequencies equal to 0 are essentially noise. Additionally the total area under the amplitude for frequencies greater than 0, was recorded for each window.

$$AUC_{p,p+1} = \frac{X_p + X_{p+1}}{2} * \frac{10}{P} \quad p > 0 \tag{5}$$

K-means clustering was then performed in order to group the transformed windows by how similar they are, in the hopes that there would be one or more clusters of only events that emerge and another set of separate clusters with no events would also be present. The areas under each section and the total area under curve were used as variables in the K-means algorithm. The K-means clustering algorithm (Leisch, 2008) is an unsupervised learning technique (i.e. the method is grouping the inputs without knowing their classification), performed in the following manner:

1. Define number of clusters (k) and randomly assign each window (i) to one of the k clusters. Calculate the centroid (c_k) of each cluster.
2. Calculate the Euclidean distance (d) between each window and each cluster centroid. Each window is made up of a set of N variables. In this application, the N variables are the areas under each section of the transformed time series in the amplitude-frequency relationship, as calculated by equation 5, as well as the total AUC for all sections.

$$d_{i-c} = \sqrt{\sum_{n=1}^{N} (X_n - X_c)^2}$$

3. Assign windows to cluster with the minimum calculated Euclidean distance.
4. Recalculate center of each cluster to include the newly classified windows.

$$c_{k-n} = \frac{1}{i_{c-k}} \sum_{i} x_{i-n}$$

$$c_k : \{c_{k-1}, c_{k-2}, \dots, c_{k-N}\}$$

5. Repeat steps 2-5 until the algorithm converges and assignments stop changing.

What is essentially happening in the clustering algorithm is the normal baseline driving with no events is consistently being classified into one group while anything else that does not fall into this category is being classified into another cluster. In some cases, this method can be sensitive to the arbitrary starting location so the K-means algorithm was rerun multiple times to test how sensitive the results were to the random starting location.

Clusters were trained on a sample of the baseline driving set (50 randomly selected baselines) and all of the event data. The remaining baseline driving consisted of about 35 hours and those time series were broken into windows and assigned to clusters without re-centering the clusters. This was done to ascertain if specifically defined cluster centers could be used in an application of this methodology without needing to perform the clustering algorithm in an ad-hoc manner, especially since an iterative process like K-means clustering is unlikely to respond well to scaling at the level envisioned, given current computing capabilities.

The time series was broken up into windows as a way to look at smaller happenings within a trip. Since each window had a constant length, breaks between windows were arbitrary and realistically could, and in some cases did, occur midway through a crash event. A successful algorithm should be able to detect at least one window within the duration of a crash event, but it does not matter if every window during the crash was detected. By flagging a single window, the entire crash can be detected, and it does not matter what the classification says is happening in the surrounding windows. It is entirely plausible that an event spanning multiple windows may only have a single window where the impact occurred classified as an event. While technically the windows during the event that go undetected are labeled false negatives, there is no loss of information by the algorithm's failure to indicate that window was positive if a neighboring window is classified as positive. So the only events that were considered false negatives were the crashes where no window spanning the length of the crash event was assigned to the event clusters. Otherwise, if one or more nearby windows indicated an event, it was considered a success.

Similarly, if consecutive windows indicate an event occurred, that is treated as a single event. So if the algorithm indicates a false positive, if consecutive or nearby windows have all been assigned to one of the event clusters, they are treated as a single false positive. How close two windows need to be in order to be considered the same event is unclear as in all cases of false positives spanning multiple windows, the false positives were next to each other. For the reasons stated, the performance will be evaluated on the percent of crash events detected (the recall), and a false positive rate (number of false positives per hour of driving time).

7. Results

The area under the amplitude-frequency relationship was calculated using equation 5 and the following breaks, presented in Table 2. Additionally, total AUC was calculated. Each subsequence had the twelve corresponding observations on Table 2 and the total AUC used in the clustering algorithm.

