Infrastructure Safety Assessment in a Connected Vehicle Environment

CVI
UTC

CONNECTED VEHICLE/INFRASTRUCTURE UNIVERSITY TRANSPORTATION CENTER (CVI-UTC)

MORGAN STATE UNIVERSITY
VirginiaTech Transportation Institute
UNIVERSITY of VIRGINIA
VDOT

Infrastructure Safety Assessment in a Connected Vehicle Environment

Prepared for the Research and Innovative Technology Administration (RITA);
U.S. Department of Transportation (US DOT)

Grant Project Title:

Advanced Operations Focused on Connected Vehicles/Infrastructure (CVI-UTC)

Consortium Members:

Virginia Tech Transportation Institute (VTTI),
University of Virginia Center for Transportation Studies (UVA CTS),
and Morgan State University (MSU).

Program Director:
Dr. Thomas Dingus
Program Director, Connected Vehicle/Infrastructure University Transportation Center
Director, Virginia Tech Transportation Institute
Professor, Department of Biomedical Engineering and Mechanics at Virginia Tech
tdingus@vtti.vt.edu
(540) 231–1501

Report Authors:
Brian L. Smith, PE, PhD
Director, University of Virginia Center for Transportation Studies
Professor and Chair, Department of Civil and Environmental Engineering
briansmith@virginia.edu
(434) 243–8585

Robert Kluger
Graduate Research Assistant,
Virginia Tech Transportation Institute
Rmk7su@virginia.edu
(540) 231-1500

Hyungjun Park, PhD
Senior Scientist
Virginia Tech Transportation Institute
hpark@email.virginia.edu

DUNS: 0031370150000 EIN: 54-6001805

Grant Funding Period: January 2012 – July 2016
Final Research Reports

December 15, 2015

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Connected Vehicle/Infrastructure UTC

The mission statement of the Connected Vehicle/Infrastructure University Transportation Center (CVI-UTC) is to conduct research that will advance surface transportation through the application of innovative research and using connected-vehicle and infrastructure technologies to improve safety, state of good repair, economic competitiveness, livable communities, and environmental sustainability.

The goals of the Connected Vehicle/Infrastructure University Transportation Center (CVI-UTC) are:

- Increased understanding and awareness of transportation issues
- Improved body of knowledge
- Improved processes, techniques and skills in addressing transportation issues
- Enlarged pool of trained transportation professionals
- Greater adoption of new technology

Abstract

The goal of the Infrastructure Safety Assessment in a Connected Vehicle (CV) Environment project was to develop a method to identify infrastructure safety "hot spots" using CV data. Using these basic safety messages to detect hot spots may allow for quicker discovery than traditional methods, such as police-reported crashes. The basic safety message may be able to detect events that police normally cannot obtain, including unreported crashes and near-crashes.

The project successfully explored some models and algorithms to detect crashes and near-crashes and also designed a methodology to apply to hot spot identification. With the data available, conclusive results were not achieved; however, the models showed some potential. Three techniques were tested to predict crashes using vehicles' kinematic data. To predict where a crash was occurring, multivariate adaptive regression splines, classification and regression trees, and a novel pattern matching approach were all tested. The models were able to identify the majority of 13 known crashes with different amounts of false positives. The pattern matching approach outperformed a simple acceleration threshold by identifying nearly 70% of crashes in a crash-only test set and 74% of near-crashes in a near-crash only test set. On the training set, it was able to identify more crashes than the thresholds without increasing the number of false positives observed. Based on the work described in this report, the CVI-UTC is fully prepared to apply the methodology to data collected on the field test bed.

Acknowledgements

The authors would like to thank the Connected Vehicle/Infrastructure University Transportation Center for funding this project. Additional thanks to Michael D. Fontaine, John Miller, Brian Byungkyu Park, and Matthew Gerber for providing feedback on various aspects on the paper.

Table of Contents

Background 1
Objectives 3
Literature Review 3
Method 11
Data 11
Models 16
Pattern Matching Algorithm 18
Classification and Regression Trees 20
Multiple Adaptive Regression Splines 22
Results 23
Conclusions and Recommendations 27
References 29

List of Figures

Figure 1. Map of crashes showing potential hot spots in Ann Arbor, Michigan. 1
Figure 2. Roadway Safety Management Process [1]. 2
Figure 3. Risk of crash events [1]. 4
Figure 4. Range rate model boundaries. [2] 7
Figure 5. Surrogate detection model. [10] 8
Figure 6. Sensitivity analysis of thresholds for conflict detection [10]. 9
Figure 7. Method 1 as recommended by Talebpour et al. [12]. 10
Figure 8. Two time series of longitudinal accelerations culminating in crashes. 12
Figure 9. Front video NDS data. 13
Figure 10. Rear video NDS data. 13
Figure 11. Trip length summary. 14
Figure 12. Time series of longitudinal accelerations in the normal driving data set. 15
Figure 13. Longitudinal and lateral accelerations from a crash event. 16
Figure 14. Sample time series of longitudinal accelerations, rear-end crash at time 170 seconds. 17
Figure 15. A sample of acceleration profiles during crashes. 18
Figure 16. Selected baselines. 19
Figure 17. Crash comparison (black dots) with baseline readings. 20
Figure 18. Decision trees excel at separating data organized in this manner. 21
Figure 19. Entropy tree. 22
Figure 20. ROC curve for CART tree. 25

List of Tables

Table 1. VTTI Event Flags Indicating a Crash or Near-crash 6
Table 2. Detection and Error Rate of Flags 7
Table 3. Crash Type Distribution (Left); Crash Conditions (Right) 14
Table 4. Pattern Matching Training Data Detection Rate 23
Table 5. Pattern Matching Results Online Data Set 23
Table 6. NDS Threshold Score Table Results 24
Table 7. CART Score Table 24
Table 8. CART Test on Normal Driving vs. NDS Triggers 25
Table 9. MARS Model Results on Training Data 26
Table 10. Results of MARS Model on Normal Driving Compared to NDS Threshold 26

Background

Transportation agencies devote significant resources to analyzing crash data collected by responding police agencies to identify “hot spots”—locations that experience a larger than average number of crashes. An example of a hot spots map created based on crash data from Ann Arbor, Michigan, is presented in Figure 1.

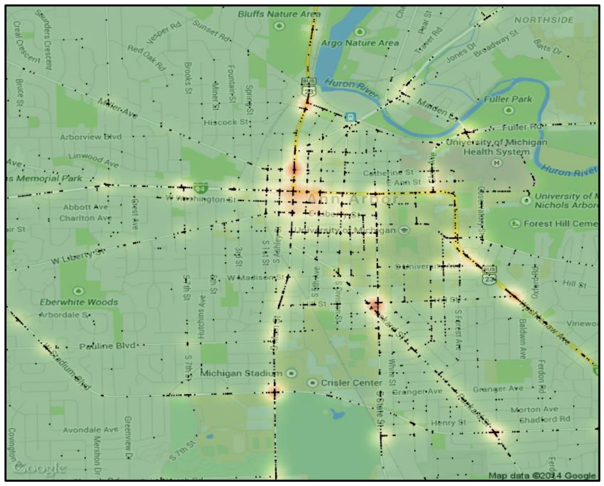

Figure 1. Map of crashes showing potential hot spots in Ann Arbor, Michigan.

In many cases, upon identification of a hot spot, a field investigation points to a particular feature of the infrastructure that contributes to the crashes, which may then be addressed specifically to improve safety. This method, detailed by the Highway Safety Manual’s (HSM’s) Roadway Safety Management Process shown in Figure 2**,** has been effectively used for many years [1]. However, this method also has significant shortcomings; for example, a large number of crashes must accumulate before a hot spot can be identified. In other words, this reactive method requires a number of crashes to occur before corrective action can be taken.

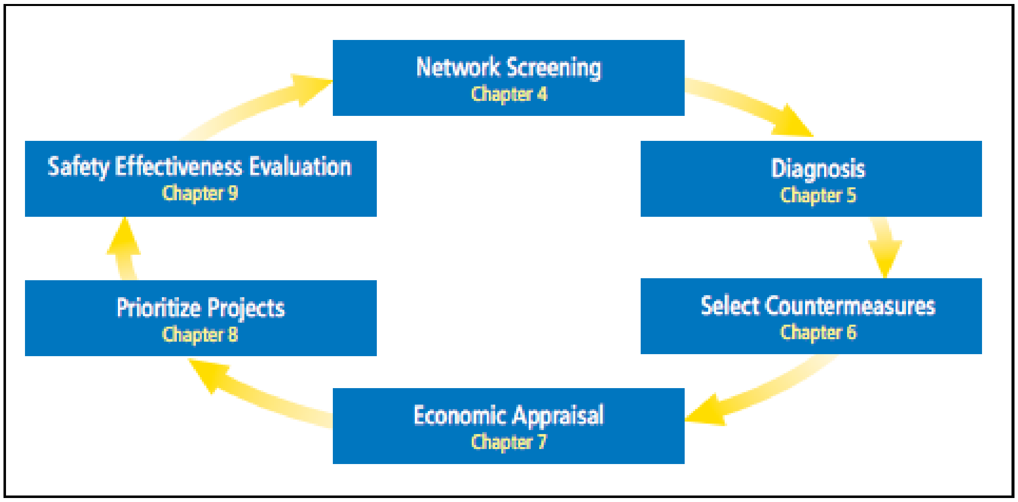

Figure 2. Roadway Safety Management Process [1].

Fortunately, there is a reason that crashes are most often referred to as "accidents." They are infrequent, even at most hot spot locations. Thus, a rather long period of time is required for a statistically significant accumulation of crashes to occur. Furthermore, accurate capture of the locations of crashes has long been a challenge in the transportation community. Even when a hot spot is identified, the exact location of the problem is often difficult to pinpoint. Police reports have been notoriously inaccurate in terms of crash location, although this has improved somewhat with the use of Global Positioning System (GPS) technology. This imprecision demonstrates a need to develop a more proactive way to accurately identify hot spots for those locations that require modifications to the transportation infrastructure in order to improve safety.

The premise of this project dictates that, for every actual crash, there also exist numerous "near-crashes" where drivers take last-second, extreme evasive action (such as swerving or rapid deceleration) to avoid a crash. These near-crashes may be as significant as actual crashes in terms of indicating potential safety problems, yet the challenge lies in identifying and compiling these near-crashes. Near-crashes have never been formally reported by the police or other agencies. However, with vehicles in a connected vehicle (CV) environment, basic vehicular operation data are available from the vehicle data bus. If significant evasive maneuvers can be extracted from this data, along with the corresponding GPS locations, a transportation agency can analyze near-crash data for hot spot identification. Using CVs instead of police reports offers the potential for a faster and more accurate network screening step (as shown in Figure 2), which, in turn, speeds up the entire Roadway Safety Management Process.

