Evaluation of Machine Learning Algorithms as Predictive Tools in Road Safety Analysis

A Thesis Submitted to the College of

Graduate and Postdoctoral Studies

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

Department of Civil, Geological, and Environmental Engineering

University of Saskatchewan

By

Saeid Tayebikhorami

Permission To Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis/dissertation in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying, publication, or use of this thesis/dissertation or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other uses of materials in this thesis/dissertation in whole or part should be addressed to:

Head of the Department of Civil, Geological, and Environmental Engineering
University of Saskatchewan
57 Campus Drive, Engineering Building
Saskatoon, Saskatchewan S7N 5A9, Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9, Canada

Abstract

The Highway Safety Manual (HSM)'s road safety management process (RSMP) represents the state-of-the-practice procedure that transportation professionals employ to monitor and improve safety on existing roadway sites. RSMP requires the development of safety performance functions (SPFs), which are the key regression tools in the Highway Safety Manual's RSMP used to predict crash frequency given a set of roadway and traffic factors. Although developing SPFs using traditional regression modeling have been proven to be reliable tools for road safety predictive analytics, some limitations and constraints have been highlighted in the literature, such as the assumption of a probability distribution, selection of a pre-defined functional form, a possible correlation between independent variables, and possible transferability issues. An alternative to traditional regression models as predictive tools is the use of Machine Learning (ML) algorithms. Although ML provides a new modeling technique, it still has made-in assumptions and their performance in collision frequency modeling needs to be studied. This research 1) compares the prediction performance of three well-known ML algorithms, i.e., Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF), to traditional SPFs, 2) conducts sensitivity analysis and compare ML with the functional form of the negative binomial (NB) model as default traditional regression modeling technique, and 3) applies and validates ML algorithms in network screening (hotspot identification), which is the first step in the RSMP. To achieve these objectives, a dataset of urban signalized and unsignalized intersections from two major municipalities in Saskatchewan (Canada) were considered as a case study.

The results showed that the ML prediction accuracies are comparable with that of the NB model. Moreover, the sensitivity analysis proved that ML algorithms predictions are mostly affected by changes in traffic volume, rather than other roadway factors. Lastly, the ML-based measure consistency in identifying hotspots appeared to be comparable to SPF-based measures, e.g., the excess (predicted and expected) average crash frequency. Overall, the results of this research support the use of ML as a predictive tool in network screening, which provides transportation practitioners with an alternative modeling approach to identify collision-prone locations where countermeasures aimed at reducing collision frequency at urban intersections can be installed.

Acknowledgments

I would like to acknowledge and give my warmest thanks to my supervisor, Dr. Emanuele Sacchi, who made this work possible and whose meticulous approach, expertise, and understanding, added immensely to my graduate experience. His guidance and advice carried me through the stages of my master's degree and writing this thesis. I would also like to thank my committee members, Dr. Haithem Soliman, and Dr. Michael Horsch, for their support and their time to read my work and make any necessary corrections.

I would like to express my appreciation to my family for their support and encouragement during my studies. Without their continuous support, it would have been impossible for me to finish my work.

Dedication

This thesis is dedicated to the family I was born into, and the family I have gained along the way.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AADT	Average Annual Daily Traffic
AASHTO	American Association of State Highway and Transportation Officials
AI	Artificial Intelligence
ANN	Artificial Neural Network
BNN	Bayesian Neural Network
BPNN	Back Propagation Neural Network
CAR	Conditional Auto Regression
CART	Classification and Regression Tree
CT	Control Type
DF	Degree of Freedom
DT	Decision Tree
EB	Empirical Bayes
FENB	Fixed Effect Negative Binomial
GDP	Gross Domestic Production
GLR	Generalized Linear Regression
GSCV	Grid Search Cross Validation
HSM	Highway Safety Manual
KNN	K-Nearest Neighbor
MAD	Mean Absolute Deviation
MJAADT	Major Average Annual Daily Traffic
ML	Machine Learning
MNAADT	Minor Average Annual Daily Traffic
MSE	Mean Squared Error
NB	Negative Binomial
Nlegs	Number of Legs
PM	Posterior Mean

PSI	Potential for Safety Improvement
RBF	Radial Basis Function
RBFNN	Radial Basis Function Neural Network
RENB	Random Effect Negative Binomial
RF	Random Forest
RSM	Road Safety Management
RSMP	Road Safety Management Process
RTM	Regression to The Mean
SD	Scaled Deviance
SGI	Saskatchewan Government Insurance
SPF	Safety Performance Function
SVM	Support Vector Machine
WHO	World Health Organization

Chapter 1

Introduction

Motor vehicle collisions are among the leading causes of death in the world, with a mortality rate of approximately 1.35 million people per year (WHO, 2018). This figure on mortality, along with injury and property damage events, indicate the enormous costs to society, both in terms of human lives and future production losses. According to WHO, the cost of motor vehicle collisions in most countries is equivalent to 1 to 3 percent of their gross domestic production (GDP).

Generally, collision events can be either due to roadway factors, vehicle factors, or human factors (or a combination of them). While all these factors can significantly contribute to the number of collisions, road safety research in civil engineering has been focused on studying the roadway and traffic factors that contribute most to collision occurrence on the roadway network.

Collisions can be reduced by implementing educational or enforcement programs or by implementing road safety engineering countermeasures (the focus of transportation engineering). Road safety analysts use manuals and guidelines such as the AASHTO's Highway Safety Manual (HSM) to systematically evaluate the current roadway systems and efficiently offer countermeasures for improvement. Amongst the most common countermeasures are installing rumble strips, improving lighting at an intersection, and adding an acceleration lane for merging major traffic streams.

