

UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

TRAFFIC ACCIDENT ANALYSIS AND PREDICTION USING THE NPMRDS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

By

MOHAMAT EIRBAN ALI BIN KAJA NAJUMUDEEN

Norman, Oklahoma

2020

TRAFFIC ACCIDENT ANALYSIS AND PREDICTION USING THE NPMRDS

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Dr. Hazem H. Refai, Chair

Dr. Choon Yik Tang

Dr. Samuel Cheng

© Copyright by MOHAMAT EIRBAN ALI BIN KAJA NAJUMUDEEN 2020
All Rights Reserved.

Dedication

*To my mom and dad,
Johara Banu and Kaja Najumudeen,
my brother, Azlan,
my beloved family
&
friends.*

ALL PRAISE IS TO THE ONE AND ONLY GOD.

Acknowledgements

My utmost gratitude to my mentor, Dr. Hazem Refai, for his never-ending support and guidance. This work would not have been completed without his encouragement and motivation. I also want to thank Dr. Thordur Runolfsson, Dr. Gregory McDonald, and Dr. Samuel Cheng for imparting their knowledge to me through the classes I have attended in OU – Tulsa. My appreciation and thanks also to Dr. Choon Yik Tang, as well as Dr. Refai and Dr. Cheng, for agreeing to be part of my master’s defense committee. My thanks to Dr. Obadah for his technical advice in approaching Machine Learning problems. I also want to acknowledge support from my colleagues and friends, especially Mohamed Afify and Abdullah. My thanks to Michelle for doing an amazing job at reviewing and editing the thesis. Finally, I want to recognize the support and great love of my family and friends.

Table of Contents

1. Introduction.....	1
1.1 Traffic Congestion	2
1.2 National Performance Management Research Data Set (NPMRDS)	3
1.2.1 Background and details of NPMRDS	3
1.3 Thesis Objective	4
2. Related Work	7
3. NPMRDS Data Acquisition and EDA.....	11
3.1 Utilizing the NPMRDS Webserver	12
3.1.1 Traffic Speed data and Incident data acquisition.....	12
3.1.2 Other features of NPMRDS Webserver.....	15
3.2 Exploratory Data Analysis	21
4. Real Time Accident Detection.....	35
4.1 Datetime parsing for Traffic Speed data	35
4.2 Feature Extraction	35
4.2.1 Incident event matching to Traffic Speed data observation.....	37
4.2.2 Supervised Machine Learning Input/Output Generation.....	38
4.3 Supervised Machine Learning Classification Models	40
4.3.1 Logistic Regression.....	41
4.3.2 Logistic Regression + Multi Adaptive Regression Spline (MARS).....	43
4.3.3 Support Vector Machine (SVM).....	45
4.3.4 Random Forest	47
4.3.5 Recurrent Neural Network (RNN).....	49

4.3.5.1 RNN Data Preprocessing	50
4.3.5.2 RNN Architecture	52
4.3.5.3 Long Short-Term Memory.....	53
4.3.5.4 Gated Recurrent Unit (GRU).....	55
4.3.6 Summary of Classification Modelling Results	57
4.3.7 Feature Importance	59
5. Post Processing Classification Modelling.....	60
5.1 Comparison between Dynamic Time Wrapping and Zero Padding.....	60
5.1.1 Data Shifting to obtain 3-hour 30-minute observation	64
5.1.2 Filter Congestion and Accident speed observation.....	65
5.1.3 Results.....	66
5.2 LSTM modelling	69
6. Conclusion and Future Work	74
References	76
Appendix	80

List of Tables

Table 4-1. Datetime parsing for the traffic speed dataset.	35
Table 4-2. Original dataset with missing timestamp.	36
Table 4-3. Generating missing timestamp observations.	37
Table 4-4. Dataset completed after matching traffic feature with incident.	37
Table 4-5. Detection and False Detection Rate for Logistic Regression.	43
Table 4-6. Detection and False Detection Rate for MARS + Logistic Regression	45
Table 4-7. Detection and False Detection Rate for SVC	47
Table 4-8. Detection and False Detection Rate for Random Forest	49
Table 4-9. Detection and False Detection Rate for LSTM.	55
Table 4-10. Detection and False Detection Rate for GRU.	57
Table 5-1. LSTM Zero-Padding Per Class Accuracy Rate.....	67
Table 5-2. LSTM Zero-Padding Per Class Accuracy Rate.....	68
Table 5-3. LSTM Oklahoma Highway I-35 Per Class Accuracy Rate.....	70
Table 5-4. LSTM Segment 1 Per Class Accuracy Rate.....	72
Table 5-5. LSTM Segment 2's Per Class Accuracy Rate.....	73

List of Figures

Figure 3-1. NPMRDS Data Acquisition Web Login.....	11
Figure 3-2. Database selection page	12
Figure 3-3. NPMRDS Dashboard.....	13
Figure 3-4. Traffic Speed dataset.....	14
Figure 3-5. Road Segment Information.....	14
Figure 3-6. Incident Analysis Generator Page.....	15
Figure 3-7. Generated Incident dataset.....	15
Figure 3-8. NPMRDS Route Analysis.....	16
Figure 3-9. Speed Distribution for Selected Segments.....	16
Figure 3-10. Travel Time distribution for selected Segments.....	17
. Figure 3-11. Congestion Analysis layout.....	17
Figure 3-12. Heatmap showing distribution of traffic congestion by traffic speed across the Highway I-35.....	18
Figure 3-13. The distribution of speed per segment across Highway I-35 for the longest occurring congestion by distance.....	18
Figure 3-14. Bar plot of Segment based on frequency of congestion by hour.....	18
Figure 3-15. Frequency plot of segment ranking.....	19
Figure 3-16. Performance Measures parameter filter.....	19
Figure 3-17. Road Performance Measures and Freight Movement information.....	20
Figure 3-18. Deployment of Snowplow Trucks and collected data.....	20
Figure 3-19. Accident distribution by Hour of the day for Highway I-35.....	22
Figure 3-20. Accident distribution by Hour of the day for Highway I-40.....	23

Figure 3-21. Accident distribution by Hour of the day for Highway I-44.....	24
Figure 3-22. Accident distribution by Day of the week for Highway I-35.....	25
Figure 3-23. Accident distribution by Day of the week for Highway I-40.....	26
Figure 3-24. Accident distribution by Day of the week for Highway I-44.....	27
Figure 3-25. Accident distribution by Month for Highway I-35.	28
Figure 3-26. Accident distribution by Month for Highway I-40.	29
Figure 3-27. Accident distribution by Month for Highway I-44.	30
Figure 3-28. Plot of most frequent accident occurring road segments (Southbound).	31
Figure 3-29. Plot of most frequent accident occurring road segments (Northbound).	32
Figure 3-30. Temporal distribution by Hour for accident prone segment (Southbound).	33
Figure 3-31. Temporal distribution by Hour for accident prone segment (Northbound).	33
Figure 3-32. Plot change in speed when no accidents occurred.	34
Figure 3-33. Plot change in speed when an accident occurred.	34
Figure 4-1. Dataset pre-processing for Supervised Learning	38
Figure 4-2. Final dataset after Incident matching.	39
Figure 4-3. Timestep at label creation.	40
Figure 4-4. An example of Logistic Regression application	42
Figure 4-5. Logistic Regression’s Test Confusion Matrix.....	43
Figure 4-6. An example of MARS classification application.....	44
Figure 4-7. MARS + Logistic Regression’s Test Confusion Matrix.....	45
Figure 4-8. An example of SVM classification algorithm	46
Figure 4-9. SVC’s Test Confusion Matrix.....	47
Figure 4-10. An example of how decision tree works	48

Figure 4-11. Random Forest’s Test Confusion Matrix.....	49
Figure 4-12.. An unrolled depiction of a single RNN.....	50
Figure 4-13. An example of data before being transformed to be suitable for RNN.	51
Figure 4-14. An example of data after transformation for RNN.	51
Figure 4-15. Dataset pre-processing for RNN.	51
Figure 4-16. RNN Model architecture.	53
Figure 4-17. Regular RNN internal structure.	54
Figure 4-18. LSTM internal structure.	54
Figure 4-19. LSTM’s Test Confusion Matrix.....	55
Figure 4-20. GRU internal structure.	56
Figure 4-21. LSTM vs GRU Gates.	56
Figure 4-22. GRU’s Test Confusion Matrix.	57
Figure 4-23. Classification model accuracies.	58
Figure 4-24. Random Forest Feature Importance.	59
Figure 5-1. DTW observation stretching with high constant sample.	62
Figure 5-2. DTW observation stretching with low constant sample.	63
Figure 5-3. Data shifting to produce 3-hour and 30-minute window.	64
Figure 5-4. LSTM Zero-Padding’s Test Confusion Matrix.....	67
Figure 5-5. LSTM DTW’s Test Confusion Matrix.....	68
Figure 5-6. LSTM I-35’s Test Confusion Matrix.	70
Figure 5-7. LSTM Segment 1’s Test Confusion Matrix.....	72
Figure 5-8. LSTM Segment 1’s Test Confusion Matrix.....	73

Abstract

Traffic accidents are incidents caused by collisions between road vehicles or a vehicle with road infrastructures or pedestrians. Traffic accidents are a common cause for non-recurring traffic bottlenecks that, in turn, cause trip delays, an increase in fuel consumption and vehicle usage, and at the worst, loss of life and property. As part of this thesis, we were granted access to the Federal Highway Association's (FHWA) National Performance Research Management Data Set (NPRMDS), which provide probe speed, average segment speed, reference speed, and travel time per segment, among other information. Statistical analysis is applied to the accident occurrence on Oklahoma roads, especially the I-35 highway corridor for the duration between 2017 and 2020 to show the effect of temporal and spatial factors, such as road segment and its geometry, time of day, day of the week, and month of the year. Multiple methodologies involving machine learning and deep learning were utilized to model accident detection using traffic speed data. Our desired outcome is ensuring a fast reaction time from an emergency response team. We produced a deployable model capable of providing a reliable detection of accident occurrences as an implementable alert system for the concerning state bodies. Using this approach, we were able to train an optimized Random Forest model, which detected 89.68 % of accidents with only a 13.92 % false detection rate. These are promising results for a real-time data environment. Speed turbulence classification was also implemented as a post processing application for classifying samples into free flow, congestion, and incident event based on historical data. The LSTM model outperformed others, especially when modelling is specified to a specific road segment. Accuracy was

measured at above 87% in classification with greater than 75% accuracy in correctly classifying congestion and accident events.

1. Introduction

Traffic accidents are a major cause of non-health related fatalities on the global stage and are the leading cause of death for children and young adults aged between 2 and 29 years old. According to World Health Organization, approximately 1.35 million people die worldwide each year due to road accidents [1]. In 2018 alone in the United States of America, the total number of fatal accidents was 33,654, accounting for 36,560 deaths [2]. In Oklahoma alone, the total number of fatal accidents was 603 with 655 deaths (or 16.6 deaths per 100,000 individuals and 1.44 deaths per million miles. These figures are higher than the national average of 11.2 deaths per 100,000 people and 1.44 deaths per million miles.

In addition to critical lives lost, traffic accidents also indirectly impact the economic health of our country, especially when considering traffic congestion. Traffic accidents lead to traffic congestion, with intensity usually dependent on the severity of the accidents, as well as the geometry and condition of the road. Congestion resulting from accidents could cause a bottleneck effect that drain fuel, causes increased wear and tear on vehicles, and leads to a decrease in road user's productivity and increase in wasted time.

Obviously, the detection and prevention of accidents could lead to much-needed improvements in road building strategies, as well as decreased fiscal spending by improving identification of road sections that are historically prone to accidents. In a more

ambitious effort, real-time accident detection could result in faster response time with increased chance of survival for accident victims.

1.1 Traffic Congestion

Traffic congestion occurs for any number of reasons. Congestion often times causes bottlenecked traffic flow, resulting in slow-downs and stops when compared to the natural flow of traffic. Causes for traffic congestion can be divided into three main categories: 1) recurring events, 2) non-recurring events, and 3) continuous events, which can be further subdivided into seven primary reasons [3]. Recurring events include:

1. Demand fluctuations when road usage changes due depending on day and hour. Because road capacity remains fixed at all times, spontaneous demand can lead to unexpected traffic congestion.
2. Repetitive events resulting from social events (e.g., concerts, Black Friday shopping). Such recurring events are known to cause high traffic volumes that far exceed standard road capacity.

Non-recurring events include:

1. Traffic Incidents (or accidents) arising from vehicle-to-vehicle, vehicle-to-infrastructure, or vehicle-to-pedestrian incidents.
2. Work Zones – are characterized as roadway construction areas affecting road infrastructure or roadside buildings; these areas force motorists to use either part or an alternate roadway to continue their travels.
3. Weather – changes due to precipitation, dim light/bright sunlight, or slippery roads cause hazardous roadways and/or visibility and resulting in decreased traffic speeds.

These weather conditions usually arise from precipitations, low light or bright light from sun, and slippery roads from accumulation of precipitations.

The last category is known as Continuous causes include:

1. Traffic road infrastructure used as traffic control devices (e.g., traffic lights, railroad crossings, and others) occasionally fail or function inefficiently, causing traffic flow disruption and/or congestion.
2. Inadequate base capacity resulting from a poorly built roadway system with inadequate amount of physical capacity (i.e., limited width, number of lanes, merge connections, and/or alignment and condition of the road). Such factors limit traffic volume.

1.2 National Performance Management Research Data Set (NPMRDS)

The Federal Highway Administration (FHWA) has long sought to quantify related metrics to traffic management and road operations, including travel time reliability and traffic congestion. As part of FHWA's initiative to encourage state departments, especially the departments of transportation (DOTs), to adapt these traffic performance metrics, the FHWA offered the National Performance Management Research Data Set (NPMRDS) that gives details of travel time measures. The NPMRDS, together with data collected by Oklahoma DOT, serves as the main focus for analysis for this thesis.

1.2.1 Background and details of NPMRDS

In 2013, the U.S. Federal government initiated a strategy to obtain a nationwide-based dataset composed of average travel time and performance measures for inclusion in the Freight Performance Measures (FPM) and Urban Congestion Report (UCR) [4]. For

optimal utilization of data for the UCR, the acquisition of NPMRDS was done by the FHWA's Office of Operations at which the dataset covers the entire National Highway System (NHS). The implementation of UCR was aimed at improving travel time reliability measures, supporting local state DOT decisions and developments, and demonstrating uses for the NPMRDS [5].

As such, the probe data providing the information for NPMRDS's dataset was initially contracted to HERE and later given to INRIX [6], NPMRDS's probe data is a spatial-temporal dataset with 5-minute granularity, which then transmits information to a central server. Unfortunately, traffic volume data is not collected. The NHS is segmented using a Geographic Information System (GIS), where time-based data are binned for every 5-minute interval per segment. Tabular information representing each road segment is also included as a separate file when traffic speed data is downloaded locally. NPMRDS's data obtained based on moving vehicle probes. Notably, the consistency and count number of data per segment per epoch is not constant with influence from traffic flow, date/time information, or location.

1.3 Thesis Objective

This thesis is written with the objective of furthering the utilization of the NPMRDS, especially when taking into consideration previous work related to congestion analysis reported in [7]. In this thesis, the focus is limited to the analysis of accidents, which are divided into two major categories for a) near real-time detection and the classification thereof, and b) accident, congestion and free-flow classification of change in speed observations using historical traffic speed data. This thesis explores data acquisition from

NPMRDS sources; various user interface-based data summaries through the NPMRDS web interface; exploratory data analysis for accident observations; multiple methodologies for preparing and modelling data; and validating such models. One objective was preparing data for supervised learning algorithms and further modifying the data for Recurrent Neural Network applications for both real-time and post-processed classification. The thesis also describes a filtering method to obtain speed turbulent observations resulting from a recurrent and non-current traffic congestion.

The main contributions of this thesis are summarized below:

- Describing the importance of utilizing data features as accident predictors, especially related to the effect of modelling using only speed features for comparison, including multifeatured modelling.
- Determining the optimal method for preparing the NPMRDS to allow for appropriate sample observations for various supervised learning algorithms) (e.g., data cleaning, pre-processing techniques, and feature engineering).
- Training multiple models for various supervised learning and recurrent neural network algorithms and comparing model performance using model validation to develop an optimal model for real-time accident detection that can be deployed in the future.
- Discussing the methodology for preparing and filtering data that shows obvious speed turbulence in a post-processing setting with a goal of producing a viable model training method to determine the difference between recurring traffic congestion and traffic accidents. This includes methods for preparing varying

lengths of data for machine learning applications using Dynamic Time Warping and Zero-padding.

As per the aforementioned stated objectives, this thesis is divided into multiple sections. Section 1 contains an introduction and background information about the NPMRDS, as well as thesis objectives. Section 2 summarizes various works that have investigated and analyzed traffic speed and accidents, including the predictive capabilities of various machine learning and deep learning frameworks for estimating various traffic parameters. Also, this section discusses the feature importance of traffic accidents, including how data has previously been handled before any analysis. Section 3 explains the exploratory data analysis performed on acquired data from NPMRDS and showcases various distribution plots that provide features relevant to the traffic accidents which, in turn, could be beneficial in establishing an approach for preparing data for modelling. Section 4 highlights information related to primary objectives of this thesis (e.g., utilize the NPMRDS to create a near real-time traffic accident detection model via a machine learning algorithm application. Section 5 focuses on data preparation for long sliding window duration in a post-processing setting aimed at distinguishing speed turbulence occurrences in historical data, and then classifying the cause of speed turbulence as a consequence of either recurring traffic congestion or non-recurring traffic accidents.

2. Related Work

As part of successful idea generation for this thesis, a specified literature research was completed on related topics, especially accident detection/prediction, congestion-based traffic analysis, and feature usage for various traffic application and parameter estimation. The resulting review was narrowed to only include works directly related to the research conducted for this thesis. It is important to note that not much attention has been granted to real-time accident prediction using real-time traffic data. Instead, most research has focused on simulation-based modelling or discovering various associations with the broad number of features and variables surrounding traffic parameters before, during or after an accident.

The first focus of this literature review is acquisition and utilization of data and features used for predicting and analyzing accidents. The authors in [8], identified four major data categories for identifying possible traffic accidents: 1) human actions, 2) human conditions, 3) environmental conditions, and 4) vehicle conditions. Researchers in [9] validated these data features, dividing them into four different categories: 1) driver factors, 2) environmental factors, 3) road factors, and 4) vehicle factors. Many of these attributed factors consist of additional characterizations, including:

- Driver factors—sex, age, driving experience, collision history, physical, and mental conditions.
- Environmental factors—weather, visibility, rain/fog/precipitations, and date/time
- Road factors—road type, location, geometry, surface condition, traffic control, maximum traffic speed, and traffic volumes.

- Vehicle factors—vehicle type, condition, and maintenance history; speed, location, maneuver type/direction.

Researchers in an Ottawa Case Study [10] collected weather, driver, vehicle, road, and event data, adding more granularity through feature engineering (e.g., datetime, solar positions, road and event features). A case study focused on Seoul City, South Korea [11] used weather variables correlation to accident severity, hypothesizing that rain was a major factor for accidents due to poor visibility and slippery road conditions. Hence, the research was focused on rainfall intensity and water level depth of rain collected. Data was collected from a nine-year period with reiteration of literature support from [12] where it was stated that rainfall may result in driving hazards. Results in [13] show the effect of rain and fog on traffic parameters. Research showed that rain has a much higher impact on the traffic than the fog. Data in [14] was acquired via a loop detector installed on the roads and analyzed against historical crash data in which features were collected and aggregated for incidents with similarities that were captured 5 minutes before the accident. Results in [15] demonstrate the importance of features like number of lanes and average speed at the intersection for predicting traffic accidents. A simulation study in [16] demonstrated best practices for using standard deviation of traffic volume, standard deviation of speed, standard deviation of occupancy, and standard deviation of travel time as accident predictors.

The second focus of this review is data processing. Regarding accident prediction reported in the Ottawa case study [10], it is important to note that data used for collision samples typically involves both real-time and historical data, although analysis for non-

collision class observations were provided through synthetic data generation. This process can be understood through the use of the following algorithm.

1. Randomly select one sample from the collision dataset (Sample1)
2. Randomly select a change to either road segment or hour of the day/day of the year.
3. Select sample with different value from collision dataset (Sample2)
4. Create non-collision dataset by combining Sample1 (i.e., change feature) with balance of features from Sample2.
5. Retain non-collision data if there is none in the collision dataset.

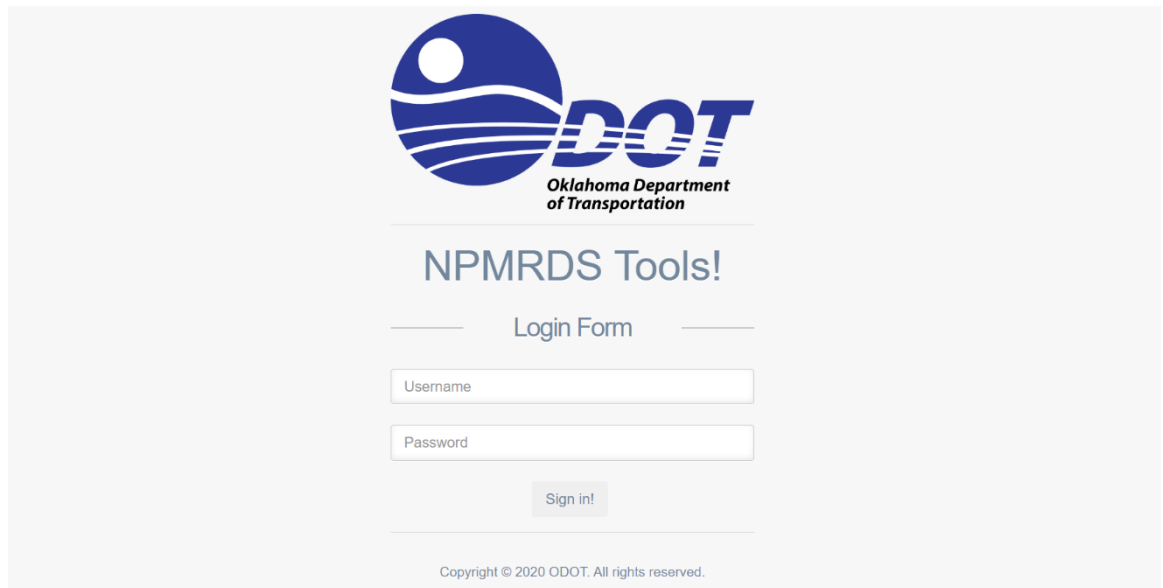
[16] described prediction of traffic accident based on multiple standard deviations of various traffic parameters using a simulation with both modelling and validation. Notably, this approach does not involve model validation using actual traffic speed data. Duration of a traffic accident prediction in [11] was determined using human observations by either a passerby or a traffic patrolman. Data processing in [14] leveraged real-time and historical data, which proved the best approach to modelling and validating the model. [16] based accident prediction using simulation data rather than real-world data.

The final focus of this review is various modelling approaches for predicting accident or other traffic parameters. Several academic papers describe an accident prediction framework using multiple machine learning approaches— the most recent case study shown in Ottawa, Canada [10]. Researchers trained a model using gradient boosted tree, which is an ensemble-type machine learning algorithm that strengthens the usually weaker prediction model (e.g., a decision tree with a accuracy of 79% and precision of 71%). However, as previously mentioned, non-collision samples were generated via simulation,

not obtained from actual observations. Researchers in [17] used a backpropagation neural network to train a model for identifying collision type, not collision occurrence. Prediction output was divided among single, rear-end, front, side, and scratch-based collisions. Accuracy was 89 %. Regarding neural networks, in [18] probabilistic neural networks were used with video data to achieve 92% accuracy and only 0.77% false rate for accident prediction. In [19], wavelet neural networks were used to predict road accident loss. While hybrid neural networks based on adaptive neuro-fuzzy technique (ANFIS) were used to predict traffic accidents [20] with 55.06% accuracy. Researchers in [21] developed a road risk index as part of a vehicle-to-vehicle (V2V) framework. The use of unsupervised learning through KNN in [22] reported 80% accident prediction accuracy using simulation.

3. NPMRDS Data Acquisition and EDA

NPMRDS is available for download in .csv format from the FWHA website [23]. Note that the URL will redirect you to a login page (See Figure 3-1) that requires authorized credentials granted to either ODOT personnel or ODOT affiliated organizations, such as the Wireless and Electromagnetic Compliance and Design (WECAD) Center at the University of Oklahoma. The website also provides visualizations and performance measure functionalities.



The image shows a web login interface for the Oklahoma Department of Transportation (ODOT). At the top center is the ODOT logo, which consists of a blue circle with a white sun-like shape and wavy lines below it, followed by the letters 'DOT' in a bold, blue, italicized font. Below the logo, the text 'Oklahoma Department of Transportation' is written in a smaller font. Underneath this is the heading 'NPMRDS Tools!' in a large, blue, sans-serif font. Below the heading is a 'Login Form' section, which is a horizontal line with the text 'Login Form' centered between two short horizontal dashes. The login form contains two input fields: 'Username' and 'Password', both with light gray borders and placeholder text. Below these fields is a 'Sign in!' button with a light gray background and a dark gray border. At the bottom of the page, there is a small copyright notice: 'Copyright © 2020 ODOT. All rights reserved.'

Figure 3-1. NPMRDS Data Acquisition Web Login.

Because part of the NPMRDS collection system was tendered to a third part contractor, the dataset has two versions. Information gathered between 2013 to 2016 were acquired by HERE, and since 2017 by INRIX (See Figure 3-2 for a screenshot of the FWHA website). Data for this paper was collected from the more recent dataset.

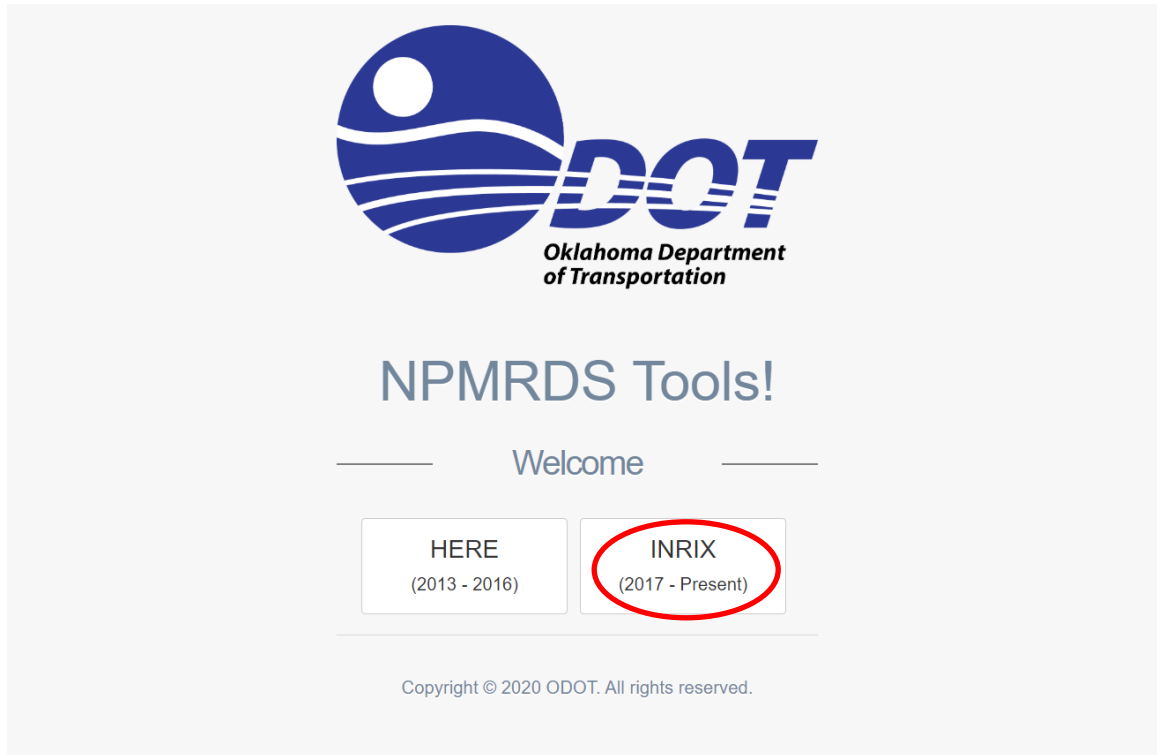


Figure 3-2. Database selection page

3.1 Utilizing the NPMRDS Webserver

3.1.1 Traffic Speed data and Incident data acquisition

The FHWA database will provide access to a number of functions designed for filtering speed data, analyzing route, congestion, and incident, measuring traffic performances; and evaluating Snowplow truck deployment. The main page (i.e., user dashboard) functions as the NPMRDS downloader (See Figure 3-3). From here, one can select the date, specific days, desired segments/highways, data source, and data averaging for a select time period. Data collection for this thesis centered on detecting an incident as close to real time as possible. Data averaging was selected and downloaded at a granularity of 5 minutes. The goal was obtaining a more granular dataset for machine learning or deep learning to characterize turbulence in speed and leverage other features to successfully

detect the incidents. The primary data source was used to train and validate models. Data was restricted to 2017 for Oklahoma highway I-35. Data was not restricted to vehicle type. Instead, each data point was considered acceptable, as traffic congestion significantly affects any localized vehicle, regardless of classification. The downloaded .csv file was stored as the primary data frame (See various features in Figure 3-4). A second .csv file containing road segment information is also included with the download (See Figure 3-5).