This project analyzed data from past field tests to develop prototype algorithms for hot spot identification from vehicular operations data. These algorithms were then demonstrated and tested on the Connected Vehicle/Infrastructure University Transportation Center (CVI-UTC)

Northern Virginia Connected Vehicle test bed to determine if they successfully extracted near-crash maneuvers. The data were then analyzed to determine if hot spots could be identified. Finally, these hot spots were examined in terms of traditional crash data to determine if a correlation existed, indicating the potential of this approach.

Objectives

The following steps were originally proposed for this project:

1. Literature Review
2. Selection of Crash/Near-Crash Identification Criteria
3. Development of Methodology to Apply Criteria to Basic Safety Message (BSM) Data
4. Application and Validation of Proposed Methodology in the CVI-UTC Northern Virginia Connected Vehicle Test Bed

The literature review revealed that only a limited amount of effort has been applied to defining a crash or near-crash in terms of kinematic data elements (acceleration, speed, yaw, etc.). This development turned Step 2 into a task involving the development of models to describe crash and near-crash events. Additionally, lack of available test bed data in the timespan of this project made Step 4 infeasible given the large data needs of this method. As a result, this report considers the following aspects of the original methodology:

1. Literature Review Conclusions
2. Development of Crash/Near-Crash Identification Algorithms
3. Suggested Approach and Requirements for Application and Testing of this Methodology

Literature Review

The literature review focused on background information about near-crashes and kinematic-based definitions for crashes. Neither of these two topics has received extensive, dedicated research for some obvious reasons. The need to detect crashes using kinematic data is fairly recent since this type of data on a large scale was never available to researchers until the 100-Car Naturalistic Driving Study (NDS) [2]. Near-crashes have traditionally been extremely difficult to study since in the past they were neither reported nor recorded. Thus, they are poorly defined occurrences that are subject to the judgments and biases of their analysts. Although crashes are actually relatively rare events even in hot spots, researchers have proposed that near-crashes, which occur more frequently, can help to identify hot spots. The HSM provides an excellent illustration of the continuum of scenarios leading up to a crash (see Figure 3**).**

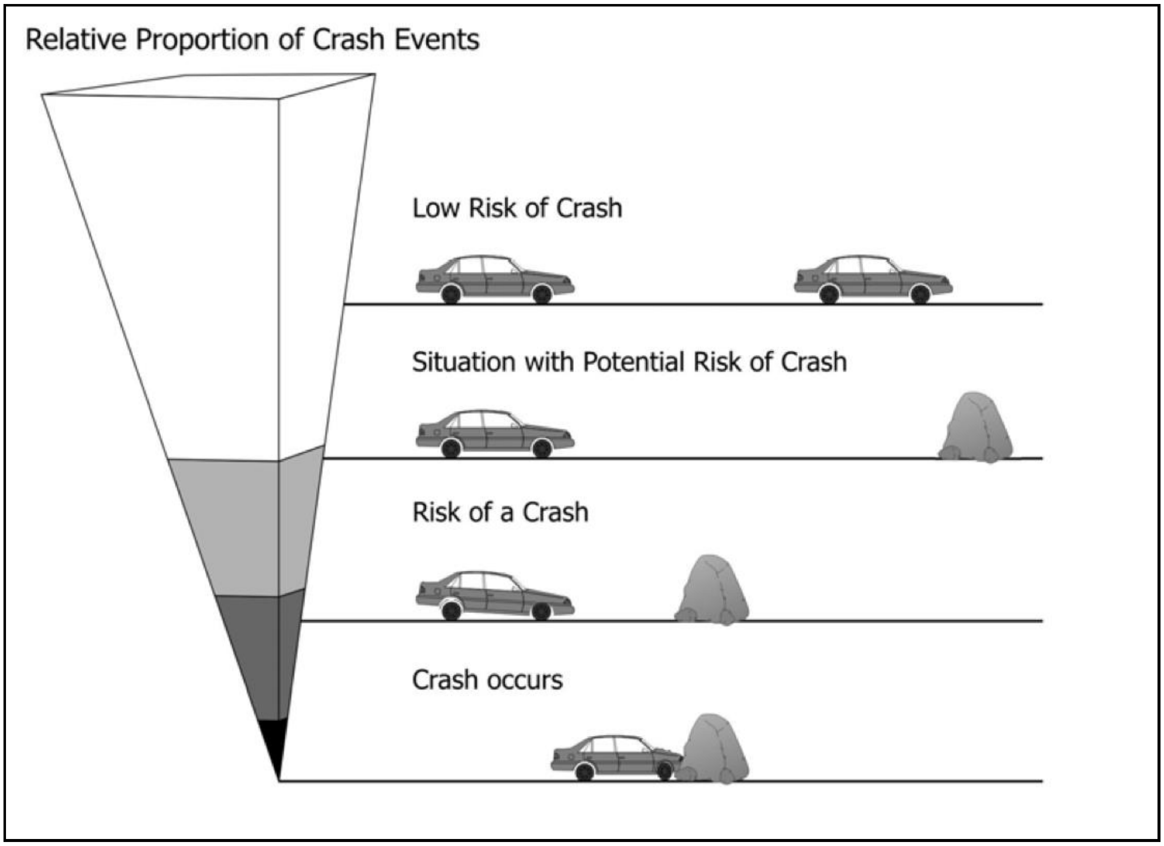

Figure 3. Risk of crash events [1].

While many safety conflicts may occur over the course of a trip, an actual crash is frequently avoided, implying that only a small percentage of events with a conflict truly result in a crash. For every crash that occurs, multiple near-crashes occur where a driver is able to avoid a crash, justifying the belief that hot spots can be detected more quickly if near-crashes are also considered. In reality, the roadway is the only contributing factor in only 3% of crashes (e.g., wet pavement, polished aggregate, steep downgrade, poorly coordinated signal systems, etc.), but an estimated 31% of crashes combine a vehicle or human factor with the roadway factor. [1]

A prevailing issue for the tracking and analysis of near-crashes is there are no specific set of charateristics that define a near crash, which has created a difficulty in tracking these events. This is because until recently, tracking these events has never been feasible, but a standard, exact definition of what events constitute a near-crash needs to be developed. In 2010, an article produced by Virginia Tech Transportation Institute (VTTI) researchers defined near-crashes as [3]:

> 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.

This definition indicates that, in near-crash situations, one or both drivers took action to avoid a crash. This is certainly a reasonable situation to define as a near-crash, yet it does not encompass events where drivers failed to take action but did not crash due to chance. Additionally, this study did not provide a concrete way to differentiate between a near-crash and a less severe event, nor did it account for the subjective nature of a rapid evasive maneuver.

While this definition is clearly not perfect, it provides a reasonable starting point. Klauer et al. [4] acknowledged its subjectivity but took it further by defining a rapid evasive maneuver as steering, braking, accelerating, or a combination of control inputs approaching a vehicle's limit. Both of these definitions were created as part of the 100-Car NDS. It is important to note that first, analysts reviewed video footage of events that were flagged by kinematic triggers and flagged a set of events, then analysts reviewed video footage from the trips around the event flags. Given the lack of opportunity for researchers and officials to study near-crashes outside of a specific, small-scale test scenario, it is difficult to evaluate the true relationship between crashes and near-crashes. Guo et. al. [3] reached the following conclusions about using near-crashes as a surrogate safety measure for crashes using the 100-Car NDS:

- No evidence exists to suggest the causal mechanisms for crashes and near-crashes are different.
- A strong frequency relationship exists between crashes and near-crashes.
- Using near-crash data will have biased results, but the direction of the bias is consistent.
- Near-crashes can improve the precision of the estimations.

New York City's police department has also made an attempt to gather information on near-crashes by crowdsourcing crash and near-crash data from witnesses with a project called Crash Stories NYC. Witnesses of a crash or near-crash were encouraged to visit a website with an interactive map and document their experience by completing a survey of date, location, and a first-hand account of the incident. Because this project is crowdsourced, it is highly subjective, especially because no training was implemented to ensure consistency. The program had low participation rates [5].

The detection of crashes as near-crashes specifically using kinematic vehicle data is a relatively new problem, as kinematic vehicle data have not typically been available on a large scale in an uncontrolled setting. This has recently begun to change, thus making the detection of safety-critical events using kinematic vehicle data a much more relevant topic to pursue, especially with the commitment to vehicle-to-vehicle (V2V) technology. Most of the completed work is either threshold based or time-to-collision-(TTC) based. While TTC metrics are likely to contribute to detecting near-crashes in a fully saturated CV environment, using TTC is not feasible in a test bed situation where only a small percentage of the vehicles are equipped with V2V. Still, even those with a TTC input are worth mentioning to include in future work, as a fully saturated CV environment is a real possibility in the near future [6].

Until now, the primary model to detect a near-crash using kinematic data was through the use of a threshold or set of conditions that must be met to flag an event. The 100-Car [1] and Second Strategic Highway Research Program (SHRP 2) [7] NDSs conducted by VTTI probably provide the best means to study near-crashes. In both of these studies, researchers equipped participant vehicles with cameras and data acquisition systems (DASs). These drivers then continued on with their daily driving lives while VTTI researchers collected kinematic and video data in order to study naturalistic driving behaviors in an uncontrolled setting.

The advantage of video data is that anything flagged by a model may be visually verified. Most studies conducted at VTTI involving crashes or near-crashes used the set of flags shown in Table 1 to indicate these possible events. Table 2 shows the percentage of valid events that were detected by each flag along with the false positive rate of each flag. It is quite apparent that when used alone, the error rate is high, although accuracy was not stated for combinations of thresholds being crossed (i.e., lateral and longitudinal accelerations both crossed).

Table 1. VTTI Event Flags Indicating a Crash or Near-crash

Event Flag	Description
Lateral Acceleration	Lateral motion equal or greater than 0.7g. Acceleration or deceleration equal or greater than 0.6g. Acceleration or deceleration equal or greater than 0.5g coupled with a forward TTC of 4 s or less.
Longitudinal Acceleration	Acceleration or deceleration between 0.4g and 0.5g coupled with a forward TTC of 4 s, and with a corresponding forward range value at the minimum TTC not greater than 100 ft.
Event Button	Activated by the driver pressing a button located by the rearview mirror when an event occurred that the driver deemed critical.
Forward TTC	Acceleration or deceleration between 0.4g and 0.5g coupled with a forward TTC of 4 s, and with a corresponding forward range value at the minimum TTC not greater than 100 ft.
Rear TTC	Any rear TTC trigger value of 2 s or less that also has 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 0.3g.
Yaw Rate	Any value greater than or equal to a ±4-degree change in heading (i.e., vehicle must return to the same general direction of travel) within a 3-s window of time.