Table 2 - Breaks for Area Calculations

Variable	Start Frequency (p)	End Frequency (p+1)
X1	0.3846154	1.1538462
X2	1.1538462	1.9230769
X3	1.9230769	2.3076923
X4	2.3076923	3.0769231
X5	3.0769231	3.8461538
X6	3.8461538	4.6153846
X7	4.6153846	5.3846154
X8	5.3846154	6.1538462
X9	6.1538462	6.9230769
X10	6.9230769	7.6923077
X11	7.6923077	8.4615385
X12	8.4615385	9.6153846

For the K-means clustering, the “flexclust” (Leisch & Dimitriadou, 2015) package and R version 3.2.4 was used. In K-means, the number of clusters needs to be defined before the algorithm can run. The starting point was varied between 2 and 6 clusters. 4 clusters were selected to represent distinct patterns in the data, as this had the best balance of classification rates in the training set and low false alarm rates in the baseline test set. The resulting cluster centroids are shown in Figure 2. Figure 3 shows a neighborhood plot of the cluster centers and the data points assigned to each cluster, projected to the first two principal components. The reason 4 clusters worked best is the decision boundary between clusters 2 and 3 was better than other values for cluster starting points.

Table 3 shows a summary of what was assigned to each cluster. The “Subsequences” column shows the total number of subsequences assigned to each cluster. The other four columns show the number of unique events represented in each cluster. This Table helps illustrate the trade-off between classification rates and false alarms. As detection rates increase, the number of false alarms also increase.