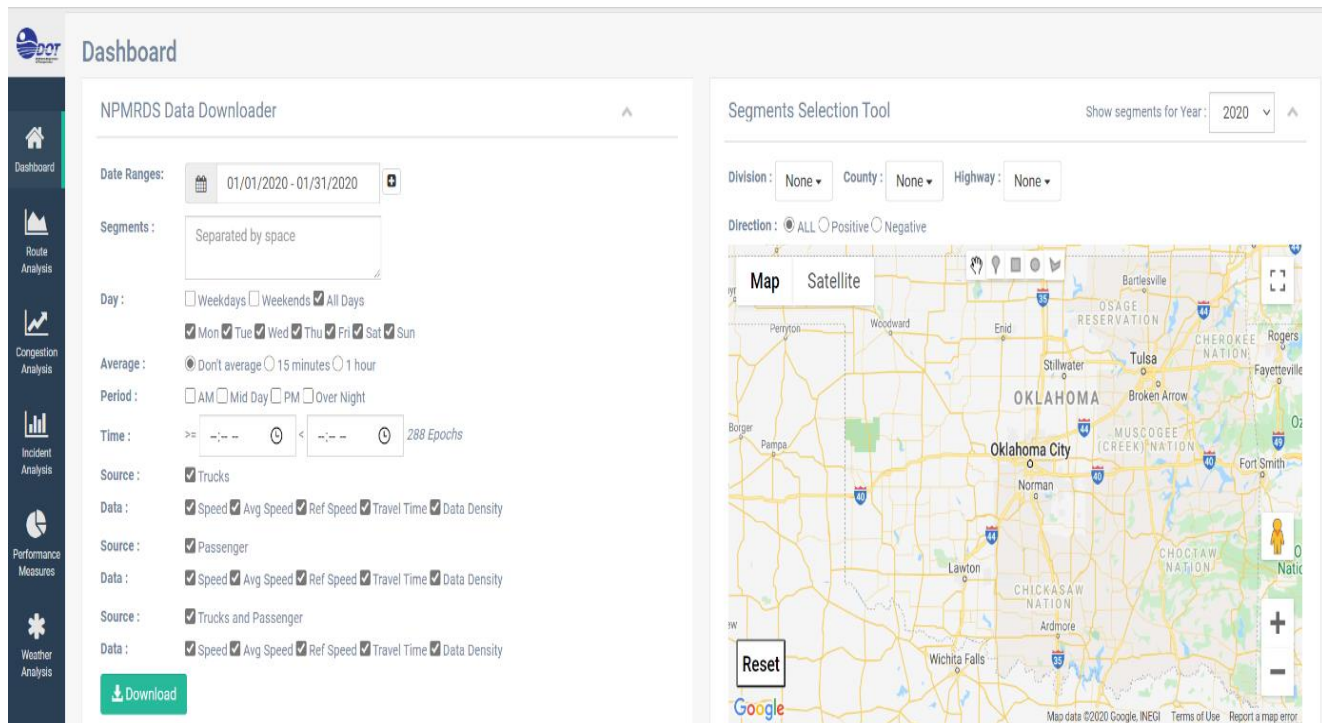


Figure 3-3. NPMRDS dashboard.

1	Date	Epoch	Segment	Speed	Average Speed	Reference Speed
2	6012017	0	111-06513	64	64	70
3	6012017	0	111P04943	58	29	66
4	6012017	0	111-05613	64	66	75
5	6012017	0	111P05088	66	64	71
6	6012017	0	111-05142	73	59	67
7	6012017	0	111N0558	69	67	72
8	6012017	0	111N0554	67	67	71
9	6012017	0	111-07522	36	54	68
10	6012017	0	111+06553	54	44	65

Figure 3-4. Traffic speed dataset

1	TMC	ROAD	DIRECTION	INTERSECTION	STATE	COUNTY	ZIP	START_LATITUDE	START_LONGITUDE	END_LATITUDE	END_LONGITUDE	MILES
2	111N17566	141ST ST	WESTBOUND	US-75/OKMUL	OK	TULSA	74033	35.959209	-96.011693	35.959167	-96.011908	0.012365
3	111N17676	141ST ST	WESTBOUND	US-75/OKMUL	OK	TULSA	74033	35.959209	-96.011693	35.959167	-96.011908	0.012365
4	111P17566	141ST ST	EASTBOUND	US-75/OKMUL	OK	TULSA	74033	35.959167	-96.011908	35.959209	-96.011693	0.012365
5	111P17676	141ST ST	EASTBOUND	US-75/OKMUL	OK	TULSA	74033	35.959167	-96.011908	35.959209	-96.011693	0.012365
6	111N17705	145TH EAST	SOUTHBOUND	OK-51	OK	TULSA	74012	36.08059	-95.815366	36.0775976	-95.81536	0.20673
7	111N17672	181ST ST	WESTBOUND	US-75/OKMUL	OK	TULSA	74047	35.901045	-96.01576	35.901052	-96.015977	0.012154
8	111P17672	181ST ST	EASTBOUND	US-75/OKMUL	OK	TULSA	74047	35.901052	-96.015977	35.901045	-96.01576	0.012154
9	111-17618	193RD EAST	SOUTHBOUND	I-44	OK	ROGERS	74015	36.191519	-95.758287	36.1638078	-95.7616606	1.92757
10	111N17618	193RD EAST	SOUTHBOUND	I-44	OK	ROGERS	74015	36.1638078	-95.7616606	36.161968	-95.761656	0.127114
11	111N17610	193RD EAST	SOUTHBOUND	CREEK TPKE	OK	WAGONEF	74014	35.99425	-95.761664	35.9926008	-95.7616696	0.113941

Figure 3-5. Road segment information.

Incident data (i.e., accident information stored digitally after hand processing) was typically based on police information. To access the data, a user must navigate to the Incident Analysis button located on the left sidebar (See Figure 3-2). As per the traffic speed downloading, date range and segment can be selected, while distance and time are minimized. These two later variables correlate to secondary incident detection. This phenomenon is beyond the scope of this thesis (See Figure 3-6). The acquired incident dataset is based on fulfilled filtering requirements and provides extensive temporal and spatial information directly related to the accident. Columns include incident ID, datetime, the severity (i.e., scale ranging from 1 to 5, with 5 indicating worst case), negative and positive road segment identification; type of collision; and geographical location (See Figure 3-7).

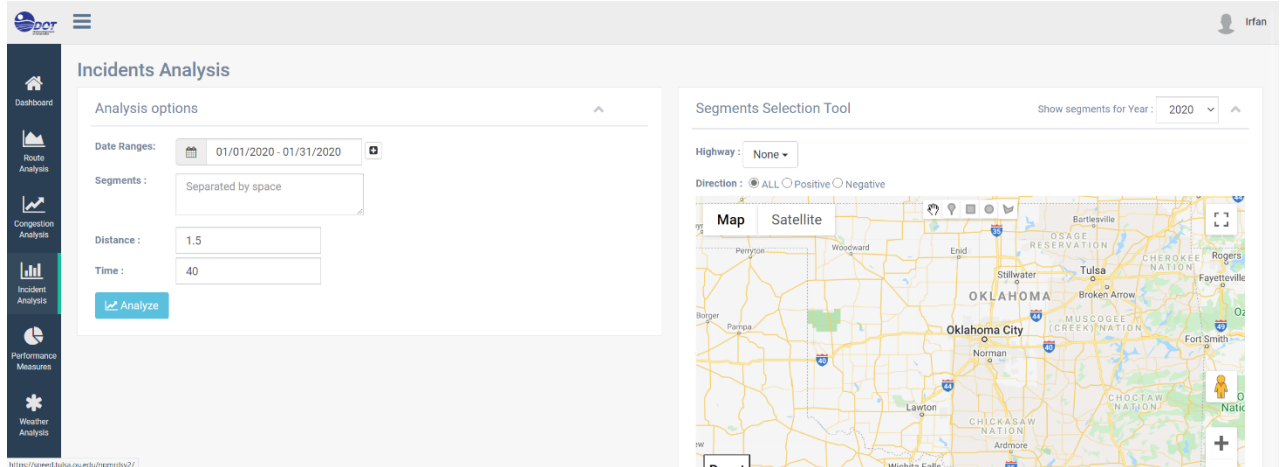


Figure 3-6. Incident analysis generator webpage.

ID	Date	Severity	tmcN	tmcP	Type of Collision	Latitude	longitude
0	1/1/2018 14:06	4	111N05649	111P05649	REAR-END	36.1379	-97.0516
3	1/1/2018 14:15	2	111N05649	111P05649	REAR-END	36.116	-97.0514
5	1/2/2018 15:15	1	111N05649	111P05649	REAR-END	36.1321	-97.0515
9	1/3/2018 17:45	1	111-09026	111+06498	F-O GUARDRL-END	35.9878	-94.5595
10	1/2/2018 19:43	1	111-04904	111P04905	SIDESWIPE-SAME	35.32	-97.4899
11	1/2/2018 15:02	1	111-05616	111+05617	ANGLE-TURNING	36.419	-94.8042
13	1/7/2018 18:53	2	111N05649	111P05649	SIDESWIPE-SAME	36.1088	-97.0514
15	1/2/2018 21:23	3	111N05649	111P05649	ANGLE-TURNING	36.107	-97.0514
17	1/3/2018 10:00	1	111N05649	111P05649	REAR-END	36.1266	-97.0515
19	1/9/2018 12:04	2	111N05649	111P05649	REAR-END	36.1342	-97.0515
21	1/9/2018 13:15	1	111N05649	111P05649	REAR-END	36.1272	-97.0515
23	1/1/2018 19:33	1	111-05114	111+05115	ANGLE-TURNING	35.1385	-97.6535

Figure 3-7. Generated incident dataset.

3.1.2 Other features of NPMRDS Webserver

The NPMRDS webserver [23] has additional features that may contribute towards this thesis's future works such as route, congestion, and performance measures of road segment and weather analyses. The route analysis webpage provides users the opportunity to filter date range, segment, averaging period, data source, and threshold filtering (See Figure 3-8). Route analysis offers speed distribution for selected road section per hour of the day, signifying distribution with maximum, minimum, and average speed (See Figure 3-9);

there is also an alternative distribution graph where travel time is reported instead of speed
 (See Figure 3-10).

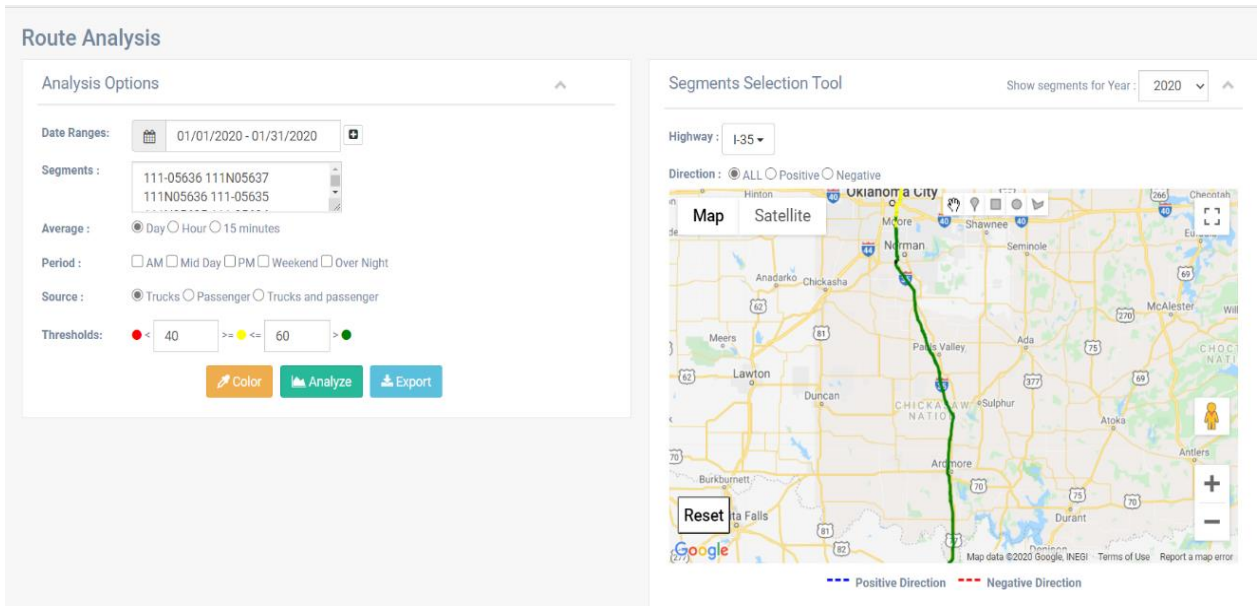


Figure 3-8. NPMRDS route analysis.

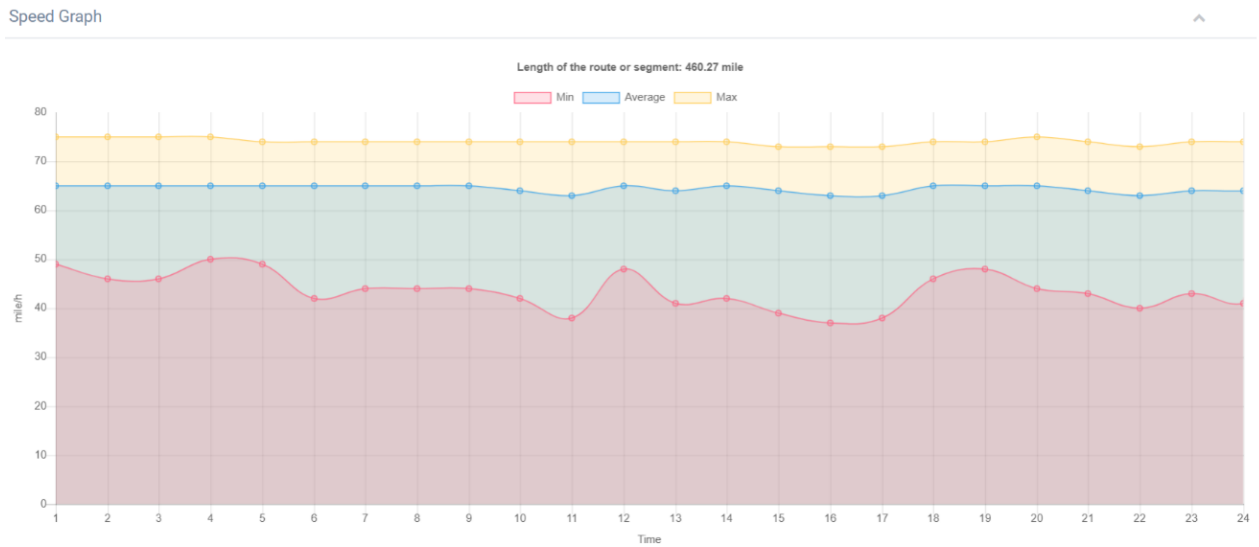


Figure 3-9. Speed distribution for selected segments.

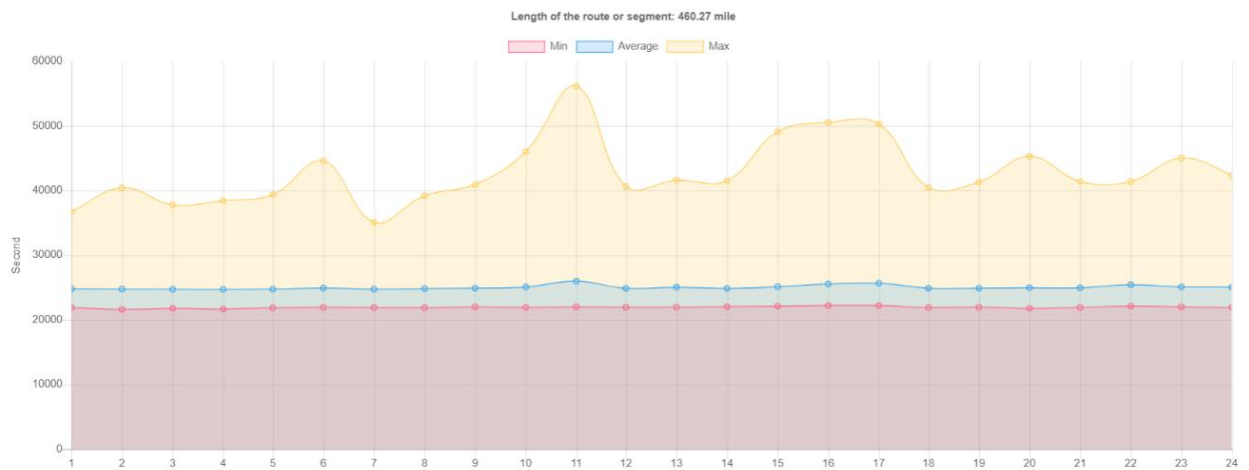


Figure 3-10. Travel time distribution for selected segments.

Congestion analysis (See Figure 3-11) gives users access to multiple analyses tools and outcomes to aid them in determining congestion. A heatmap plot (See Figure 3-12) assists users in finding the longest spanning congestion occurrence in a selected date range (See Figure 3-13); producing bar plots of the top 10 segments experiencing congested by frequency of hours congested (See Figure 3-14); and segment ranking based on occurrence count and average duration of congestion (See Figure 3-15).

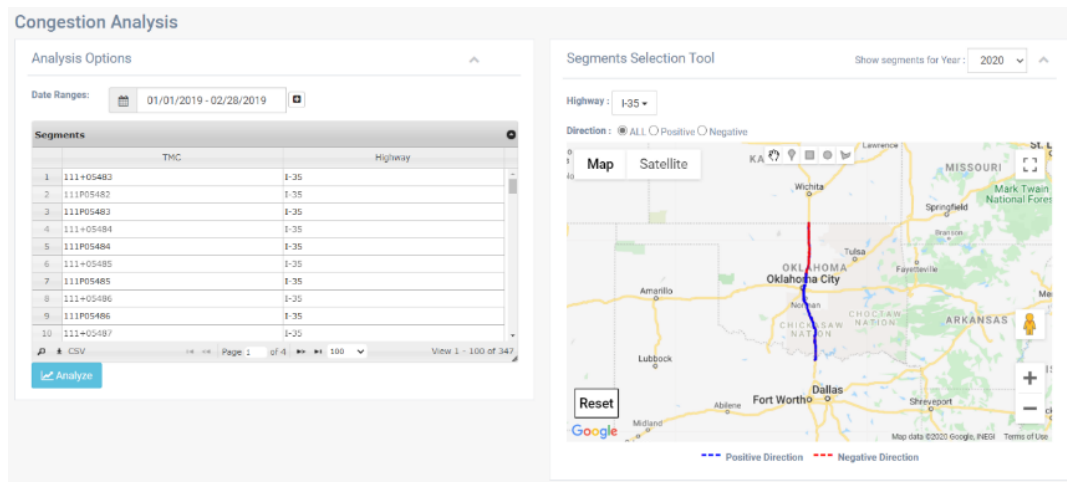


Figure 3-11. Congestion analysis layout.

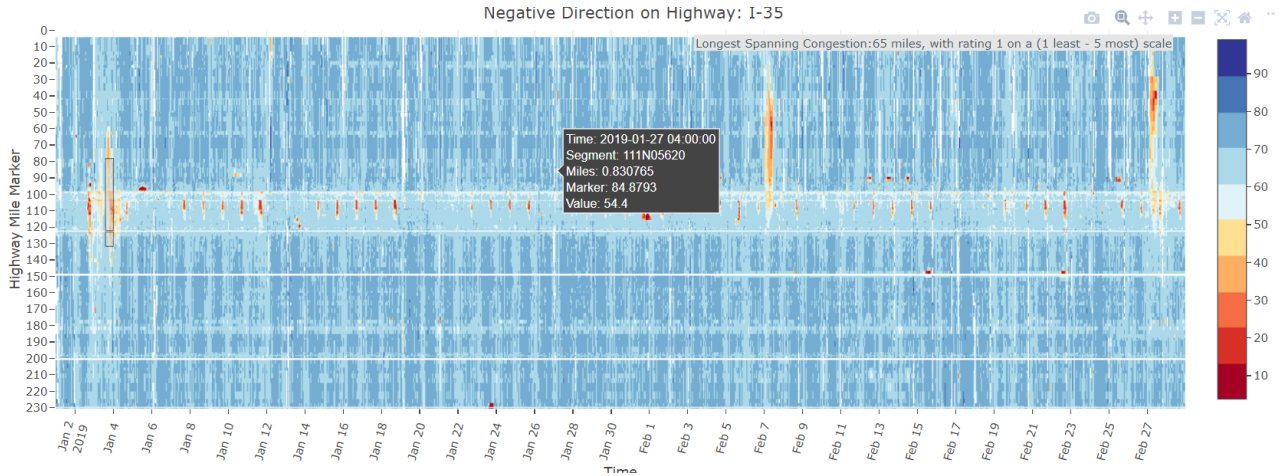


Figure 3-12. Heatmap showing distribution of traffic congestion by traffic speed across Oklahoma Highway I-35.

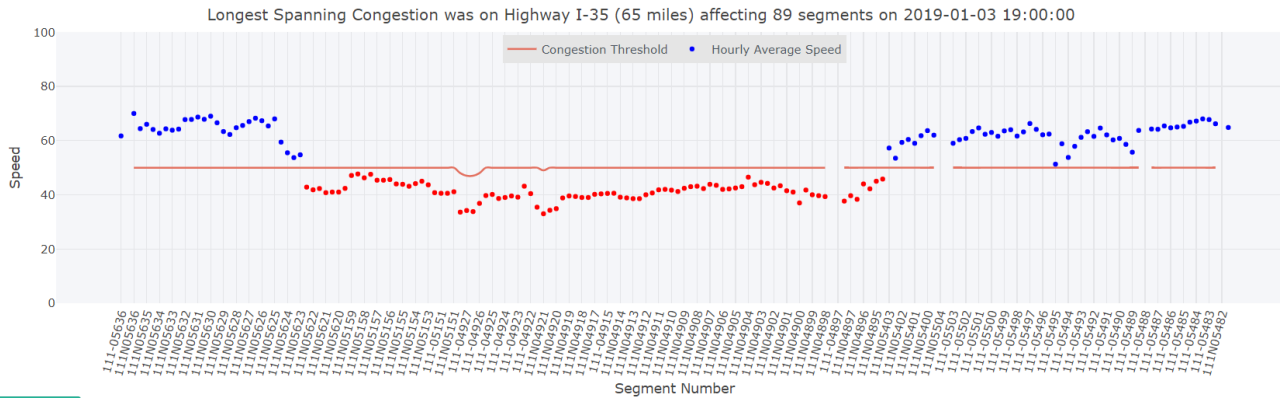


Figure 3-13. The distribution of speed per segment across Oklahoma Highway I-35 for the longest occurring congestion by distance.

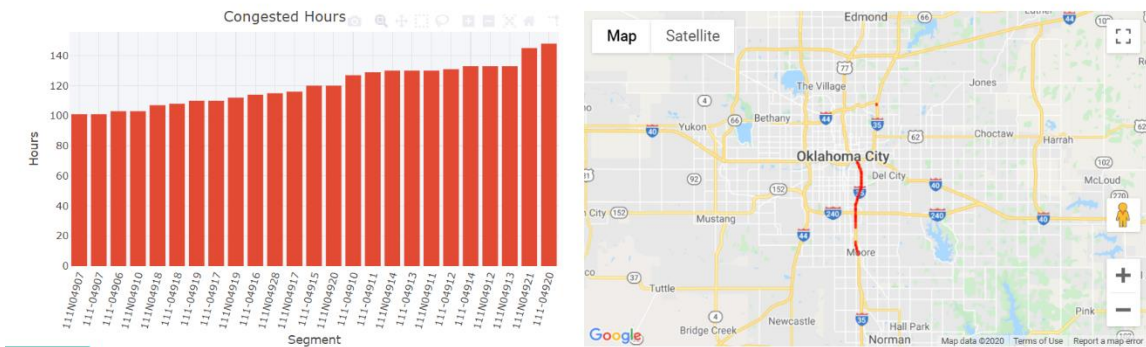


Figure 3-14. Bar plot of segment based on frequency of congestion by hour.

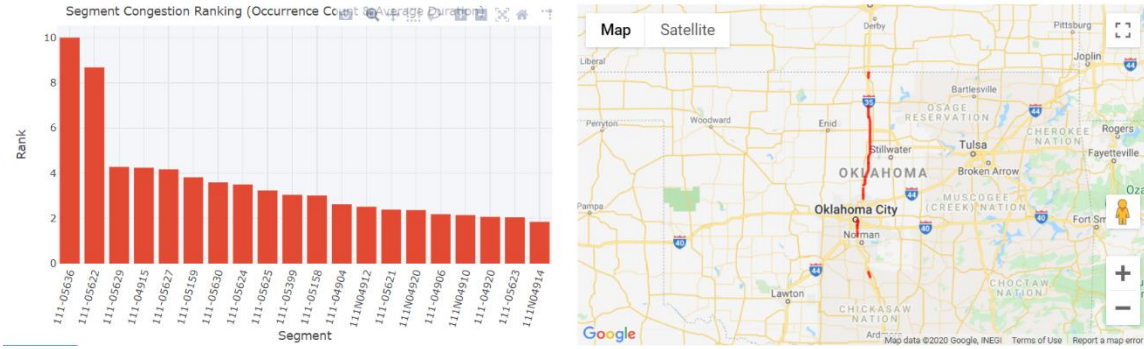


Figure 3-15. Frequency plot of segment ranking.

The NPMRDS webserver also provides a performance measure analysis (See Figure 3-16) for selected date ranges, segments, and other performance measures related parameters, including a derivative of the FHWA guideline used to determine efficiency and usage of selected road sections (See Figure 3-17). The website also offers users weather analysis, in particular data regarding snowplow truck deployment with datetime and location (See Figure 3-18).