Table 2. Detection and Error Rate of Flags

	Percent of Valid Flagged Events	Percent of Invalid Flagged Events
Lateral Acceleration	3.5	91.3
Longitudinal Acceleration	44.7	66.4
Event Button	8.4	69.9
Forward TTC	56.4	86.4
Rear TTC	4.6	59.9
Yaw Rate	21.7	91.1

The majority of studies involving crashes or near-crashes discussed their criteria for detection using kinematic data. Since many of these studies were conducted by VTTI, the above thresholds were either the first filter or the primary filter used to find events. An additional look at some 100-Car data found range rate (radar) to be a good predictor of crashes and near-crashes, but the error rate was still relatively high. The relative error for each boundary model (shown Figure 4) increased as the detection rate went up. The model with minimum error identified 10 out of 11 crash events, had a false alarm rate of 20%, and a valid hit rate of 74% for crashes and near-crashes [2]. Range rate and TTC are very similar and can be a great metric for collisions with lead vehicles; however, the method is still prone to error since the radar does not always target the correct location. This problem was acknowledged in another VTTI report that estimated range rate based on video data and the width of the lead vehicle [8].

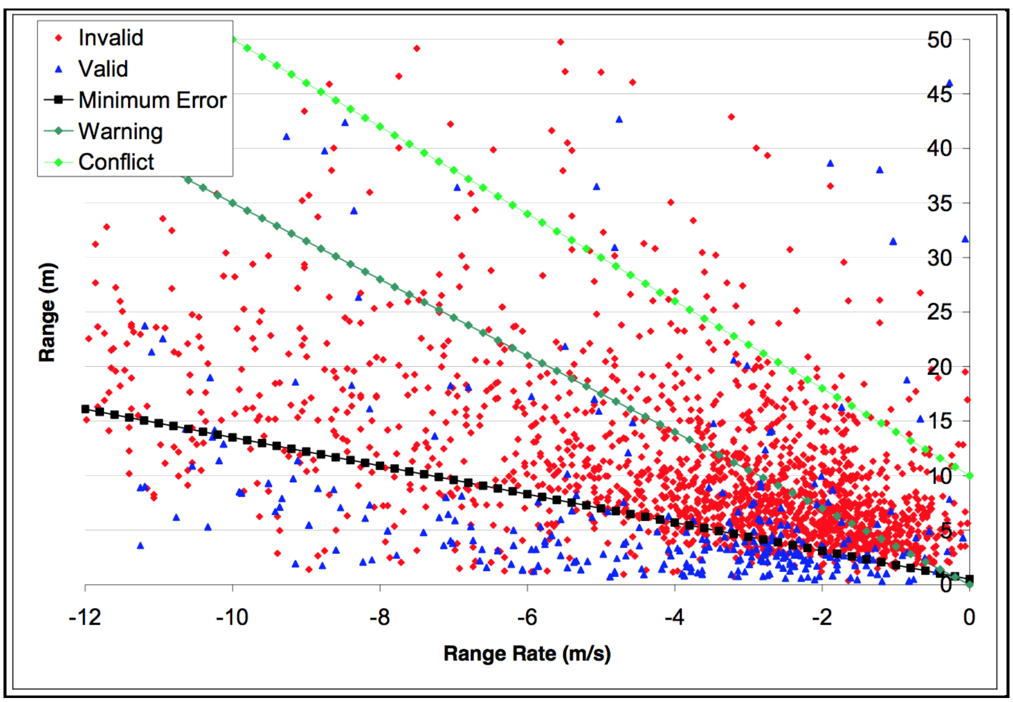

Figure 4. Range rate model boundaries. [2]

Another study from the SHRP 2 NDS [9] used the same verbal definition of a near-crash but filtered candidate events by a braking force of 0.5g or a steering input that resulted in a lateral acceleration of 0.4g. This study admitted that upon completion of the filtering process, it was up to the analyst's subjective judgment to define the candidate event as a near-crash or a less severe crash-relevant event.

A series of studies by Wu and Jovanis [10, 11] suggested a method to use naturalistic driving to detect surrogate events (defined as crashes or near-crashes in this report), through a series of screenings (Figure 5). This study used data posted in VTTI's online data warehouse, which will be further discussed in the Method section. The first screening was a simple threshold optimized to the sensitivity and specificity of the lateral accelerations (Figure 6) that resulted in a set of candidate surrogate events for classification before a second screening. For classification, the goal was to separate events by the structure of the crash's lateral acceleration progression. In this case, a Chow test was conducted to test for a structural difference between intersection and non-intersection crashes. After this, a second screening was performed, after which a model was developed indicating if an event could (or could not) be used as a surrogate for crashes [10]. A conversion factor could then be calculated using conditional probabilities to essentially get the value of the surrogate in terms of a crash (i.e., one surrogate event is worth 0.13 crashes) [11].

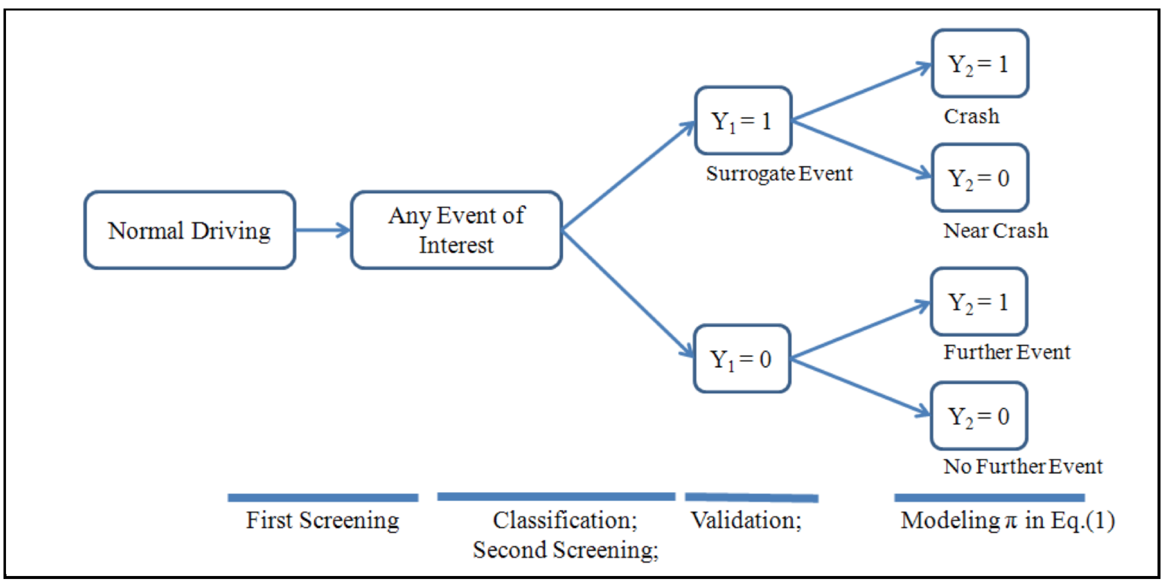

Figure 5. Surrogate detection model. [10]

	Maximum Lateral Acceleration (LATM)		Maximum Difference in Lateral Acceleration (LATD)	
Cut-off point	Sensitivity	Specificity	Sensitivity	Specificity
>= 0.0g	100.00%	0.00%	100.00%	0.00%
>= 0.1g	100.00%	4.80%	100.00%	2.40%
>= 0.2g	98.41%	22.40%	100.00%	13.60%
>= 0.3g	92.06%	33.60%	96.83%	28.00%
>= 0.4g	71.43%	60.80%	93.65%	41.60%
>= 0.5g	46.03%	77.60%	84.13%	56.80%
>= 0.6g	38.10%	84.00%	63.49%	72.80%
>= 0.7g	26.98%	89.60%	49.21%	80.00%
>= 0.8g	12.70%	95.20%	41.27%	83.20%
>= 0.9g	7.94%	96.80%	36.51%	84.80%
>= 1.0g	4.76%	98.40%	25.40%	90.40%
>1.0g	0.00%	100.00%	0.00%	100.00%

Figure 6. Sensitivity analysis of thresholds for conflict detection [10].

Another report by Talebpour et al. [12] used the NGSim data set and proposed two methods for detecting near-crashes, both of which used TTC metrics. The NGSim data set contains vehicle trajectory data at two locations collected via video data reduction [13]. The recommended method by Talebpour et al. is shown in Figure 7. This approach calculated a normal distribution for each driver's longitudinal acceleration and flagged any event when the acceleration's probability of occurrence was less than the predefined value. Then, the situation was examined for hard braking due to a conflict with a lead vehicle or hard braking by a following vehicle. The approach seems reasonable; yet their recommendation to use this method was based on results that seemed most realistic to the authors, which is subjective. It illustrates the importance of individual driver preferences in detection of near-crashes using acceleration and TTC information [12].