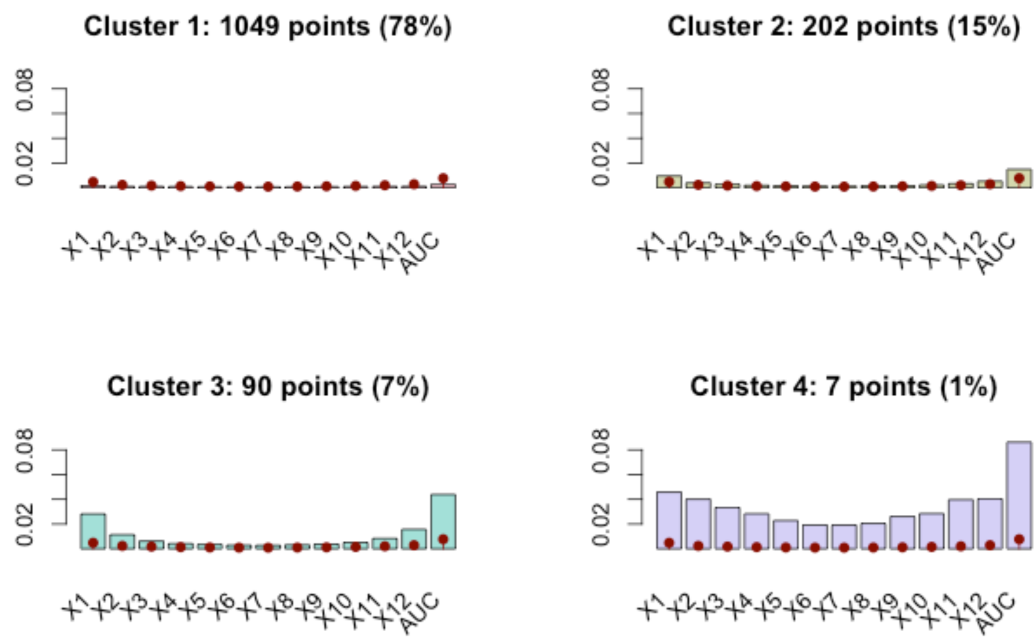

Figure 2 - Cluster Centroids

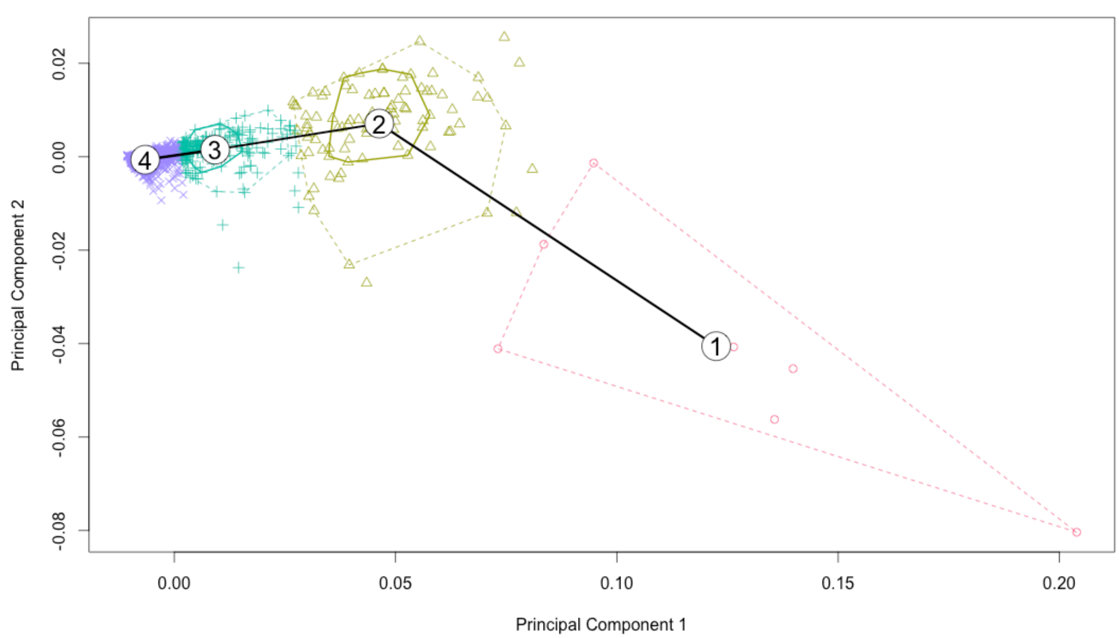

Figure 3 - Neighborhood Plot of K-means Analysis

Table 3 - Clustering Classifications

Cluster	Subsequences	SCEs	Crash	Near-Crash	Baselines
1	1049	91	42	49	50
2	202	76	41	41	9
3	90	69	40	29	0
4	7	7	7	0	0
3&4	97	71	31	40	0
2&3&4	299	90	41	49	9

Using Table 3, clusters 3 & 4 were selected to be the "event clusters". The events were considered flagged if there was at least one subsequence spanning a portion of the event was placed in one of the two event clusters. An event was considered missed (false negative) if none of the subsequences spanning the event were placed in either of the event clusters. A false positive was any baseline subsequence, or group of consecutive baseline sequences, placed into the event cluster. The additional recall gained by including cluster two does not outweigh the addition of 9 false alarms, especially considering baseline driving is generated at a considerably higher rate than SCEs. This will be discussed further in section 8.

The centroids of these clusters were recorded and an additional 35 hours of baseline driving was used to get a realistic sense of this algorithm's false alarm rate in application. During the baseline driving, a total of 13 false alarms were identified. This translates to about 1 false positive per 2.7 hours.

Table 4. Event Cluster Breakdown

Clusters	**Number of Data Sets**	**Number of Unique Events**	**Flagged by Algorithm**
Event Total	91 Epochs	91	71
Crash	42 Epochs	42	31
Near-Crash	49 Epochs	49	40
Baseline	50 Epochs	0	0
Baseline Test	35 Hours	0	13

False positives can be explained by a variety of possibilities including:

- Certain drivers behave more aggressively and the algorithm needs to take that into account when making a prediction.
- The equipment recording certain drivers' actions may have been calibrated differently, located poorly within the vehicle, or malfunctioning.
- Different vehicles are prone to different dynamics based on various factors such as the vehicle's weight, tire quality, tire pressure, road conditions, etc.
- Site characteristics like poor pavement quality could be responsible for the unusual action
- Clusters may not be using a set of ideal baseline samples, though they were randomly sampled for each run and the results only changed minimally.

There is also a possibility that some of the false positives are actually near-crashes that were not categorized by the NDS study, and since the authors do not have access to video to review what was happening, that will remain unknown. All of the false alarms were assigned to the same event cluster, but the majority of events also were classified in that cluster as well.

The code was run 100 times with different random seeds to determine how sensitive the methodology was to the random starting point. The size of each cluster was the same for all 100 runs showing that the method is not sensitive to random starting point.