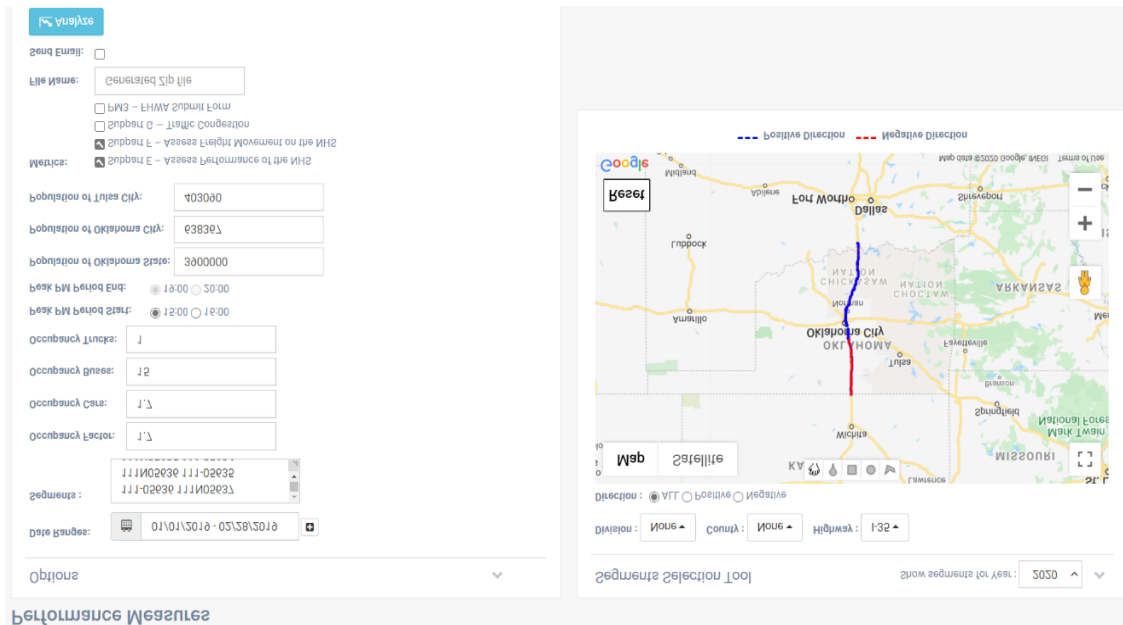


Figure 3-16. Performance measures parameter filter.

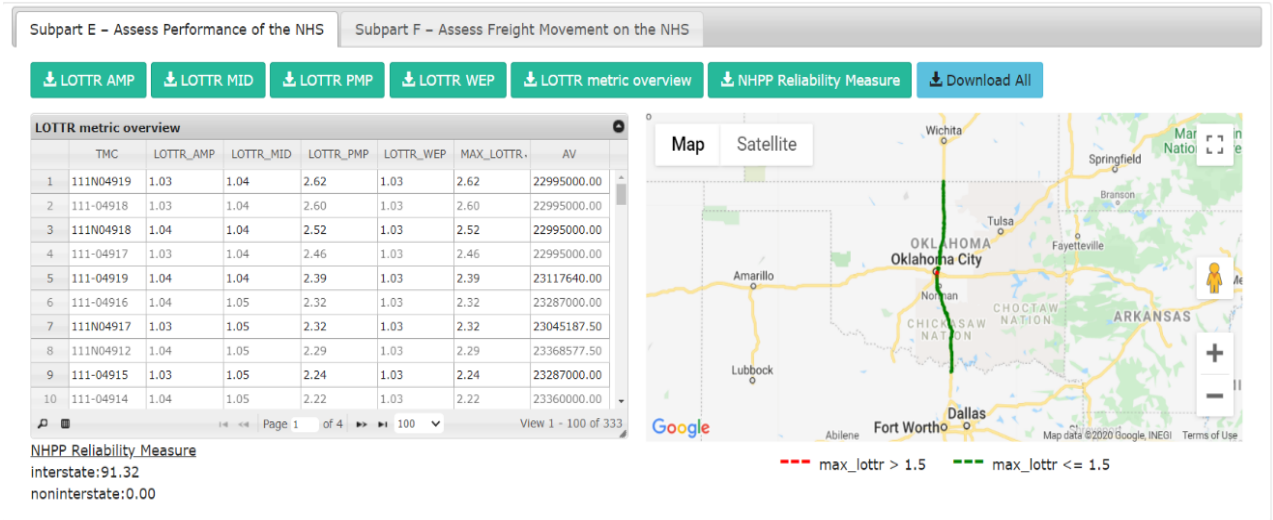


Figure 3-17. Road performance measures and freight movement information.

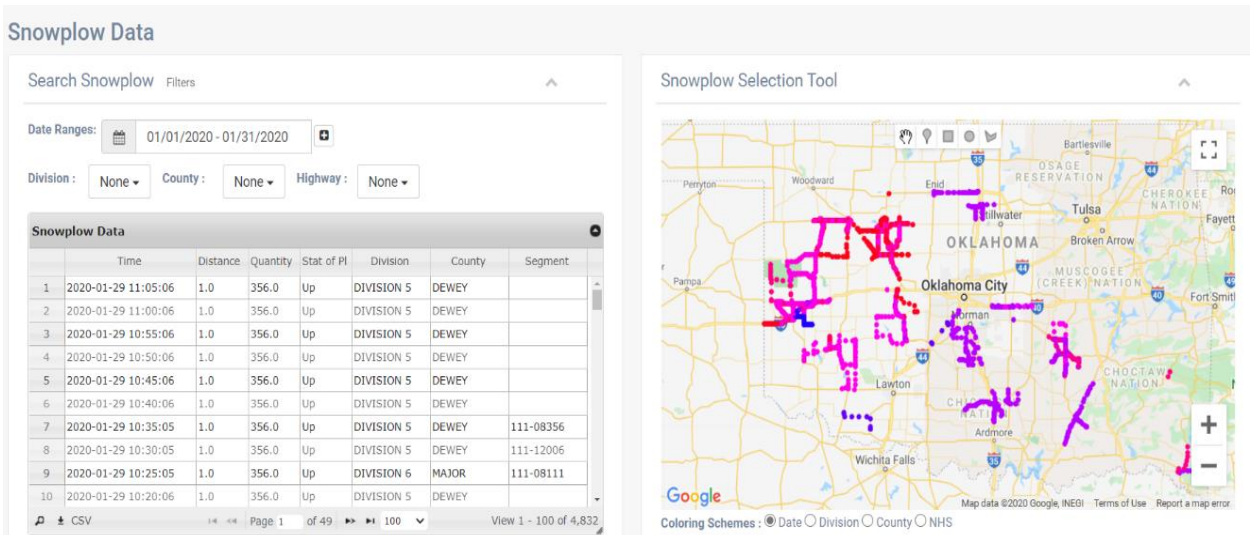


Figure 3-18. Deployment of snowplow trucks and collected data.

3.2 Exploratory Data Analysis

Like any data analysis, the acquired data for this thesis were first processed and scripted to generate Exploratory Data Analysis (EDA) output for further understanding of the dataset. The first step is reviewing the incident dataset to understand and analyze road accident occurrences. The distribution of accidents by hour-of-the-day (note hour is detailed in military format: 0 -23) from 2017 to 2019 is visualized as a frequency bar plot per each year (See Figure 3-19). As the figure shows, the plot shows the occurrence of peaking or a binomial pattern centered at hour 7 (i.e., 7:00 am) and at hour 17 (i.e., 5:00 pm). This exhibited pattern corresponds closely to the estimated traffic rush hours which are 6:00 to 10:00 am and 3:00 to 7:00 pm. To determine if this distribution is not only confined to our case study of Oklahoma Highway I-35, the same distribution plot was created for Oklahoma Highways I-40 and I-44 (See Figure 3-20 and Figure 3-21) and, as expected, their respective plots closely resemble the distribution plot for Oklahoma Highway I-35, suggesting the hourly distribution of accident is not largely influenced by spatial factors. Instead, traffic flow plays a major role in the cause for an accident. Even more, we can infer that the greater traffic flow during the two traffic rush hour periods is cause for the higher potential in accident occurrence. The increased risk is not only due to greater number of vehicles but also the human behavior tied to the rush hour period when it is more likely for a person to drive recklessly.

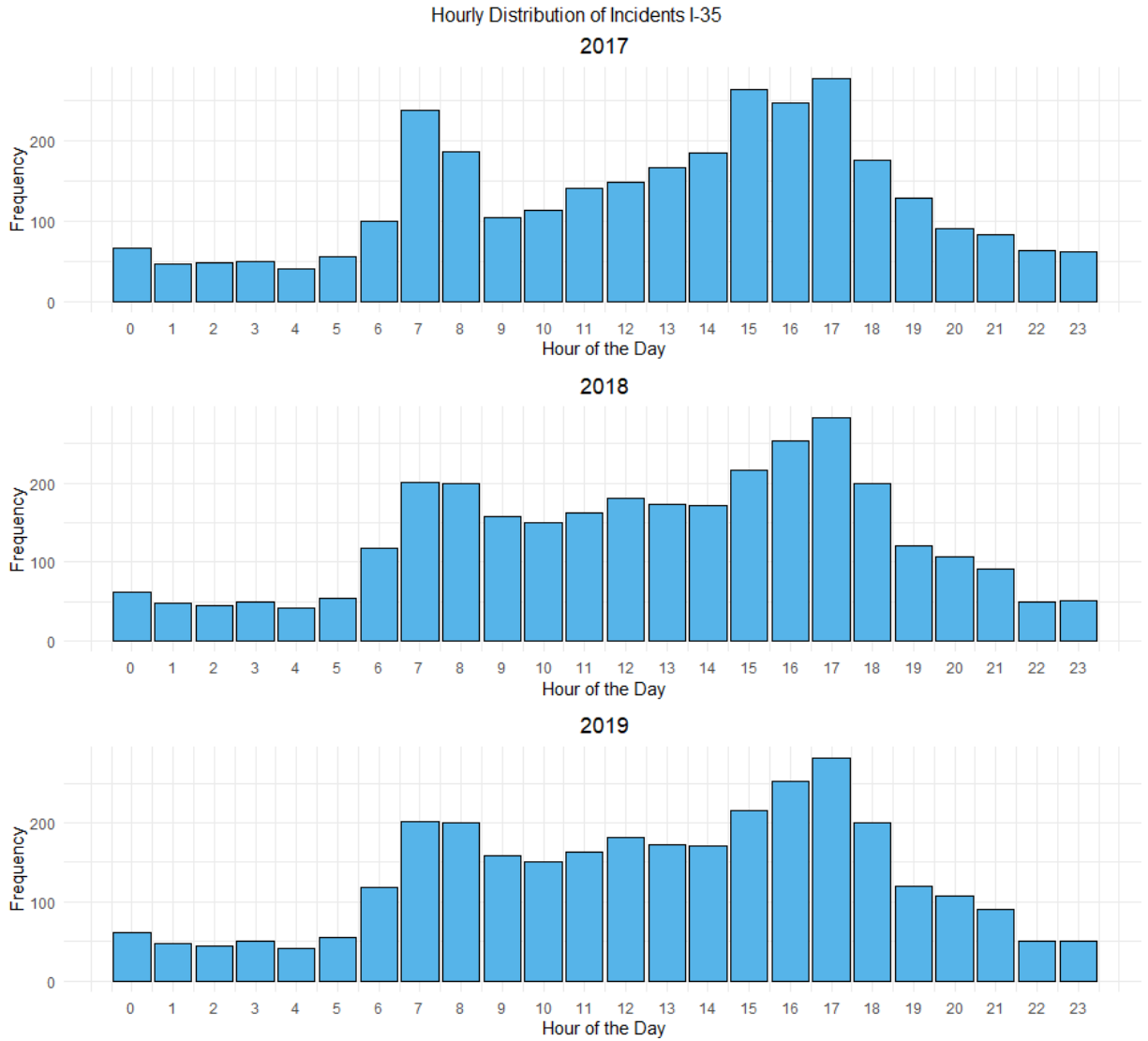


Figure 3-19. Accident distribution by hour of the day for Oklahoma highway I-35.

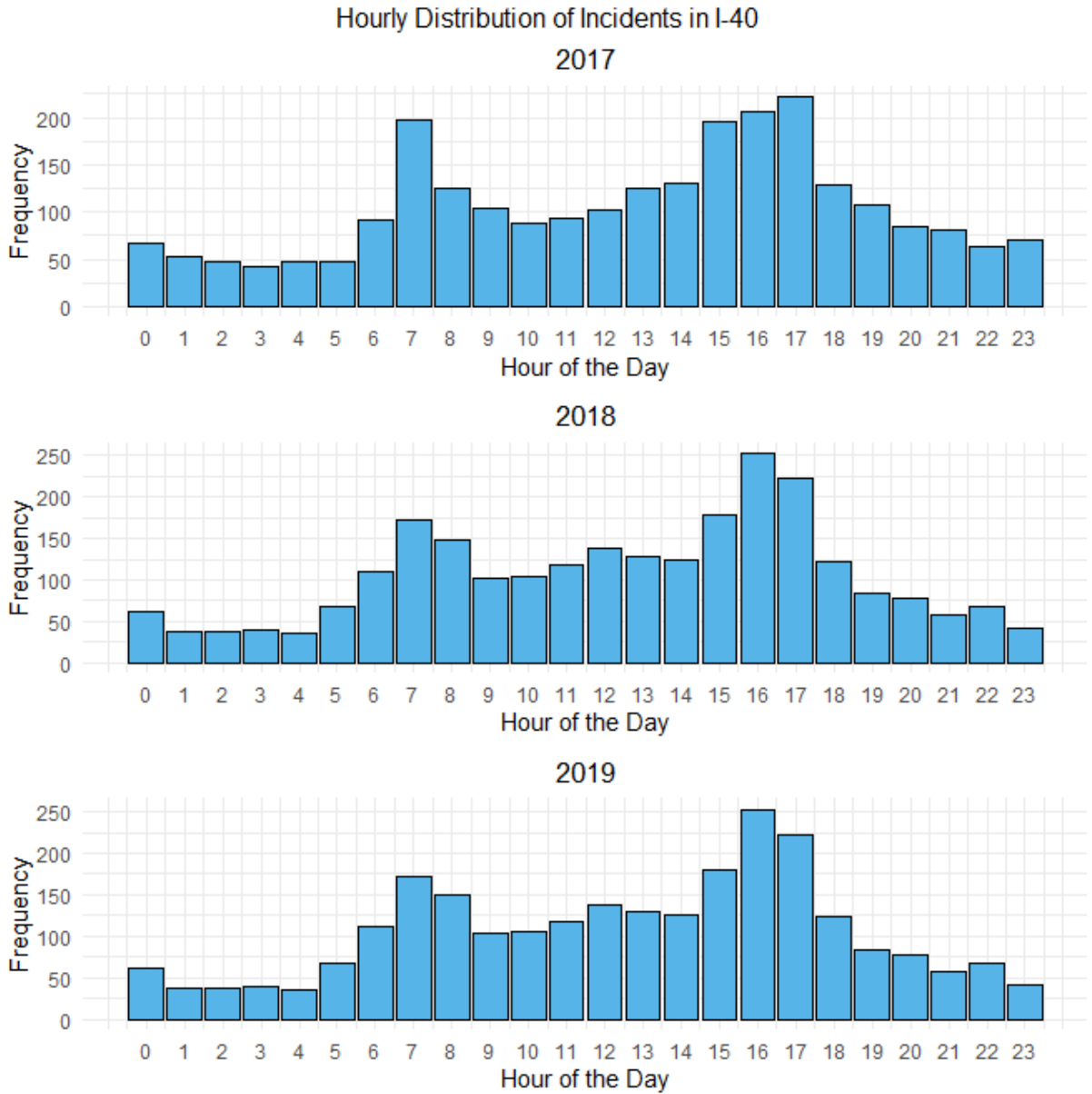


Figure 3-20. Accident distribution by hour of the day for Oklahoma highway I-40.

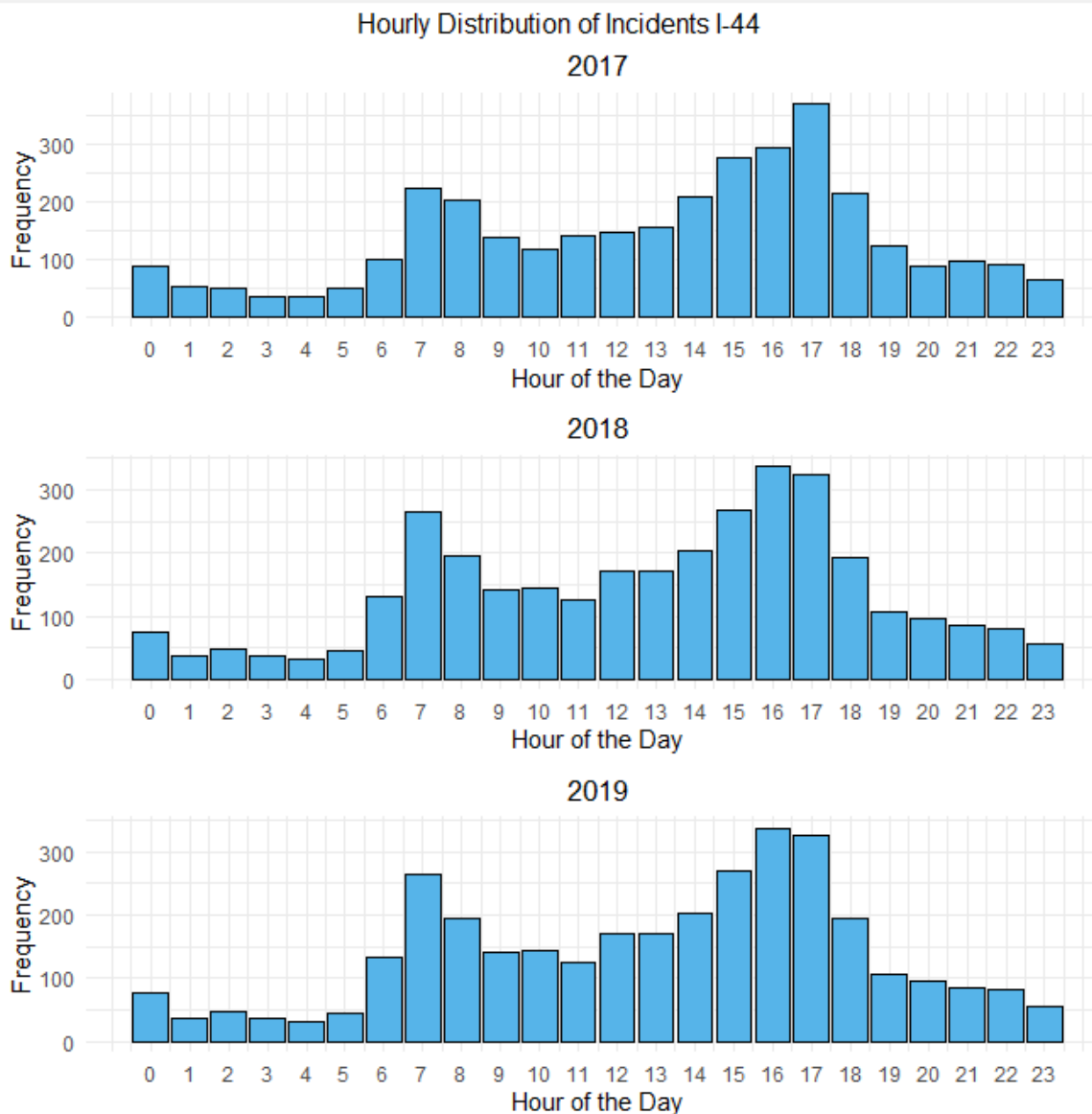


Figure 3-21. Accident distribution by hour of the day for Oklahoma highway I-44.

To investigate the effect of day of the week in accident distribution, the frequency bar plot of accidents based of this criterion is plotted for Oklahoma highway I-35 (See Figure 3-22). This figure demonstrates that traffic is not heavier one day over others, with the exception of Fridays, which has slightly heavier traffic. From this information, one can surmise that heavier traffic on Friday could be because it is the day before the weekend. Increased traffic flow might be indicative of drivers anticipating the small break from work

or travel to social events. This distribution is corroborated by the distribution for Oklahoma highway I-40 (See Figure 3-23), as well as the distribution for Oklahoma highway I-44 for 2018 and 2019. In the latter case, the mode of the distribution falls on Wednesday (See Figure 3-24).

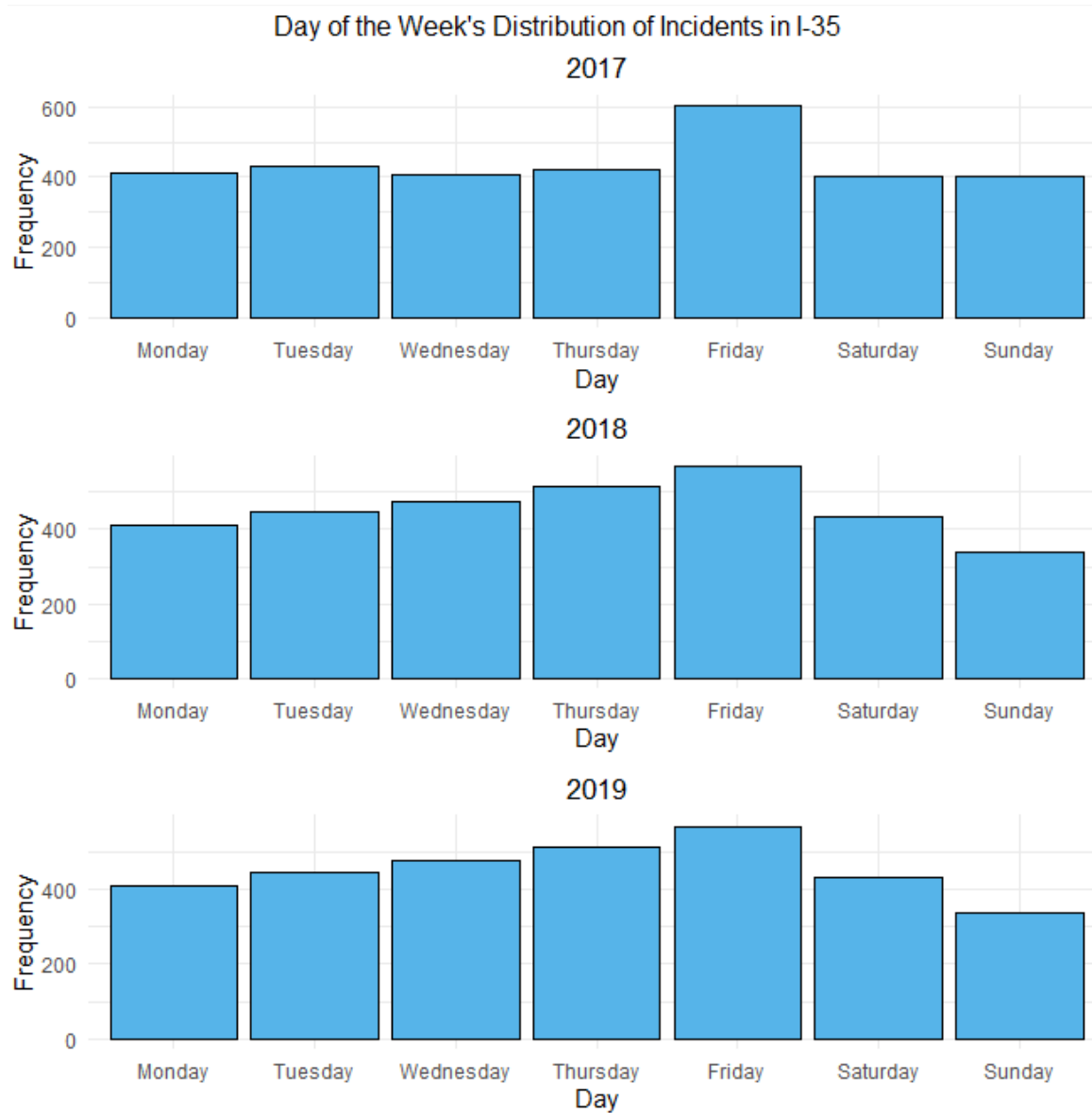


Figure 3-22. Accident distribution by day of the week for Oklahoma highway I-35.

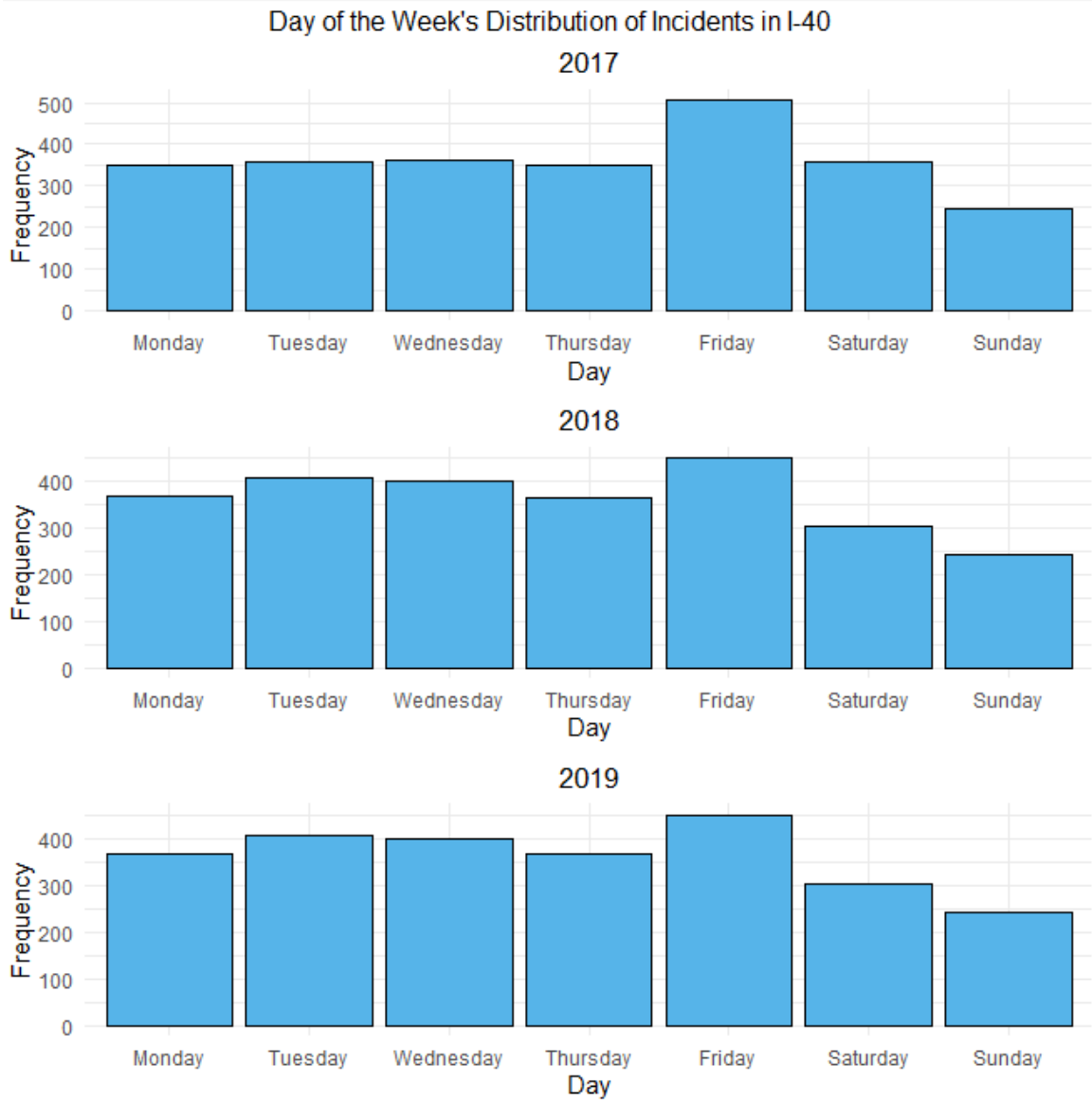


Figure 3-23. Accident distribution by day of the week for Oklahoma highway I-40.