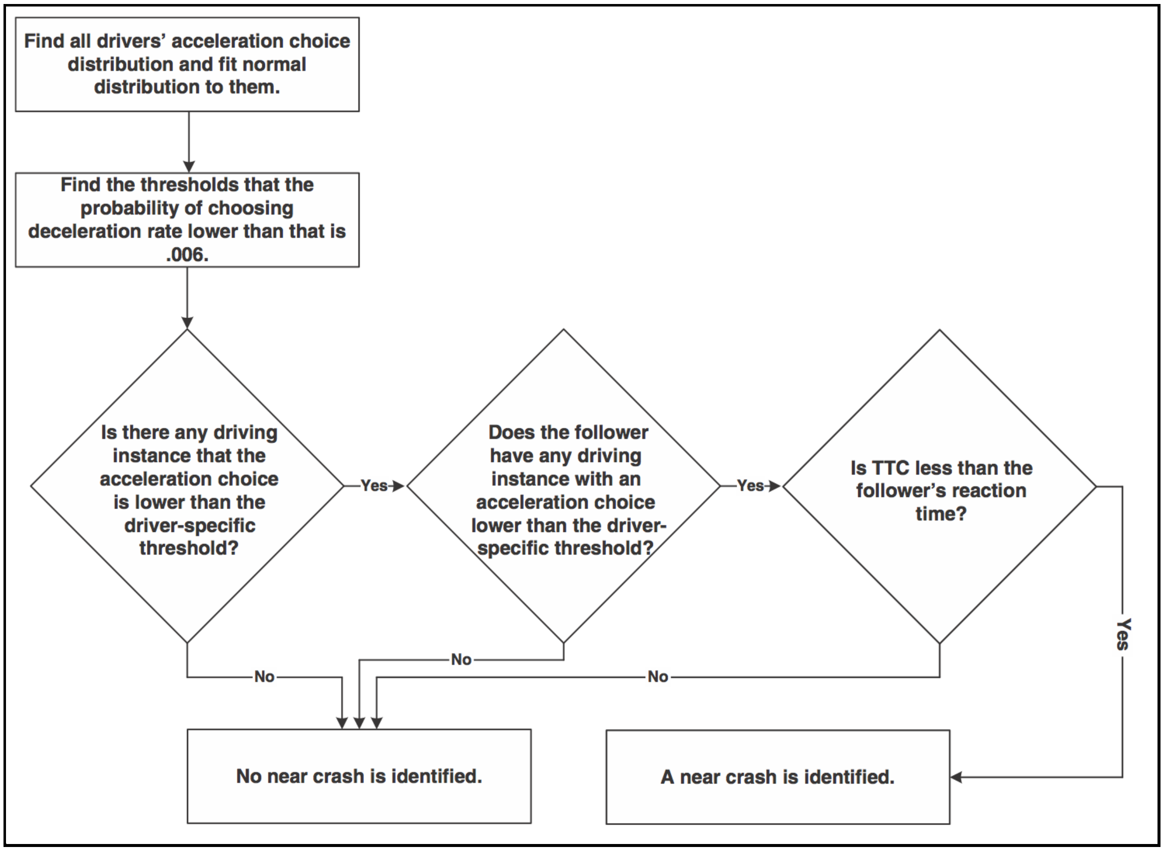

Figure 7. Method 1 as recommended by Talebpour et al. [12].

Another research study considering NDS data to study risky driving in teens used a simple threshold trigger that was lower than the other NDS studies listed; this is because the data were used to detect "risky driving" and not near-crashes [14]. Smith et al. also examined accelerations in crash-imminent scenarios to quantify different situations for collision avoidance systems [15]. Lastly, an alternative approach was taken by Gordon et al. Measurements of known run-off-the-road hot spots in northern Virginia were taken and NDS data were examined for differences in speed entering and exiting the segments, as well as for yaw rates at different points. This report showed the viability of performing this type of analysis to learn more about vehicle trajectories at hot spot locations [16].

Based on the findings of the literature review, a method for detecting crashes and near-crashes was designed using components in the BSM. Due to the low market penetration of V2V technology in the test bed setting, TTC was not considered for the preliminary algorithms. Instead, various modeling techniques were explored to predict if a BSM reading occurred during a crash.

Method

Some of the techniques for detecting crashes and near-crashes in a CV environment that have already been discussed are actually difficult to apply. Many of these algorithms use TTC as a metric for detection, which is not really feasible due to the small percentage of vehicles with V2V technology. The TTC metric is likely to provide insight, but the goal is to have a method that is feasible with a relatively small number of V2V vehicles. Additionally, some of these flags were developed with the knowledge that video data were available to check the results; this may have influenced the designers to choose slightly more liberal criteria for flagging events.

Thus, the objective of this research was to develop a model that does not use TTC and that has a minimal error rate to detect crashes using BSM elements. In the following sections, the data sources used in building the models will be described, followed by the methods used for modeling crash and near-crash events.

Data

All of the data were acquired from the NDS conducted by VTTI. Developing the models based on NDS data is beneficial because it provides video capture of the events that occurred, while still containing the key kinematic elements that are present in the BSM. Unfortunately, due to limited data availability, the training and test data sets were not derived from the same unified data set.

The training data set consisted of 14 crashes that occurred during the 100-Car NDS. Three data files were associated with each crash: trip log, front video, and rear video. Descriptions of each are followed by visual representations in Figure 9**,** Figure 10, and Figure 11, respectively:

- *Trip Log* – This table contained various dynamic, geographic, and time-related data, which were collected from each vehicle's DAS at a frequency of 10 Hz. Data elements collected included speed, three-direction acceleration, and yaw rate. Figure 8 shows a time series of longitudinal accelerations for two of these events.

- *Front Video* – This video showed the driver's view through the front windshield. A timestamp appeared in the bottom left corner of each video for reference to provide a connection between events or actions observed in the video and the corresponding data in the trip log. Figure 9 shows a screenshot from one of the front video files.

- *Rear Video* – This video showed the view of the trip through the back windshield of the vehicle. Rear video footage was black and white and lacked a timestamp; whenever anything of interest was captured in this video, the corresponding time in the front video would need to be examined to extract the timestamp. Figure 10 shows a screenshot from one of the rear video files.

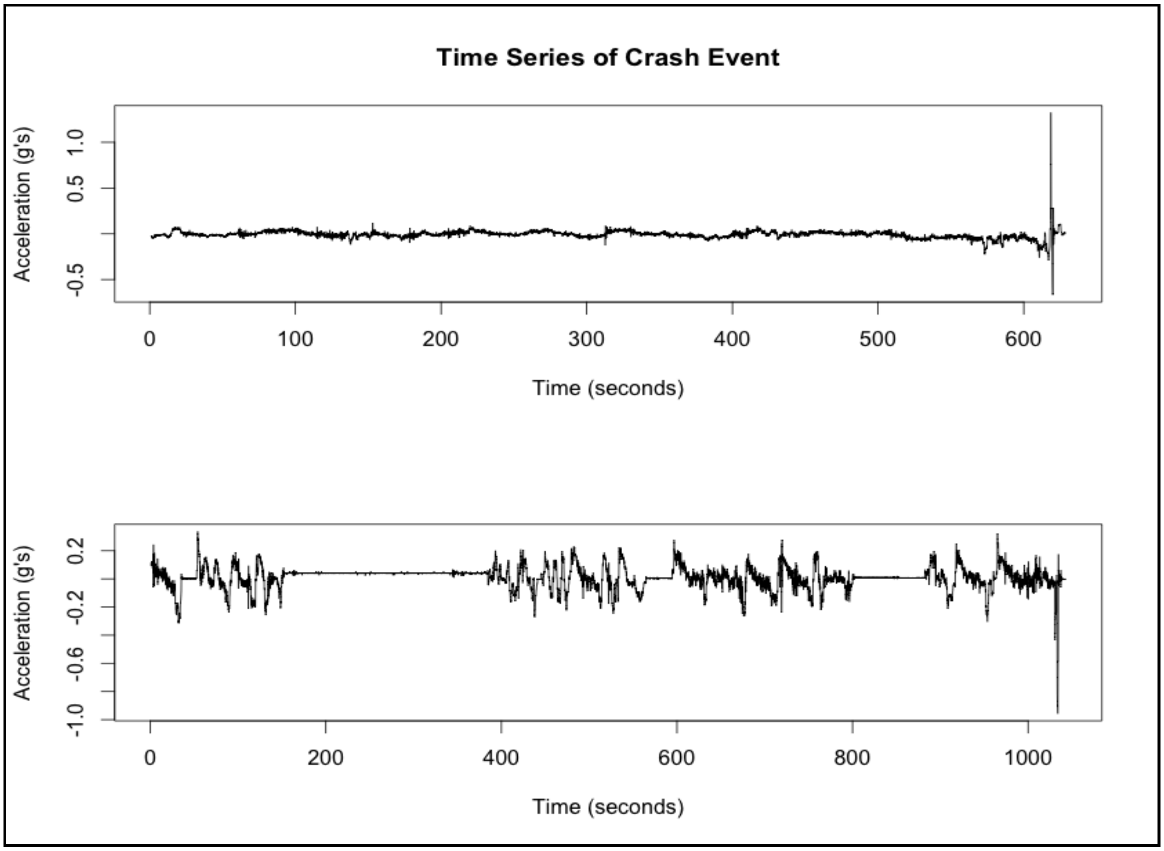

Figure 8. Two time series of longitudinal accelerations culminating in crashes.

Figure 9. Front video NDS data.

Figure 10. Rear video NDS data.

The trips varied in crash type, time of day, length of trip, and outdoor conditions. Researchers removed the trip origins in an effort to make the personally identifying information unavailable to analysts. Since all of the trips culminated in a crash, the removal of destinations was not necessary. The shortest video received was 22 seconds long (the next shortest was over 3 minutes), while the longest video was 35 minutes. The distribution of trip lengths, in 5-minute increments, is shown in Figure 11. The average trip length was 10.4 minutes.

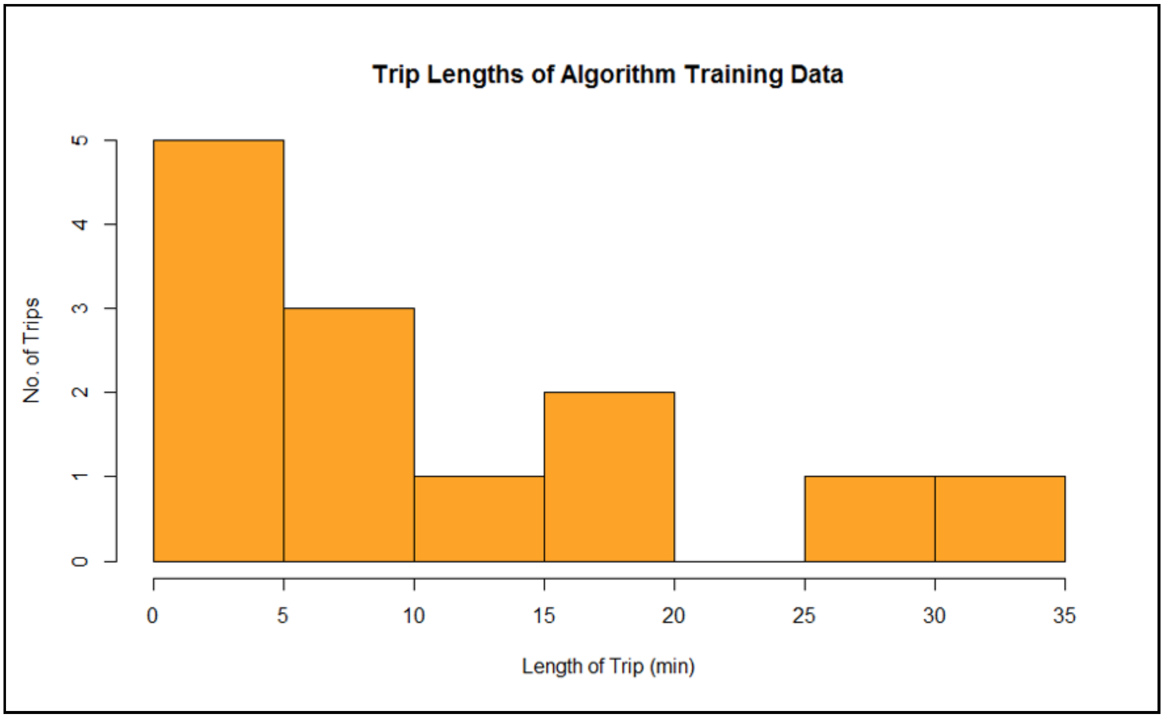

Figure 11. Trip length summary.

