Infrastructure Safety Assessment in a Connected Vehicle Environment
UTC Title Page

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 (UVA) Center for Transportation Studies,
and Morgan State University (MSU).

Submitted by:
Virginia Tech Transportation Institute
3500 Transportation Research Plaza
Blacksburg, VA 24061

Program Director:
Dr. Thomas Dingus
Director Virginia Tech Transportation Institute
Director National Surface Transportation Safety Center for Excellence
Newport News Shipbuilding Professor of Engineering at Virginia Tech
tdingus@vtti.vt.edu
(540) 231 – 1501

Name of Submitting Official:
Brian L Smith, PE, PhD
Director UVA Center for Transportation Studies
Professor and Chair CEE Dept.
briansmith@virginia.edu
(434) 243 - 8585

Robert Kluger
Graduate Research Assistant

DUNS: 0031370150000
EIN: 54-6001805

Grant Funding Period: January 2012 – January 2014
Reporting Period End Date: January 1, 2017
Final Research Reports
November 7, 2014
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 Vehicles-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 Environment project was to develop a method to identify infrastructure safety “hot spots” using connected vehicle data. The premise was to show that using connected vehicle basic safety messages to detect hot spots could allow for a quicker detection than traditional methods using police-reported crashes. This is because the basic safety message may be able to detect events that police normally cannot obtain, including unreported crashes and near-crashes. The project successfully developed a model to detect crashes/near-crashes, and also designed a methodology to apply hot-spot identification. Based on the work described in this report, the UTC is fully prepared to apply the methodology to data collected on the field test bed.

Background

Transportation agencies devote significant resources to analyzing crash data collected by responding police agencies to identify “hot spots” – locations that experience larger than normal numbers of crashes. An example of a hot spots map created based on crash data from Ann Arbor Michigan is presented in Figure 1. In many cases, upon identification of a hot spot, field investigation will point to a feature of the infrastructure that is contributing to the crashes. This feature may then be addressed specifically to improve safety. This method, detailed in the Highway Safety Manual’s (HSM) Roadway Safety Management Process (Figure 2), has been used for many years, and has proven to be effective (1). However, this method also has significant shortcomings. One of the key shortcomings is that the agency must wait for a large number of crashes to accumulate before a hot spot may be identified. In other words, this is a very reactive method that requires a number of crashes to occur before corrective action may be taken.

Figure 1. Mapping of Crashes With Potential Hot Spots in Ann Arbor.
Fortunately, there is a reason that crashes are most often referred to as “accidents.” They are infrequent, even at most hot spot locations. Thus, for a statistically significant accumulation of crashes to occur requires a rather long period of time. Furthermore, accurate capture of the location of crashes has long been a challenge in the transportation community. Police reports have been notoriously inaccurate in terms of crash location—although this is improving somewhat with the use of GPS. Thus, even when a hot spot is identified, the exact location of the problem is often difficult to pinpoint.

Thus, there is a need to develop a more proactive way to accurately identify “hot spots”—locations that require modifications to the transportation infrastructure to improve safety. The premise behind this project is that for every actual crash, there also exist numerous “near misses” where drivers’ take last second, extreme evasive action (such as swerving or rapid deceleration) to avoid a crash. These near misses may be as significant as actual crashes in terms of indicating potential safety problems. The challenge lies in identifying and compiling these near misses (since they have never been formally reported by individuals or through the police). However, with vehicles in a connected vehicle (V2V) environment, basic vehicular operation data will be available from the vehicle data bus. If significant evasive maneuvers may be extracted from this data, along with the corresponding GPS location, this near miss data may be analyzed by a transportation agency in a manner similar to current police crash reports to identify hot spots. Using connected vehicles, instead of police reports, offer the potential for a much quicker and more accurate network screening step as shown in Figure 2, which in turn speeds up the entire Roadway Safety Management Process.