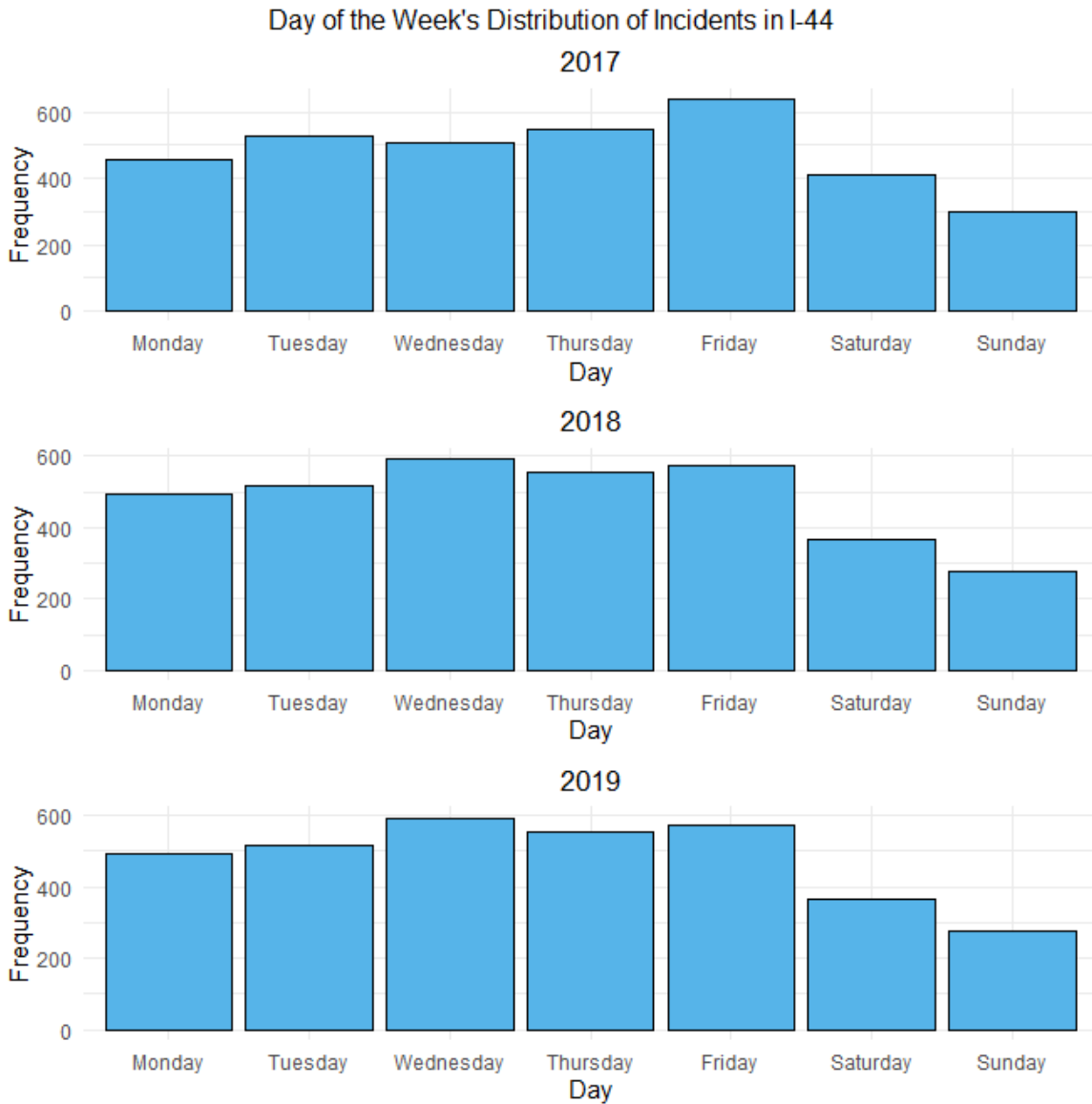


Figure 3-24. Accident distribution by day of the week for Oklahoma highway I-44.

The previous temporal distribution was also analyzed according to months in a year. Oklahoma highway I -35 showed no sign of deviation or skewness to the distribution that would provide conclusive inference for cause and effect. The month during which the observation occurred was not considered to have a primary effect on the possibility of an accident occurring (See Figure 3-25). Similarly, an analysis of Oklahoma highways I-40 and I-44 yielded similar output with no discernible distribution pattern (See Figure 3-26

and Figure 3-27). Prior to the distribution analysis, a hypothesis was formulated, suggesting that an increase in accidents from October to January was due to worsening weather conditions in Oklahoma with the arrival of winter. However, data analysis did not suggest such a causation.

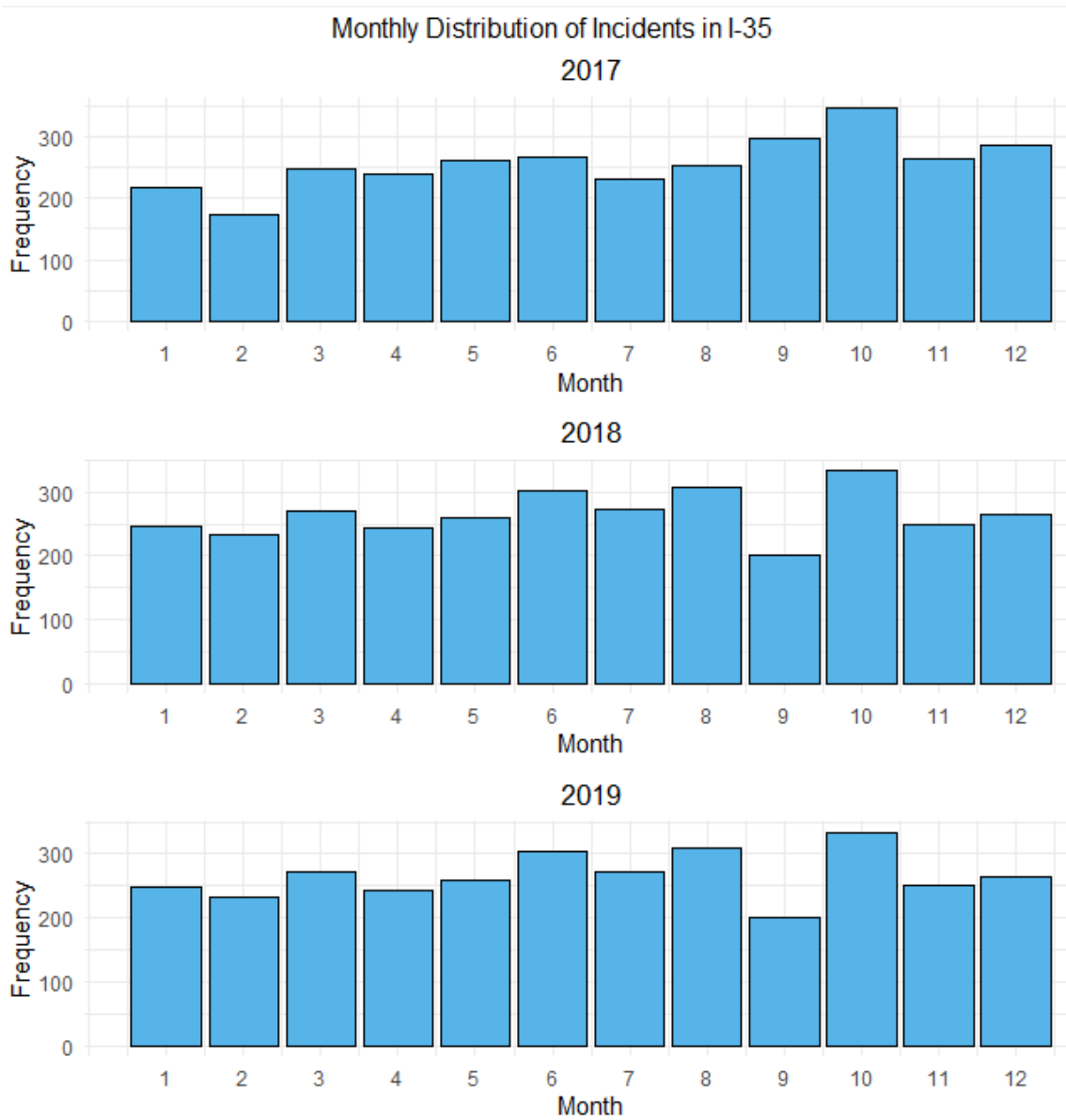


Figure 3-25. Accident distribution by month for Oklahoma highway I-35.

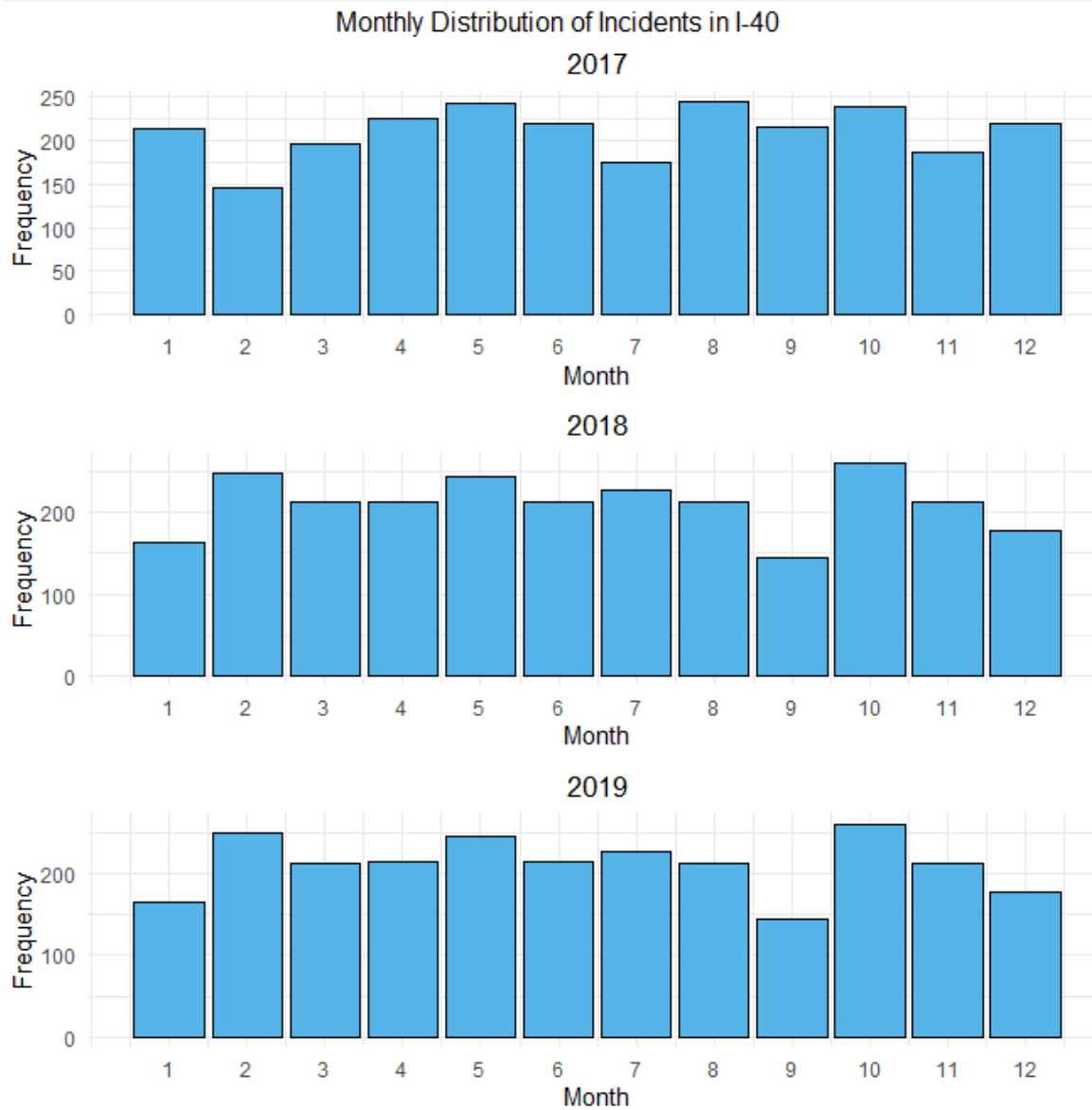


Figure 3-26. Accident distribution by month for Oklahoma highway I-40.

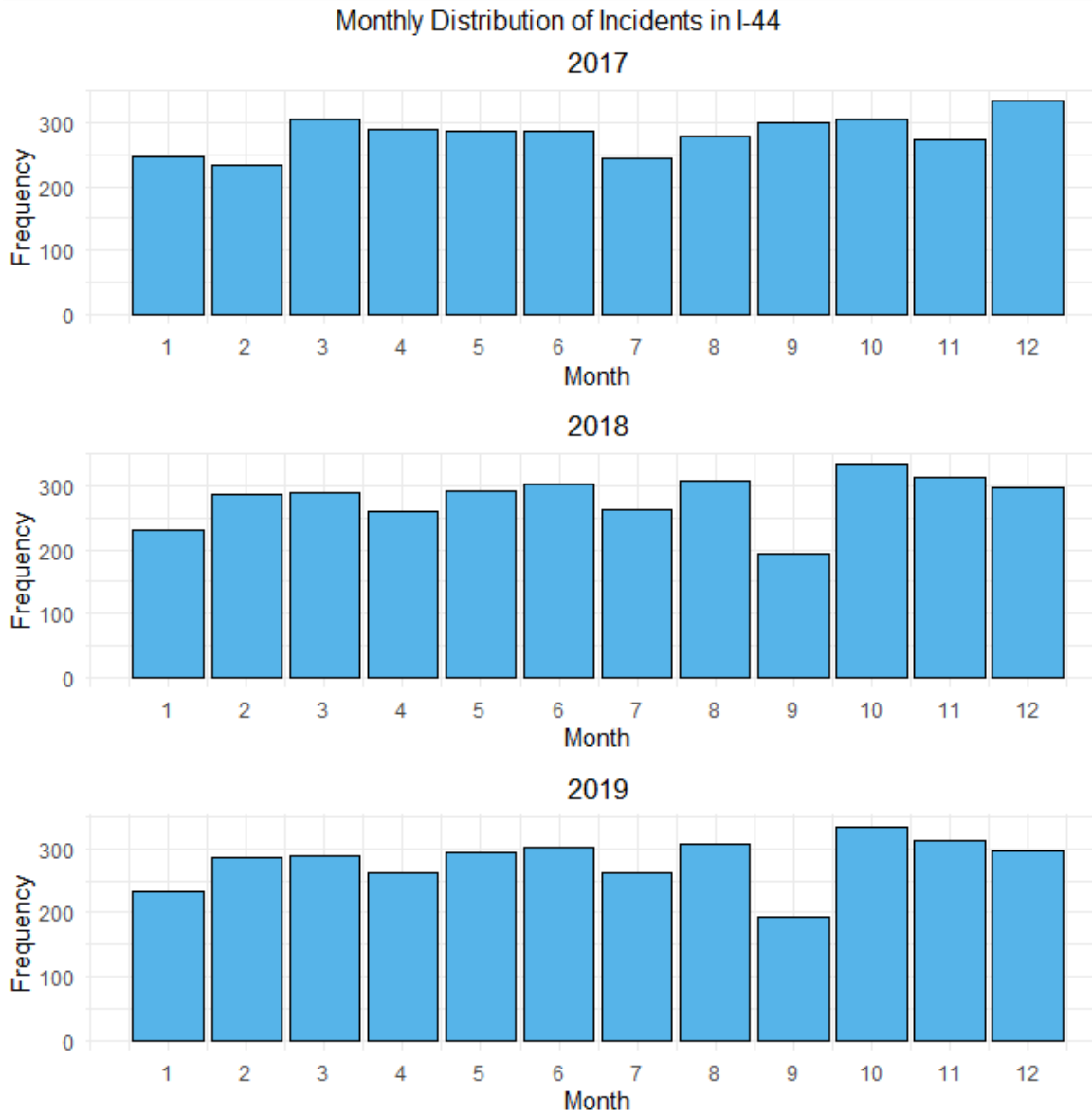


Figure 3-27. Accident distribution by month for Oklahoma highway I-44.

Visualizations of data to this point showed the temporal aspect of accident occurrences. Accident distribution could also be determined spatially based on road segment where accidents occurred. Clearly, the top 10 road segment that are prone to accidents can be determined through filtering based on highest frequency of accidents. Number of accidents per segment is plotted in this paper with different color bars to denote the year of accident occurrence, where road segment 111N04912 had the highest incident rate on southbound Oklahoma highway I-35, and segment 111P04912 had the highest incident rate on the northbound direction of the same highway(See Figure 3-28 and Figure 3-29).

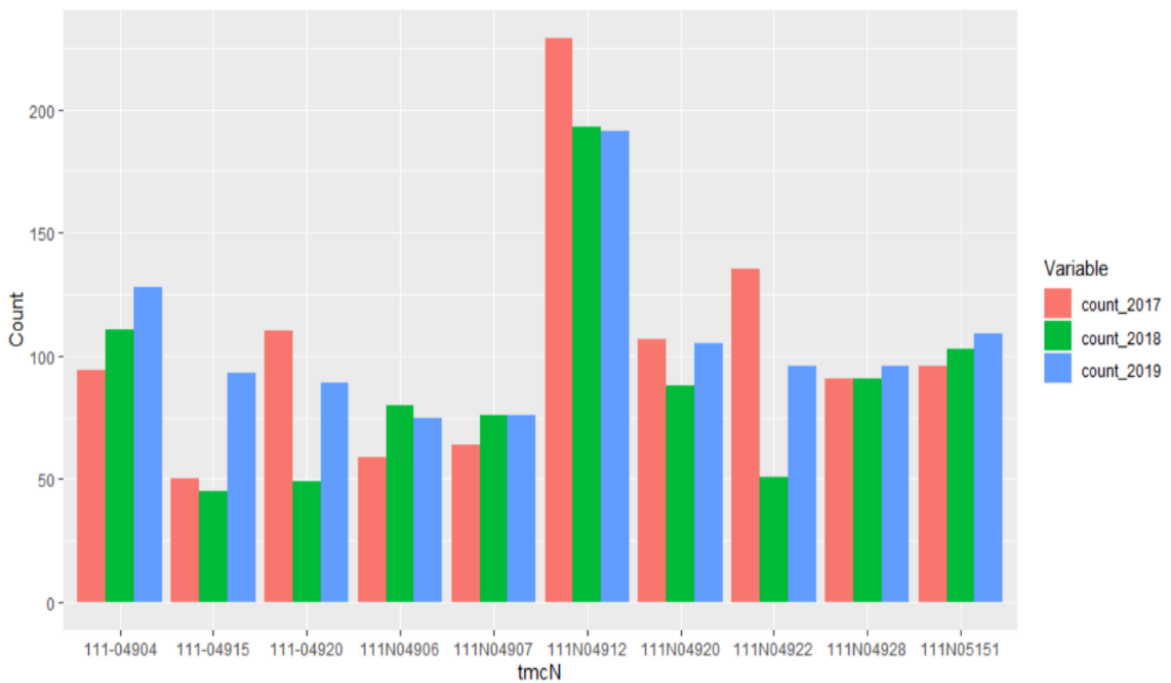


Figure 3-28. Plot of most frequent accident occurring road segments (southbound).

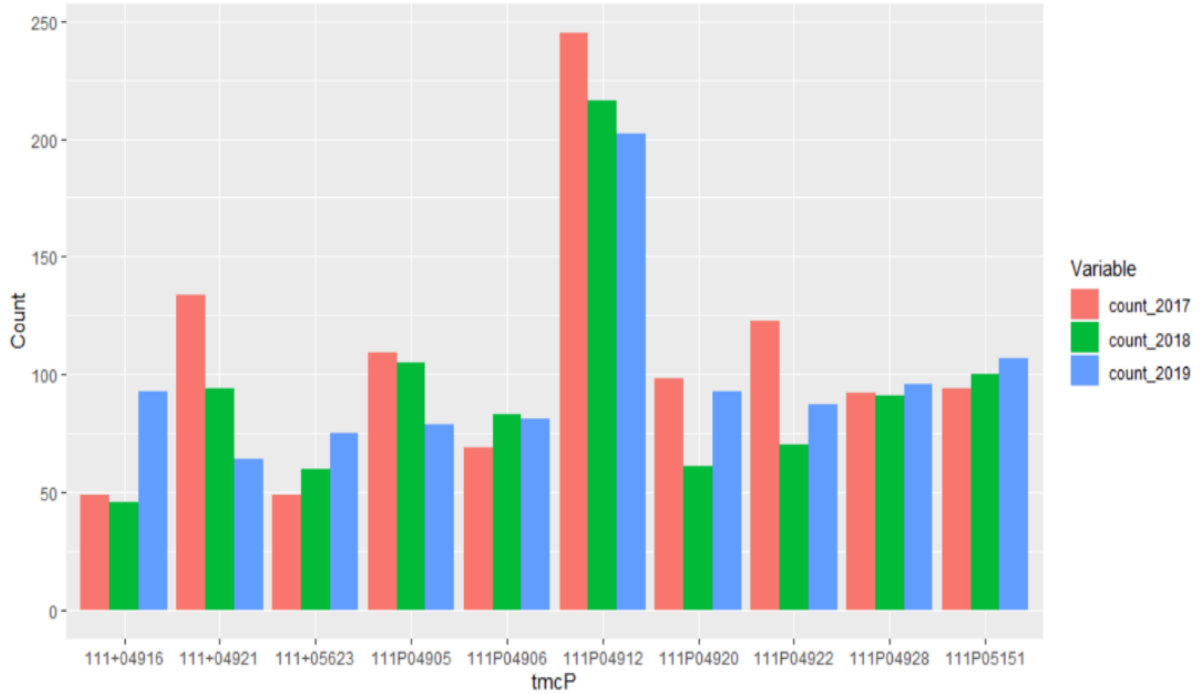


Figure 3-29. Plot of most frequent accident occurring road segments (northbound).

A graph of hourly distribution for these particular segments was plotted (See Figure 3-30 and Figure 3-31) to analyze accident occurrence for hour of day, as reported in Figure 3-25 and Figure 3-26. Figure 3-16 shows that vehicle distribution mostly fit the expected pattern, with the exception of certain segments where mode was more concentrated between 11 am and 2 pm.

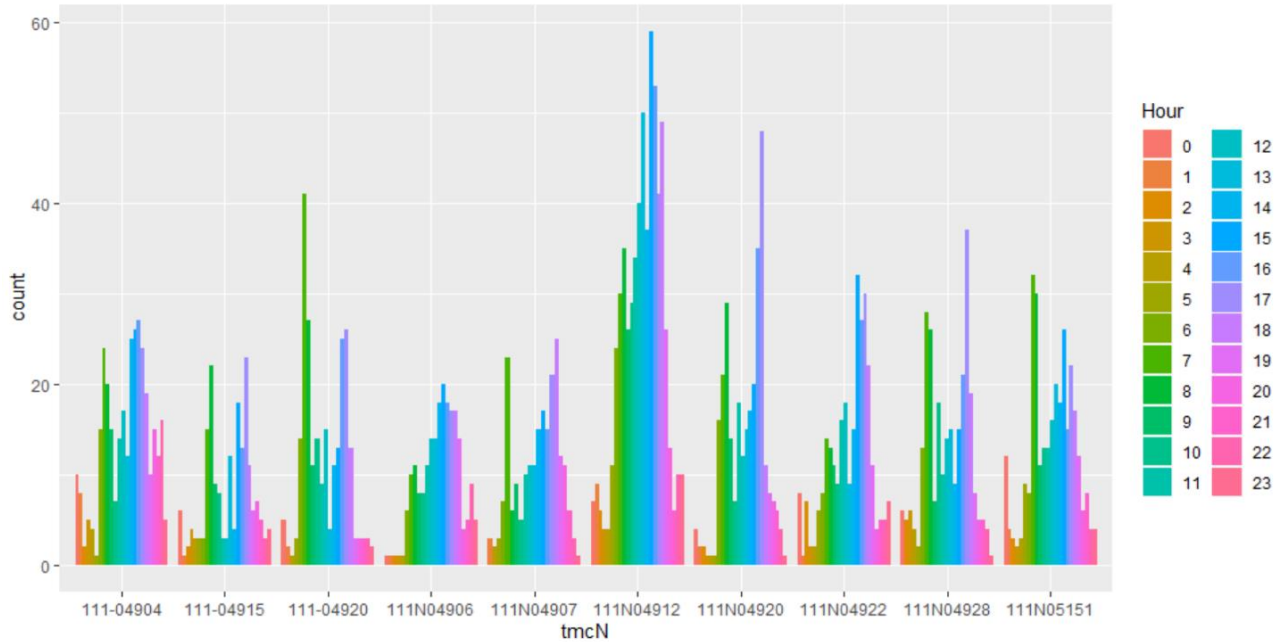


Figure 3-30. Temporal distribution by hour for accident prone segment (southbound).

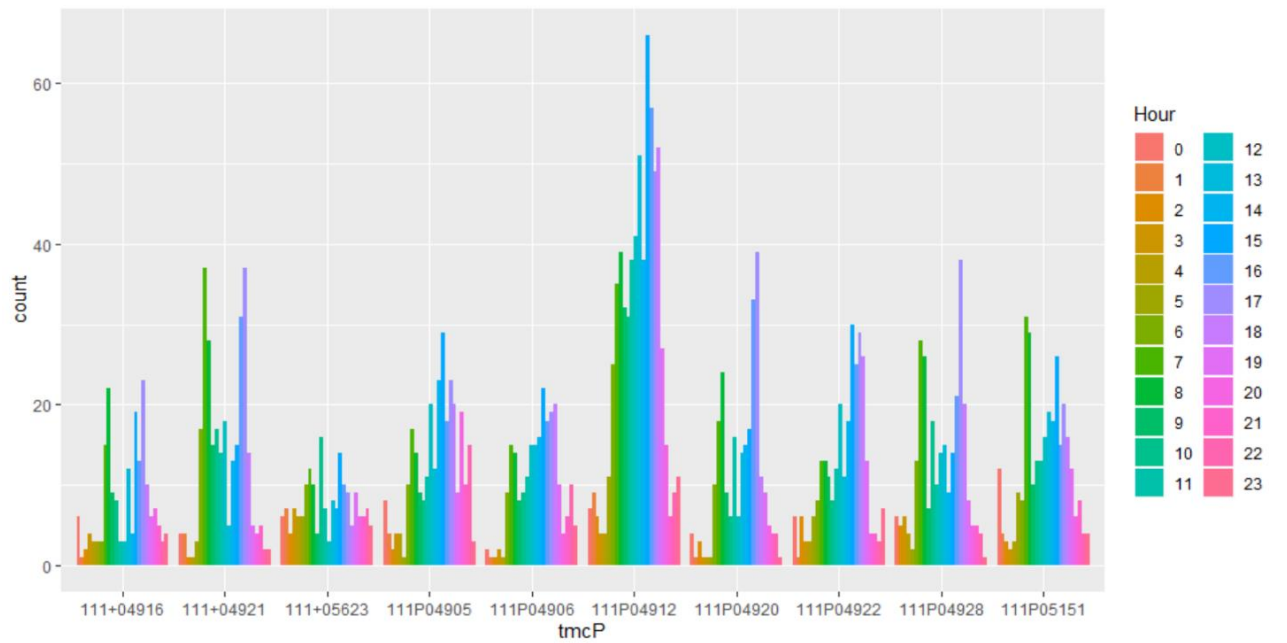


Figure 3-31. Temporal distribution by hour for accident prone segment (northbound).

To observe the change in speed after an accident, change in speed was plotted in relation to the accident event. The first plot shows change in speed with no known event occurring

during the specified time period for 100 randomized samples (See Figure 3-32). The second plot shows change in speed for 100 randomized samples after an accident occurred (See Figure 3-33).