As suggested by HSM, Road Safety Management (RSM) is a process of deciding on whether a facility (i.e., roadway segment, or intersection) has safety issues, and whether to implement a road safety countermeasure. Overall, RSM assists with making decisions related to the design, operation, and maintenance of roadway networks. As illustrated on figure 1-1, this process starts with network screening for hazardous locations (hotspots), which is to rank and identify the hotspots in the road network. Then, in the diagnosis step, the top hotspots are being assessed for potential safety problems. In this assessment, the collision data at hotspots is to be reviewed and specific patterns for collision types and/or severity are identified. Afterward, road safety engineering countermeasures are selected depending on the type of collisions to be targeted

(identified in diagnosis) and moving on to the next step, the most economically viable countermeasures are prioritized. Finally, road safety countermeasures are implemented, and their actual safety effectiveness is evaluated in the post-treatment period.

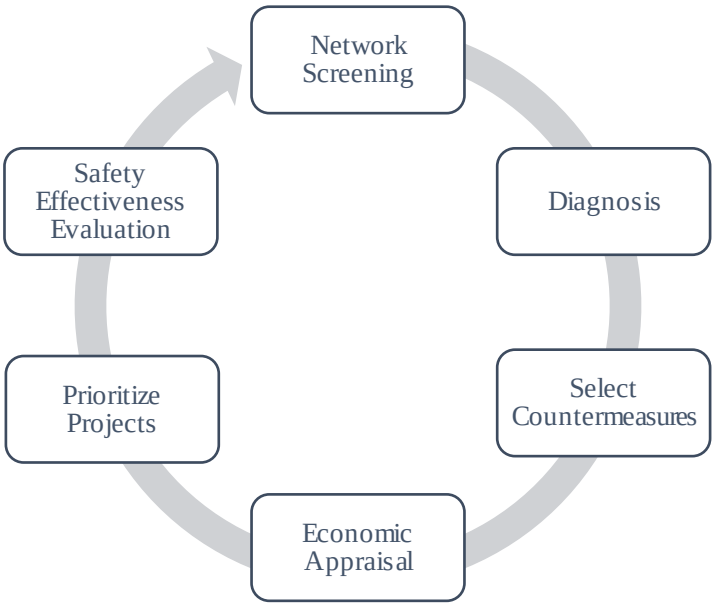


Figure 1-1 Overview of The Road Safety Management (RSM) Process (HSM, 2010)

1.1 Network Screening

According to the HSM, network screening is the process of reviewing the transportation network to identify and rank sites from most likely to least likely to experience a reduction in collision frequency after implementing a countermeasure. Network screening is a crucial step to initiate the RSM process. The HSM suggests a 5-step framework to carry out network screening (HSM, 2010):

1. "Establish Focus: to identify the purpose or intended outcome of the network screening analysis. This decision will influence data needs, the selection of performance measures, and the screening methods which can be applied.
2. Identify Network and Establish Reference Populations: Specify the type of sites or facilities being screened (i.e., road segments, intersections, at-grade rail crossings) and identify groupings of similar sites or facilities.

3. Select Performance Measures: There are a variety of performance measures available to evaluate the potential to reduce collision frequency at a site. In this step, the performance measure is selected as a function of the screening focus and the data and analytical tools available.
4. Select Screening Method: There are a variety of methods (i.e., ranking, sliding window, and peak searching) that can be used after obtaining the results of the performance measures for the identification of the hotspots.
5. Screen and Evaluate Results: The final step in the process is to conduct the screening analysis and evaluate results".

Many specific safety improvement programs can be considered while conducting network screening. For example, an agency desires to perform network screening for selecting roadway projects based on the available budget as part of a capital improvement program. Another example, a specific collision type is concerning an agency and it is desired to implement a system-wide improvement program to reduce that specific collision type. Although the HSM has introduced network screening as the initial task for RSM, when it comes to implementing this process, some requirements have made this task expensive and time-consuming. For instance, network screening requires collision counts as well as the traffic volume records for a period of time, usually more than a year, which is not always available, especially in small jurisdictions.

1.2 Performance Measures

As mentioned before, network screening is a process of identifying those sites where the potential for reducing collisions is maximized. To be able to carry out this task, performance measures are usually estimated for each site in the network to quantitatively measure its safety level. This measure can be the average collision frequency, expected average collision frequency, a critical collision rate, etc. The key criteria to select the performance measures are data availability, regression-to-the-mean bias, and performance threshold, which are reviewed below (HSM, 2010).

1.2.1 Data Availability

Typical data required for network screening consist of collision records, traffic volumes, and roadway information, such as the number of lanes, control type at intersections, etc. While the process of gathering the data needs time and resources, the size of data and inputs limits the selection of performance measures. Small jurisdictions usually struggle to have reliable data available for network screening and, therefore, may be forced to select from limited performance measures due to a lack of sufficient data.

1.2.2 Regression-to-the-Mean Bias

According to Hauer (1997), safety is a property of an entity (i.e., road intersection, etc.) and if the entity remains unchanged, or in other words, if the users, the level of use, the geometrical features, and the environment do not change, then it is expected for the safety to remain unchanged. With this understanding that the number of collisions at a specific site has natural fluctuations over time, the safety of an entity is defined as some "average in the long run". To be more specific, the long-term average number of collisions, also known as collision frequency, represents the safety of an entity. However, long-term collision frequency may be different from what is seen in the shorter term. The randomness of collision occurrence indicates that long-term collision frequency cannot be obtained by looking at its short-term amount. Statistically, when in the short-term, a period of high collision frequency is observed in a site, there is a tendency to experience a period of low collision frequency in the subsequent period. This tendency is known as the regression-to-the-mean (RTM) bias. Failure to account for RTM bias may introduce short-term flaws that will lead to a selection bias. In other words, sites may be wrongly prioritized based on their short-term records for safety improvement, and sites, where improvements could be most cost-effective, can be neglected from the analysis. A list of performance measures and whether they account for RTM bias or not is provided in table A-1 of Appendix A.