This project will analyze data from past field tests to develop prototype algorithms for hot spot identification from vehicular operations data. These algorithms will then be demonstrated and tested in the CVI-UTC Northern Virginia connected vehicle test bed to determine if they successfully extract “near miss” maneuvers. This data will then be analyzed to determine if
hotspots may be identified. Then, finally, these hot spots will be examined in terms of traditional crash data to determine if there is a correlation – thus pointing to 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 the Basic Safety Message (BSM) Data
4) Application and Validation of Proposed Methodology in the CVI-UTC Northern Virginia Connected Vehicle Testbed

The literature review of this project revealed that a limited amount of background work had been done to define a crash or near-crash in terms of kinematic data elements (acceleration, speed, yaw, etc.). This development turned step 2 into a task that involved the development of models to describe crash and near-crash events. Additionally a lack of available testbed data in the timespan of this project made step 4 infeasible given the large data needs of a method like this. As a result this report will focus on 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

The literature review will be outlined in this section while the method section will discuss the models and algorithms used to develop algorithms in the identification process in addition to the data. The results section will discuss the models further and the discussion section will elaborate on how the models performed relative to both a set benchmark found in the literature, and to each other. The conclusion will re-outline the key points from the method, discussion, and results while also discussing the most natural next steps to be taken, namely, how this work can be applied to testbed data in a proof-of-concept study.

**Literature Review**

Two areas of focus in the literature review phase were background information on near-crashes and kinematic-based definitions for crashes. Neither of these two topics has had extensive research dedicated to them for some obvious reasons. The need to detect crashes using kinematic data is fairly recent since this type of data has never been available to researchers on a large scale until the 100-Car Naturalistic Driving Study (2). As far as near-crashes are concerned, they have traditionally been extremely difficult to study since they are not events that get reported. This has left them as poorly defined occurrences that are very subjective to analysts’ judgments.

Since crashes are relatively rare events, even in hot spots, the premise of this research is to show that near-crashes(sometimes also “near-misses”), which occur much more frequently, can be used
to help locate hot spots. The HSM illustrates the continuum of scenarios leading up to a crash very well. It can be seen in Figure 3 that while many conflicts may occur, an actual crash is frequently avoided. It implies that only a small percentage of events with a conflict truly result in a crash. This means that for every crash that occurs, there are multiple close calls where a driver was able to avoid getting in a crash, which is the justification for believing that hot spots can be detected more quickly if near-crashes are also considered. In reality, only 3% of crashes occur with the roadway as the only contributing factor, but an estimated 31% of crashes had the roadway as a contributing factor in combination with a vehicle and/or human factor. Some examples of roadway factors listed include wet pavement, polished aggregate, steep downgrade, and poorly coordinated signal systems (1).

One prevailing issue is that near-crashes have been very poorly defined due to the difficulty involved in tracking events that do not get reported to police. Guo et al. (3) defined it as:

"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
A maneuver is defined as steering, braking, accelerating, or any combination of control inputs that approaches the limits of the vehicle capabilities.

This definition implies that one or both drivers took action to avoid a crash, which certainly is a reasonable situation to define as a near-crash but also is not going to encompass an event where drivers failed to take action but didn't crash due to luck. Additionally, this study does not provide a concrete way to differentiate between a near-crash and a less severe event. A rapid evasive maneuver is also a subjective occurrence. While these are clearly not perfect, they are a reasonable starting point. Klauer et. al (4) acknowledge that it is subjective but take it further by defining a rapid evasive maneuver as steering, braking, accelerating, or a combination of control inputs that approach the vehicles’ limits. Both of these sources were part of the 100-Car Naturalistic Driving Study (NDS) where a set of event flags were implemented and analysts reviewed video footage of the points in trips where the flags were triggered.