After a closer review of the trip logs, one crash had to be removed due to a block of missing data at the time of the crash. This left researchers with 13 usable crashes. The video files were also examined to understand information about the exterior conditions and crash types. Table 3 shows the distribution of crash type, lighting, and weather for the 13 remaining crashes. In all cases of precipitation, the type of precipitation was rain; also, one of the nighttime crashes experienced precipitation as well.

Table 3. Crash Type Distribution (Left); Crash Conditions (Right)

Crash Type	Count
Rear End	9
Sideswipe	1
Lane Departure	1
Angle	2

Conditions	Count
Day	11
Night	2

Conditions	Count
Clear	11
Precipitation	2

Once the video data were reviewed, the point of the crash and the points spanning the entirety of the crash event were labeled in the trip logs. This was done manually, using personal judgment, by taking the timestamp at the point of collision and the timestamp when the vehicle came to a stop and filling in a crash indicator at all the points between them.

Due to the small number of crashes available in the training set, it did not make much sense to set aside data for testing from the original data set; as a result, two additional data sets, both originating from the NDS, were used for testing. The first set for testing was a set of another 14 trips with no crashes or near-crashes. These trips contained significantly more driving time, amounting to roughly 10 hours combined across the 14 trips. This set of data was used to test the false positive rate (false alarms per hour) for the models developed from the first trip. Since no video data were available, it was not possible to verify that a near-crash did not occur if the model flagged an event. However, since VTTI said that no crashes or near-crashes occurred when the data were acquired, the assumption that any detection is a false positive seems reasonable, at least for exploratory purposes. Figure 12 shows the longitudinal acceleration profile of a trip from this normal driving data set. Notice the difference in scales on both axes.

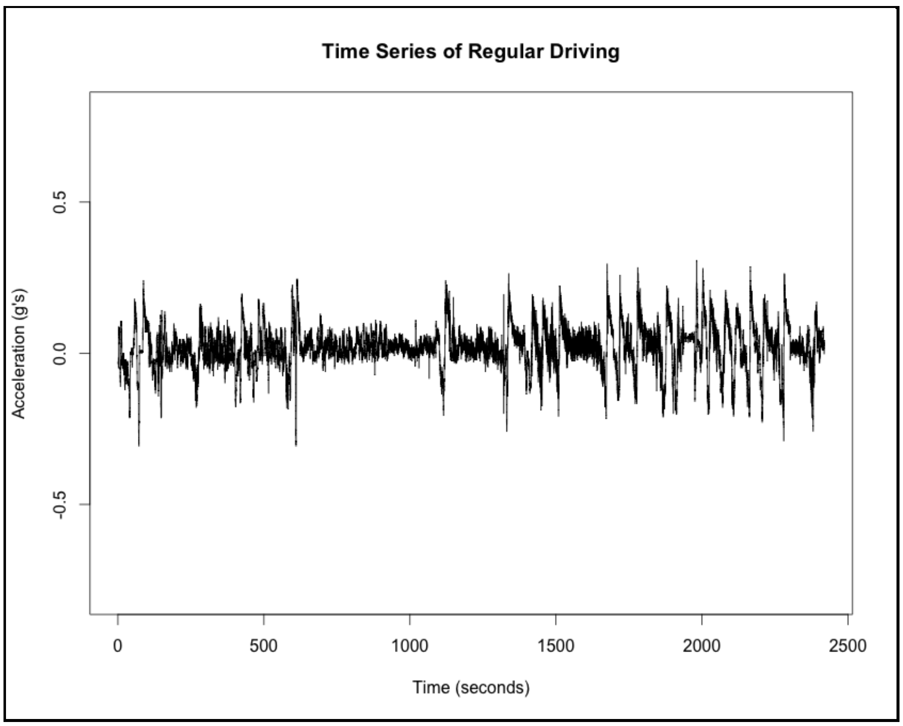

Figure 12. Time series of longitudinal accelerations in the normal driving data set.

The third set of data contained 68 crash events and 760 near-crash events [17]. This data set included 30 seconds of pre-event and 10 seconds of post-event data, amounting to slightly more than 40 seconds worth of data per event [18]. This dataset contained fewer attributes than the previous full trips acquired from VTTI and had no video; however, due to the short duration of the time series, it can be assumed that any point where a crash or near-crash was indicated by a model was indeed the point of the crash or near-crash. Additionally, a short account of what occurred during each crash and near-crash was recorded in an event narrative file [19]. This data set was used to test the sensitivity (false negative rate) of the model. A sample time series of longitudinal and lateral accelerations is shown in Figure 13**.**

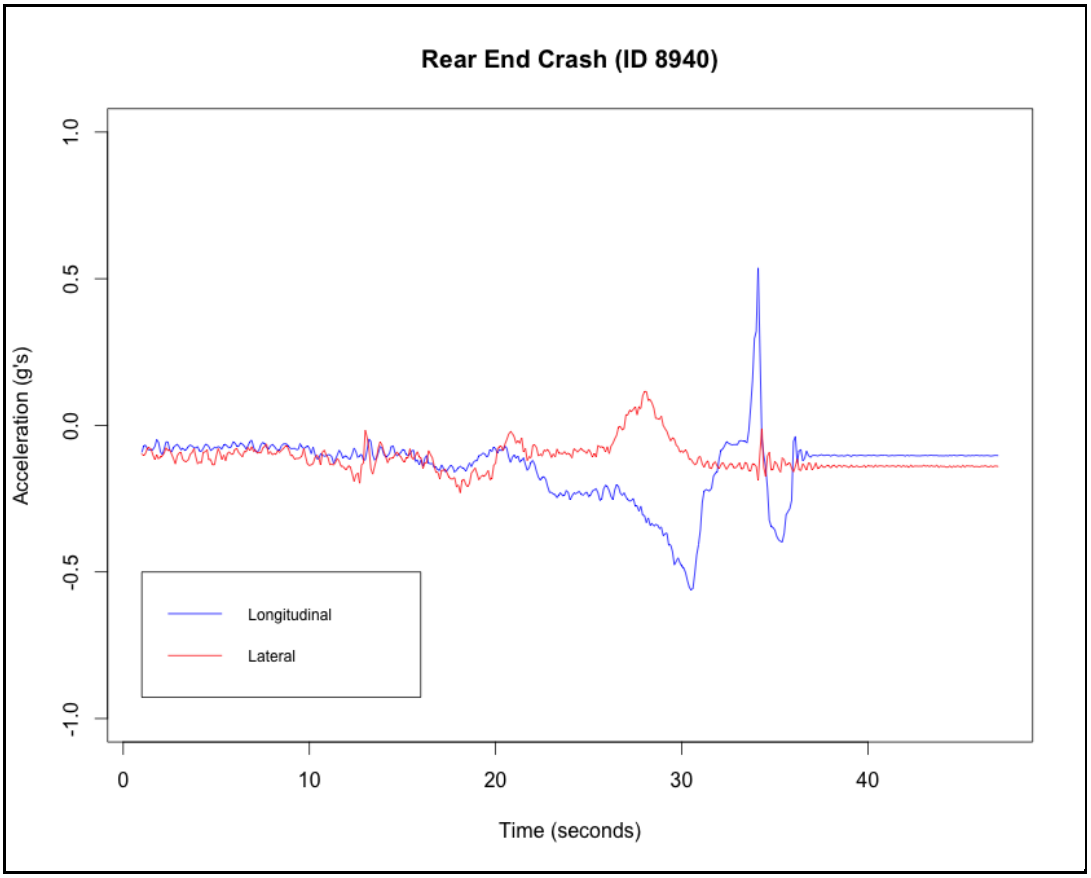

Figure 13. Longitudinal and lateral accelerations from a crash event.

Models

A few different modeling techniques were applied to predict if a data point was part of a crash or near-crash event. The first ones were the threshold-based models VTTI uses as its first screening method; these results were used as the benchmark for comparison to estimate model performance. Others tested included Classification and Regression Trees (CART), Multivariate Adaptive Regression Splines (MARS), and a pattern matching algorithm.

A vehicle's current acceleration depends on immediate past actions, especially for normal driving tasks when actions are deliberate and repeated throughout a trip. Seeing an acceleration drop from −0.2g to −0.3g is very different than an acceleration dropping from 0.3g to −0.3g over the same time period. In most cases of a crash, there is a large spike in acceleration that oscillates briefly around zero while decaying to commonly observed values shortly after the impact. Using pattern recognition, these differences can be captured, and employing a reasonable threshold will not identify what is occurring at surrounding points. So, if a threshold is exceeded, you wouldn't know how sharp of an increase there was leading up to the event without information on the surrounding points. Additionally, the type of crash and the point and direction of collision will impact the pattern of the accelerations. For these reasons, the models required some sort of data aggregation and manipulation before construction. A slightly different approach to aggregation was taken in each modeling technique.

Figure 14 shows the longitudinal acceleration (g's) for a trip that took place primarily on the highway. During this trip, the vehicle stopped suddenly on the freeway around time 125 seconds, which is represented 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 −3g upon impact.