1.2.3 Performance Threshold

To better identify hotspots, a performance threshold can be calculated to act as the reference point. After a threshold is identified, all sites with a value greater than the threshold will be identified as

collision-prone locations and will be further analyzed. Selection of the performance measure affects the choice of performance threshold and therefore, performance measures are to be selected based on the requirements of the projects and the performance threshold that is going to be used. For example, performance measures could be the average of the observed collision frequencies in the reference population, predicted collision frequency from the safety performance functions (SPFs), and the expected collision frequency using the empirical Bayes (EB) method. However, not all the performance measures calculate a performance threshold for identifying collision-prone locations. A list of performance measures and whether they calculate a performance threshold is provided in table A-2 of Appendix A.

1.2.3.1 Safety Performance Function (SPF)

As mentioned in the previous section, an SPF is among the options for determining the performance threshold in network screening. SPF is a statistical regression model that reflects the predicted long-term collision frequency at a facility type with specific characteristics. The use of an SPF as a performance threshold has the advantages of accounting for the RTM bias and the non-linearity that exists between the collision frequency and the traffic volume. An example of SPF for a typical urban intersection is shown in equation 1.1(HSM, 2010):

$$N_{Spf} = a_0 \times {V_1}^{a_1} \times {V_2}^{a_2} \times \exp\left(\sum_{j=1}^{n} b_j x_j\right) \quad 1.1$$

Where, N_{Spf} is the predicted average collision frequency, V_1 is the average annual daily traffic on the major approach of the intersection, V_2 is the average-annual-daily traffic on the minor approach of the intersection, and x_j is any of the other variables that are not the traffic volume, such as number of legs, control type at the intersection, etc., and a_i and b_j are the regression parameters. The regression parameters of the SPF are calculated based on the assumption that the collision count follows a negative binomial distribution. The negative binomial is an extension of the Poisson distribution. It has been observed that collision data are usually over dispersed, meaning that the variance typically exceeds the mean. Therefore, over dispersed data cannot be modeled using the Poisson distribution, where the mean and variance of the data are made equal, and this

sets the need for using the negative binomial distribution. The degree of overdispersion in a negative binomial model is represented by a statistical parameter, known as the overdispersion parameter that is estimated along with the coefficients of the regression equation. The larger the value of the overdispersion parameter, the harder it is to fit a proper function and as a result, the less reliable is the SPF predictions for reflecting the long-term collision frequency. Therefore, the empirical Bayes method has been introduced to account for possible issues associated with the interpretations made by using SPFs.

1.2.3.2 Empirical Bayes (EB) Method

The expected collision frequency of a road site (i.e., its true level of safety as long-term average collision frequency) can be estimated by combining the observations and the predictions. The EB method combines the two estimates into a weighted average using a weight factor, which is a function of the SPF overdispersion parameter. Therefore, the estimated collision frequency using the EB method – also known as the expected average collision frequency at a site - is not only dependent on the validity of the observed data but also dependent on the variance of the SPF reflected in the overdispersion parameter. The expected average collision frequency using the EB method is shown in equation 1.2 (HSM, 2010):

$$N_{expected} = w \times N_{spf} + (1 - w) \times N_{observed} \quad 1.2$$

Where, w is the weighted adjustment to be placed on the SPF and is calculated as (HSM, 2010):

$$w = \frac{1}{1 + k \times (\sum_{all\,years\,of\,study} N_{spf})} \quad 1.3$$

Where, k is the overdispersion parameter of the associated SPF.

HSM suggests a number of performance measures based on the availability of data and the method used to calculate the performance threshold. Amongst the performance measures offered by the HSM, three of them were selected in this study. These methods are called excess predicted average collision frequency using SPFs, expected average collision frequency with EB Adjustment, and

Excess expected average collision frequency with EB Adjustment. Summary of the performance measures and their data needs are provided in table A-2 of Appendix A.

The process of network screening is finalized using the screening method (i.e., ranking, sliding window, and peak searching). These methods are dependent on the reference population and its characteristics. For instance, the sliding window is a method applied for screening the roadway segments. In this method, a short length (e.g., 0.1 to 0.5km) of a road segment is established and referred to as a window. The window is conceptually moved each time by a specific distance (0.05 to 0.25km) along the entire stretch of the road segment. The performance measure will be calculated for each position of the window and will be compared/ranked. For screening the intersections, however, ranking the sites based on their performance measure is the common approach. In the final step of the network screening, the results of previous steps are further analyzed for initiating the next steps of the RSM process.

1.3 Machine Learning

An alternative modeling technique to traditional regression models (e.g., SPFs in road safety) are machine learning (ML) models, which can be used to estimate the predicted (long-term) collision frequency. Due to the extensive progress that has been made in the computation technology in the recent decades, the use of ML in place of traditional regression models has become more common than before. In the road safety analysis, scientists have explored and compared the ML models' performance with the intention to increase the prediction accuracy and the generalization abilities of the road safety predictive models. ML is a technique in artificial intelligence that allows automating the analytical model building based on the idea that models can be trained through learning from the data and identifying the patterns. ML has a variety of subsets, such as supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, etc. Each of these subsets is introduced below:

- Supervised Learning – both outputs and inputs are available in this type of learning. The supervised learning algorithms generate functions that map the inputs to desired outputs by looking at several input-output examples in the data. The data used in this supervised learning is called labeled data, where each observation has a label or a value for the output variable.

- Unsupervised Learning – learning the patterns within a set of inputs that lacks the output labels or values.
- Semi-supervised Learning – learning the patterns within a set of inputs that are a combination of both labeled and non-labeled observations.
- Reinforcement Learning – instead of learning to map the inputs to their desired outputs, in this type of ML, the algorithm learns a policy of how to act given the observations. Each action has some impacts, and the algorithm learns by the feedback provided after observing the impact.