Since researchers and officials have lacked the ability 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) came to the following conclusions about using near-crashes as a surrogate safety measure for crashes, using the 100 Car Naturalistic Driving Study:

- there is no evidence suggesting the causal mechanisms for crashes and near-crashes are different,
- there is a strong frequency relationship between crashes and near-crashes
- using near-crashes 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 also made an attempt gather information on near-crashes by crowd-sourcing crash and near-miss data from people who witnessed them in a program called Crash Stories NYC. In this program, witnesses of a crash or near-crash could go to a website with an interactive map, and fill out a survey of date, location, and a first-hand account of the incident. Again since this is crowd sourced, it is very subjective, especially considering that in this case no training was done to try and ensure consistency. This program appears to have very low participation in recent months with only 2 accounts in September 2014 (5).

The detection of crashes as near-crashes specifically using kinematic vehicle data is a relatively new problem since kinematic vehicle data has not really been available on a large scale in an uncontrolled setting. This has recently begun to change making the detection of safety-critical events using kinematic vehicle data a much more relevant topic to pursue, especially with the commitment to V2V technology. Most of the work that has been done is either threshold-based or time-to-collision (TTC) based. While TTC metrics are likely to be able to contribute to detecting near-crashes in a fully saturated connected vehicle environment, but in a testbed situation where a small percentage of the vehicles are equipped with V2V, using TTC is not feasible. Still, even the ones with TTC as an input are worth mentioning to include in future work, with a fully saturated connected vehicle environment being a real possibility in the near future (6).

Up through now, the primary model for detecting a near-crash using kinematic data has been
through the use of a threshold or set of conditions that must be met to flag an event. The 100-Car (1) and SHRP2 (7) Naturalistic Driving Studies conducted by Virginia Tech Transportation Institute (VTTI) probably provide the best way to study near-crashes. In both these studies, drivers were recruited to have their vehicles equipped with cameras and data-acquisition systems (DAS). These drivers then continued on with their daily lives while VTTI was able to collect kinematic and video data for real driving in an uncontrolled setting. The video data allows anything that gets flagged by a model to be verified.

Most VTTI studies involving near-crashes or crashes used the set of flags shown in Figure 4 to indicate a possible near-crash or crash event. Table 1 shows the percent of valid events that were detected by each flag, and 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>
<thead>
<tr>
<th>Trigger Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Lateral Acceleration</td>
<td>Lateral motion equal to or greater than 0.7 g.</td>
</tr>
<tr>
<td>2. Longitudinal Acceleration</td>
<td>Acceleration or deceleration equal to or greater than 0.6 g.</td>
</tr>
<tr>
<td></td>
<td>Acceleration or deceleration equal to or greater than 0.5 g coupled with a forward TTC of 4 s or less.</td>
</tr>
<tr>
<td></td>
<td>All longitudinal decelerations between 0.4 g and 0.5 g coupled with a forward TTC value of ≤4 s, and with a corresponding forward range value at the minimum TTC not greater than 100 ft.</td>
</tr>
<tr>
<td>3. Event Button</td>
<td>Activated by the driver by pressing a button located by the rearview mirror when an event occurred that the driver deemed critical.</td>
</tr>
<tr>
<td>4. Forward TTC</td>
<td>Acceleration or deceleration ≥0.5 g coupled with a forward TTC of 4 s or less.</td>
</tr>
<tr>
<td></td>
<td>All longitudinal decelerations between 0.4 g and 0.5 g coupled with a forward TTC value of ≤4 s, and with a corresponding forward range value at the minimum TTC not greater than 100 ft.</td>
</tr>
<tr>
<td>5. Rear TTC</td>
<td>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 0.3 g.</td>
</tr>
<tr>
<td>6. Yaw Rate</td>
<td>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.</td>
</tr>
</tbody>
</table>