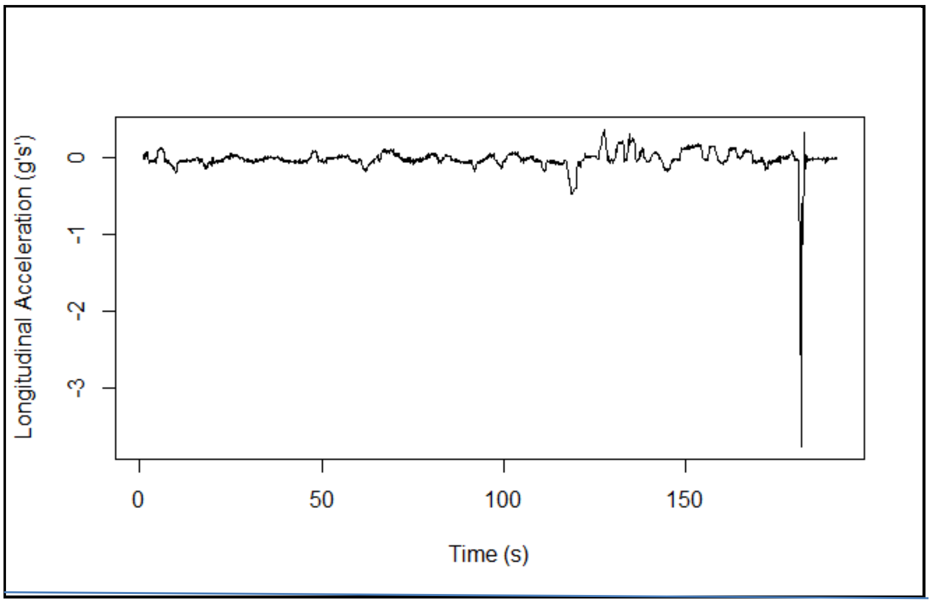

Figure 14. Sample time series of longitudinal accelerations, rear-end crash at time 170 seconds.

Figure 15 shows a panel of four other crash events. The *y*-axis values vary from crash to crash and depend on a combination of crash type, type of vehicle involved, and speed at impact. Thus, simply implementing the threshold can lead to issues when trying to detect crashes in this and similar data sets. These issues can force the analyst into a trade-off between 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.

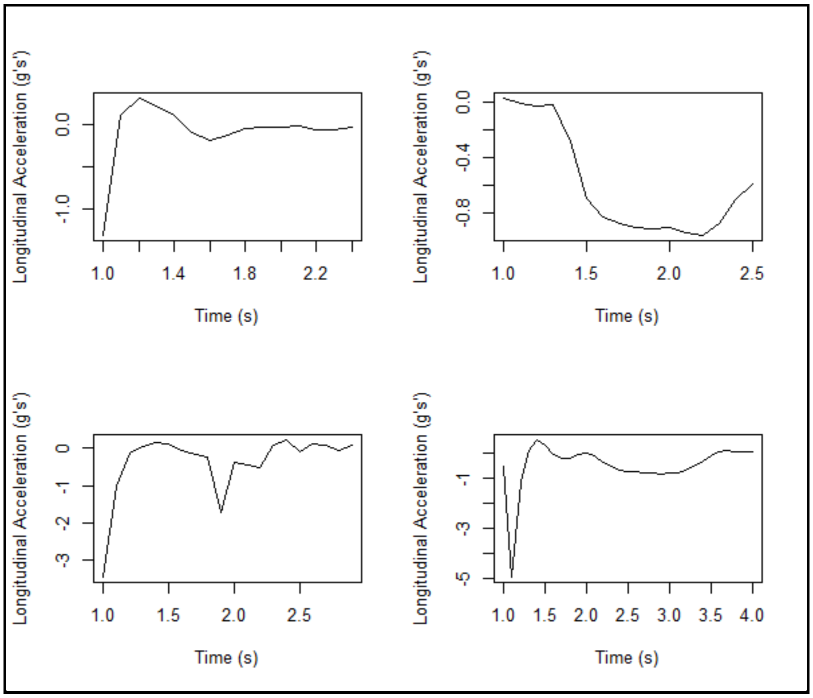

Figure 15. A sample of acceleration profiles during crashes.

Pattern Matching Algorithm

By examining Figure 14 again, we see that certain patterns appear to repeat 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 do not follow a consistent pattern.

Researchers inspected video data from three different trips and determined five baseline time series to represent five different common driving actions:

- 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 16. Sensitivity analysis was completed with different series selected as baselines and with different numbers of baselines. Using too few baselines resulted in numerous unidentifiable stretches, while too many baselines led to confusion about what type of action happened at a specific point.

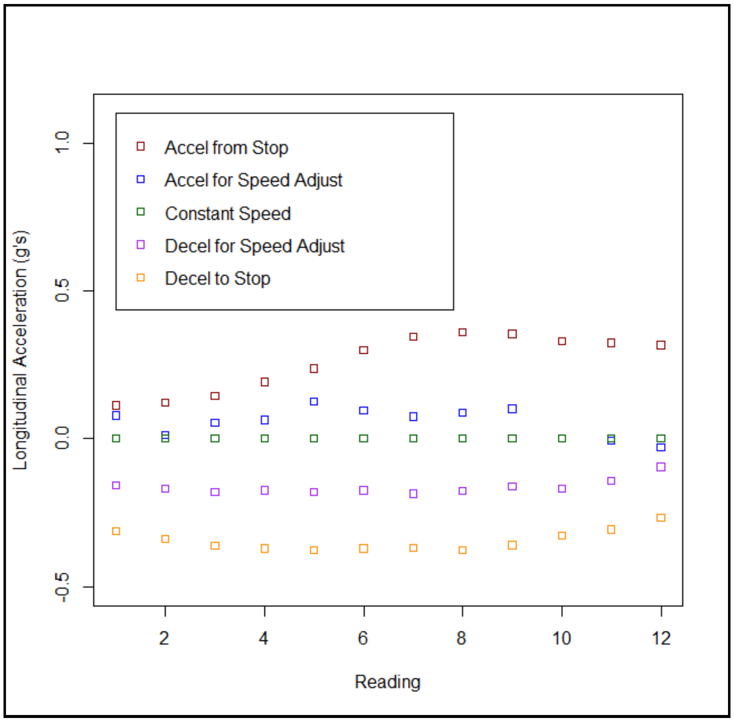

Figure 16. Selected baselines.

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 that performed 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.

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

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 decided after testing multiple candidate baselines. After no apparent difference in the results between different baselines tested, the candidate baseline 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; thus, each point could have been assigned to more than one pattern. In the event of a point being classified into multiple actions, the action that had the most assignments was selected. In the event that a point did not have at least 6 of the 12 pattern matches relate to a single action, the point was listed as an unidentified action. Six was settled on as a threshold through sensitivity analysis and because it ensured that the majority of the pattern-matches indicated the point was a part of the action.

Figure 17 shows a crash, represented by black dots, compared to the baselines from Figure 16. Simply through visual 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.

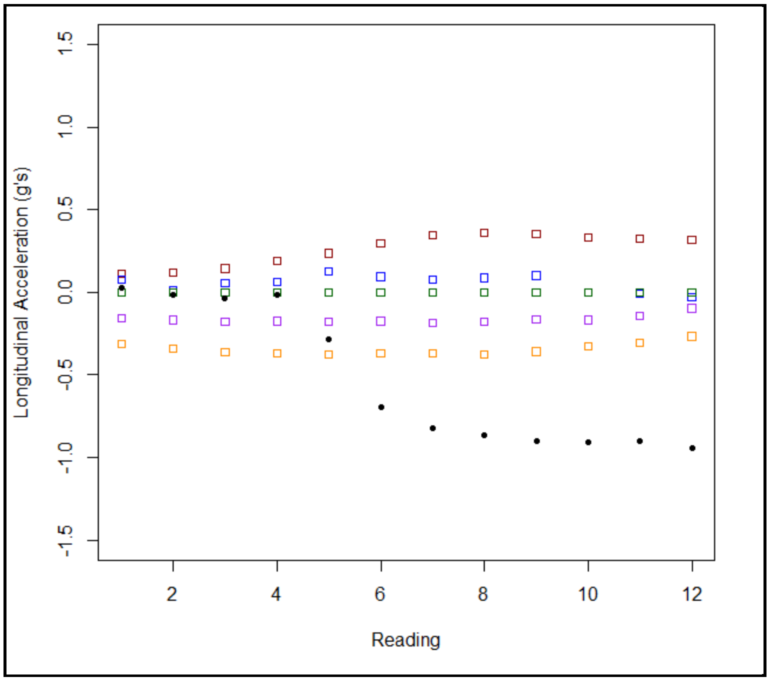

Figure 17. Crash comparison (black dots) with baseline readings.

Classification and Regression Trees

The first attempt at modeling the data was conducted by inputting data points collected at a frequency of 10 Hz into a model, without any aggregation whatsoever. However, this was highly unsuccessful since the longitudinal acceleration values decay quickly to seemingly normal-level readings. The next step was to test the point of contact—just the few readings where the crash actually started. This was also unsuccessful likely due to a lack of data (usually less than 10 readings per crash). The next solution was aggregating the data into intervals and calculating some descriptive statistics for each interval. Those values were calculated as follows:

- Maximum longitudinal acceleration
- Minimum longitudinal acceleration
- Mean longitudinal acceleration
- Variance longitudinal acceleration
- Maximum lateral acceleration (after taking absolute value)
- Mean lateral acceleration
- Variance lateral acceleration

- Maximum *z*-direction acceleration
- Minimum *z*-direction acceleration
- Median *z*-direction acceleration
- Mean speed
- Maximum speed
- Minimum speed
- Median speed minimum
- Euclidean distance from set vehicle maneuver trajectories

CART is a recursive technique that chooses the best variable to split the data during each step based on a variety of proposed metrics for impurity, generally either using the Gini or Information (sometimes "Entropy") values—both were tested in the modeling process. The result of each phase is an exhaustive search and then a split based on the optimal value of the selected metric. The benefits of CART include an easily interpreted decision tree and the ability to make decisions for data organized in the manner shown in (see Figure 18). Unfortunately, while it provides the optimal solution at each phase, the overall optimal solution is not necessarily reached. Additionally, trees do not make splits based on variable relationships.