In the context of road safety, the task to develop a model for predicting the long-term collision frequency is categorized under supervised learning. A schematic overview of supervised learning is provided in the flowchart shown in figure 1-2.

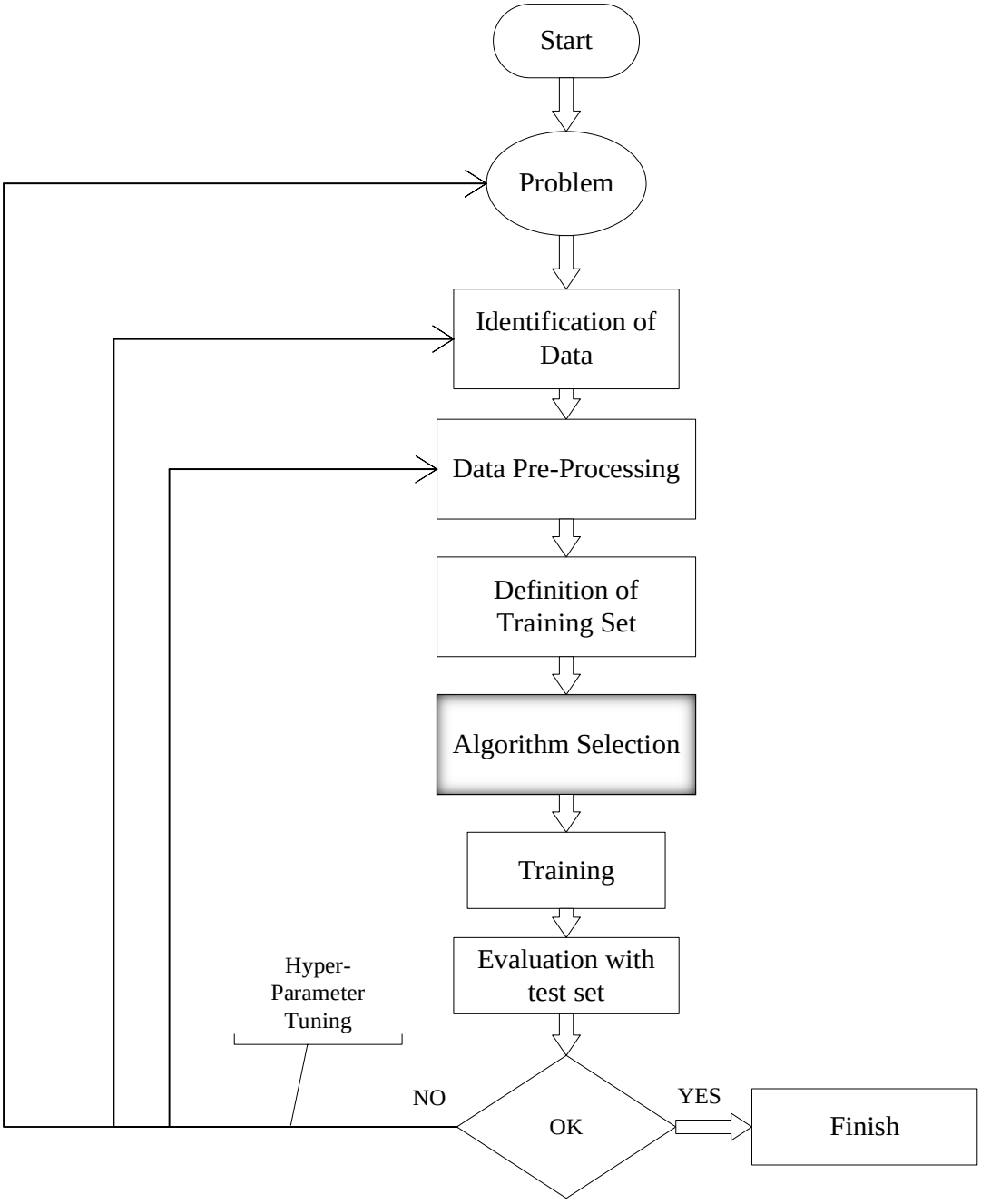


Figure 1-2 Flowchart of the supervised learning process (adopted from Ayodele, 2010)

Supervised learning is used both for classification and regression problems in road safety analysis. For example, it could be used for developing models to classify road collisions by severity (Delen et al., 2006; Chang and Wang, 2006; Oña et al., 2011; Iranitalab and Khattak, 2017). On the other hand, supervised learning can be used to develop regression models for estimating the frequency of road collisions (Chang, 2005; Chang and Chen, 2005; Xie et al., 2007; Li et al., 2008; Çodur

and Tortum, 2015; Zeng et al., 2016). Study of the literature reveals that, the most common algorithms used in road safety are artificial neural network (ANN), decision tree (DT), random forest (RF), classification and regression tree (CART), and support vector machine (SVM).

1.4 Problem Statement

Developing an accurate SPF for a specific road facility can sometimes be a complex task. Common issues are associated with collision frequency modeling, such as treating over-dispersion of collision data (larger variation than expected), under-dispersion (smaller variation than expected), explanatory variables varying over time (i.e., traffic volume), temporal and spatial correlation (i.e., sites in a specific area experiencing a high number of collisions), low sample-mean and small sample size and injury-severity and collision-type correlation (Lord and Mannering, 2010). Although research has demonstrated the reliability of using statistically developed SPFs for collision frequency estimation, a number of model limitations have been highlighted. First, statistical models need a pre-defined functional form, usually with fixed parameters, meaning that one should know a priori whether the collision frequency is related to roadway and traffic factors linearly, exponentially, etc. Second, overfitting is likely to happen. In fact, collision data samples are small as their occurrences are rare, and the developed model would act poorly when transferred to other sites and jurisdictions. Third, tracking the correlation between explanatory variables might be time-consuming, and therefore, sometimes, correlation studies are neglected, which can affect the adequacy of the results. Last, traditional statistical methods are usually weak in assigning the "right" weight to outliers during the fitting process. Alternatively, ML algorithms are used to develop collision frequency predictive models. As explained in 1.3, ML is a different modeling technique comparing to statistical methods and while it might not have the mentioned limitations of the statistical methods, there are other limitations associated with them. It is important to understand that ML is an alternative method, not necessarily a superior one comparing to statistical methods.