Figure 4 - VTTI event flags (2)
<table>
<thead>
<tr>
<th></th>
<th>Percent of Valid Events Detected</th>
<th>Percent of Flagged Events that Are Invalid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lateral Acceleration</td>
<td>3.5</td>
<td>91.3</td>
</tr>
<tr>
<td>Longitudinal Acceleration</td>
<td>44.7</td>
<td>66.4</td>
</tr>
<tr>
<td>Event Button</td>
<td>8.4</td>
<td>69.9</td>
</tr>
<tr>
<td>Forward TTC</td>
<td>56.4</td>
<td>86.4</td>
</tr>
<tr>
<td>Rear TTC</td>
<td>4.6</td>
<td>59.9</td>
</tr>
<tr>
<td>Yaw Rate</td>
<td>21.7</td>
<td>91.1</td>
</tr>
</tbody>
</table>

The majority of studies involving crashes near-crashes discussed their criteria for detecting them using kinematic data. Since many of them 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 in Figure 5) increased as the detection rate went up. The model with minimum error was able to identify 10/11 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 it is clearly still prone to error since the radar may not necessarily be targeting the correct location. This problem was acknowledged and another VTTI report (8) said that in some cases, range rate was estimated based on video and the width of the lead vehicle due to this issue.
Another study from the SHRP2 NDS (9) uses the same verbal definition of a near-crash but filtered candidate events by 0.5g’s braking force or a steering input that resulted in a lateral acceleration of 0.4g’s. This study admitted that once the filtering process was done, 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 suggested a method to use Naturalistic Driving to screen for Surrogate Events (defined as crashes or near-crashes in these papers), through a series of methods (Figure 9). This study used data posted in VTTI’s online data warehouse, which will be discussed at a later point in this proposal. The first screening method was a simple threshold optimized to the sensitivity and specificity of the lateral accelerations (Figure 10). The result is a set of candidate surrogate events for a 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 can be used as a surrogate for crashes (10). After this is completed, a conversion factor can 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 6. Surrogate Detection Model (10)

Figure 7 - Sensitivity Analysis of Thresholds for Conflict Detection, Jovanis Northwestern Presentation
Another paper, this one by Talebpour et al. (12) used the NGSim dataset and proposed two methods for detecting near-crashes, both using TTC metrics. The NGSim dataset contains vehicle trajectory data at two locations, collected via video data reduction (13). The recommended method by Talebpour et al. is shown in Figure 8. This approach is done by calculating a normal distribution for each driver’s longitudinal acceleration and flagging any event when an acceleration with a probability of occurrence less than the predefined value. Then the situation was examined for hard braking with due to a conflict with a lead vehicle or hard braking by a following vehicle. The approach seems reasonable but their recommendation to use this method was based on results that seemed most realistic, although they did not say why the thought it was realistic. It illustrates the importance of individual driver preferences in detection of near-crashes using acceleration and TTC information (12).

![Figure 8 - Method 1](image)

Another paper have used NDS data to study risky driving in teens used a simple threshold trigger that was lower than the NDS studies listed since it was used to detect “risky driving” and not near crashes (14). Smith et al. also examined accelerations in crash imminent scenarios to quantify different regimes for collision avoidance systems (15). Lastly, an alternative approach was taken by Gordon et al. where know run-off-the-road hot spots in Northern Virginia were taken and NDS
data was examined for difference in speed entering and exiting the segments and yaw rates at different points. This paper showed that it is viable to do this type of analysis to learn more about vehicle trajectories at hot spot locations (16).

Based on previous findings, a method for detecting crashes and near-crashes will be designed using components in the basic safety message. Due to the low market penetration of V2V technology in the testbed setting TTC will not be considered for the preliminary algorithms. Various modeling techniques will be explored to predict if a BSM reading occurred during a crash.

**Method**

The goal of this section is to describe the methods used to develop a way to detect crash/near-crash events using elements in the Basic Safety Message. While some techniques have been discussed to detect crashes and near-crashes in a connected vehicle environment, the actual application of those techniques is difficult. Many of these algorithms use TTC as a metric for detection, which is not really feasible with a small percentage of vehicles with V2V technology. TTC information is likely to provide insight but the goal is to have a method that is feasible with a relatively small number of vehicles equipped with V2V. Additionally some of these flags were developed with the knowledge that video data was available to check the results, which may have influenced the designers to choose slightly more liberal criteria for flagging events.