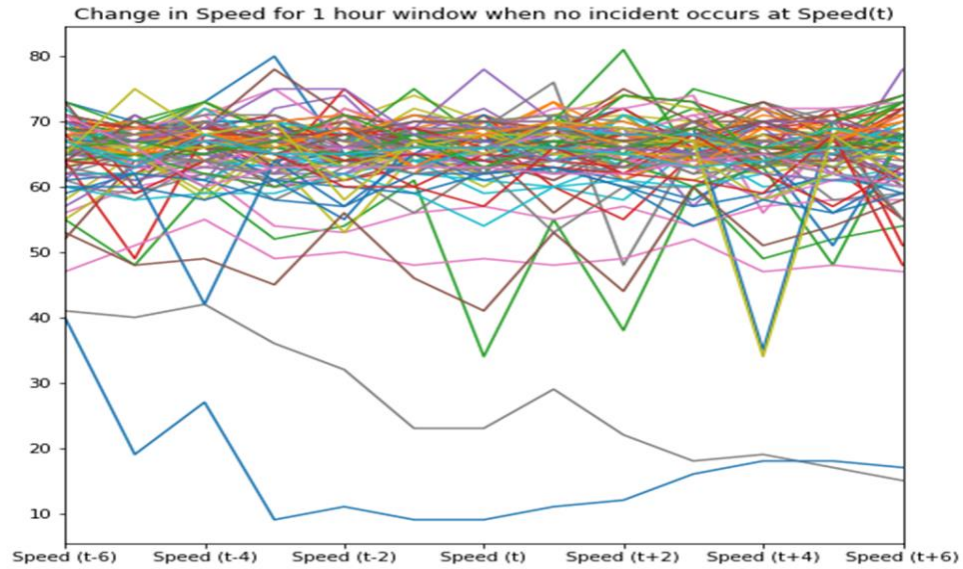


Figure 3-32. Plot change in speed when no accidents occurred.

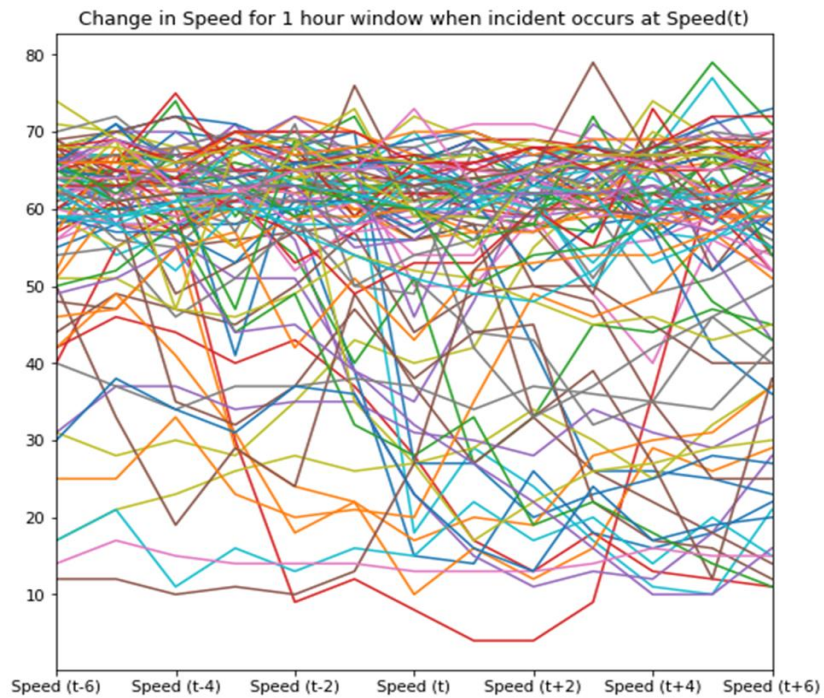


Figure 3-33. Plot change in speed when an accident occurred.

4. Real Time Accident Detection

4.1 Datetime Parsing for Traffic Speed Data

The traffic speed dataset must be matched with incident data to be suitable for feature and label creation for machine learning or deep learning purposes. To ensure accuracy, both datasets must be matched with the correct corresponding temporal and spatial information. For temporal matching, date and time information should be in the same format. However, data in the traffic speed dataset can be represented in two columns (i.e., date and epoch), where date is represented by unformatted numbers (e.g., the first one or two digits represent the day, the next two the month, and the last four the year and epoch values range from 0 to 287 in increments representing an addition of 5 minutes to the start time of each day, (e.g., 12:00 am) (See Table 4-1).

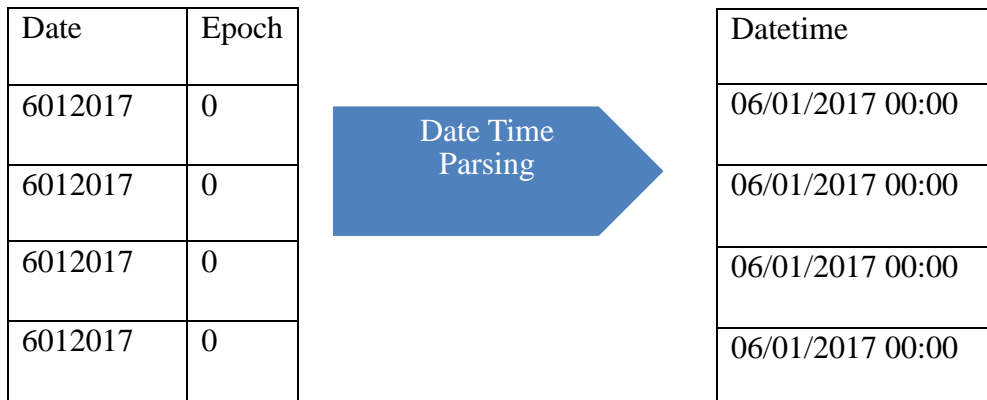


Table 4-1. Datetime parsing for the traffic speed dataset.

4.2 Feature Extraction

Before feature engineering can be applied, data must be processed for uniformity in robust time series analysis. Missing timestamps were generated with NaN values in feature data

(See Table 4-2 and Table 4-3). Traffic speed data and road segment data indicate speed, average speed, reference speed, date, time, longitude, latitude, travel time, and segment length in miles. [10] explains that some features (e.g., position of the sun that could be extracted from already available temporal and spatial data). Utilizing the pytz [24] and pysolar library [25], the solar azimuth and solar altitude can be generated from input of road segment location and observation time. To apply the supervised machine learning algorithm, necessary features for input include speed, hour, day, month, longitude, latitude, travel time, solar azimuth, and solar altitude. Features such as reference speed, average speed, and segment length were not included, as they did not demonstrate variation or correlation with the occurrence of a road traffic event.

Datetime	Speed	Longitude	Latitude	Travel Time
06/01/2017 00:00	x	x	x	X
06/01/2017 00:05	x	x	x	X
06/01/2017 00:25	x	x	x	X
06/01/2017 00:30	x	x	x	X

Table 4-2. Original dataset with missing timestamp.

Datetime	Speed	Longitude	Latitude	Travel Time
06/01/2017 00:00	x	x	x	X
06/01/2017 00:05	x	x	x	X
06/01/2017 00:10	NaN	NaN	NaN	NaN
06/01/2017 00:15	NaN	NaN	NaN	NaN

06/01/2017 00:20	NaN	NaN	NaN	NaN
06/01/2017 00:25	x	x	x	X
06/01/2017 00:30	x	x	x	X

Table 4-3. Generating missing timestamp observations.

4.2.1 Incident event matching to traffic speed data observation

An observation formed using traffic speed data was matched with the incident dataset using datetime and location as the inner join key. Observations without a successful match were classified as a non-accident occurring observation, while matched observations were classified as accident occurring (See Table 4-4). Appendix B-1 reports snippets of the matched data.

Datetime	Segment	Speed	Hours	Day	Month	Travel Time
01/01/2017 00:00	0	60.0	0	6	1	191.0345
01/01/2017 00:05	0	67.0	0	6	1	171.0752
01/01/2017 00:10	0	64.0	0	6	1	179.0943

Longitude	Latitude	altitude	azimuth	Incident
-97.43	35.10	-31.51	97.38	0
-97.43	35.10	-30.50	98.01	0
-97.43	35.10	-29.49	98.65	0

Table 4-4. Dataset completed after matching traffic feature with incident.

4.2.2 Supervised Machine Learning Input/Output Generation

This thesis introduces two methodologies, namely machine learning and deep learning, for: introduced with the first one being the accident occurrence detection in near real time. The two approach for the near real-time detection. This section details speed feature shifting (sliding window) to create a 20-minute window observation, where speed (t-2), (t-1), (t), (t+1), and (t+2) in time represents a five-minute shift. This was accomplished using a simple algorithm (See Appendix B-2) to generate the supervised learning dataset (See Figure 4-2). After the shift was completed and appended to the original observation, rows with NaN values were dropped, ensuring all observations have correct time shift information; hence, the importance of the earlier timestamp generation with NaN values.

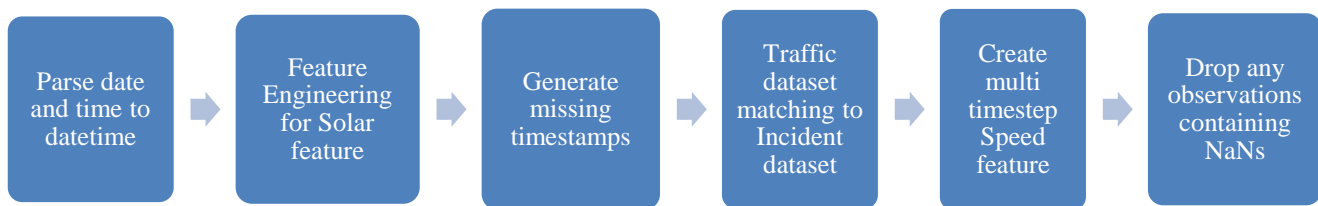


Figure 4-1. Dataset pre-processing for supervised learning

Date	Speed (t-2)	Speed (t-1)	Speed (t)	Speed (t+1)	\
2017-03-17 16:15:00	34.0	30.0	30.0	23.0	
2017-03-15 15:50:00	61.0	67.0	63.0	67.0	
2017-09-07 22:40:00	60.0	71.0	62.0	62.0	
2017-04-05 13:05:00	64.0	61.0	62.0	62.0	
2017-06-17 06:30:00	64.0	64.0	68.0	61.0	
2017-06-07 16:50:00	14.0	11.0	12.0	14.0	
2017-05-10 23:50:00	58.0	57.0	58.0	56.0	
2017-03-02 13:10:00	57.0	61.0	62.0	60.0	
2017-03-14 17:35:00	53.0	57.0	59.0	52.0	
2017-11-18 00:20:00	68.0	65.0	67.0	68.0	

Date	Speed (t+2)	Segment	Average Speed	Hours	Day	Miles	\
2017-03-17 16:15:00	24.0	199	60.0	16	4	0.421389	
2017-03-15 15:50:00	61.0	128	62.0	15	2	3.786800	
2017-09-07 22:40:00	63.0	182	51.0	22	3	0.349596	
2017-04-05 13:05:00	62.0	6	63.0	13	2	0.813567	
2017-06-17 06:30:00	72.0	159	64.0	6	5	0.570332	
2017-06-07 16:50:00	18.0	199	61.0	16	2	0.421389	
2017-05-10 23:50:00	60.0	143	61.0	23	2	3.402040	
2017-03-02 13:10:00	58.0	198	60.0	13	3	0.353945	
2017-03-14 17:35:00	51.0	185	62.0	17	1	0.373674	
2017-11-18 00:20:00	67.0	128	58.0	0	5	3.786800	

Date	Longitude	Latitude	TravelTime	altitude	azimuth	\
2017-03-17 16:15:00	-97.459626	35.522479	50.566680	-20.119979	283.907031	
2017-03-15 15:50:00	-97.374964	35.013281	216.388571	-15.651527	279.097439	
2017-09-07 22:40:00	-97.490616	35.334350	20.299123	-38.613430	44.088626	
2017-04-05 13:05:00	-97.485400	35.227073	47.239374	21.551167	262.469788	
2017-06-17 06:30:00	-97.327833	36.282752	30.194047	61.083651	108.069625	
2017-06-07 16:50:00	-97.459626	35.522479	126.416700	-11.993269	309.744762	
2017-05-10 23:50:00	-97.138037	34.427608	211.161103	-18.438778	51.472116	
2017-03-02 13:10:00	-97.460773	35.507320	20.551645	2.694344	259.823911	
2017-03-14 17:35:00	-97.494992	35.364672	22.800447	-35.981860	297.388537	
2017-11-18 00:20:00	-97.374964	35.013281	203.469851	-21.870385	98.869287	

Date	Target
2017-03-17 16:15:00	0
2017-03-15 15:50:00	1
2017-09-07 22:40:00	0
2017-04-05 13:05:00	0
2017-06-17 06:30:00	0
2017-06-07 16:50:00	1
2017-05-10 23:50:00	0
2017-03-02 13:10:00	1
2017-03-14 17:35:00	1
2017-11-18 00:20:00	0

Figure 4-2. Final dataset after incident matching.

4.3 Supervised Machine Learning Classification Models

As part of modeling to capture the correct feature information that leads us to the best fit a model to predict a possible occurrence of an accident at near real time with live data streaming. Modelling and implementation explained in this thesis serve as a prototype version for demonstrating possible methods to correct modelling implementation that will eventually lead to successful real time model deployment, Higher probability of possible accident detection requires that accident labelling will be reported in a ten minute window (e.g., time t, t+1 and t+2) with the feature inclusive of speed (t-2) to speed (t+2). Thus, modelling will not be exactly real time, as the possibility of detecting an accident occurring at time t will require the future time input at t+2 (i.e., corresponds to 10-minute delay), as shown in Figure 4-3. In summary, the model will detect accidents ranging from real time to a 10-minute delay of the accident occurrences.

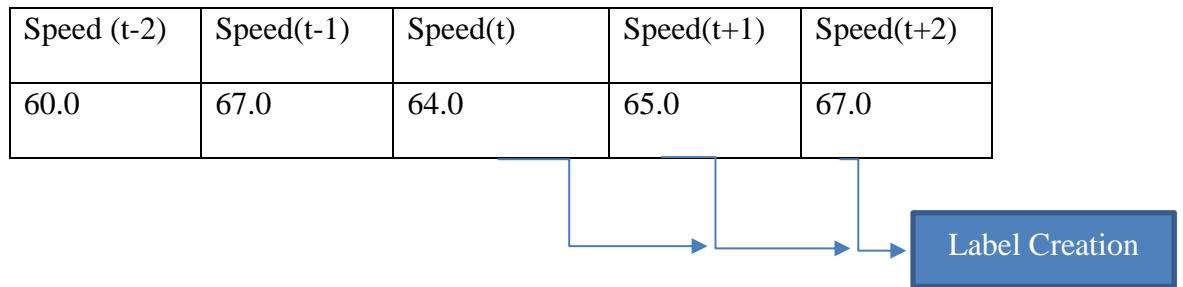


Figure 4-3. Timestep at label creation.

Model performance was mainly attributed by accident detection rate (See eqn. 1) and false detection rate (See eqn. 2) with inclusion of accuracy (See eqn. 3) and specificity (See eqn. 4) scoring measure. Accuracy provides the overall correct classification of the model, while the specificity provides the misclassification rate.

$$Accident\ Detection\ Rate\ (\%) = \frac{True\ Positives}{True\ Positives + False\ Negatives} \times 100 \quad (1)$$

$$\text{False Detection Rate (\%)} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} \times 100 \quad (2)$$

$$\text{Accuracy (\%)} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}} \times 100 \quad (3)$$

$$\text{Specificity (\%)} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Negatives}} \times 100 \quad (4)$$

4.3.1 Logistic Regression

The first model implemented was the Logistic Regression statistical model, which uses a linear combination of parameterized feature weight to create a binary output—in this case with 0 indicating no accident and 1 indicating an accident occurred. Logistic Regression application is visualized in Figure 4-4). Two process are required before model training to normalize the data by minimizing biased feature weights and to resample a balanced label as in our case where observations for non-accident far outnumber those for accident. To normalize the training feature, the MinMaxScaler library was used with a range of 0 to 1. The fitted scaler was retained as well, to apply normalization on the test data. Logistic regression was trained using k cross-validation to obtain the best penalty and cost, C values using the Grid Search algorithm. From the Grid Search, the optimal parameters for Logistic Regression were penalty= 'l2' and C=1.7575106248547894. The 'liblinear' algorithm served as the solver—one of few that supports training with L2 penalty. Results are summarized in a confusion matrix (See Figure 4-5) and Table 4-5, which indicates accuracy of correct prediction coupled with false prediction rate. Detection rate indicates the model's ability to accurately classify accidents; false detection rate indicates the percentage of non-accident observations classified as accidents. The model indicates an **accuracy of 96.49 %** and **specificity of 3.51 %**.

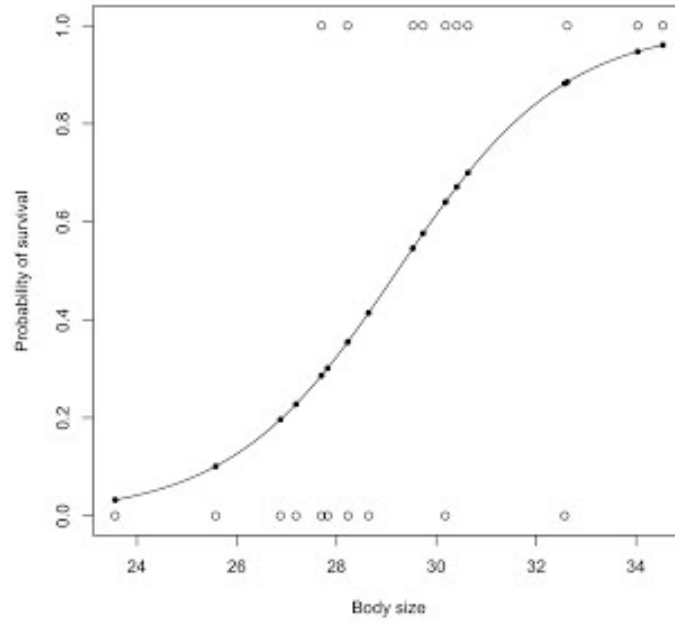


Figure 4-4. An example of logistic regression application [26].

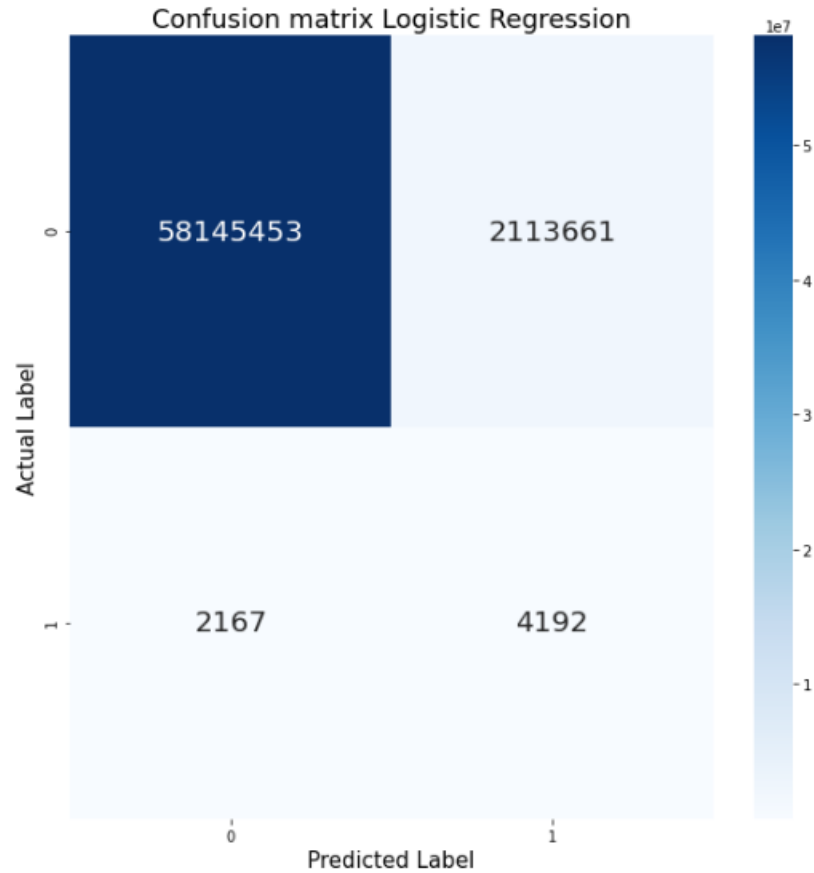


Figure 4-5. Logistic regression test confusion matrix.

Table 4-5. Detection and False Detection Rate for Logistic Regression.

Accident Detection Rate	False Detection Rate
65.92 %	35.08 %

4.3.2 Logistic Regression + Multi Adaptive Regression Spline (MARS)

To improve the logistic regression model, processes reported in this thesis included using a pipeline to train the model by adding a MARS model before forwarding output to the logistic regression model for classification output. MARS is a modelling technique that has traditionally been used for regression problems. Logistic regression

models primarily introduce a linear function, while MARS creates hinges at various knot values to produce a non-linear function that is actually a combination of linear functions that changes according to determined feature values. (e.g., how MARS produces the statistical fitting [See Figure 4-5]). The pipeline was trained with k cross-validation of 5 with grid search to obtain optimal parameters: penalty= '11' and C=3237.45754281764 for the logistic regression model and max_degree=4 for the MARS model. The confusion matrix showed a slight improvement in the accident detection rate and a decrease in the false detection rate, although these results are far from desirable (See Figure 4-7 and Table 4-6). The MARS model delivered **accuracy of 73.15 %** and **specificity of 26.85 %**.

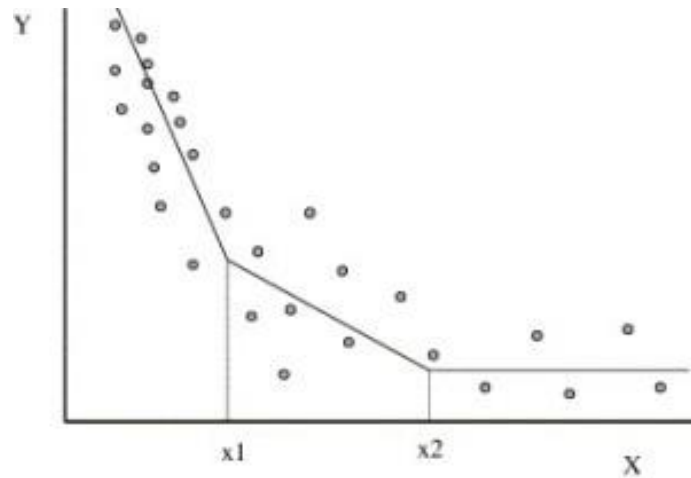


Figure 4-6. An example of MARS classification application.

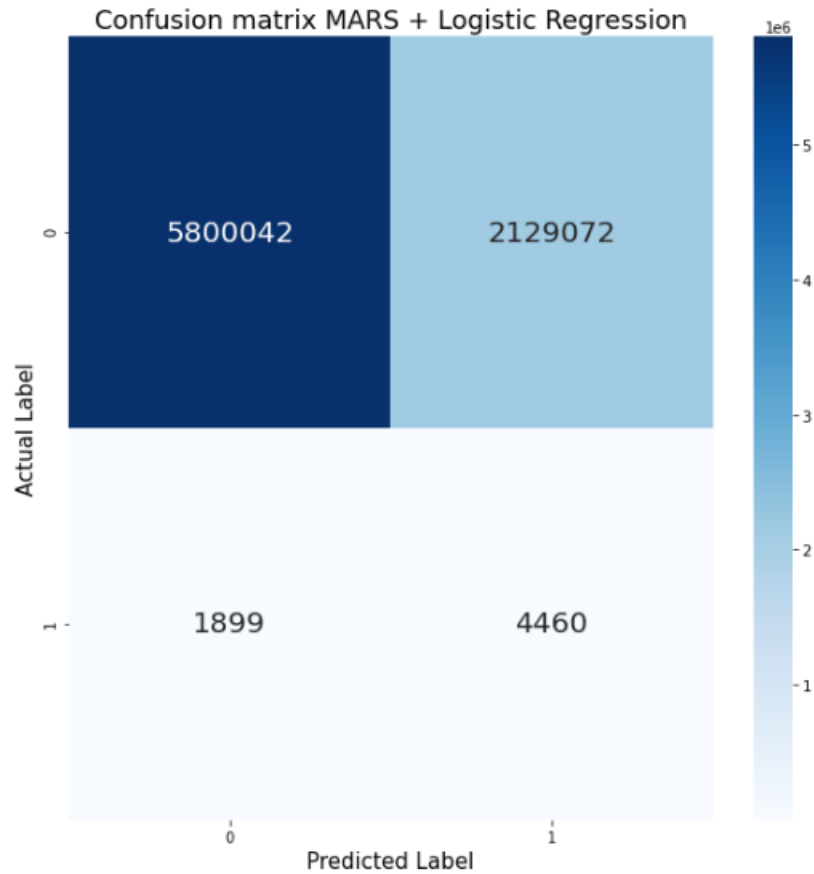


Figure 4-7. MARS + logistic regression test confusion matrix.

Table 4-6. Detection and False Detection Rate for MARS + Logistic Regression

Accident Detection Rate	False Detection Rate
70.14 %	26.85 %

4.3.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) classifier is an algorithm that defines decision boundary between features by attempting to widen the gap between labels. Points that are nearest the lines are support vectors. The goal of the algorithm is maximizing the gap (i.e., margin) between the support vectors and decision boundaries (See Figure 4-8). As part of hyperparameter optimizations, the grid search algorithm with k cross-over validation of 5 was used to determine optimal regularization parameter (C), gamma value for kernel, and

kernel type for the algorithm application. The best performing hyperparameter from the optimization function was kernel='rbf', C=500, and gamma =1. Results for SVM based classification can be found in Figure 4-9 and Table 4-7). Model **accuracy was 94.08 %**, and **specificity was 5.92 %**. Model accuracy was high primarily because the model is biased in classifying more non-accidents as the non-balanced test set, favoring non-accident data. Hence, an accident detection rate and false detection rate that are more reliable to evaluate model performance.

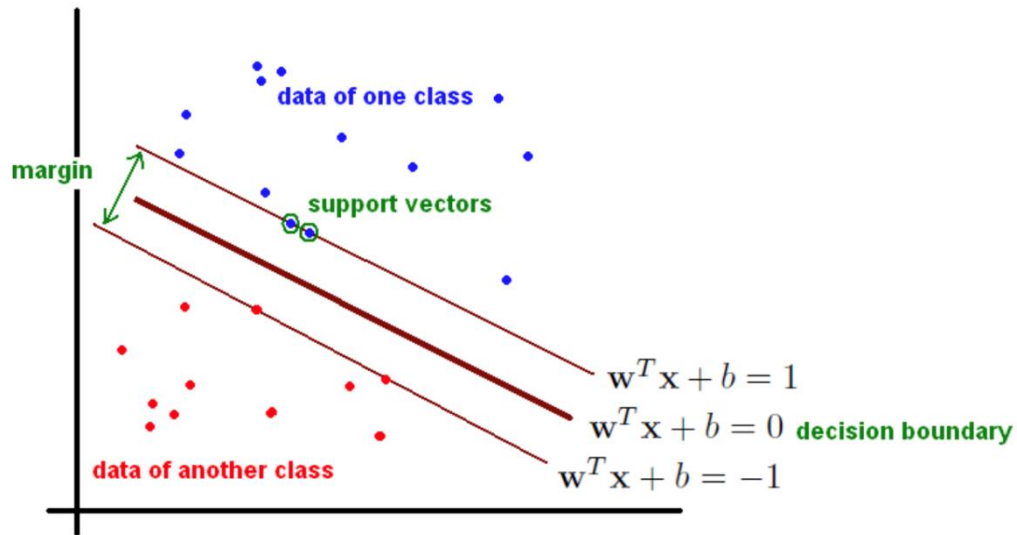


Figure 4-8. An example of SVM classification algorithm [27].