8. Discussion

The results using this proposed algorithm to detect safety-critical events were markedly better than anything found in the literature review while using a reasonably complete sample of 91 total crash and near-crash events. Findings indicate that roughly 78% of all safety-critical events can be identified with a high frequency time series of longitudinal acceleration data alone.

As discussed in the introduction, the authors envision a variety of possible applications for a methodology like the one presented. Those applications include:

- Allowing infrastructure providers to identify SCEs in connected vehicle environments and evaluate road network safety
- Allowing taxis and shared ride service providers to monitor their drivers and ensure they provide safe rides to customers
- Allowing insurance companies to monitor their customers' driving tendencies and better evaluate risk
- Allowing agencies to monitor fleet vehicles (e.g. buses, snow plows, etc.) for both driver performance and tort liability claims
- Allowing transportation management agencies to monitor traffic and provide emergency response when necessary
- Providing real-time alerts to emergency response services in connected vehicle environments
- Identifying events in large-scale naturalistic driving studies

Each potential application will likely have slightly different inputs depending on the purpose of the application, sensor quality, allowable false positive rate, vehicle type, and numerous other factors. This work was done with high quality sensors and for light vehicles driven by the public and is applicable as presented in connected vehicle environments according to current standards as well as large-scale naturalistic driving studies.

There are a variety of ways that longitudinal accelerations can be collected in vehicles, including dedicated sensors like accelerometers or through cellular technology and GPS data. The vehicles used in this study were equipped with high-quality sensors using an accelerometer, but it is unclear how well this carries over using accelerations derived from GPS technology in cell phones. It is possible that the cluster centers may look different and there may be more false positives as errors in GPS readings can lead to larger accelerations since they are derived from positions and that positional error will carry over.

Allowable false positive rate and definition of an SCE is also going to determine how usable this algorithm is in application and what some of the inputs need to be for it to be successful. While the false positive rate is low, baseline driving gets generated at a much higher rate than SCEs. As a result, there are still likely to be some false positives due to exposure despite the small percentage of baselines that get flagged, although their significance is application dependent. While false positives may be completely unacceptable in some situations, they also likely still indicate an unexpected driving action such as hitting a pot-hole or approaching the end of a queue in congestion, which could be useful to document.

The exact methodology presented in this paper is only applicable for finding SCEs in light vehicles with accelerations collected at a frequency of 10 Hz. The exact inputs, like the 2.5 second window length and number of pre-defined clusters were selected because, of the values tested, the results were the best on a consistent basis when varying other variables.

The general process using a DFT and K-means clustering could work, but likely requires different window lengths, data processing needs, numbers of clusters, resulting cluster centers, and error rates and therefore will need to be altered accordingly. Thus experimentation was performed on the number of predefined clusters and the window length used in order to assess sensitivity. Of the ones testes, the values that worked best in this application with the stated goals in mind were selected. Specifying crash type and severity while considering location is a clear next step to this work, as is examining other kinematic data like speed and lateral acceleration.

9. Conclusion

Presented in this paper was a unique approach to detecting safety-critical events using vehicles' longitudinal acceleration. It used the Discrete Fourier Transform in combination with K-means clustering to flag windows that were likely to be crashes or near-crashes. The algorithm was able to detect almost 78% of crashes and near-crashes that were acquired for this study, while generating about 1 false positive every 2.7 hours. This algorithm had excellent performance in comparison to what is currently being used in application, and can also be easily expanded upon as other advances are made and other data types are collected.

Further points of emphasis in future studies on this subject will be to include additional variables frequently collected, such as lateral acceleration to improve the recall and the false positive rate. The most glaring issue with this methodology is the inability to differentiate between crashes and near-crashes as well as the inability to determine crash type for events that were crashes. One additional check to differentiate between crashes and near-crashes could be a heuristic to check if the driver continues to drive. Other than that, including different variables, and placing additional emphasis on the relationship between location of peaks in the frequency domain and the type or severity of event is a logical next step to improve this methodology's performance. Some examples of additional sensors that could provide more information include noise sensors and multi-directional radar. Additional focus on driver-specific or site-specific trajectories could also improve classification rates and reduce false positives.