The goal here is to develop a model that does not use TTC and has a minimal error rate to detect crashes using BSM elements. The first step is to discuss the data sources used in building the models. Next the methods used for modeling crash/near-crash events will be described.

**Data**

This section will provide a detailed description of the training and test sets used for the model building process. All of the data was acquired from the Naturalistic Driving Study conducted by VTTI. Developing the models off 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 is a set of fourteen crashes that occurred during the 100-Car NDS. Three data files were associated with each crash:

*Trip Log* – Table containing various dynamic, geographic, and time-related data. Data was collected from the vehicles’ data bus through a Data Acquisition System (DAS) designed by VTTI at a frequency of 10 Hz. Data elements collected included speed, 3-
direction acceleration, and yaw rate. Figure 9 shows a time-series of longitudinal accelerations for two of these events.

**Time Series of Crash Event**

![Graph showing time series of crash event with acceleration values ranging from -0.5 to 1.0 over time (seconds) from 0 to 600.]

**Figure 9. Two Time Series of Longitudinal Accelerations Culminating in Crashes**

*Front Video* – Video file containing the driver’s view through the front windshield. A timestamp was present 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 10 shows a screen shot from one of the front video files.
Rear Video – Video file containing footage of the trip out the back windshield of the vehicle. In this case footage was black and white, and a timestamp was not present, so in the event that 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 11 shows a screen shot from one of the rear video files.
The trips varied in crash type, time of day, length of trip, and outdoor conditions. The data was cleaned by VTTI, by removing the trip origins making the personally identifying information unavailable to us as analysts. Since, all of the trips culminated in a crash, so 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 bins, is shown in Figure 12. The average was 10.4 minutes.

![Trip Lengths of Algorithm Training Data](image)

**Figure 12. Trip Length Summary**

After a closer review of the trip logs, one of the crashes had to be removed due to a block of missing data at the time of the crash. This left thirteen useable crashes with which to model. The video files were also examined to understand information about the exterior conditions and crash types. Table 2 shows the distribution of crash type, lighting, and weather for the thirteen remaining crashes. In all cases of precipitation, the type of precipitation was rain. It should be noted that one of the nighttime crashes had precipitation present as well.
Table 2 - Left: Crash Type Distribution, Right: Crash Conditions

<table>
<thead>
<tr>
<th>Crash Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rear End</td>
<td>9</td>
</tr>
<tr>
<td>Sideswipe</td>
<td>1</td>
</tr>
<tr>
<td>Lane Departure</td>
<td>1</td>
</tr>
<tr>
<td>Angle</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day</td>
<td>11</td>
</tr>
<tr>
<td>Night</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear</td>
<td>11</td>
</tr>
<tr>
<td>Precipitation</td>
<td>2</td>
</tr>
</tbody>
</table>

Once the video data was 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 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. This was done manually using personal judgment.

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 same set. As a result, two different data sets were used for testing, both also from the NDS. The first set used for testing was a set of another 14 trips, this time of trips with no crashes or near-crashes. These trips contained significantly more time of driving, amounting to roughly 10 hours combined across the 14 trips. This set of data will be used to test the false positive rate (False alarms per hour) for the models developed from the first trip. Since there is no video data, it is not possible to go verify that a near-crash did not occur if the model flags an event, but since VTTI said that no crashes or near-crashes occurred when the data was acquired, the assumption that any detection is a false positive seems reasonable, at least for exploratory purposes. Figure 13 shows the longitudinal acceleration profile of a trip from this data set. Notice the difference in scales on both axes.