Several studies have employed ML algorithms to predict the long-term average collision frequency and compared their performances with the SPFs (Mussone et al., 1999; Abdelwahab and Abdel-Aty, 2002; Chang, 2005; Xie et al., 2007; Lie et al., 2008). Although these studies prove the

adequacy of using ML algorithms in predicting the long-term collision frequency, the application of ML techniques in the practical steps of the RSM process remains unexplored.

1.5 Research Statement

Overall, this research bridges the gap that exists in the literature by evaluating and validating the use of ML algorithms in network screening as part of the RSM process. To achieve the objectives of this research, SVM, DT and RF are selected amongst the most common supervised learning algorithms that are used in road safety analysis. An SPF using negative-binomial (NB) distribution is developed as the default statistical approach to compare the performance of ML with traditional regression models used in the network screening.

The advantage of using ML in place of traditional statistical modeling allows avoiding biases arising from the selection of functional forms to relate roadway and traffic characteristics to collision frequency and the choice of selecting a probability distribution to model collision data. However, ML algorithms are regarded to act as black boxes due to their nature of non-parametric models, that is, they cannot explicitly explain the relations between the explanatory variables and the outcome. As a response to this criticism, a sensitivity analysis is conducted to quantify the effects of each input variable on the output. Overall, the objectives of this research are:

- To compare the prediction accuracy of ML algorithms with the NB approach in predicting the long-term collision frequency
- To conduct a sensitivity analysis and compare/interpret ML with the functional form of NB modeling, and
- To validate the use of ML as a predictive tool in network screening.

The results of this research will support the work of road safety practitioners who seek to implement the RSM process in full, or in part, with ML algorithms.

1.6 Thesis Outline

Chapter 1 introduced the road safety management process. In particular, the network screening task was introduced, the models used to complete a network screening were explained, and the scope of this thesis work was specified.

Chapter 2 provides a description of the fundamental characteristics of collision data and methodological issues, as well as a literature review of different statistical methods used for crash frequency analysis and the use of ML as an alternative to statistical methods.

Chapter 3 provides information about the collected data and the predictive models used in this study. Each predictive method is described and the criteria to compare these methods are identified.

Chapter 4 develops a crash frequency model using both statistical modeling and ML algorithms. the results of the model implementations are provided, sensitivity analysis is conducted and the use of ML in network screening is evaluated.

Finally, Chapter 5 reports the conclusions derived from the study, limitations, and directions for future work in this field.

1.7 Publications

Research conducted for this thesis generated a manuscript entitled "Validation of Machine Learning Algorithms as Predictive Tool in the Road Safety Management Process: The Case of Network Screening" by Tayebikhorami and Sacchi, submitted to the Journal of Transportation Engineering: Part A, Systems – ASCE in September 2021.

Chapter 2

Literature Review

Road safety issues can be systematically addressed by the means of the RSM process (Nodari and Lindau, 2007), which requires the development of SPFs for the facility of interest. SPFs attempt to fit a function to collision data by incorporating the most significant geometric and operational characteristics of the road, and in some cases also environmental conditions (Chang, 2005; Hauer, 2004). To better represent the properties of collision data and the level of severity of collisions, various modeling techniques have been proposed. Traditionally, statistical modeling has been used to predict collision frequency and to classify the severity of them (Persaud and Nguyen, 1998; El-Basyouny and Sayed, 2006; Xie and Zhang, 2008; Malyshkina and Mannering, 2010; Savolainen et al., 2011; Kidando et al., 2019). However, this approach has some limitations and machine learning (ML) represents an attractive alternative for researchers. Overall, collision modeling can provide two types of results. First, an estimate of collision frequency (i.e., the total number of collisions, or the number of collisions at some specific injury level), given the specific characteristics of the infrastructure, and second, understanding the collision contributing factors and how they affect the collision event, and its severity. This study can be categorized as the first type of study with the focus on producing an estimate of the total collision frequency given the characteristics of the facility (in this study, urban intersections). In this chapter, the fundamental characteristics of collision data and modeling issues are presented. Subsequently, a review of the most common statistical approaches, used for modeling collision frequency is provided and the strengths and weaknesses of each approach are discussed. Last, an overview of ML literature in modeling the collision data and the main input variables used by similar studies are provided.

2.1 Fundamental Characteristics of Collision Data and Modeling Issues

Part of road safety research focuses on developing analytic approaches to study the factors that affect the number of collisions occurring at some facilities (i.e., a roadway segment or intersection) over a specific time period (week, month, year, number of years). A key concept required for modeling is that collision data are non-negative integers and requires the application of count-data

regression models or other approaches that can properly account for the integer nature of these data. A review of data and methodological issues in modeling road safety is presented below. The issues, discussed below, have been identified to be the main sources for potential bias in terms of defining statistical models that may lead to erroneous collision-frequency predictions and incorrect inferences.

2.1.1 Over-dispersion

Collision data usually have a variance that exceeds the mean of collision counts, making it problematic to use the most common count-data modeling approach (Poisson regression) since the Poisson distribution requires the mean and the variance to be equal. As a result, using the common Poisson regression approach to describe the collision data can lead to erroneous conclusions (Miaou, 1994; Maher and Summersgill, 1996; Cameron and Trivedi, 1998; Park and Lord, 2007).