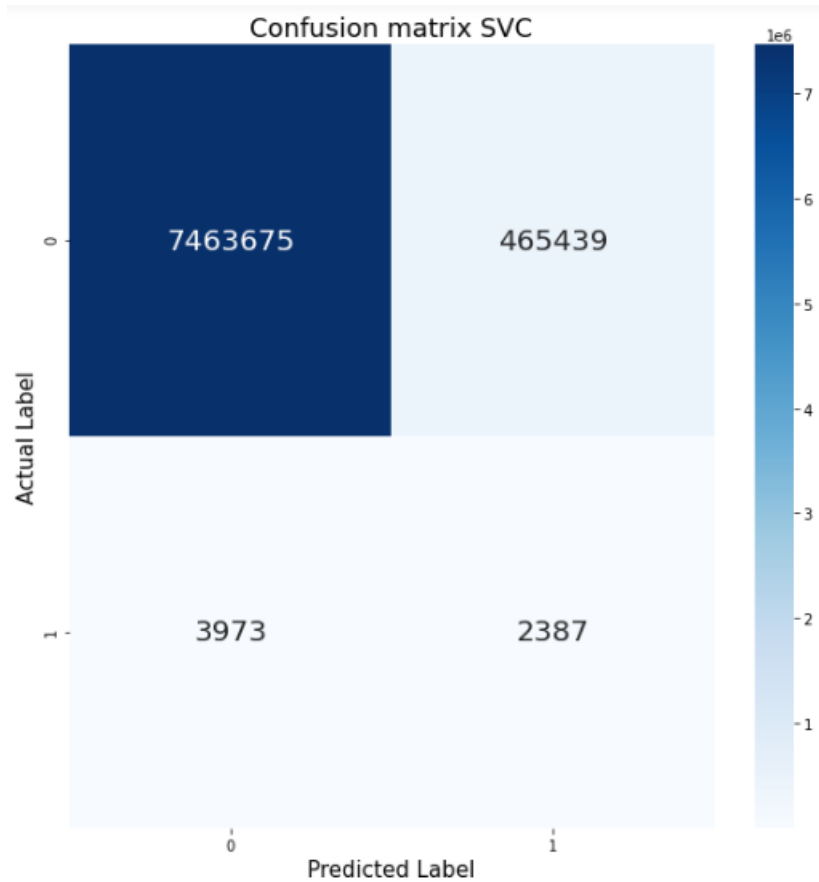


Figure 4-9. SVC’s test confusion matrix.

Table 4-7. Detection and False Detection Rate for SVC

Accident Detection Rate	False Detection Rate
37.52 %	5.87 %

4.3.4 Random Forest

Random Forest is an algorithm that combines multiple outputs from a weak classifier—typically decision trees—and determines final classification based on the majority vote of classification from the individual decision trees. Decision tree is an algorithm that can be represented as a flowchart, with an internal node representing the test

of an attribute with an outgoing branch from the node representing test outcome on the attribute. See Figure 4-10 for an example of decision tree generation. The random forest algorithm uses several uncorrelated decision trees to create a weighted classification decision for overcoming an individual decision tree error. The resulting confusion matrix for Random Forest, as well as the calculation that summarizes accident detection rate and false detection rate, shows significant capability for predicting accidents, as reported in [10]. The research in that paper also used a similar ensemble algorithm through a gradient boosting tree algorithm (See Figure 4-11 and Table 4-8). Random Forest **accuracy was 86.09 %**, and **specificity was 13.91 %**.

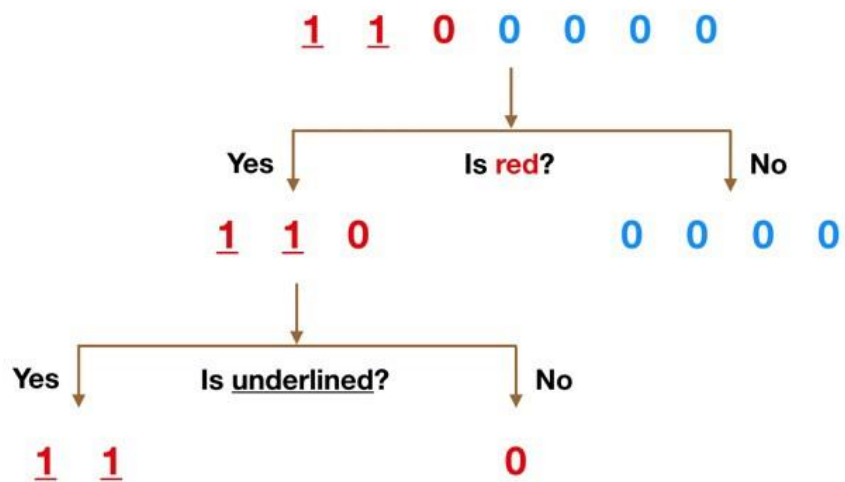


Figure 4-10. An example of the way in which a decision tree works [28].

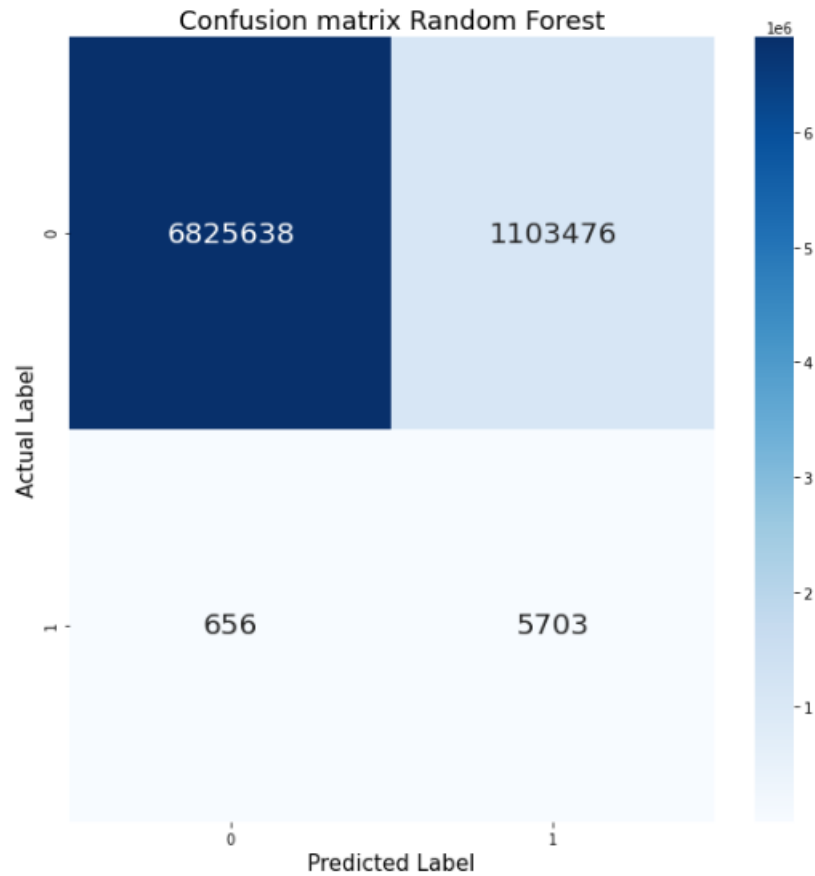


Figure 4-11. Random Forest’s test confusion matrix.

Table 4-8. Detection and False Detection Rate for Random Forest

Accident Detection Rate	False Detection Rate
89.68 %	13.92 %

4.3.5 Recurrent Neural Network (RNN)

One of the best ways to predict a time series dataset into a classification model is leveraging a recurrent neural network (RNN). RNN is recognized as an optimal neural network method for classification, as it functions well with sequential data (e.g., time series data). This feedback mechanism ensures that current output is dependent upon previous information and its order [29]. On the contrary, a normal neural network is unable to do

perform this basic training, as the system accepts only the decision of the outcome based on current information—regardless of dependency on past information. An RNN is merely an extension of regular neural network, where the self-loop function permits the equivalent of multiple copies of the same network (See Figure 4-12). Because RNN depends on backpropagation through time, either vanishing or an exploding gradient becomes an issue. Hence, RNN architecture will depend on using long short-term memory (LSTM) and gated-resistance unit (GRU) for handling the gradient problem by enforcing a constant error flow. This process has proved its ability to handle a complex, long time-lag based problem [30].

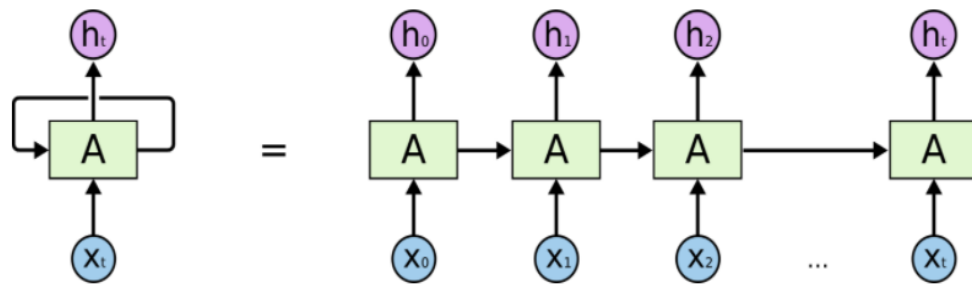


Figure 4-12.. An unrolled depiction of a single RNN.

4.3.5.1 RNN Data Preprocessing

While data preprocessing for RNN is quite similar to that for supervised machine learning described in Section 4.1, shifting must be applied to all features rather than speed alone. Regarding the supervised learning method, features were not shifted, as change in features are either negligible or primarily constant. Notably, for RNN training, data input requires a 3D shape, representing samples, features and timesteps. An example of such transformation is where average speed, speed, travel time, and others are shifted to produce $var1(t-2)$, $var2(t-2)$, $var3(t-2)$, ..., $var1(t-1)$, $var2(t-1)$, $var3(t-1)$, ... $var1(t+2)$, $var2(t+2)$,

var3(t+2) (See Figure 4-13 and Figure 4-14). Lastly, data must be normalized, as neural networks are sensitive to bias in feature weights.

	Average Speed	Reference Speed	Segment	Speed	Travel Time	Date	Incident	Hours
2017-01-01 00:00:00	NaN	NaN	NaN	NaN	NaN	2017-01-01 00:00:00	0	0
2017-01-01 00:00:00	67.0	71.0	111+04895	60.0	191.03	2017-01-01 00:00:00	0	0
2017-01-01 00:05:00	67.0	71.0	111+04895	67.0	171.08	2017-01-01 00:05:00	0	0
2017-01-01 00:10:00	67.0	71.0	111+04895	64.0	179.09	2017-01-01 00:10:00	0	0
2017-01-01 00:15:00	67.0	71.0	111+04895	65.0	176.34	2017-01-01 00:15:00	0	0
2017-01-01 00:20:00	67.0	71.0	111+04895	67.0	171.08	2017-01-01 00:20:00	0	0
2017-01-01 00:25:00	67.0	71.0	111+04895	70.0	163.74	2017-01-01 00:25:00	0	0
2017-01-01 00:30:00	67.0	71.0	111+04895	66.0	173.67	2017-01-01 00:30:00	0	0
2017-01-01 00:35:00	67.0	71.0	111+04895	79.0	145.09	2017-01-01 00:35:00	0	0
2017-01-01 00:40:00	67.0	71.0	111+04895	64.0	179.09	2017-01-01 00:40:00	0	0
2017-01-01 00:45:00	67.0	71.0	111+04895	61.0	187.90	2017-01-01 00:45:00	0	0
2017-01-01 00:50:00	67.0	71.0	111+04895	58.0	197.62	2017-01-01 00:50:00	0	0
2017-01-01 00:55:00	67.0	71.0	111+04895	76.0	150.82	2017-01-01 00:55:00	0	0
2017-01-01 01:00:00	67.0	71.0	111+04895	68.0	168.56	2017-01-01 01:00:00	0	1
2017-01-01 01:05:00	67.0	71.0	111+04895	62.0	184.87	2017-01-01 01:05:00	0	1
2017-01-01 01:10:00	67.0	71.0	111+04895	62.0	184.87	2017-01-01 01:10:00	0	1
2017-01-01 01:15:00	67.0	71.0	111+04895	62.0	184.87	2017-01-01 01:15:00	0	1
2017-01-01 01:20:00	67.0	71.0	111+04895	60.0	191.03	2017-01-01 01:20:00	0	1

Figure 4-13. Example of data before suitable transformed for RNN.

var1(t-1)	var2(t-1)	var3(t-1)	var4(t-1)	var5(t-1)	var6(t-1)	var7(t-1)	var1(t)	var2(t)	var3(t)	var4(t)	var5(t)	var6(t)	
NaN	NaN	NaN	NaN	NaN	NaN	NaN	67	71	0	60	191.03	0	
67	71	0	60	191.03	0	0	0	67	71	0	67	171.08	0
67	71	0	64	179.09	0	0	0	67	71	0	64	179.09	0
67	71	0	67	171.08	0	0	0	67	71	0	64	179.09	0
67	71	0	65	176.34	0	0	0	67	71	0	65	176.34	0
67	71	0	64	179.09	0	0	0	67	71	0	65	176.34	0
67	71	0	67	171.08	0	0	0	67	71	0	67	171.08	0
67	71	0	65	176.34	0	0	0	67	71	0	67	171.08	0
67	71	0	70	163.74	0	0	0	67	71	0	70	163.74	0
67	71	0	67	171.08	0	0	0	67	71	0	70	163.74	0
67	71	0	66	173.67	0	0	0	67	71	0	66	173.67	0
67	71	0	70	163.74	0	0	0	67	71	0	66	173.67	0
67	71	0	79	145.09	0	0	0	67	71	0	79	145.09	0
67	71	0	66	173.67	0	0	0	67	71	0	79	145.09	0
67	71	0	64	179.09	0	0	0	67	71	0	64	179.09	0
67	71	0	79	145.09	0	0	0	67	71	0	64	179.09	0

Figure 4-14. Example of data after transformation for RNN.

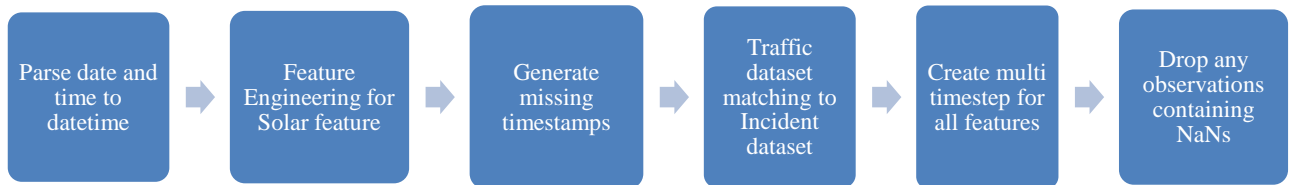


Figure 4-15. Dataset pre-processing for RNN.

4.3.5.2 RNN Architecture

Similar to any neural network architecture, RNN architecture depends on a multi-layer of connecting neurons. The architecture shown in Figure 4-16 was discovered through trial and error by comparing scoring metrics (i.e., accuracy and precision between training and validation data) for determining whether adding layers and neurons would significantly improve model performance. After input, the data passes through four layers of neurons, the first three consisting of different numbered RNN neurons and the last being a dense neuron layer with a 'relu' activation function. Additionally, two dropout layers exist between the first two RNN layers, which prevent overfitting by randomly setting input units to zero with a default rate during model training. Training criteria is also set to 1000 epochs via the early monitoring function, which stops the training when performance does not increase more than the established threshold of 5×10^{-4} for the chosen scoring metric. For this model training, the preferable scoring metric is precision rather than accuracy. The latter is a metric that defines percent of correct classification from total prediction for optimizing the model for a real-life application, whereas occurrence of accidents is far outnumbered by non-events. Instead it is preferable to approach optimization through precision scoring. Precision is defined as the ratio of correct positive identification, which, in the work for this thesis, was the ratio of all correctly classified accident samples over all samples classified as accidents. This approach prevents the model from bias in classifying positive class to achieve highly accurate results, where the chance for false positive occurs higher than usual. Model training for each epoch is based on precision scoring of the model for non-balanced validation data.

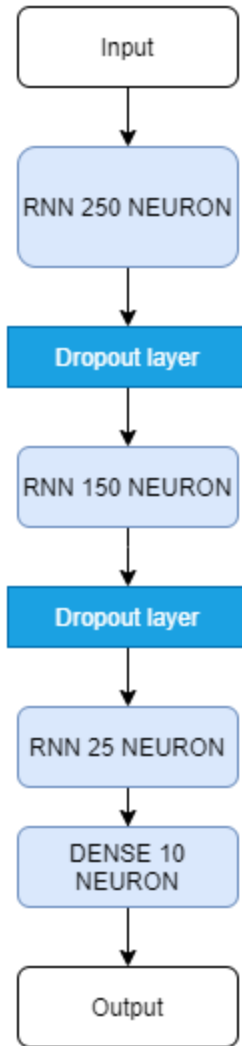


Figure 4-16. RNN model architecture.

4.3.5.3 Long Short-Term Memory

LSTM networks were introduced by [31] to overcome long term dependencies in RNN. LSTM is the most popular deep learning method in time series analysis and used even for text and memory analysis. Unlike RNN, LSTM remembers information for an extended period of time as default behavior without altering parameters. The difference in the regular RNN and LSTM is the internal gate system of the LSTM, where three gates determine the weight and importance of each previous time step values for information

flow. Gates keep past information relevant for current analysis. On the contrary, RNN information flow merely a pointwise addition that passes through tanh function. As such, RNN cannot efficiently retain prior information, especially the increasing lag values (See Figure 4-17 and See Figure 4-18). The Forget gate determines information importance. The input gate advances the hidden state and current input gates through a sigmoid function. The output gate determines the next hidden state, which will also be used to make the predictions. Results for near real-time accident detection were predicted to be superior when using LSTM architecture; (See Figure 4-19 and Table 4-9). The model's **accuracy was 80 %**, and **specificity was 20 %**.

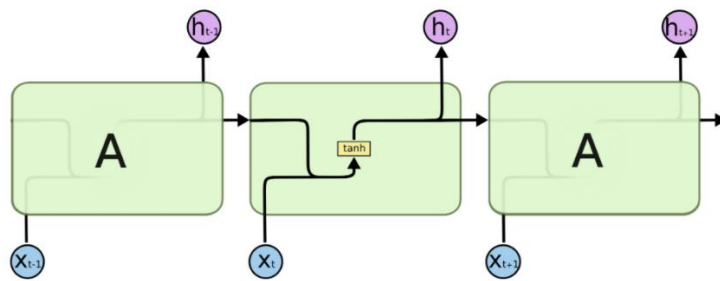


Figure 4-17. Regular RNN internal structure.

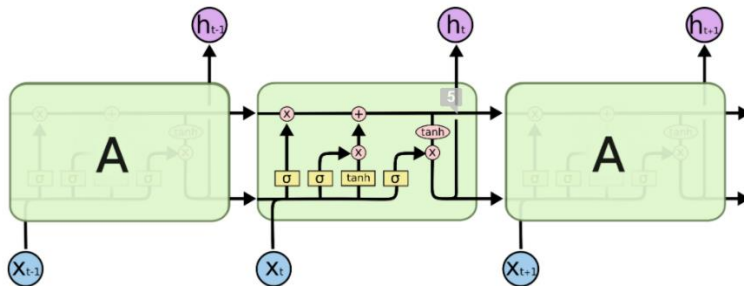


Figure 4-18. LSTM internal structure.

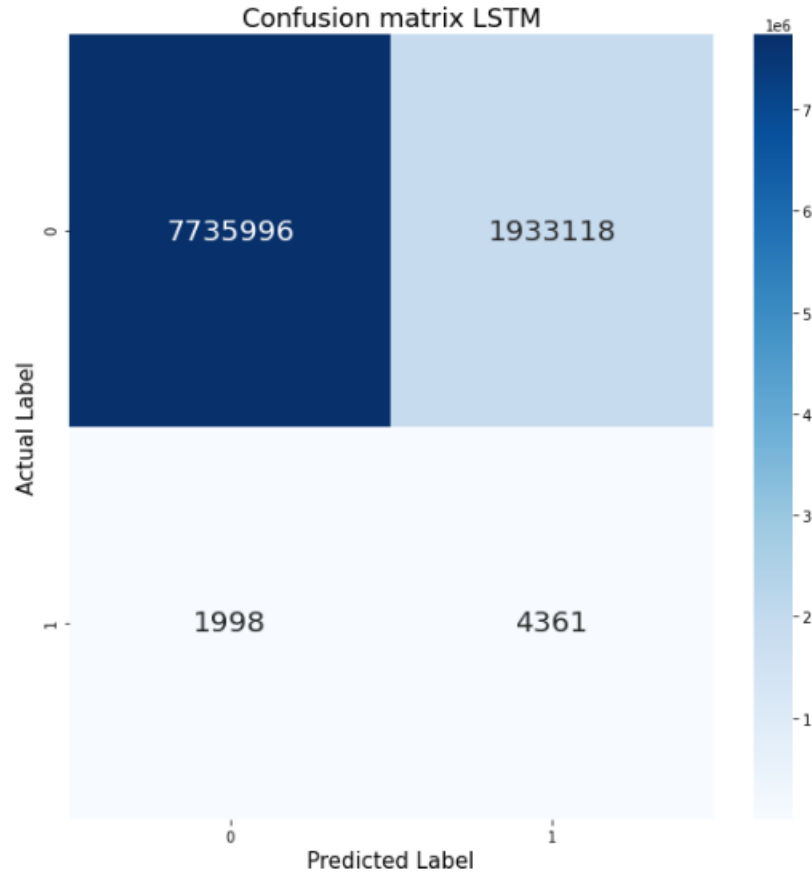


Figure 4-19. LSTM’s test confusion matrix.

Table 4-9. Detection and False Detection Rate for LSTM.

Accident Detection Rate	False Detection Rate
68.58 %	19.99 %

4.3.5.4 Gated Recurrent Unit (GRU)

Another popular variant of LSTM is known as GRU, which was first introduced in [32]. In summary, GRU is a simpler version of the LSTM model, where the forget and input are combined as the update gate; also, other changes facilitated simple internal state processes (See Figure 4-20 and Figure 4-21). The resulting confusion matrix and tabulation of the accident and false detection rates delivered a slightly worsening

performance when compared to original LSTM (See Figure 4-22 and Table 4-10). The GRU model had an **accuracy of 76.5 %**, and **specificity of 23.5 %**.

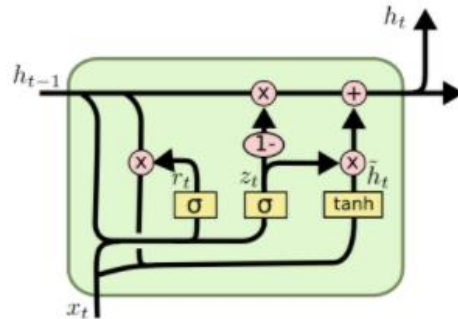


Figure 4-20. GRU internal structure.

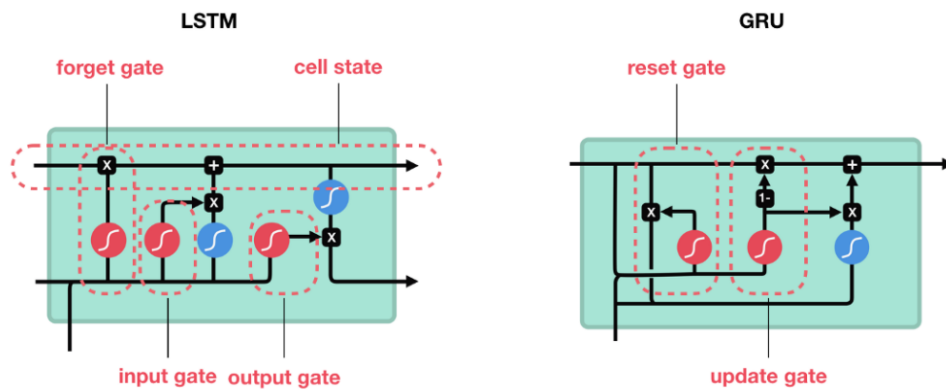


Figure 4-21. LSTM vs. GRU gates.

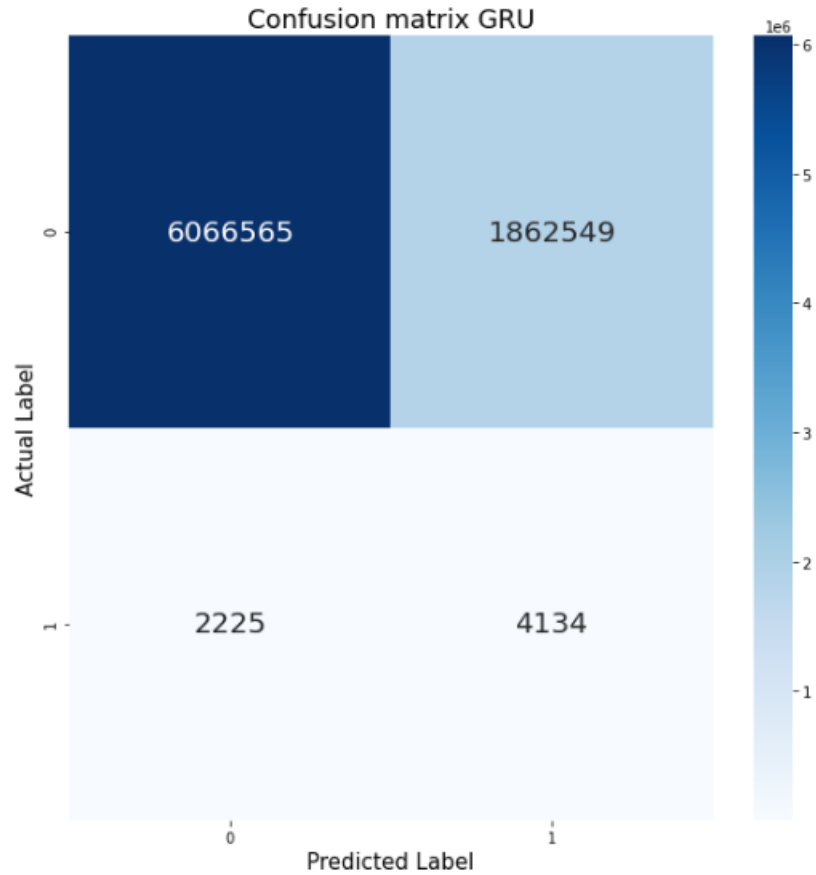


Figure 4-22. GRU’s test confusion matrix.

Table 4-10. Detection and False Detection Rate for GRU.

Accident Detection Rate	False Detection Rate
65.01 %	23.49 %

4.3.6 Summary of Classification Modelling Results

Because the test dataset is a subset of a real-world dataset, therefore a representative of it without any resampling, the non-accident samples far outnumber accident samples. Depending purely on accuracy scoring metric as the norm, the classification problem will be not a true representative of the model performance. Given that a model is biased to predict a sample as non-accident, the accuracy metric of that model will increase. This is

true for most actual accident samples, as shown when using the SVM classifier, as demonstrated in Section 4.2.3. Even so, the correct accident detection rate—when coupled with false detection rate—can be used to evaluate model performance. Optimal performance will be characterized with a high accident detection rate and low false detection rate. When comparing the performance of all the models, Random Forest model was superior, followed by LSTM, GRU, MARS + Logistic Regression, Logistic Regression, and SVC (See Figure 4-23). Although expected results were not conformed, LSTM was shown to be outperformed, yet at acceptable levels. It is acceptable to say at this specific test set, the Random Forest outperformed all other model but without a balanced sample of Negative and Positive class, our model performance metric can not be the final verdict as metrics such as ROC and AUC are more all composing in describing the model performance at various threshold values instead of just the optimized values as shown in this thesis.