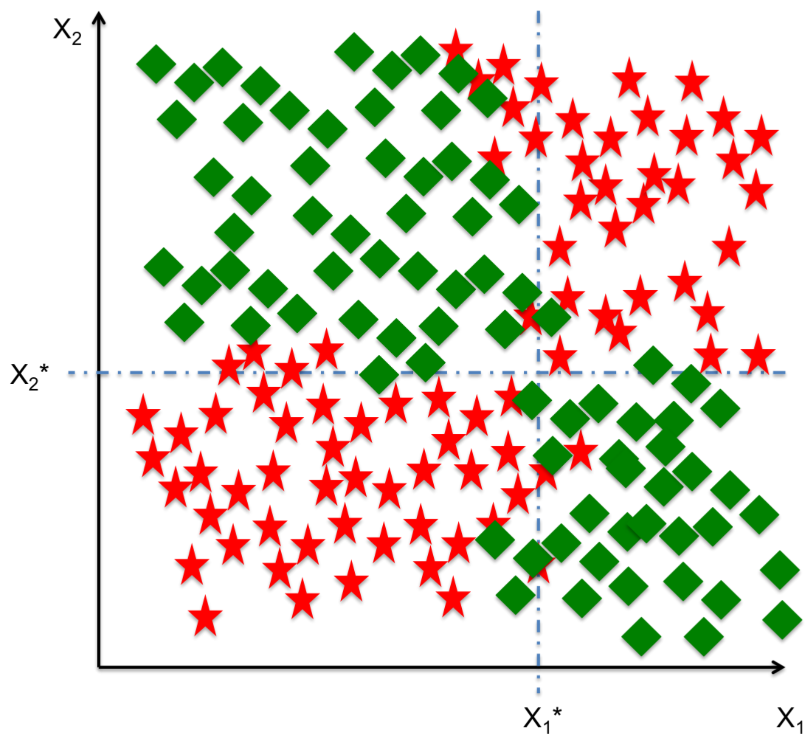

Figure 18. Decision trees excel at separating data organized in this manner.

Using the "rpart" package in R [20], a few candidate trees were constructed, with different subsets of the above variables and both impurity metrics. The best tree was the entropy metric with the variables presented in Figure 19. The tree was pruned using the minimum complexity parameter value. The two numbers below each branch (*N*/*M*) are the predicted 0 and 1 values at each endpoint (*N* = number of predicted 0's or non-events, *M* =number of predicted 1's or crashes/near-crashes). The Receiver Operating Characteristic (ROC) curve is shown Figure 20 and the score table is shown in Table 6 for the training data set. The false positive rate is shown in Table 7. Clearly the CART model produces more false positives, but that is likely due to the overlapping nature of the windows (i.e., if one point has a minimum longitudinal acceleration and maximum lateral acceleration that implies a crash, it will be a part of 12 windows, all of which will be labeled crashes). This issue can likely be improved upon by not overlapping the windows, which will be tested in the future.

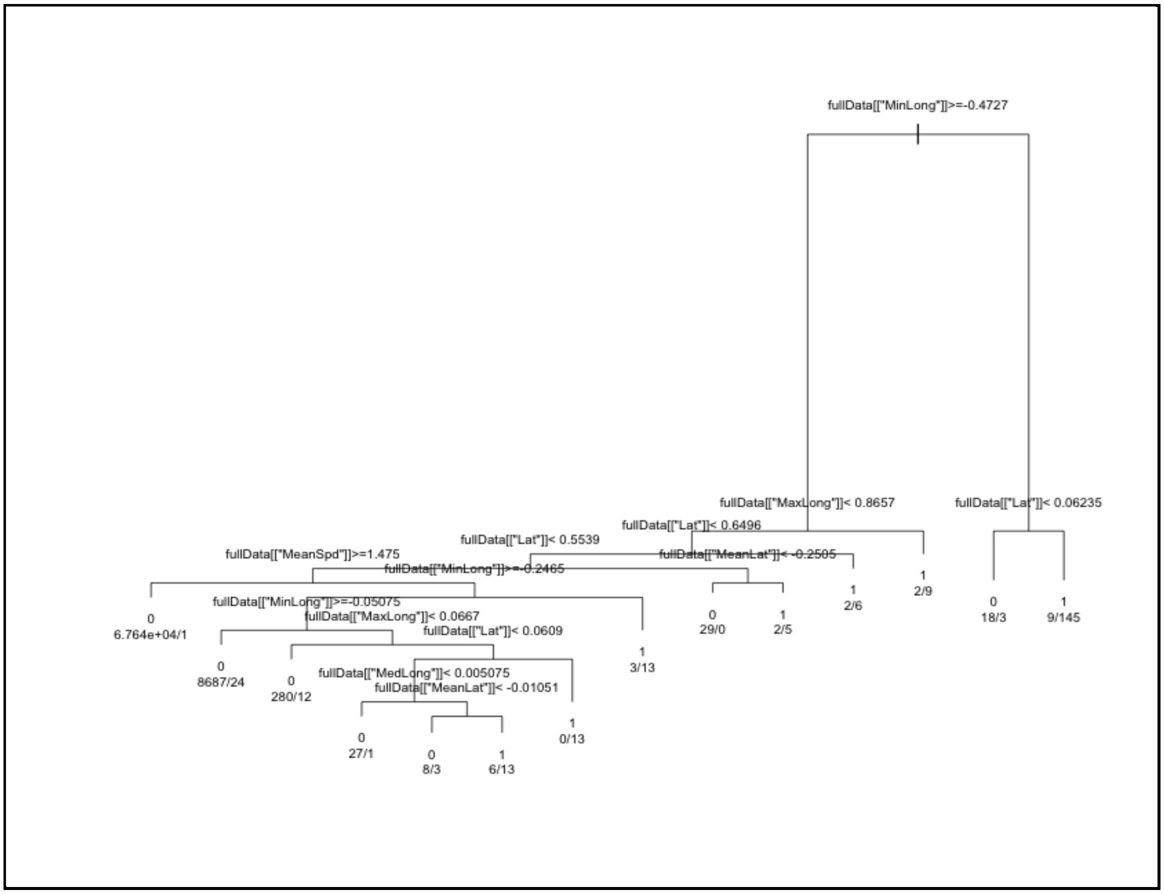

Figure 19. Entropy tree.

Multiple Adaptive Regression Splines

Using the same aggregated data used in the CART model, a MARS model was constructed. MARS is a technique proposed by Freidman [21] in which a piecewise regression function can be built by creating hinge functions that can change the trajectory of the model based on the data's trajectory. MARS has the benefit of being able to provide very good fits to the data. The process also outputs a way to determine variable importance, at the cost of some interpretability.

To build the MARS model, the "earth" package from R [22] was used. MARS models predict a probability for each data point being true, so a threshold was the probability for calling a window a crash. The best one was around 0.8, based on total error as a metric. On the training data, the model predicted 9 false positives and 56 false negatives out of a total of 76,960 windows. Two hundred forty-eight of the total windows spanned a positive crash reading.

Results

This section will discuss the predictive capacity of each of the three methods on the training data and the false positive rate on the test data. The online crash and near-crash event data described previously have yet to be tested in the CART and MARS models because some of the data elements used in the models were not available in that data set. The NDS thresholds were used as a baseline for comparison.

Pattern matching had very positive preliminary results, identifying 12 out of 13 crashes while producing only three false positives. Upon further inspection, the unidentified crash was a low-speed rear-end collision that did not exceed a deceleration of −0.3g at the point of impact. Additionally, the false positives occurred at explainable points upon reviewing the video; the first occurred when the vehicle went over a speed bump and the second occurred when the vehicle (this particular acceleration series is shown earlier in Figure 8) was forced to stop suddenly on the freeway, which could be defined as a safety-critical event and thus should be inspected to see if it happened frequently 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 results derived from using two different thresholds as event identifiers. The results were favorable for the pattern matching technique. Table 4 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 for 0.6g and 0.4g. The detection rate results for the online data, shown in Table 5, showed no statistically significant difference between the 0.5g threshold and the pattern matching with 95% confidence, with n = 68. However, given the benefits of a reduced false positive rate, the pattern matching algorithm may still be worthwhile.

Table 4. Pattern Matching Training Data Detection Rate

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

Table 5. Pattern Matching Results Online Data Set

	Crash	**Near-Crash**
Pattern Match	0.691	0.737
0.5g Threshold	0.603	0.663
0.6g Threshold	0.485	0.486

Pattern matching and thresholds did not always detect the same false positives. In general, the false positives from the pattern matching technique tended to be either near-crashes or safety-critical events, while this was not the case for thresholds, especially with the lower threshold of 0.4g. An additional 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. The pattern matching algorithm was developed with a logical way to account for successive indications of crashes.

For the CART model and the MARS model, the windows created overlap so there were nearly as many windows as total readings, which yielded 248 positive readings even though there were only 13 actual crashes. Table 5 and Table 7 show the detection rate for each algorithm; however, a more robust algorithm still needs to be developed with a decision rule about how close two positive readings can be in order to be considered the same.

Table 6 shows the score table for the VTTI NDS thresholds, the benchmark comparison for the other models. Table 7 shows the predictions and error rates for the CART model. The recall (sensitivity) of the CART model is 199 out of 248 and the precision is 199 out of 221. The ROC curve (Figure 20) shows this model very easily detects the majority (~90%) of the windows that contain a crash, but getting the additional windows almost linearly increases the false positive rate after that. To check for false positive rate, the model was tested on normal driving events; Table 8 shows that the model did not carry over to the test data as well as the NDS threshold.