2.1.2 Under-dispersion

Although rare, under-dispersion or having a variance lower than the mean of the collision counts can exist, especially when we have a large sample mean. Using the common count-data modeling approaches can be problematic in the presence of under-dispersed data since it may lead to producing incorrect parameter estimates (Oh et al., 2006).

2.1.3 Time effect on explanatory variables

Collison data modeling is usually done over a period of time and all the variations of the explanatory variables over this time are considered to be negligible. While this period could be months or even years, lack of detailed data to describe the within time-period variations in the explanatory variables may result in the loss of potentially important information. For instance, traffic volume is an explanatory variable that has variations in each day and the distribution of traffic volume (by day or even by hour) is likely to be highly influential on the collision occurrence. However, this information is not provided when modeling the collision data for a yearly period, and consequently, important information is lost by using an annual average daily traffic. This can

introduce errors in collision modeling estimation as a result of unobserved heterogeneity (Washington et al., 2010).

2.1.4 Low sample mean and small sample size

Gathering the collision data is an expensive and time-consuming task and therefore, collision data are usually characterized by a low number of samples. In addition, collision occurrence is a rare event and hence the records include a considerable amount of zero counts. Low sample size and mean may cause deficiencies in count-frequency statistical modeling. For instance, estimation of the model parameters using the common maximum likelihood approach is more reliable using a larger number of observations. Also, a low sample mean can cause erroneous interferences due to the choice of collision count distribution that is skewed excessively toward zero (Lord and Mannering, 2010).

2.1.5 Injury-severity and collision-type correlations

Collision data is commonly classified based on either the severity of collisions, such as fatal collision, severe injury, injury, and no-injury, or the collision type, such as right-angle, rear-end, single-vehicle run-off-the-road, etc. It is most common to model the total collision frequency (including all severity levels and collision types) and separately deal with collision severity and type after determining the total frequency of collisions. However, developing collision frequency models in each severity level or for each collision type, which is done by some researchers, may result in potentially serious statistical problems. That is because of the correlation that exists between the various collision severity levels, and also the collision types. For example, having developed separate models for different levels of severity, one can notice that an increase in the collision frequency of a certain severity level could also have some changes in the frequency of other severity levels. This requires the use of more complex models known as multivariate models (Miaou and Song, 2005; Bijleveld, 2005; Song et al., 2006; Ma and Kockelman, 2006; Park and Lord, 2007; Ma et al., 2008; and El-Basyouny and Sayed, 2009a).

2.1.6 Under reporting

Another problem associated with collision data is the fact that less severe collisions are less likely to be reported and vice versa (Aptel et al., 1999). It is difficult to determine the level of underreporting and, therefore, its rate is usually unknown. Researchers have shown that underreporting is usually a function of severity levels as well as the reporting agencies, such as cities, regions, etc. (Hauer and Hakkert, 1988; and James, 1991).

2.1.7 Omitted -variables bias

Overly simplified models that use only one variable, such as traffic volumes, increases the risks of biased estimation for the statistical model parameters, which can lead to incorrect inferences (Washington et al., 2003, 2010).

2.1.8 Endogenous variables

Carson and Mannering (2001) studied the problem of endogeneity associated with explanatory variables that are selected to form a collision frequency model. Endogeneity is when the changes in the dependent variable force some changes in the explanatory variables as well. For instance, it may be seen that signalized intersections are experiencing a higher number of collisions compared to unsignalized intersections. However, it should be understood that the potential for a higher number of collisions is the reason for placing signals at the intersections. In other words, the number of collisions and the presence of signals at intersections are endogenous. If this endogeneity is ignored, the parameter estimates might be biased. In the case of signals, it can be wrongly concluded that signalizing the intersection increases the frequency of collisions.

In traditional least-squares regression models, accounting for endogeneity is relatively straightforward (Washington et al., 2003, 2010). However, for count-data models, the modeling process does not have the flexibility to borrow the traditional endogenous-variable correction techniques (such as instrumental variables). Consequently, accounting for endogenous variables adds considerable complexity to the count-data modeling process (Kim and Washington, 2006).

2.1.9 Functional form

As a critical step in the modeling process of the count-data, a functional form, such as linear, polynomial, exponential, etc. needs to be selected to relate the explanatory variables to the dependent variable. Most count-data models assume a linear relation as their functional form, however, plenty of studies suggest that collision frequency is non-linearly related to the explanatory variables (i.e., traffic volume and road segment length). These non-linear functions can often increase the complexity level and may require involved estimation procedures (Miaou and Lord, 2003; Bonneson and Pratt, 2008).

2.1.10 Fixed Parameters

To avoid further complexity that arises from considering varying parameters over the observations to the predictor, model parameters are usually constrained to fix amounts (Anastasopoulos and Mannering, 2009; El-Basyouny and Sayed, 2009b; Washington et al., 2010). However, this is not the case in collision occurrence and the parameters may vary from one observation to another and, neglecting this, would result in biased estimation of the model parameters.

2.2 Collision Frequency Modeling Using Statistical Methods

A wide variety of modeling techniques have been presented in the literature to efficiently model collision frequency. In the following subsections, a summary of the most common statistical approaches along with their strengths and weaknesses is provided.

2.2.1 Poisson Regression

Collision data are non-negative integers (count data), and they cannot be appropriately modeled using the least-squares regression, which assumes a continuous dependent variable. Therefore, most of the thinking in the literature have chosen the Poisson regression to be the starting point for modeling the collision frequency as it suits the modeling of non-negative integers (Jovanis and Chang, 1986; Joshua and Garber, 1990; Jones et al., 1991; Miaou and Lum, 1993; and Miaou,

1994). In Poisson regression, the probability of having y_i collisions per some time period in the roadway entity (road segments, intersections, etc.) i is defined as:

$$P_{(y_i)} = \frac{EXP\ (-\lambda_i)\lambda_i^{y_i}}{y_i!} \qquad 2-1$$

Where, λ_i is the Poisson parameter for roadway entity i, which equals to the expected number of collisions on entity i, $E(y_i)$. In Poisson regression, it is desired to find the λ_i (expected collisions per time) as a function of the explanatory variables. the most common functional form being $\lambda_i = EXP(\beta X_i)$, where X_i is a vector of explanatory variables and β is a vector of estimable parameters.