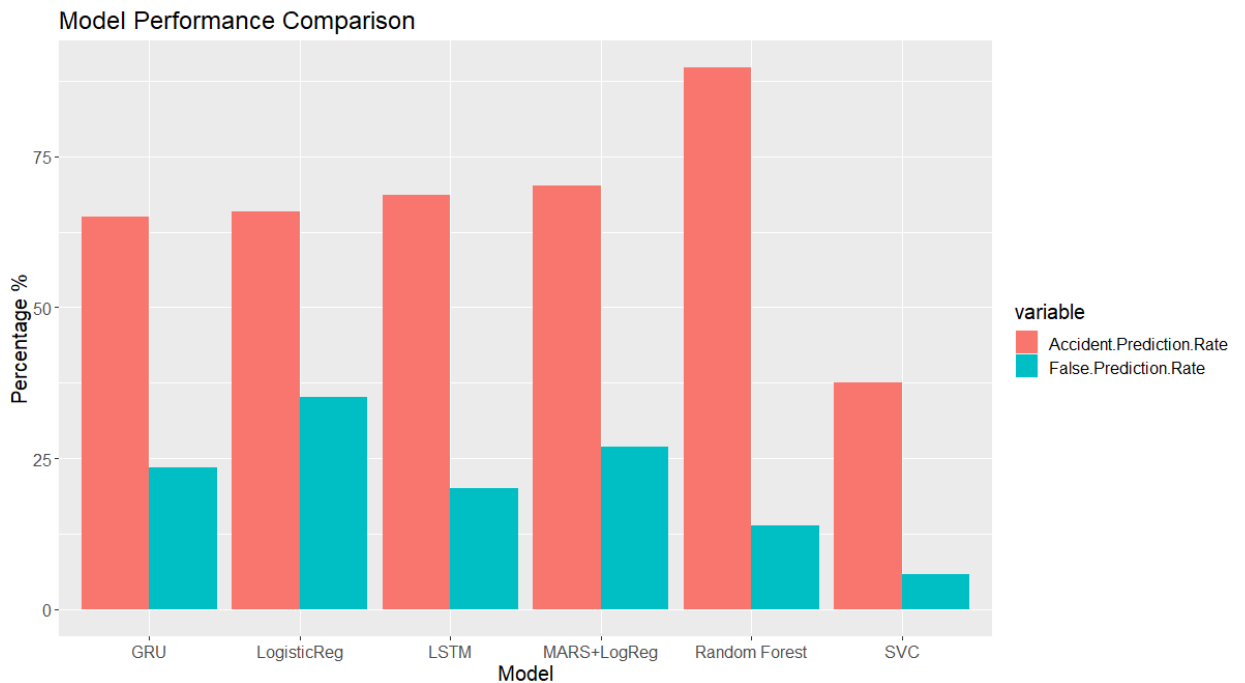


Figure 4-23. Classification model accuracies.

4.3.7 Feature Importance

In validating aforementioned previous work, this section shows the importance of certain features for predicting accident occurrences. The analysis of feature importance was performed on the Random Forest model. Feature importance is a built in-functionality on its sklearn-based modelling which shows the most important features are altitude of the sun, azimuth of the sun, speed features, travel time, and temporal information proved to be the most important features (See Figure 4-24). Longitude and Latitude of the road segment, segment length, and segment ID features were of minimal importance in the model.

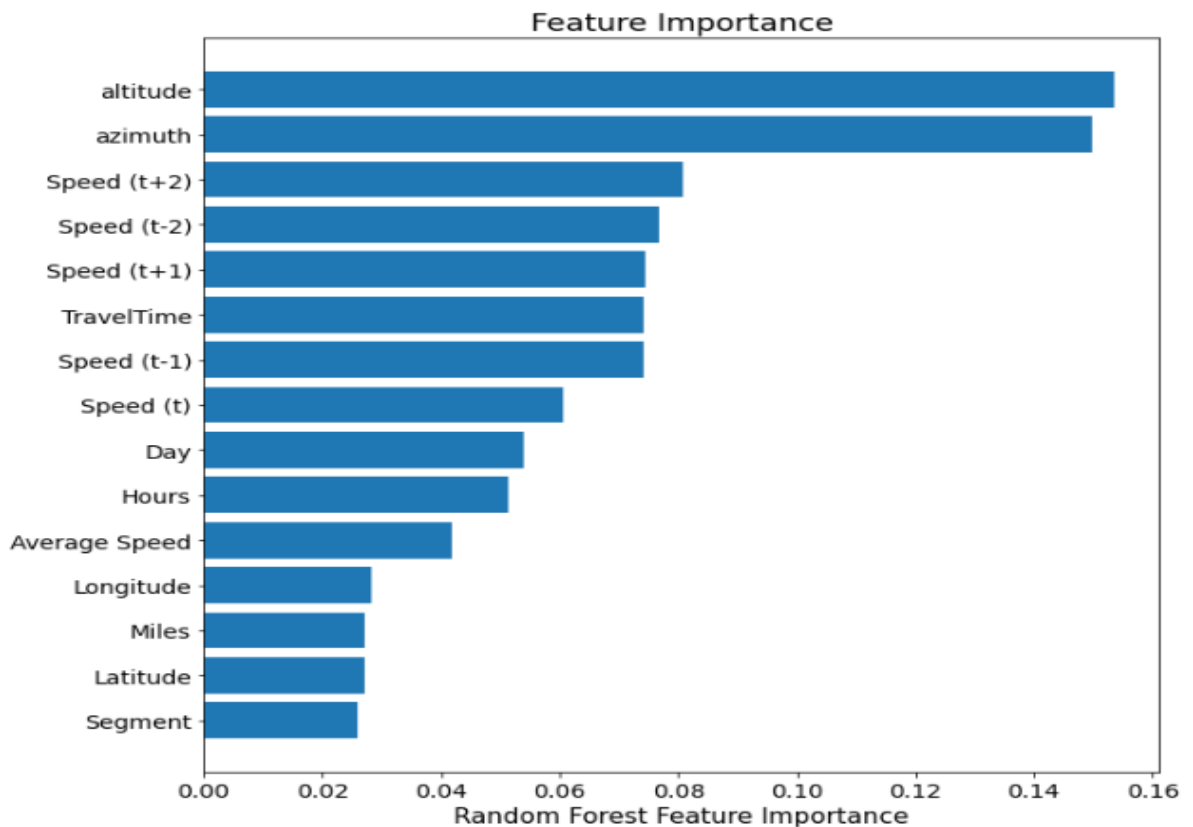


Figure 4-24. Random Forest feature importance.

5. Post Processing Classification Modelling

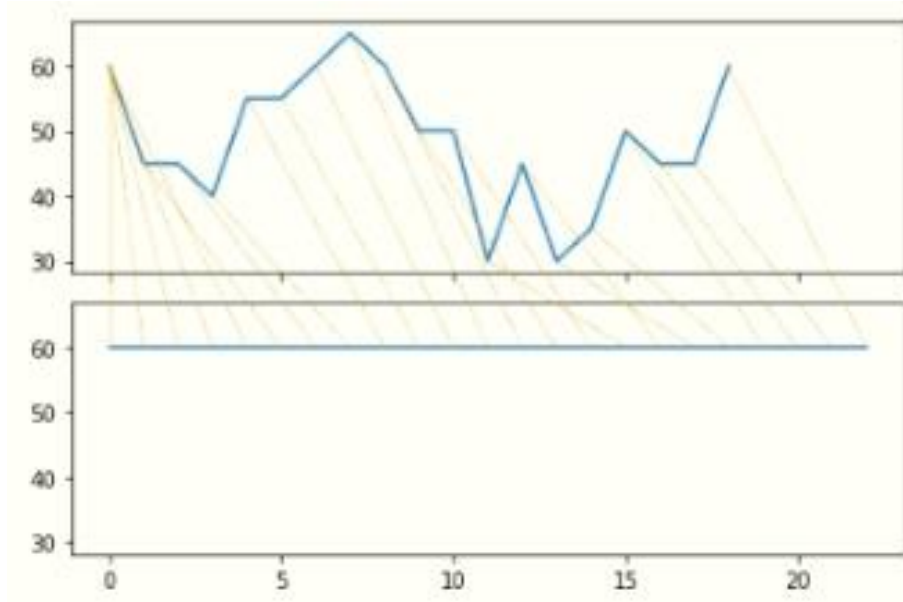
This section explains the preferable methodology for analyzing speed data in a post-processing setting. The objective is analyzing and classifying occurrences of speed drop, which are filtered according to a pre-determined speed percentage drop. The goal is differentiating between events causing traffic speed drop between regular rush hour congestion and traffic accidents. Free flow observations serve as our control class and is based on observations that demonstrate no prominent drop in speed. Validation of the correct methodology for distinguishing traffic events from historical traffic speed data will provide insights into speed turbulence occurrences that if modeled could become a valuable tool for researchers and state entities for correctly identifying and developing road maintenance plans, especially for cases where road incident data are not available.

5.1 Comparison between Dynamic Time Wrapping and Zero Padding

The first step in an effort to prepare and filter data observations for speed turbulence occurrence is to first extract the speed observations between initial time of observations to the occurrence of speed returning back close to the initial observance after the set percentage drop. One challenge to this approach is filtered observations produce data with varying time periods that cannot be solved using a regular machine learning or deep learning approach for model training. As such, several methods were investigated, including LSTM, which accepts varying data length, given as long as the batch training includes similar data length. This, however, will present another challenge, as a varying length sample doesn't necessarily produce a similar amount of data samples for each time varying length data (i.e., variance in data length is completely random). Two solutions to overcome this problem were 1) zero padding the data to create data samples of equal length

or 2) applying Dynamic Time Warping (DTW) to each data with a reference observation sample set at constant speed of 60 mph and set length. Dynamic time warping in this application will stretch observation samples to the required observation length without greatly distorting original observations. While the zero padding is the most convenient and often applied method in the literature for such problems, DTW was explored in this thesis as an alternative and perhaps more reliable data processing method for varying length samples.

DTW is a similarity measure algorithm that works like a Euclidean distance algorithm, although it was developed to measure the similarity between two observations of different lengths. Unlike Euclidean distance, which calculates the distance between observations using one on one matching, DTW first generates one-on-many or many-on-one matching between two observations to determine the distance between the two observations [33]. Implementation of the DTAIDistance library for python-based application [34] enabled the discovery of an optimum warping path between two varying length observations. Further application of the warp path as a function converted the originally shorter observation to a reference observation. For this thesis, the determined reference observation sample was one of constant speed 60 for a 3-hour 30-minute window. Multiple constant values were considered as reference sample. A high-valued constant, rather than a low-, outperformed others for optimally stretching the original sample without significant distortion (See Figure 5-1 and Figure 5-2).



After DTW.

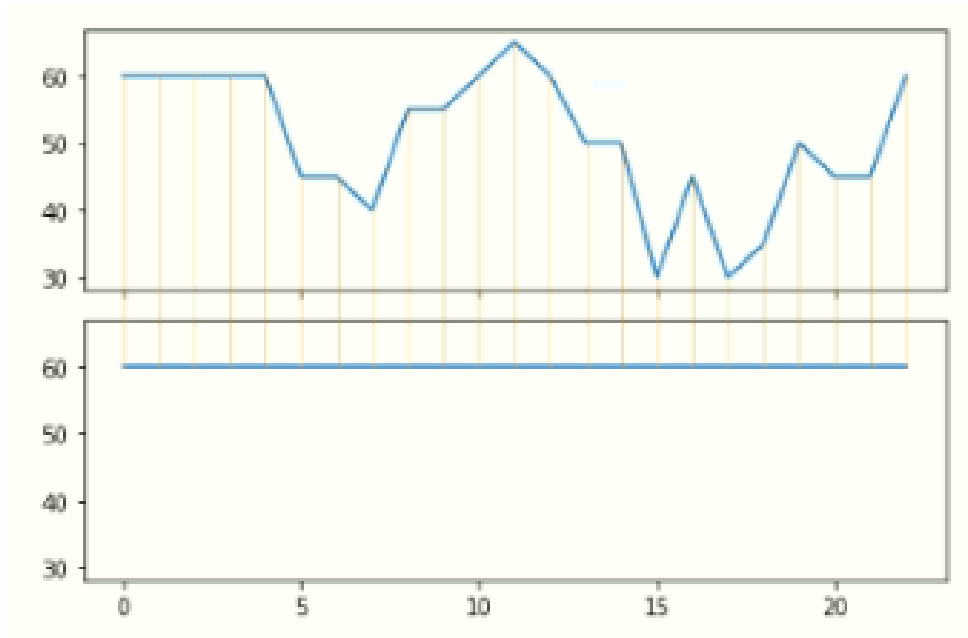
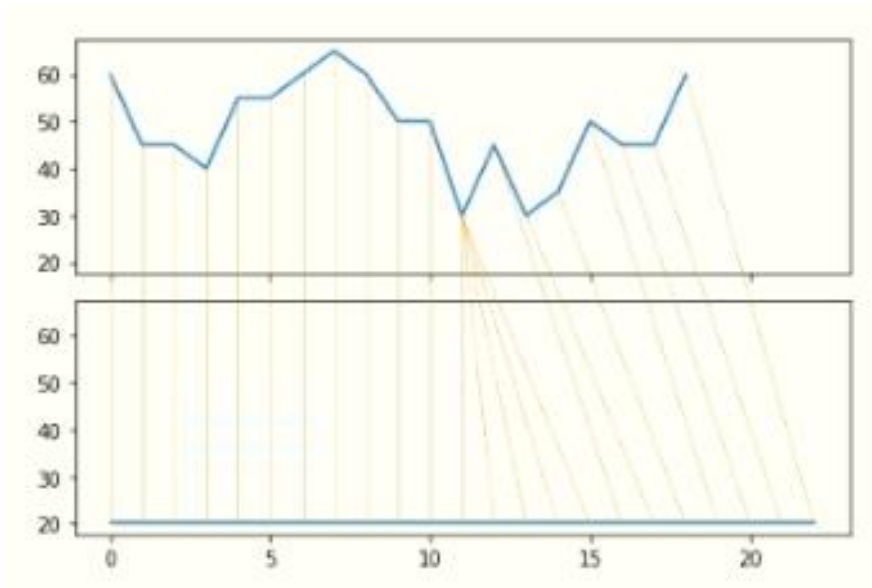


Figure 5-1. DTW observation stretching with high-constant sample.



After DTW.

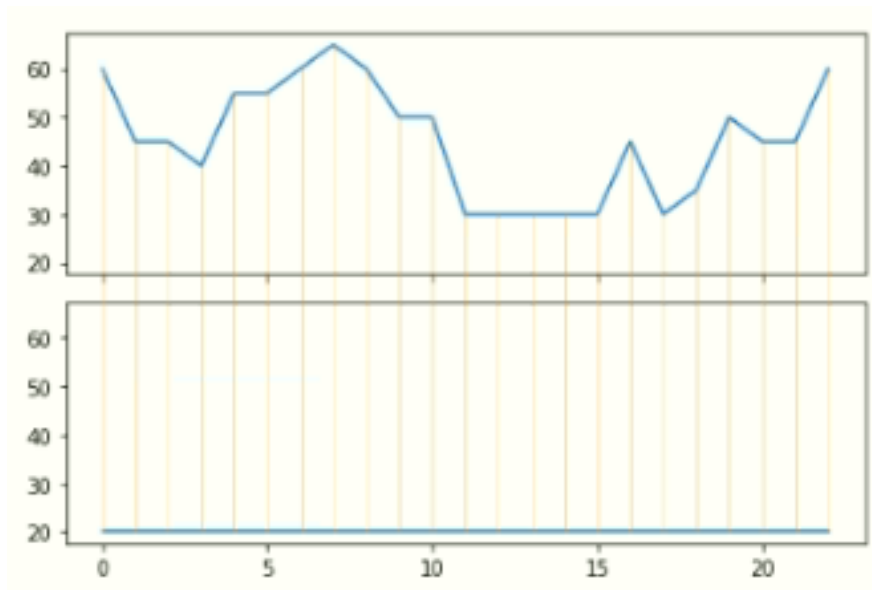


Figure 5-2. DTW observation stretching with low-constant sample.

5.1.1 Data Shifting to obtain 3-hour 30-minute observation

Data for LSTM modelling were prepared as detailed in section 4.2.6.1. The only exception was shifting data to $\text{data}(t-2)$, $\text{data}(t-1)$, ..., $\text{data}(t+2)$. Post processing required the shift to create a three-and-a-half-hour window instead of the more typical 20-minute windows, such that $\text{data}(t-12)$, $\text{data}(t-11)$, ... $\text{data}(t+30)$ were created. Difference in the window is due to the goal of the modelling where instead of real-time prediction, post-processing which does not have time limitations. Results are shown (See Figure 5-3).

Date	var1(t-12)	var2(t-12)	var3(t-12)	var4(t-12)	var1(t-11)	\
2017-01-01 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN
2017-01-01 00:05:00	NaN	NaN	NaN	NaN	NaN	NaN
2017-01-01 00:10:00	NaN	NaN	NaN	NaN	NaN	NaN
2017-01-01 00:15:00	NaN	NaN	NaN	NaN	NaN	NaN
2017-01-01 00:20:00	NaN	NaN	NaN	NaN	NaN	NaN

Date	var2(t-11)	var3(t-11)	var4(t-11)	var1(t-10)	var2(t-10)	\
2017-01-01 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN
2017-01-01 00:05:00	NaN	NaN	NaN	NaN	NaN	NaN
2017-01-01 00:10:00	NaN	NaN	NaN	NaN	NaN	NaN
2017-01-01 00:15:00	NaN	NaN	NaN	NaN	NaN	NaN
2017-01-01 00:20:00	NaN	NaN	NaN	NaN	NaN	NaN

Date	...	var3(t+28)	var4(t+28)	var1(t+29)	var2(t+29)	\
2017-01-01 00:00:00	...	6	0	59.0	2	
2017-01-01 00:05:00	...	6	0	63.0	2	
2017-01-01 00:10:00	...	6	0	66.0	2	
2017-01-01 00:15:00	...	6	0	67.0	2	
2017-01-01 00:20:00	...	6	0	66.0	2	

Date	var3(t+29)	var4(t+29)	var1(t+30)	var2(t+30)	var3(t+30)	\
2017-01-01 00:00:00	6	0	63.0	2	6	
2017-01-01 00:05:00	6	0	66.0	2	6	
2017-01-01 00:10:00	6	0	67.0	2	6	
2017-01-01 00:15:00	6	0	66.0	2	6	
2017-01-01 00:20:00	6	0	66.0	2	6	

Date	var4(t+30)
2017-01-01 00:00:00	0
2017-01-01 00:05:00	0
2017-01-01 00:10:00	0
2017-01-01 00:15:00	0
2017-01-01 00:20:00	0

Figure 5-3. Data shifting to produce 3-hour and 30-minute window.

5.1.2 Filter Congestion and Accident speed observation

There are several steps required before suitable data can be generated for LSTM model training. The first data to be gathered is in regard to observations that shows speed turbulence in the observations. For the purpose to show the validity of this methodology, the required observed speed change considered for turbulence was set to - 15 percent. Zero padding (or DTW) was required for algorithm processing (See Appendix B-3). The process was as follows.

1. Extract only speed variable observations for the entire dataset.
2. Separate the dataset for accident and non-accident observations.
3. Separate non-accident data between rush hour and non-rush hour periods [Creating Congestion and Free Flow observations].
4. Create a dataset of percent change in speed from initial speed for each step-in time [Percent Change dataset].
5. Determine which observations to retain from the Percent Change Dataset based on the percent decrease in minimum required speed. Use filtered Percent Change Dataset index to filter the accident dataset and retain only observations with significant speed drop.
6. Note that for each sample wherein a column represents an increment of speed per time step (e.g., 5-minutes), observations from the initial column (t-12) are compared with a column showing an allowance of two time steps from the point at which the Percent Change Dataset reports percent drop is returning to zero is retained.

7. Remaining columns are padded with zero (or undergo DTW) to ensure that each observation has 33 columns [(t-12) to (t+30)].
8. Step 4 thru Step 7 are repeated for the congestion dataset.
9. Three classes of observation were prepared to solve the classification problem.

LSTM model training was performed for a classification problem predicting three classes that were created according to the algorithm described above. Only the speed feature was considered for model performance comparison between zero padding and DTW processing methods.

5.1.3 Results

The LSTM model training architecture and parameters are identical to the one shown and described in Section 4.2.5.2. Data collection was limited to Oklahoma Highway I-35, like described in Chapter 4. The zero-padding data processing in the first model reported model **accuracy of 75.5 %** and **specificity of 24.5 %** (See Figure 5-4 and Table 5-1). The model trained using data warped with DTW indicated model **accuracy of 68.7 %** and **specificity of 31.3 %** (See Figure 5-5 and Table 5-2). The zero-padding based model had an overall improved performance and lower false prediction for most classes; the DTW model showed only a slight improvement in detecting accident class (e.g., increase of 0.9 %). DTW, therefore, is not an ideal candidate for solving the problem of varying length of data.

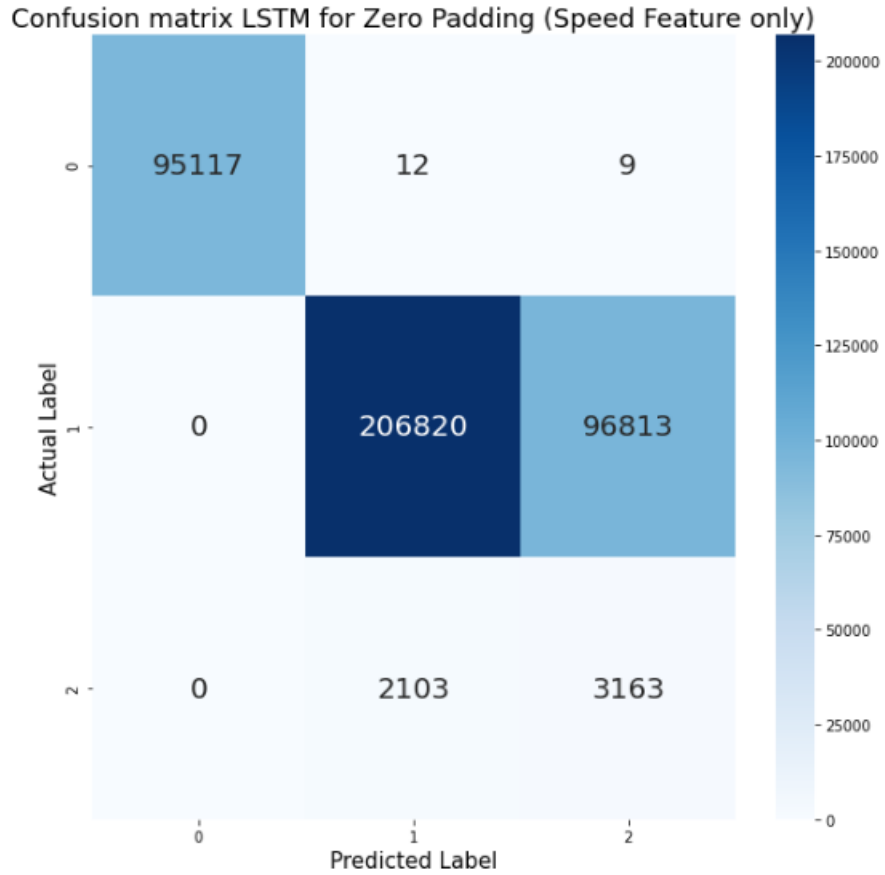


Figure 5-4. LSTM zero-padding test confusion matrix.

Table 5-1. LSTM Zero-Padding Per Class Accuracy Rate

Predicted Class	Accident Prediction Rate and False Prediction Rate		
	Free Flow	Congestion	Accident
Free Flow	99.97 %	0.00 %	0.00 %
Congestion	0.01 %	68.12 %	39.94 %
Accident	0.01 %	31.88 %	60.06 %

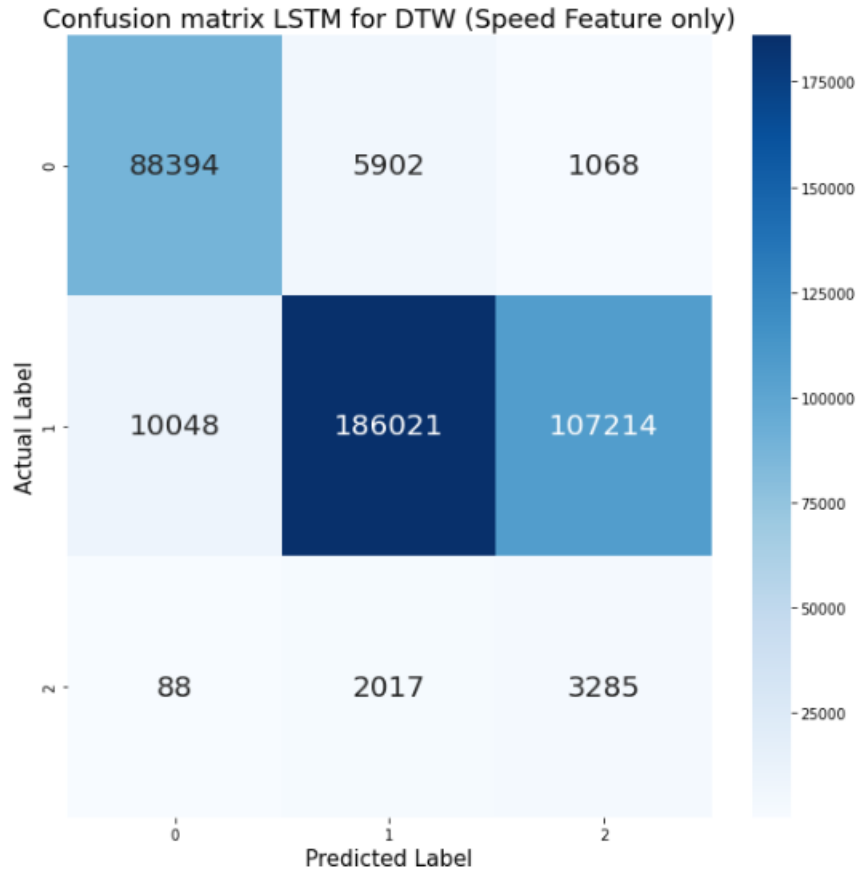


Figure 5-5. LSTM DTW’s test confusion matrix.

Table 5-2. LSTM Zero-Padding Per Class Accuracy Rate.

Predicted Class	Accident Prediction Rate and False Prediction Rate		
	Free Flow	Congestion	Accident
Free Flow	92.69 %	3.31 %	1.63 %
Congestion	6.19 %	61.34 %	37.42 %
Accident	1.12 %	35.35 %	60.95 %

5.2 LSTM modelling

Based on Section 5.1 results, the model comparison between DTW and zero padding demonstrated that the zero-padding method performs slightly better. Model improvement enables classification between free flow, congestion, and accident observation. These will be continued on the zero padding-based data processing, as this method is the norm for handling varying length data for a deep learning approach. However, this thesis showed DTW is able to process varying length data, especially for a traffic speed classification application. To improve classification, features like travel time, temporal data (i.e., month, day, hour), spatial data i.e., longitude, latitude) and solar position (i.e., azimuth and altitude) were included the model training. An early model was developed using the aforementioned features gathered from all road segments on Oklahoma Highway I-35. Results show a slight improvement when compared to initial model results reported in Figure 5-5, which only used the speed feature. Model accuracy was 91.27 % and specificity was 8.73 %, which is a near 15 percent improvement of the initial model for scoring accuracy and congestion classification, even though accident detection rate worsened (See Figure 5-6 and Table 5-3).