Table 6. NDS Threshold Score Table Results

VTTI NDS	**Actual**	
Prediction	0	1
0	76680	120
1	32	128

Table 7. CART Score Table

CART Model	**Actual**	
Prediction	0	1
0	76690	49
1	22	199

*Recall = 199/248 = 80.24%, Precision = 199/221 = 90.05%

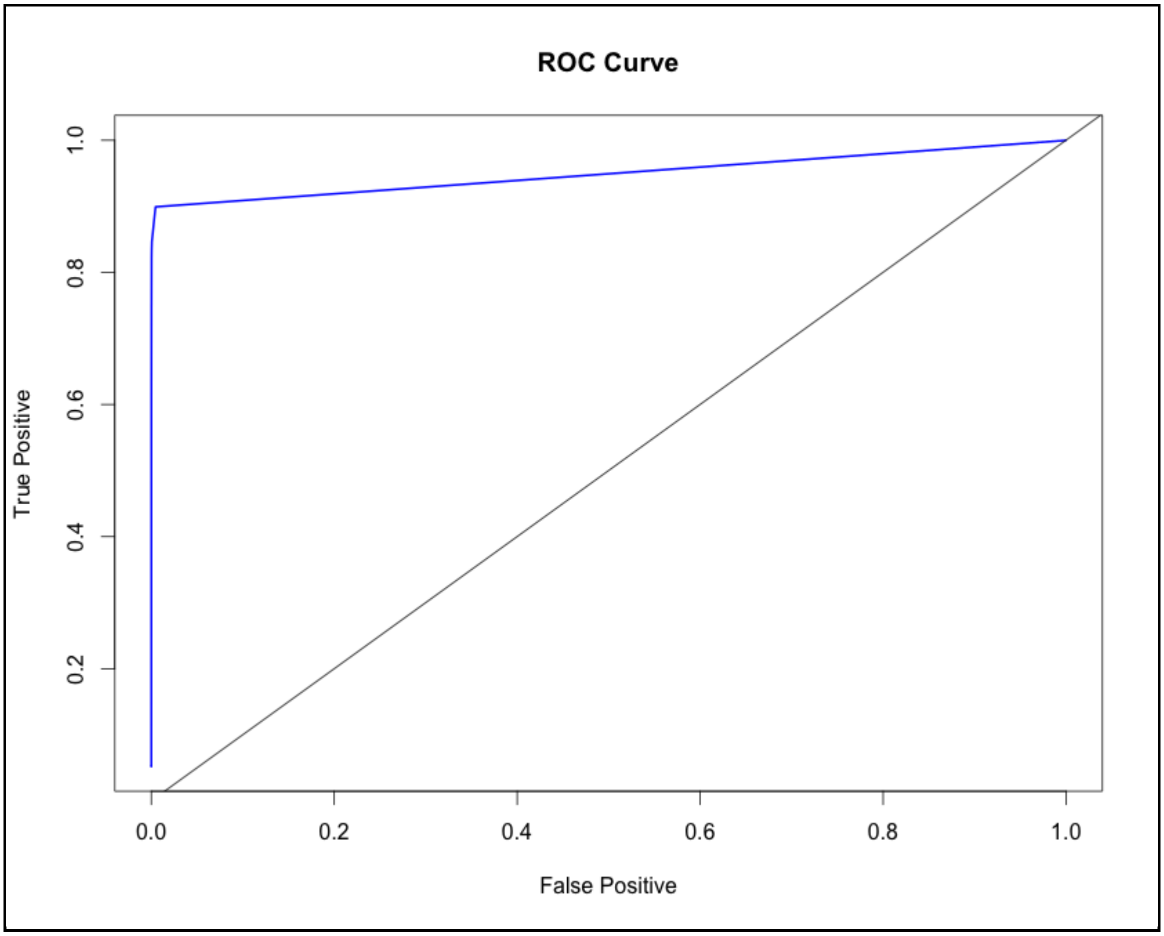

Figure 20. ROC curve for CART tree.

Table 8. CART Test on Normal Driving vs. NDS Triggers

	Actual = 0	
Prediction	**CART**	**VTTI NDS Study**
0	368056	369051
1	1025	30

As shown in Table 9, the MARS model's performance on the training data improved on both the precision and the recall over the CART model and the VTTI thresholds. It also resulted in only 16 false positives on the normal driving data set, as compared to the 30 false positives from the VTTI thresholds.

Table 9. MARS Model Results on Training Data

MARS Model	**Actual *t* = 0.8**	
Prediction	0	1
0	76703	56
1	9	192

Recall = 192/248 = 87.42%, Precision = 192/201 = 95.52%

Table 10. Results of MARS Model on Normal Driving Compared to NDS Threshold

	Actual = 0	
Prediction	**MARS**	**VTTI NDS Study**
0	369065	369051
1	16	30

Conclusions and Recommendations

Researchers proposed the question of which detection method should be carried out in an application setting. In the pattern matching method, sensitivity analysis was used as the justification for numerous decisions that were made. A complete sensitivity analysis was necessary, but it is important to keep in mind that the algorithm was designed using 13 crashes, so finding the optimal values to use in the algorithms was not possible on a general basis. However, researchers had to make decisions about what distance metric to use, the number of baselines to use and their value, length of windows, and criteria for the classification of unidentified points, among a few other things. After the sensitivity analysis was completed, it was discovered that many of the choices made at different points of the algorithm had a range of acceptable decisions, but were, for the most part, related. For example, the selected length of the window impacted later steps in the algorithm, such as the value of the Euclidean distance.

With pattern matching, 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 and prevent the false positive. This was purely demonstrated as a proof of concept, and while it appeared promising, the computing cost required to complete this algorithm was fairly high without a practical or a statistical significant benefit over a simple threshold.

For the CART and MARS models, the MARS model appeared to perform well with both the training and the test data, likely at the cost of easy interpretability. Again, given the limited amount of data available, it is difficult to say that the MARS model is definitely a better selection than a simple threshold, but based on this preliminary work it certainly has the potential to be an improvement. Additional work can be done to look at the best way to segregate the time series data to optimize the model results. This includes determining if the time series should be broken up into windows, and if so, where the optimal locations for breaking those windows are. Additionally, the models could benefit from an algorithm that takes the model results and interprets them into events, since an event could span a period of time. Currently, the models classify the points, but do not recognize that multiple points in a row are part of a single event.

Some additional approaches can also be used to predict crashes. Autoregressive Integrated Moving Average (ARIMA) models are regression models used on time series data to forecast future values while taking into account either the value of the time series at a previous time or the error at a previous time. An ARIMA model for each driver could be developed to forecast the value of a certain variable (e.g., longitudinal acceleration) for time ($t + 1$) and calculate a confidence interval around the forecast. Then, if the variable's true value at ($t + 1$) falls out of the confidence interval of the forecast, the event can be flagged. Additionally, the aggregated data used in the MARS and CART models can also be used to develop other types of predictive models, including logistic regression, neural networks, and support vector machines.

The next important step to this research is to design a methodology to detect crashes and use the crashes detected in applications. This can be performed on the CVI-UTC Northern Virginia CV test bed data once a sufficient volume of data has been collected (i.e., enough “rare” crash and safety-critical events have occurred). The current state of the network screening process requires at least three years of crash data, making it a highly reactive process that subjects the public to sub-optimal or dangerous driving conditions until enough crash data accumulate. Using CV BSMs provides infrastructure providers with a new type of data that can provide more insight than ever into what is actually occurring on the roads.

By inputting BSM data collected in the test bed into these models, researchers can obtain a list of readings that are likely crashes and near-crashes. Then the GPS coordinates for each crash and near-crash can be plotted on the network and compared to known hot spots. Known hot spots can be determined using traditional methods in order to establish a ground truth. This work will also provide more information on near-crashes and their relationship to crashes. Depending on these results, this study will have contributed by:

- Developing a model to use kinematic data to detect crashes and near-crashes;
- Developing a prototype method to use BSMs to detect hot spots;
- Providing an avenue for researchers to understand near-crashes;
- Showing the relationship between crashes and near-crashes.

References

[1] American Association of State Highway and Transportation Officials. *Highway Safety Manual*. AASHTO, Washington D.C., 2010.

[2] 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. National Highway Traffic Safety Administration, 2006.

[3] Guo, F., S. G. Klauer, J. M. Hankey, and T. A. Dingus. Near Crashes as a Crash Surrogate for Naturalistic Driving Studies. T*ransportation Research Record: Journal of the Transportation Research Board, No. 2147*, TRB, National Research Council, Washington, D.C., 2010, pp. 66-74.

[4] Klauer, S. G., T. A. Dingus, V. L. Neal, J. Sudweeks, and D. Ramsey. *The Impact of Driver Inattention on Near-Crash/Crash Risk: An analysis using the 100-Car Naturalistic Driving Study*. Report No. DOT HS 810 594. National Highway Traffic Safety Administration, 2006.

[5] *New York City Crash Stories*. crashstories.org. Accessed September 2014.

[6] Neale, V. L., T. A. Dingus, S. G. Klauer, J. Sudweeks, and M. Goodman. An Overview of the 100-Car Naturalistic Driving Study and Findings. In *Proc. 19th Int. Tech. Conf. ESV*, 2005

[7] Strategic Highway Research Program (SHRP2). http://www.shrp2nds.us.

[8] Victor, T., J. Bargman, C. Boda, M. Dozza, J. Engstrom, C. Flannagan, J. D. Lee, and G. Markulla. *Analysis of Naturalistic Driving Study Data: Safer Glances, Driver Inattention, and Crash Risk*. SHRP 2 Safety Project S08A. Publication Draft.

[9] Rakha, H., J. Du, S. Park, F. Guo, Z. Doerzaph, D. Viita, G. Golembiewski, B. Katz, N. Kehoe, and H. Rigdon. *Feasibility of Using In-Vehicle Video Data to Explore How to Modify Driver Behavior That Causes Nonrecurring Congestion*. SHRP2 Report S2-L10-RR-01. Transportation Research Board of the National Academies, Washington D.C., 2011.

[10] Wu, K. and P. Jovanis. Defining, Screening, and Validating Crash Surrogate Events Using Naturalistic Driving Data. 3rd International Conference on Road Safety and Simulation, September 14-16, Indianapolis, USA, 2011.

[11] Wu, K. and P. Jovanis. The Relationship Between Crashes and Crash-Surrogate Events in Naturalistic Driving Data. *Accident Analysis and Prevention*, Vol 45, 2012, pp. 506-516.

[12] 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.

[13] FHWA. *NGSim Interstate 80 Data Set Fact Sheet*. http://www.fhwa.dot.gov/publications/research/operations/06137/index.cfm. Last Accessed August 2014.

[14] Simons-Morton, B, K. Cheon, F. Guo, and P. Albert. Trajectories of Kinematic Risky Driving among Novice Teenagers. *Accident Analysis & Prevention*, Vol. 51, March, 2013, pp. 27-32.

[15] Smith, D., W. Najm, and A. Lam. *Analysis of Braking and Steering Performance in Car Following Scenarios*. SAE Technical Paper, 2003-01-0283. SAE, 2003.

[16] Gordon, T.J., L. P. Kostyniuk, P. E. Green, M. A. Barnes, D. F. Blower, S. E. Bogard, A. D. Blankespoor, D. J. LeBlanc, B. R. Cannon, and S. B. McLaughlin. *A Multivariate Analysis of Crash and Naturalistic Driving Data in Relation to Highway Factors*. Report S2-S01C-RW-1. Transportation Research Board of the National Academies. Washington, D.C., 2013.

[17] Virginia Tech Transportation Institute. *VTTI Data Warehouse*. http://forums.vtti.vt.edu. Last Accessed September 2014.

[18] Virginia Tech Transportation Institute. *Event Time Series Data Dictionary*. Unpublished attachment to dataset. March 2010.

[19] Virginia Tech Transportation Institute. *Event Narratives*. Unpublished attachment to dataset. March 2010.

[20] Therneau, T., B. Atkinson, and B. Ripley (2014). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-8. 2014. http://CRAN.R-project.org/package=rpart

[21] Friedman, J. H. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, Vol. 19, No.1, 1991. pp 1-67.

[22] Milborrow, S. Derived from mda:mars by Trevor Hastie and Rob Tibshirani. Uses Alan Miller's Fortran utilities with Thomas Lumley's leaps wrapper. *earth:Multivariate Adaptive Regression Spline Models*. R package version 3.2-7. 2014.