Poisson regression is a basic model that is easy to estimate, however, scientists have found it problematic to apply this method and some of its extensions to model the collision frequency. Poisson regression can produce biased results when modeling over- or under-dispersed data. Also, small sample size and low sample mean, which is the case in most of the collision data, can adversely affect the outcome of this regression technique.

2.2.2 Negative-binomial (Poisson-gamma) regression

As an extension to the Poisson regression model, the negative-binomial model is used to overcome the issue of over-dispersed data. While in the Poisson regression it is assumed that the mean and the variance of the sample data are the same, in the negative binomial this assumption is relaxed. The model results in a closed-form equation and the mathematics to manipulate the relationship between the mean and the variance structures are relatively simple. The difference between the two methods is embodied in the definition of the estimation parameter, which in negative binomial is defined as $y_i = x_i\beta + \varepsilon_i$ for each observation i. The term ε_i is known as the error term, which follows a gamma distribution with mean 1 and variance φ. This variation in the model allows the variance to differ from the mean and the most common function for defining the variance that is used in the highway safety analysis is defined as:

$$VAR(y_i) = E(y_i)[\,1 + \varphi E(y_i)] = E(y_i) + \varphi E(y_i)^{\,2} \qquad 2.2$$

The value of φ, being known as the over-dispersion parameter, is playing an important role in this modeling technique. It can be observed that the Poisson model is a special case of the negative

binomial distribution when φ approaches zero, which means that the selection between these two models is dependent upon the value of φ. The details of the parameters used in this section are explained in section 3.2.

The negative-binomial regression model appropriately accounts for over-dispersion that exists in most of the collision datasets. However, it does not account for under-dispersion and can be potentially biased in case of low sample size and low sample mean. While the use of this model is usually considered with a fixed value for the over-dispersion or its inverse, studies have shown that variance function can also be dependent on the value of the explanatory variables (Miaou and Lord, 2003; Cafiso et al., 2010). Although more sophisticated modeling techniques are proposed in the literature to develop collision frequency models, negative binomial is the most common approach that is being used within the scope of the RSM process to develop SPFs. Since negative-binomial is going to be the default tool for collision frequency modeling, explaining more sophisticated techniques is considered to be out of the scope of this thesis work. Therefore, in Table 2-1, a summary of the common approaches for modeling the collision frequency, as well as relative studies are provided.

Table 2-1 Representative summary of the previous studies for statistical modeling of the collision data (obtained from Lord and Mannering, 2010)

Model Type	Related Studies
Poisson	Jovanis and Chang (1986), Joshua and Garber (1990), Jones et al. (1991), Miaou and Lum (1993), and Miaou (1994)
Negative binomial/Poisson-gamma	Hauer et al. (1988), Bonneson and McCoy (1993), Miaou (1994), Persaud (1994), Miaou and Lord (2003), Amoros et al. (2003), Hirst et al. (2004), Abbas (2004), Lord et al. (2005a), El-Basyouny and Sayed (2006), Lord (2006), and Kim and Washington (2006)
Poisson-lognormal	Lord and Miranda-Moreno (2008), and Aguero-Valverde and Jovanis (2008)
Zero-inflated Poisson and negative binomial	Shankar et al. (1997), Carson and Mannering (2001), Lee and Mannering (2002), Qin et al., (2004), Lord et al. (2007), and Malyshkina and Mannering (2010)
Conway–Maxwell–Poisson	Lord et al. (2008), and Lord et al. (2010)
Gamma	Oh et al. (2006), and Daniels et al. (2010)
Generalized estimating equation	Lord and Persaud (2000), Halekoh et al. (2006), Wang and Abdel-Aty (2006), and Lord and Mahlawat (2009)
Generalized additive	Xie and Zhang (2008), and Li et al. (2009)
Random-effects[1]	Johansson (1996), Flahaut et al. (2003), MacNab (2004), Noland and Quddus (2004), Miaou et al. (2005), Aguero-Valverde and Jovanis (2009), Wang et al. (2009) and Guo et al. (2010)
Negative multinomial	Hauer (2004), and Caliendo et al. (2007)

[1]includes the spatial statistical models

Despite the progress that has been made in the statistical modeling of collision data, the limitations of these models are acknowledged in the literature since each method has its assumptions and pre-defined functions (Zeng et al., 2016a). Therefore, efforts have been made to explore the use of machine learning as an alternative modeling technique in which, instead of assuming a pre-defined relation between the risk factors and the collision frequency, the estimates are made after learning from the observed data.

2.3 Collision Data Modeling Using Machine Learning Algorithms

Machine learning is a branch of artificial intelligence (AI) concerned with the design and analysis of algorithms that enable computers to learn from data, make predictions, or act without direct human supervision. In the road safety analysis, whether a classification study (i.e., predicting the severity of collisions) or a regression study (i.e., predicting the frequency of collisions) is conducted, collision data can be used within various modeling techniques, however, the majority of models used in the literature are k-nearest neighbors (KNN), decision trees (DT), support vector machines (SVM), and artificial neural networks (ANN).

KNN is a simple and pioneering technique in ML used mostly for classification problems. It identifies the K-points in the training data that come closest to a given observation and tests the value of the calculated distance for each category. Therefore, the class of the observation of interest should include the majority of k closest observations (Devroye et al., 1994). The KNN algorithm is a non-parametric algorithm which means that it makes no assumptions about the underlying data. It is also called a lazy learning algorithm because it does not learn from the training set, stores the data set at the time of classification, and does not take any action on the data. In road safety, KNN has mostly been used for classifying the severity of collisions.