Comment [b1]: This is confusing – you seem to imply that you are using the same data set for training as you do for testing – which I do not believe is the case, correct?
The third set of data is set of 68 crash events and 760 near-crash events and is posted on VTTI’s data warehouse (17). This data set includes 30 seconds of pre-event and 10 seconds of post-event 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. This set also had no video, but due to the short duration of the time series, it can be assumed that any point where a crash or near-crash indicated by a model was indeed the point of the crash. Additionally, there was an event narrative file where VTTI staff wrote a short account of what occurred during each crash and near-crash (19). This dataset was used to test the sensitivity (false negative rate) of the model on a test set. A sample time series of longitudinal and lateral accelerations are shown in Figure 14.
Figure 14. Longitudinal and Lateral Accelerations from and Crash Event Online

Models

A few different modeling techniques will be applied to predict if a data point is a part of a crash/near-crash event. The first ones will be the threshold-based models VTTI uses as its first screening method. The threshold-based results will be used as a benchmark for comparison to estimate model performance. Other models that have been tested include Classification and Regression Trees (CART), Multivariate Adaptive Regression Splines (MARS), and a pattern matching algorithm.

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. 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, while employing a reasonable threshold will not identify what is occurring at 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 it is constructed. A slightly different approach to
aggregation was taken in each modeling technique so this will be discussed further in each section.

Figure 15 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.

![Sample Time Series of Longitudinal Accelerations, Rear End Crash at Time 170 Seconds](image)

Figure 16 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.
Pattern Matching Algorithm

By examining figure 13 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 common 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 17. 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 16. Some Acceleration Profiles during Crashes
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 18 shows a crash (black dots) compared to the baselines from figure 17. 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.

![Comparison of Crash with Baseline Readings](image)

**Figure 18. Crash comparison (Black Dots) with baseline readings**

**Classification and Regression Trees**

The first attempt at modeling the data was done 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:

- Max Longitudinal Acceleration
- Min Longitudinal Acceleration
- Mean Longitudinal Acceleration
- Variance Longitudinal Acceleration
- Max Lateral Acceleration (after taking absolute value)
• Mean Lateral Acceleration
• Variance Lateral Acceleration
• Max Z-direction Acceleration
• Min Z-direction Acceleration
• Median Z-direction Acceleration
• Mean Speed (Due to sampling rate I had to interpolate values for speed, used R’s spline function)
• Max Speed
• Min Speed
• Median Speed
• Min Pattern Matching Distance (pattern matching paper)

CART is a recursive technique that chooses the best variable to split the data by 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 a very interpretable decision tree and the ability to make decisions for data organized in the manner shown in Figure 19. One drawback is that 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 19. Decision Trees Separating Excel at Data Organized in This Manner, (20)

Using the “rpart” package in R (21), 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 present below in Figure 20. 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 ROC curve is shown in figure 21 and the score table is shown in table 5 for the training data set. The false positive rate is shown in table 6. Clearly there are a lot more false positives in the CART model, but that is likely due to the overlapping nature of the windows (i.e. if one point has a min longitudinal acceleration and max 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 soon.
Figure 20. Entropy Tree
**Multiple Adaptive Regression Splines**

Using the same aggregated data as was used in the CART model, a MARS model was constructed. MARS is a technique proposed by Freidman (22) in which a piecewise regression function can be built. It works by building hinge functions that can change the trajectory of the model based on what trajectory the data follows MARS has the benefits of being able to provide very good fits to the data and 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 (23) was used. MARS models predict a probability for each data point being true, so a threshold was 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 76960 windows. 248 of the total windows spanned a positive crash reading.

**Results**

The Result section will discuss the predictive capacity of each of the three methods above on the training data and the false positive rate on the test data. The online data has 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 12/13 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 3 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. The detection rate results for the online, shown in Table 4, data showed no statistically significant difference between the 0.5g threshold and the pattern matching with 95% confidence, with an n=68. However, given the benefits of a reduced false positive rate the pattern matching algorithm may still be worth while.