References

Agrawal, R., Faloutsos, C., & Swami, A. (1993). Efficient similarity search in sequence databases. *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms (FODO '93)*, *8958546*, 69–84. http://doi.org/10.1007/3-540-57301-1_5

Amundsen, F., & Hyden, C. (1977). Proceeding of First Workshop on Traffic Conflicts. In *First Workshop on Traffic Conflicts.* Oslo, Norway.

Chin, H. C., & Quek, S. T. (1997). Measurement of traffic conflicts. *Safety Science*, *26*(3), 169–185. http://doi.org/10.1016/S0925-7535(97)00041-6

Cooley, J. W., & Tukey, J. W. (1965). An Algorithm for the Machine Computation of the Complex Fourier Series. *Mathematics of Computation*, *19*, 297. http://doi.org/10.2307/2003354

Dingus, T., Klauer, S. G., Neale, V. L., Petersen, A. D., Lee, S. E., Sudweeks, J., … Knipling, R. R. (2006). *The 100-Car Naturalistic Driving Study Phase II – Results of the 100-Car Field Experiment*.

Engstrom, J., & Victor, T. (2005). System and Method for real-time recognition of driving patterns. United States.

Guo, F., Klauer, S. G., McGill, M. T., & Dingus, T. A. (2010). *Evaluating the Relationship Between Near-Crashes and Crashes: Can Near-Crashes Serve as a Surrogate Safety Metric for Crashes? Contract No. DOT HS* (Vol. 811).

Halkias, J., & Colyar, J. (2006). Next Generation SIMulation Factsheet. Retrieved May 18, 2016, from http://www.fhwa.dot.gov/publications/research/operations/its/06135/index.cfm

Harding, J., Powel, G., Yoon, R., Fikentscher, J., Doyle, C., Sade, D., … Wang, J. (2014). *Vehicle-to-vehicle communications: Readiness of V2V technology for application. (Report No. DOT HS 812 014)*. Washington,.

Kluger, R., & Smith, B. L. (2014). Pattern Matching Longitudinal Time Series Data To Identify Crashes in Naturalistic Driving Data. In *ITS World Congress* (pp. 3329–3339). Detroit, Michigan.

Leisch, F. (2008). A Toolbox for K-Centroids Cluster Analysis, (1999), 1–22.

Leisch, F., & Dimitriadou, E. (2015). Package " flexclust ."

Mcdonald, A. D., Lee, J. D., Aksan, N. S., Dawson, J. D., Tippin, J., & Rizzo, M. (2013). The Language of Driving: Advantages and Applications of Symbolic Data Reduction for Analysis of Naturalistic Driving Data. *Transportation Research Record*, *2392*, 22–30. http://doi.org/10.3141/2392-03.The

Saunier, N., Sayed, T., & Ismail, K. (2010). Large-Scale Automated Analysis of Vehicle Interactions and Collisions. *Transportation Research Record: Journal of the Transportation Research Board*, *2147*, 42–50. http://doi.org/10.3141/2147-06

Sayer, J., LeBlanc, D., Bogard, S., Hagan, M., Sardar, H., Buonarosa, M. L., & Barnes, M. (2008). *Integrated Vehicle-Based Safety Systems Field Operational Test Plan*. Ann Arbor. Retrieved from http://deepblue.lib.umich.edu/bitstream/2027.42/62108/1/102281.pdf

Singleton, R. (1969). An algorithm for computing the mixed radix fast Fourier transform. *IEEE Transactions on Audio and Electroacoustics*, *17*(2), 93–103. http://doi.org/10.1109/TAU.1969.1162042

Smith, J. O. (2007). Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications, Second Edition. Retrieved from https://ccrma.stanford.edu/~jos/mdft/

Talebpour, A., Mahmassani, H., Mete, F., & Hamdar, S. (2014). Near-Crash Identification in a Connected Vehicle Environment. *Transportation Research Record: Journal of the Transportation Research Board*, *2424*, 20–28. http://doi.org/10.3141/2424-03

Virgina Tech Transportation Institute. (2013). The SHRP2 Naturalistic Driving Study.