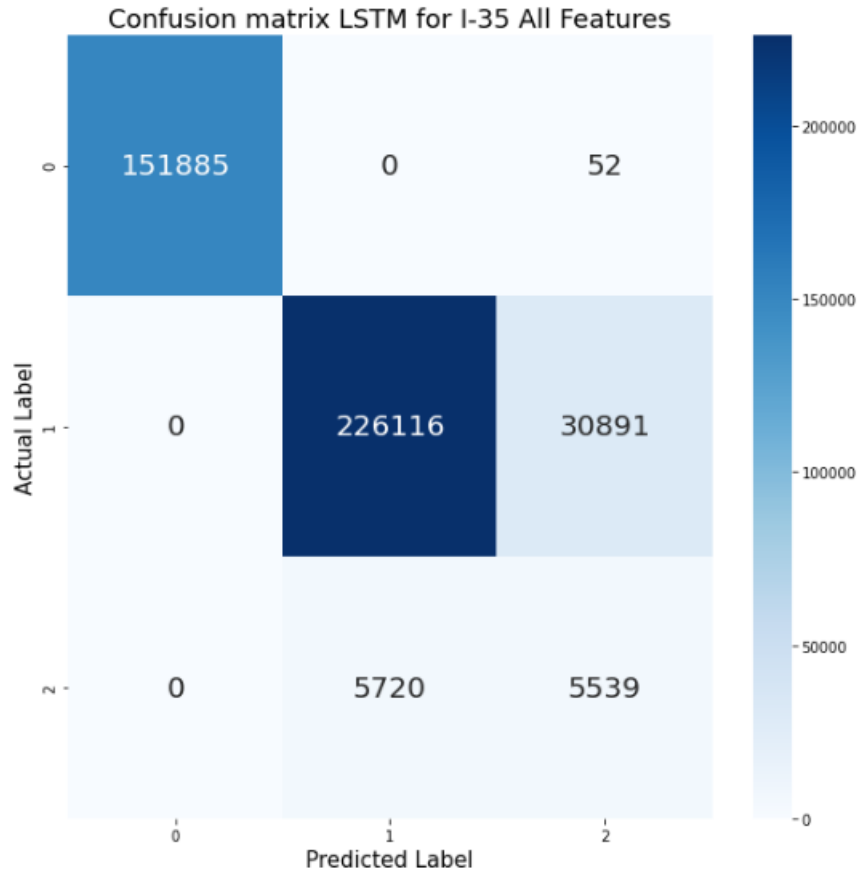


Figure 5-6. LSTM Oklahoma Highway I-35 test confusion matrix.

Table 5-3. LSTM Oklahoma Highway I-35 Per Class Accuracy Rate.

Predicted Class	Accident Prediction Rate and False Prediction Rate		
	Free Flow	Congestion	Accident
Free Flow	99.97 %	0.00 %	0.00 %
Congestion	0.00 %	87.98 %	50.80 %
Accident	0.03 %	12.02 %	49.20 %

To validate that a localized spatial modelling would yield improved classification results, two datasets were prepared using two sections of Oklahoma Highway I-35. The first section contained road segments 04910 to 04914; the second section contained 04914 to 04918. The first localized model for the first segment yielded an accuracy of 88.46 % with specificity of 11.54 % (See Figure 5-7 and Table 5-4); the second localized model for the second section yielded an accuracy of 87.81 % and specificity of 12.13 % (See Figure 5-8 and Table 5-5). Overall, the localized model reported significant performance improvement over models that considered utilizing the entire Oklahoma Highway I-35. This can be explained by considering the difficulty of processing location information. For example, label encoding does not correctly encode data based on occurrences of classification. LSTM was initially created to capture changes in temporal data. Hence the reason for the localized model reporting better results. The localized model can be validated for its ability to correctly classifying speed turbulence as based on either traffic accident or congestion.

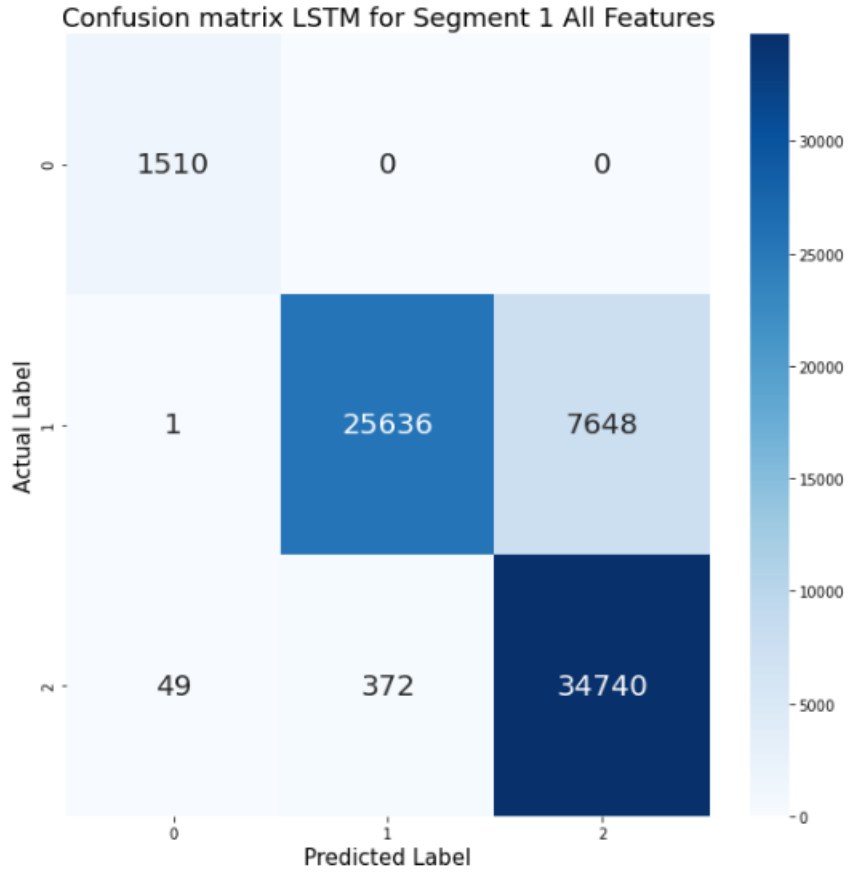


Figure 5-7. LSTM Segment 1 test confusion matrix.

Table 5-4. LSTM Segment 1 Per Class Accuracy Rate.

Predicted Class	Accident Prediction Rate and False Prediction Rate		
	Free Flow	Congestion	Accident
Free Flow	100.00 %	0.00 %	0.14 %
Congestion	0.00 %	77.02 %	1.06 %
Accident	0.00 %	22.98 %	98.8 %

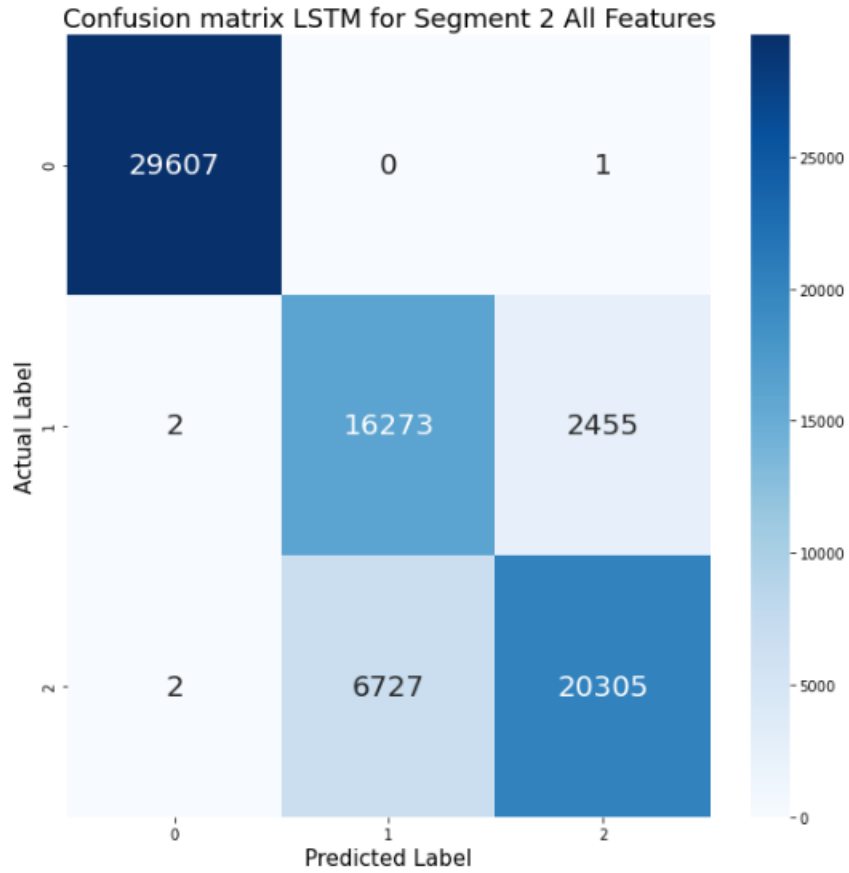


Figure 5-8. LSTM Segment 2 test confusion matrix.

Table 5-5. LSTM Segment 2's Per Class Accuracy Rate.

Predicted Class	Accident Prediction Rate and False Prediction Rate		
	Free Flow	Congestion	Accident
Free Flow	99.99 %	0.01 %	0.01 %
Congestion	0.00 %	86.88 %	24.88 %
Accident	0.00 %	13.11 %	75.11 %

6. Conclusion and Future Work

The work presented in this thesis covers several traffic accident-based analyses, methodology of analyses, model training, and model validation. The work herein demonstrates the best possible approach for acquiring and preparing necessary data from NPMRDS for use with supervised learning and RNN model training to execute near real-time accident detection. Research showed that the ensemble-based Random Forest algorithm was the best performing model with an accident detection rate of 89.68 % and false detection rate of 13.92 %. The LSTM model delivered 68.58 % accident detection and 19.99 % rate of false detection. Most others reported in the literature did not achieve this level of performance. Of those that did, results were achieved only through implementation of simulation or synthetic data usage for model training and validation. Results reported in this thesis showed that NPMRDS offered reliable real-world data with the possibility of real-time implementation. With regard to classification of historical traffic data for identifying speed turbulent occurrences resulting from either traffic accidents or regular traffic congestion, the best process included localized modelling, where accident detection accuracy ranged from 87 % to 88 % and congestion detection rate was well above the 75 %.

NPMRDS has tremendous undiscovered potential, especially for traffic accident analysis and traffic parameter detection problems, where additional features and modelling methods can be implemented. It is recommended that in order to improve and resolve current methodologies, researchers should implement data collection abilities that can always be able to collect the necessary traffic data per epoch per segment instead of relying on unreliable speed probe-based data collection. Features like road geometry can also be

collected via test drives on a roadway under investigation using a road geometry analysis tool. Likewise, NOAA weather stations could be utilized to include weather information and analyze effects of weather and road condition on traffic accidents and congestion. ODOT and WECAD are contributing to this advancement by deploying multiple Road Weather Information System (RWIS) across Oklahoma highways in an effort to provide real-time weather information. LSTM is a well-suited approach for capturing temporal aspects; however, this model was not intended to capture spatial aspects. Hence, new techniques, such as CNN-LSTM, 3D-based LSTM and others, should be considered.

References

- [1] World Health Organization, "Road Traffic injuries",WHO, 7 February 2020. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>. [Accessed 14 November 2020]
- [2] United States Department of Transportation, National Highway Traffic Safety Administration CrashStats, NHTSA, 13 November 2019. [Online]. Available: <https://www-fars.nhtsa.dot.gov/Main/index.aspx>. [Accessed 14 November 2020]
- [3] G. F. List, B. Williams, N. Roupail, R. Hranac, T. Barkley, E. Mai, A. Ciccarelli, L. Rodegerdts, A. F. Karr, X. Zhou, J. Wojtowicz, J. Schofer, and A. Khattak, "Guide to Establishing Monitoring Programs for Travel Time Reliability: SHRP 2 Report S2-LO2-RR-2," FHWA, Washington, D.C., 2014.
- [4] FHWA Office of Operations, "National performance management research data set (NPMRDS) information," FHWA, 23 June 2015. [Online]. Available: http://www.ops.fhwa.dot.gov/perf_measurement/. [Accessed 9 September 2020].
- [5] FHWA Office of Operations, "2013 urban congestion trends," FHWA, 23 April 2015. [Online]. Available: <http://www.ops.fhwa.dot.gov/publications/fhwahop15005/index.htm>. [Accessed 9 September 2020].
- [6] INRIX Press Releases, "INRIX Selected by The U.S. Federal Highway Administration for National Traffic Data Set," INRIX, 5 June 2017. [Online]. Available: <https://inrix.com/press-releases/npmrds/>. [Accessed 14 November 2020].
- [7] Bitar, Naim. Big Data Analytics in Transportation Networks Using the NPMRDS / by Naim Bitar. (2016). Web.
- [8] SGI Canada, "2017 Saskatchewan Traffic Accident Facts", 2017. [Online]. Available: <https://www.sgi.sk.ca/documents/625510/627017/TAIS+2017+Annual+Report/6112c88a-c088-44e6-9aa5-415450cc0ebf>. [Accessed 14 November 2020].
- [9] R. Tian, Z. Yang and M. Zhang, "Method of road traffic accidents causes analysis based on data mining", 2010 Int. Conf. Computational Intelligence and Software Engineering, pp. 1-4, 2010.
- [10] Reveron, Enrique, and Cretu, Ana-Maria. "A Framework for Collision Prediction Using Historical Accident Information and Real-time Sensor Data: A Case Study for the City of Ottawa." 2019 IEEE International Symposium on Robotic and

Sensors Environments (ROSE) (2019): 1-7. Web.

- [11] Lee, Jonghak, Chae, Junghyo, Yoon, Taekwan, and Yang, Hojin. "Traffic Accident Severity Analysis with Rain-related Factors Using Structural Equation Modeling – A Case Study of Seoul City." *Accident Analysis and Prevention* 112 (2018): 1-10. Web.
- [12] Mondal, P. "Are Road Accidents Affected by Rainfall? A Case Study from a Large Indian Metropolitan City." *British Journal of Applied Science & Technology* 1.2 (2011): 16-26. Web.
- [13] Peng, Yichuan, Jiang, Yuming, Lu, Jian, and Zou, Yajie. "Examining the Effect of Adverse Weather on Road Transportation Using Weather and Traffic Sensors." *PloS One* 13.10 (2018): E0205409. Web.
- [14] Zhang, Hui, Li, Siyao, Wu, Chaozhong, Zhang, Qi, and Wang, Yafen. "Predicting Crash Frequency for Urban Expressway considering Collision Types Using Real-Time Traffic Data." *Journal of Advanced Transportation* 2020 (2020): 1-8. Web.
- [15] K. Xie, X. Wang, H. Huang, and X. Chen, "Corridor-level signalized intersection safety analysis in Shanghai, China using Bayesian hierarchical models," *Accident Analysis & Prevention*, vol. 50, pp. 25–33, 2013.
- [16] Wen Huiying, Luo Jun, Chen Xiaolong, and Quo Xiaohui. "Real-time Highway Accident Prediction Based on Grey Relation Entropy Analysis and Probabilistic Neural Network." 2011 International Conference on Electric Technology and Civil Engineering (ICETCE) (2011): 1420-423. Web.
- [17] Lv Yuejing, Zhou Xing-lin, Zhang Haixia, Liu Ming, and Li Jie. "Research on Accident Prediction of Intersection and Identification Method of Prominent Accident Form Based on Back Propagation Neural Network." 2010 International Conference on Computer Application and System Modeling (ICCASM 2010) 1 (2010): V1-434-1-438. Web.
- [18] Ju-Won Hwang, Young-Seol Lee, and Sung-Bae Cho. "Hierarchical Probabilistic Network-Based System for Traffic Accident Detection at Intersections." 2010 7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (2010): 211-16. Web.
- [19] Sheng Li, and Dongmei Zhao. "Prediction of Road Traffic Accidents Loss Using Improved Wavelet Neural Network." 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering. TENCOM '02. Proceedings 3 (2002): 1526-529 Vol.3. Web.

- [20] Polat, Kemal, Polat, Kemal, Durduran, S Savaş, and Durduran, S Savaş. "Automatic Determination of Traffic Accidents Based on KMC-based Attribute Weighting." *Neural Computing & Applications* 21.6 (2012): 1271-279. Web.
- [21] Zhaojian Li, Kolmanovsky, Ilya, Atkins, Ella, Jianbo Lu, Filev, Dimitar P, and Micheleni, John. "Road Risk Modeling and Cloud-Aided Safety-Based Route Planning." *IEEE Transactions on Cybernetics* 46.11 (2016): 2473-483. Web.
- [22] Yisheng Lv, Shuming Tang, and Hongxia Zhao. "Real-Time Highway Traffic Accident Prediction Based on the K-Nearest Neighbor Method." 2009 International Conference on Measuring Technology and Mechatronics Automation 3 (2009): 547-50. Web.
- [23] Oklahoma Department of Transportation, NPMRDS Tools, 2020. Available: <https://speed.tulsa.ou.edu/npmrdsv2/login/log>. [Accessed 14 November 2020].
- [24] Bishop, Stuart. "Pytz." PyPI, Nov. 2020. Available: pypi.org/project/pytz/. [Accessed 14 November 2020].
- [25] Zebner, Holger, et al. "Staring Directly at the Sun since 2007." Pysolar, 2014. Available: pysolar.readthedocs.io/en/latest/. [Accessed 14 November 2020].
- [26] School of Biological Science, University of Nebraska-Lincoln, "Plotting logistic regression in R", Shizuka Lab. [Online]. Available: <https://sites.google.com/site/daishizuka/toolkits/plotting-logistic-regression-in-r>. [Accessed 14 November 2020].
- [27] Machine Learning with R, Chapter 13: Support Vector Machine. Available: https://www.google.com/url?sa=i&url=https%3A%2F%2Ffderyckel.github.io%2Fmachinelearningwithr%2Fsvm.html&psig=AOvVaw2GjgX4A23jqcJDuzdb3c3q&ust=1603827105188000&source=images&cd=vfe&ved=0CAkQjhxqFwoTCMDO093_0uwCFQAAAAAdAAAAABAJ. [Online]. [Accessed 14 November 2020].
- [28] Yiu, Tony, "Understanding Random Forest: How the Algorithm Works and Why it is so effective", 12 June 2019. [Online]. Available: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>. [Accessed by 14 November 2020].
- [29] Smagulova, Kamilya, Kazybek Adam, Olga Krestinskaya, and Alex Pappachen James. "Design of CMOS-memristor Circuits for LSTM Architecture." (2018): IEEE International Conferences on Electron Devices and Solid-State Circuits, 2018. Web.
- [30] Gers, Felix A., Schmidhuber, Jürgen, and Cummins, Fred. "Learning to Forget: Continual Prediction with LSTM." *Neural Computation* 12.10 (2000): 2451-471.

Web.

- [31] Hochreiter, Sepp, and Schmidhuber, Jürgen. "Long Short-Term Memory." *Neural Computation* 9.8 (1997): 1735-780. Web.
- [32] Cho, Kyunghyun, Van Merriënboer, Bart, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. "Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation." *Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation* (2014). Web.
- [33] Zhang, Jeremy, "Dynamic Time Warping", 1 February 2020. [Online]. Available: <https://towardsdatascience.com/dynamic-time-warping-3933f25fcdd>. [Accessed 14 November 2020].
- [34] Meert, Wannes, DTW library for Python, 2017. [Online]. Available: <https://dtaidistance.readthedocs.io/en/latest/modules/dtw.html>. [Accessed 14 November 2020].

Appendix

Appendix A:

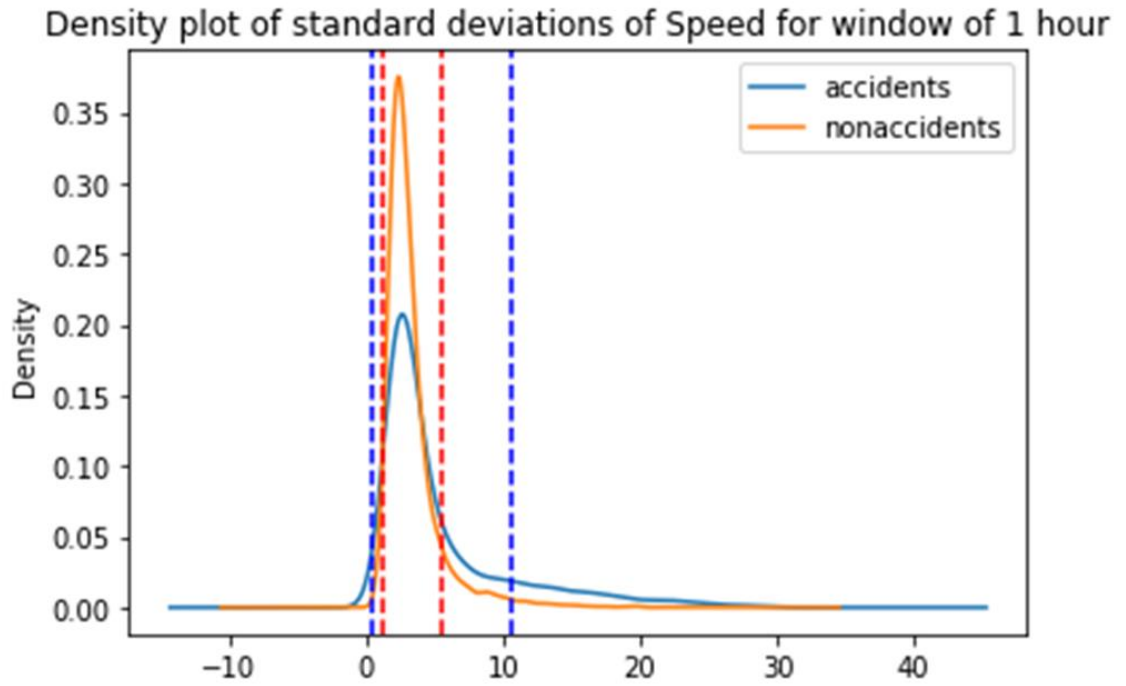


Figure A-1. Density plot of standard deviations of speed change observations

Appendix B:

Date	Segment	Speed	Average Speed	Hours	Day	Miles	\
2017-01-01 00:00:00	0	60.0	67.0	0	6	3.1839	
2017-01-01 00:05:00	0	67.0	67.0	0	6	3.1839	
2017-01-01 00:10:00	0	64.0	67.0	0	6	3.1839	
2017-01-01 00:15:00	0	65.0	67.0	0	6	3.1839	
2017-01-01 00:20:00	0	67.0	67.0	0	6	3.1839	
2017-01-01 00:25:00	0	70.0	67.0	0	6	3.1839	
2017-01-01 00:30:00	0	66.0	67.0	0	6	3.1839	
2017-01-01 00:35:00	0	79.0	67.0	0	6	3.1839	
2017-01-01 00:40:00	0	64.0	67.0	0	6	3.1839	
2017-01-01 00:45:00	0	61.0	67.0	0	6	3.1839	
2017-01-01 00:50:00	0	58.0	67.0	0	6	3.1839	
2017-01-01 00:55:00	0	76.0	67.0	0	6	3.1839	
2017-01-01 01:00:00	0	68.0	67.0	1	6	3.1839	
2017-01-01 01:05:00	0	62.0	67.0	1	6	3.1839	
2017-01-01 01:10:00	0	62.0	67.0	1	6	3.1839	
2017-01-01 01:15:00	0	62.0	67.0	1	6	3.1839	
2017-01-01 01:20:00	0	60.0	67.0	1	6	3.1839	
2017-01-01 01:25:00	0	63.0	67.0	1	6	3.1839	
2017-01-01 01:30:00	0	NaN	NaN	1	6	3.1839	
2017-01-01 01:35:00	0	NaN	NaN	1	6	3.1839	

Date	Longitude	Latitude	TravelTime	altitude	azimuth
2017-01-01 00:00:00	-97.427137	35.099841	191.034000	-31.513458	97.377400
2017-01-01 00:05:00	-97.427137	35.099841	171.075224	-30.500195	98.013353
2017-01-01 00:10:00	-97.427137	35.099841	179.094375	-29.488509	98.646099
2017-01-01 00:15:00	-97.427137	35.099841	176.339077	-28.478514	99.276127
2017-01-01 00:20:00	-97.427137	35.099841	171.075224	-27.470326	99.903907
2017-01-01 00:25:00	-97.427137	35.099841	163.743429	-26.464060	100.529888
2017-01-01 00:30:00	-97.427137	35.099841	173.667273	-25.459831	101.154501
2017-01-01 00:35:00	-97.427137	35.099841	145.089114	-24.457755	101.778163
2017-01-01 00:40:00	-97.427137	35.099841	179.094375	-23.457947	102.401275
2017-01-01 00:45:00	-97.427137	35.099841	187.902295	-22.460525	103.024225
2017-01-01 00:50:00	-97.427137	35.099841	197.621379	-21.465606	103.647390
2017-01-01 00:55:00	-97.427137	35.099841	150.816316	-20.473310	104.271138
2017-01-01 01:00:00	-97.427137	35.099841	168.559412	-19.483757	104.895824
2017-01-01 01:05:00	-97.427137	35.099841	184.871613	-18.497070	105.521797
2017-01-01 01:10:00	-97.427137	35.099841	184.871613	-17.513375	106.149399
2017-01-01 01:15:00	-97.427137	35.099841	184.871613	-16.532796	106.778964
2017-01-01 01:20:00	-97.427137	35.099841	191.034000	-15.555465	107.410819
2017-01-01 01:25:00	-97.427137	35.099841	181.937143	-14.581511	108.045289
2017-01-01 01:30:00	-97.427137	35.099841	NaN	-13.611070	108.682692
2017-01-01 01:35:00	-97.427137	35.099841	NaN	-12.644280	109.323344

Date	Incident
2017-01-01 00:00:00	0
2017-01-01 00:05:00	0
2017-01-01 00:10:00	0
2017-01-01 00:15:00	0
2017-01-01 00:20:00	0
2017-01-01 00:25:00	0
2017-01-01 00:30:00	0
2017-01-01 00:35:00	0
2017-01-01 00:40:00	0
2017-01-01 00:45:00	0
2017-01-01 00:50:00	0
2017-01-01 00:55:00	0
2017-01-01 01:00:00	0
2017-01-01 01:05:00	0
2017-01-01 01:10:00	0
2017-01-01 01:15:00	0
2017-01-01 01:20:00	0
2017-01-01 01:25:00	0
2017-01-01 01:30:00	0
2017-01-01 01:35:00	0

Figure B-1. Example of Data being prepared for Supervised Learning

```

# Function to convert series to supervised learning
def series_to_supervised(data, n_in=1, n_out=1, dropnan=True):
    n_vars = 1 if type(data) is list else data.shape[1]
    df = DataFrame(data)
    cols, names = list(), list()
    # input sequence (t-n, ... t-1)
    for i in range(n_in, 0, -1):
        cols.append(df.shift(i))
        names += [('var%d(t-%d)' % (j+1, i)) for j in range(n_vars)]
    # forecast sequence (t, t+1, ... t+n)
    for i in range(0, n_out):
        cols.append(df.shift(-i))
        if i == 0:
            names += [('var%d(t)' % (j+1)) for j in range(n_vars)]
        else:
            names += [('var%d(t+%d)' % (j+1, i)) for j in range(n_vars)]
    # put it all together
    agg = concat(cols, axis=1)
    agg.columns = names
    # drop rows with NaN values
    if dropnan:
        agg.dropna(inplace=True)
    return agg

```

Figure B-2. Data Shifting function to convert time series data to supervised learning

```

import re
numericonly = re.compile(r'^\d+-$')
df1['drop'] = 0
accident_list = []
i=0
for index, row in df.iterrows():
    idx = (row <= -0.15).idxmax(axis=1)
    temp = df.iloc[index]
    temp = temp.loc[idx:]
    idx1 = (temp > -0.05).idxmax(axis=1)
    temp_ind = idx1[6:9]
    xtemp = numericonly.sub('',temp_ind)
    if not xtemp:
        new_index = 3
    elif int(xtemp) == -3:
        new_index = int(xtemp) + 4
    else:
        new_index = int(xtemp) + 3
    if new_index >= 30:
        new_index = 30
    if new_index >= 0:
        new_index = str(new_index)
        index_str = 'var1(t'+new_index+)'
    else:
        new_index = str(new_index)
        index_str = 'var1(t'+new_index+)'
    if idx == idx1:
        df1.loc[index, 'drop'] = 1
    else:
        df1.loc[index,index_str] = 0
    #array = df1.loc[index,:].to_numpy()
    #array = array[~np.isnan(array)]
    #c = dtw.warp(array, template60)
    #array1 = c[:,0]
    #accident_list.append(array1)

```

Figure B-3. Algorithm function to apply data filtering for zero padding/DTW