<table>
<thead>
<tr>
<th>Detection Rate</th>
<th>Pattern Matching</th>
<th>Threshold 0.6g's</th>
<th>Threshold 0.4g's</th>
</tr>
</thead>
<tbody>
<tr>
<td>False Positives</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 3 - Pattern Matching Training Data Detection Rate
Table 4 - Pattern Matching Results Online Data Set

<table>
<thead>
<tr>
<th></th>
<th>Crash</th>
<th>Near-Crash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern Match</td>
<td>0.691</td>
<td>0.737</td>
</tr>
<tr>
<td>0.5 Threshold</td>
<td>0.603</td>
<td>0.663</td>
</tr>
<tr>
<td>0.6 Threshold</td>
<td>0.485</td>
<td>0.486</td>
</tr>
</tbody>
</table>

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.

For the CART model and the MARS models, the windows created overlap so there are nearly as many windows as total readings. That is why the total number of positive readings is 248, even though there were only 13 actual crashes. The following score tables show the detection rate for each algorithm, but 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 5 shows the score table for the VTTI NDS Thresholds, used as a benchmark for the other models. Table 6 shows the predictions and error rates for the CART model. The recall (sensitivity) of the CART model is 199/248 and the precision is 199/221. The ROC (Figure 21) curve shows that 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 and the table 7 shows that the model did not carry over to the test data as well as the NDS threshold.

Table 5 - NDS Threshold Score Table Results

<table>
<thead>
<tr>
<th>VTTI NDS</th>
<th>Actual</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pred</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>76680</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>32</td>
<td>128</td>
<td></td>
</tr>
</tbody>
</table>
Table 6 - CART Score Table

<table>
<thead>
<tr>
<th>CART Model</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pred</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>76690</td>
</tr>
<tr>
<td>1</td>
<td>22</td>
</tr>
</tbody>
</table>

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

Figure 21. ROC Curve for CART Tree
Table 7 - CART Test on Normal Driving vs NDS Triggers

<table>
<thead>
<tr>
<th>Actual</th>
<th>Pred</th>
<th>CART</th>
<th>VTTI NDS Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>368056</td>
<td>369051</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1025</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

The MARS model’s performance on the training data is shown in Table 8 improved both the Precision and the Recall over the CART model and the VTTI Thresholds on the training data. It also resulted in only 16 false positives on the normal driving compared to the 30 false positives from the VTTI Threshold.

Table 8 - MARS Model Results on Training Data

<table>
<thead>
<tr>
<th>MARS Model</th>
<th>t=0.8</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>76703</td>
<td>56</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>192</td>
</tr>
</tbody>
</table>

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

Table 9 - Results of Model on Normal Driving Compared to NDS Threshold

<table>
<thead>
<tr>
<th>Actual = 0</th>
<th>Pred</th>
<th>MARS</th>
<th>VTTI NDS Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>369065</td>
<td>369051</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

Discussion

This section will be devoted to discussing which detection method should be carried out in an application setting. In the pattern matching methodology, sensitivity analysis was brought up as the justification for numerous decisions that were 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 that 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. 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 practically nor a statistically 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.

Some additional approached can also be tried 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. The idea behind this is for an ARIMA model for each driver to 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. Those could include logistic regression, neural networks, and support vector machines.

Conclusions and Recommendations
In the discussion section, the models’ benefits and weaknesses were discussed. The next important step to this research is to design a methodology that can be carried out on CVI-UTC Northern Virginia connected vehicle testbed data once there has been a sufficient volume of data collected. Network screening is a process that, in its current state, requires at least three years of crash data, making it a highly reactive process requiring the public to be subject to sub-optimal or dangerous driving conditions until enough crashes build up. Using connected vehicle basic safety messages provides infrastructure providers with a new type of data that will provide more insight than ever into what is actually occurring on the roads.

By inputting BSM data collected in the testbed into these models, a list of readings that are likely crashes and near-crashes can be obtained. Then the GPS coordinates for each crash/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/near-crashes
- Developing a prototype method to use BSMs to detect hot spots
- Providing an avenue for researchers to understand near-crashes
- Showing a relationship between crashes and near-crashes
References