Wu, K. F., & Jovanis, P. P. (2013). Defining and screening crash surrogate events using naturalistic driving data. *Accident Analysis and Prevention*, *61*, 10–22. http://doi.org/10.1016/j.aap.2012.10.004

Wu, K., & Thor, C. P. (2015). Method for Using Naturalistic Driving Study Data to Analyze Rear-end Crash Sequences. *Transportation Research Record: Journal of the Transportation Research Board*, *2518*. http://doi.org/http://dx.doi.org/10.3141/2518-04

Wu, K.-F., & Jovanis, P. P. (2013). Screening Naturalistic Driving Study Data for Safety-Critical Events. *Transportation Research Record: Journal of the Transportation Research Board*, (2386), 137–146. http://doi.org/10.3141/2386-16

RESEARCH CONTRIBUTION

This compendium of work has a few key research contributions. First, it includes a comprehensive discussion of the findings related to how connected vehicle technology can benefit highway safety analyses. These findings are interesting and should be treated as a starting point for future methods and ideas. The true benefit to using connected vehicles to carry out safety analyses remains to be seen, and requires more data than is currently available. Identifying hot spots with any level of confidence or accuracy using connected vehicles is not likely feasible at the current level of market penetration, even in pilot deployments like the safety pilot. However, described were some considerations as connected vehicle technology grows and changes that are better to be examined now rather than after the technology is deployed. This includes how this process fits within the connected vehicle standard to ensure it is feasible once the technology is widely deployed. Falling into that realm is: ensuring the algorithms will operate properly under the security and de-identification measures taken by the standard message set to ensure privacy, ensuring the vehicles are collecting the correct data elements at the highest quality available while utilizing everything to which the vehicle has access, ensuring that the message set standard has a mechanism in place to deliver event flags that may be produced by vehicle OBEs, and finally ensuring that, as research progresses and algorithms are developed, there is a means to test advancements and improvements to algorithms that do detect events. With these measures in place, there should be no issues evaluating how to identify hot spots once technology reaches a critical level of market penetration.

The second contribution is the exploration of how crashes and near-crashes can be detected in connected vehicle environments. All of the hypothesized benefits of using connected vehicles for hot spot identification hinge on the ability to successfully detect crash and crash-surrogate events. As a result, a major focus of this research was modeling crashes and near-crashes in order to describe them in terms of connected vehicle data elements. Three creative methods were proposed as possible approaches to identifying these types of events, including a pattern matching approach, a speed prediction time series based approach, and a discrete Fourier transform approach. Each of them have benefits and drawbacks in terms of both complexity and accuracy, but serve as excellent starting points for further research and the lessons learned are applicable and should be considered as additional models are proposed.

It is also important to consider that the event detection work is not just applicable to connected environments. The benefits to being able to identify crash and surrogate events automatically from an incoming data stream can easily be seen throughout the transportation community. Infrastructure providers can use this to monitor fleet vehicles and contractors to ensure proper driving behavior while vehicles are in-use for government purposes. Similarly, this can be done by insurance providers to try and ascertain driver risk, rental car agencies for audits, and taxi-like services to ensure drivers meet a certain safety benchmark. Additionally, in connected environments, assuming the model accuracy can be improved, models can be developed to automatically alert emergency response units and traffic management agencies so the incident can be managed as quickly as possible.

FURTHER RESEARCH

EVENT DETECTION

For event detection, there are numerous directions further research can take. In this dissertation a few different general models were proposed to identify generic events regardless of type or severity while only using longitudinal acceleration and speed as key indicators. This was partly an artifact of the fact that the data sets were still on the small side which made it difficult to create specific models with any amount of confidence. In reality, it is likely that regional or local models would be required anyways, to account for regional dependencies that may be present. Local driver characteristics, tendencies by infrastructure providers, and regional characteristics like weather effects can all impact how a prediction model performs. This is similar to the way current crash prediction models are used in practice today. There are still a variety of questions that must be answered before this is viable for real-life applications.

First, it is clear that trajectory data needs to be reliably collected or processed. It is preferable to use the vehicle's built-in network rather than GPS-based trajectories, especially if event detection is the goal. If detecting other more common actions is the goal, having consistently high quality data becomes less critical as 1) it is going to be relatively easy to identify outliers, and 2) removing the outliers shouldn't hugely impact sample size. When this becomes an issue is for event detection where events are the outliers. Observed outliers could just as easily be an event as they could be a GPS error or other bad data that was collected erroneously if the data collection mechanism is not trustworthy.

Second, a key area for additional improvement is to utilize other BSM elements that are currently being collected. This is an obvious next step that may be able to help improve accuracy and discern crash types. Some of those data elements include lateral acceleration, pitch, roll, and yaw. Additionally, there are some other databases that could be merged with the BSM to improve understanding of what is occurring. For example, information on the geometric design of each area can be used to understand if certain observed actions are expected or even legal in the vicinity.

Third, in situations where vehicles do have radar and other equipment that may improve event detection capabilities, there is no reason for it to not be utilized. For connected vehicles, the current standard message set is restrictive. While constantly projecting radar data may be too taxing on the system, a generic BSMP2 trigger could be added to allow the vehicle network to communicate likely near-crashes to the OBE so that it can, in turn, be projected to the RSE. This would allow vehicle manufacturers to develop their own triggers using their vehicles' full repertoire of equipment while maintaining corporate privacy. Similarly, non-connected vehicle settings may also be able to utilize radar, sound sensors, and other equipment that may be installed. In general, more equipment will likely increase the accuracy of the flags and many vehicles are being manufactured with radar for other applications so it is really a matter of effectively utilizing the technology for additional purposes.

For pattern recognition approaches, establishing a known and controlled ground truth would likely be highly beneficial to prediction accuracy. Particularly for safe driving, having a bank of time series patterns for each action in different types of vehicles and driving conditions would allow for nuanced differences between conditions that indicate a safety-critical event or other conditions that are observed regularly in driving to be identified. For example, in the pattern recognition application, the approach was to identify events that did not fit into known patterns. However, one of the observed false positives turned out to be a vehicle going over a speed bump. Knowing what a speed bump looks like in terms of vehicle dynamic time series data and testing it against the data as it gets generated could prevent false positives due to this specific occurrence.

There could be a couple approaches to developing this bank of known actions. Controlled experimental field tests where drivers go out on a test track and collect trajectory data would be a large undertaking but would also likely provide the best ground truth data for matching to observed data. In these controlled experiments, inputs such as vehicle type or weight, road surface conditions, approach speed, reaction time, etc. can be varied to collect a complete bank of observable vehicle actions. While this is clearly the more expensive option to pursue, it could have other research and application benefits that may justify endeavor.

Perhaps the more economic approach would be to wait for connected vehicle technology to be deployed and mine a more complete dataset for commonly occurring patterns. Restraining this to specific sites and RSE would control for specific local characteristics (i.e. if a speed bump is present within the range of the RSE, the trajectories at that specific location all reflect its presence).

Additional heuristic indicators can also be used to differentiate between a crash or near-crash. For example, if the vehicle keeps driving after the flagged events, it may lower the likelihood that the flagged event was a crash. Data from other databases, such as operations data collected by loops and Bluetooth sensors, may also be able to verify or invalidate events shortly after a flag is detected.

SAFETY HOT SPOT IDENTIFICATION

This work is a preliminary look at how the evaluation of transportation safety can be carried out. Current transportation safety evaluation research and applications are heavily entrenched (in the best case scenario) in following the HSM's process of developing crash prediction models using police-reported crash data, and using those models to determine if the location is safe or not. While historically this has been the only data set readily available on a large scale, technological advancements in both vehicle technology and cellular technology make it potentially beneficial for the entire process of safety evaluation to shift to utilizing the new data sources.

In terms of further research for hot spot identification, a few necessary studies must take place before any implementation can occur. Specifically, the entire EB-method in centered around correcting and adjusting crash predictions to account for yearly variation and regression to the mean. At a minimum, the method must be adjusted for yearly variations in safety-critical events (rather than just crashes) and event prediction functions would need to replace the currently used safety performance functions (crash prediction models). Additionally, this method was developed specifically to account for regression to the mean bias present in yearly crash counts, a bias that is significant due to the relative scarcity of crash events. It is unclear if this issue will persist if safety-critical events, which occur more frequently than police reported crashes, can be accurately detected on a large scale.

Much of this will be impacted by the exact relationship between safety-critical events and crash risk, the primary metric that needs to be minimized to improve safety. Right now, observed crashes are used to estimate crash risk and years of study have helped researchers and practitioners understand the relationship between observed crash counts and crash risk. One of the ways the known bias in the data can be counteracted is to wait for a proper sample size, something that usually takes three years. Studying the relationship between safety-critical events and crash risk, rather than just police reported crashes, may lead to findings where crash risk can be estimated quickly helping to solve the problems quicker and prevent unnecessary exposure to high-risk locations. But, the working assumption with this concept is that safety-critical events are indeed a good surrogate for crashes. This means that locations with a high crash risk also have a high safety-critical event risk as a byproduct of that crash risk.

It is possible that the analysis methods change altogether. If regression to the mean is no longer an issue that needs to be corrected, it is entirely possible that large-scale spatial analysis techniques could be preferable to the EB method. For example, spatial scan statistics can be used to identify if the rate of event occurrences over a certain space is higher than the rest of the locations. For spatial scan statistics, the null hypothesis is that the rate of event occurrence over a predefined space is the same as every other similar space and it is rejected if the rate is different. A methodology for applying a spatial scan statistic to a relatively uniform region could be as follows:

1. Ensure the AADT of the region is fairly uniform. Exposure is still going to be the leading cause of events and high exposure doesn't necessarily mean the location is "hot" so it is best to test locations independent of exposure.
2. Flag events that occur in the region using event detection algorithms and map them within the selected region.
3. Define the event rate at each cluster *i,* as θ_i. The null hypothesis is that $\theta_i = \theta$, where θ is the population event rate, which is unknown. The clusters are circles of varying sizes being moved across a spatial area.
4. A X^2-distributed statistic can then be calculated to test the hypothesis using traditional hypothesis testing concepts.

All clusters that are significantly different can be considered hot spots as the risk is statistically higher in those specific clusters. Further measures could perhaps be taken as well to constrain the scan statistic to a grid or network, rather than using a circle. After this, potentially "hot" locations would be sent into a diagnosis phase where the root cause of the problem would be identified through further study, just like the one outline in the HSM. This could, in theory, make the network screening portion of the process, easier to automate as it requires less manual input then the EB method.

Validation will be particularly challenging for studies like this and will likely be time consuming. The primary metric needs to be improvement in targeting high-crash locations over the EB method. The ideal way to handle validation of new methods would be to forecast high crash locations using both methods and simply wait to see which ones turn out to be more accurate in the long run. Obviously, certain factors make this very difficult to accomplish in a real-life scenario. For example, any changes in travel patterns caused by construction at the site, construction on parallel routes, or changes to route or mode choice patterns will impact how many crashes a site experiences, in addition to the random element already observed in crash data. Determining the best approach to validating all hot spot studies, including the ones described in this study utilizing connected vehicle data, requires attention and will be up to researchers in the safety community.

CONCLUDING REMARKS

The compendium addressed two research ideas to varying extents:

1. Can crash and crash-surrogate events be identified using vehicle trajectory data that would be collected in connected vehicle environments?
2. Can connected vehicle technology be used by infrastructure providers to identify hot spots?

Both questions, and a variety of sub-questions that arose from examination of those two primary topics are explored. Both topics are expected to be of critical importance to the transportation community once connected vehicle technology is implemented on a wide scale.

Findings were preliminary, though appeared positive. First, a general research paper outlining how and why connected vehicle technology could be beneficial to hot spot identification was presented. Next a pattern matching approach was described to identify safety-critical events on a small naturalistic driving study data set. This same data set with some added baseline driving was used to create a speed prediction algorithm for event detection. Finally, a clustering approach of time series segments using a discrete Fourier transform was tested on a significantly larger and more robust data set with apparent success in terms of both event detection capabilities as well as avoiding false alarms. Additionally, lessons learned and further research topics were described