DT is very convenient for classification tasks. However, it can model regression problems as well. To build the tree, input data (i.e., road and environmental features) and the output data (i.e., collision frequency) are fed to the model in a training set. The decision tree begins by creating the root node and continues with decision nodes that divide an input feature and form ramifications, and "leaves" that contain the classification or regression information. Each node represents the test of a feature and the criterion for ramification is the feature's utility for classification. Thus, the selected feature, one of the tree nodes, generates the greatest information gain (entropy); i.e., it provides the best quality for classification. In decision trees, the induction algorithms seek the features that better generate the examples, generating sub-trees.

SVM model is built using statistical learning theory (Scholkopf and Smola, 2002). The model attempts to learn a hyper plane, known as a decision surface, which is used to maximize the margin of separation between observations. The learned hyperplane is used to discriminate the test set into two groups, namely, positive samples and negative samples. Although it was originally developed

for classification tasks, it has been extended to solve regression problems and problems with non-linearly separable data (Burges, 1998; Smola and Scholkopf, 2004; Trafalis and Gilbert, 2006; Üstün et al., 2005).

ANN is a highly complex, non-linear[1], parallel[2] processor with a natural propensity for storing experimental knowledge and making it available afterward (Haykin, 2009). A multi-layer perceptron ANN is typically made up of three types of layers: an input layer, an output layer, and one or more hidden layers. A perceptron is a neural network unit that does certain computations to detect features or intelligence in the input layer. The input layer obtains the values of the input features, i.e., the road features. The hidden layer, made up of m neurons, adds up the weights of the input values of the various input features and calculates the complex association patterns. A single hidden layer is usually enough for road safety analysis applications, but the number of neurons in it is generally the object of experimentation (Chang, 2005; Villiers and Barnard, 1993). For the output layer, the values of the various hidden neurons are summed and the network's output values are presented. Feedforward is the most common type of network architecture, in which the propagation of signals is always from the previous layers to the posterior ones. In terms of training, the back propagation algorithm is the most used to minimize errors by adjusting the weights of the network (Haykin, 2009). In this case, the cost function is in the direction in which the function's variation rate is minimal and it guarantees that the network surface trends in the direction that leads to the greatest error reduction. Lastly, the main activation function used is related to the representational capacity of the neural network and it introduces a non-linear component.

Table 2-2 provides a summary of the most prominent studies in road safety that have employed the ML algorithms. In this table, facility type, percentage of data used for training and test set, and types of tasks (classification or regression) are identified and studies are sorted in chronological order.

[1] Non-linear processor is a processor whose output is not a linear function of the inputs
[2] Parallel processors can run two or more processing units for handling separate parts of the overall task, simultaneously.

Table 2-2 Representative summary of the previous studies that used ML to model the collision data (obtained from Silva et al., 2020)

ML Algorithm	Facility Type	Dependent Variable	Training/test percentage	Task	Reference
ANN, DT	urban streets	Injury; property damage only	60/40	Classification	Sohn and Lee (2003)
ANN	urban streets and highway	No injury; possible injury; evident injury; incapacitating/fatal injury	51.9/48.1	Classification	Abdel-Aty and Abdelwahab (2004)
ANN	multi-lane highway segments	Number of collisions per segment per year	75/25	Regression	Chang (2005)
ANN	urban streets and highway	No injury; possible injury; non-incapacitating injury; incapacitating injury; fatal injury	N/M[1]	Classification	Delen et al. (2006)
ANN, BNN[2]	Two-way two-lane highway segments	Number of collisions per segment	60/40, 70/30, 80/20	Regression	Xie et al. (2007)
SVM	Two-way two-lane highway segments	Number of collisions per segment	60/40, 70/30, 80/20	Regression	Li et al. (2008)
ANN	highway	Injury; property damage only	80/20	Classification	Alikhani et al. (2013)
ANN	highway	No injury/property damage only; possible injury; non-incapacitating injury; incapacitating/fatal injury	80/20	Classification	Zeng and Huang (2014)

ANN	multi-lane highway	Number of collisions per segment	70/30	Regression	Çodur and Tortum (2015)
ANN	highway	Number of collisions per segment per year	N/M	Regression	Zeng et al. (2016a)
ANN	highway	Number of collisions with slight injuries per segment per year; Number of collisions with severe or fatal injuries per segment per year	N/M	Regression by class[3]	Zeng et al. (2016b)
KNN, SVM, RF[4]	local, interstate, and highway	Property damage only; possible injury; severe injury; disabling/fatal injury	70/30	Classification	Iranitalab and Khattak (2017)
KNN, DT, RF, SVM	freeway	No injury; possible injury; non-capacitating injury; incapacitating injury; fatal injury	75/25	Classification	Zhang et al. (2018)
DT, RF, KNN	urban streets	Damage injury; injured; hospitalized; fatal injury	10-fold cross validation[5]	Classification	Wahab and Jiang (2019)
ANN	highway	Property damage only; complaint of pain; visible injury; severe injury; fatal injury	70/30	Classification	Amiri et al. (2020)

[1]Not Mentioned
[2]Bayesian Neural Network
[3]Regression has been used in each class of collision severity to obtain the frequency of collisions by each class
[4]Random Forest model, an extension to DT
[5]cross validation is a technique that is used to avoid data selection bias

As shown in Table 2-2, several studies have investigated the use of ML algorithms as a regression tool, in place of traditional statistical modeling, for collision frequency prediction. Chang (2005) developed an ANN model for the prediction and classification of collisions on road segments and compared the ANN collision frequency predictions to the ones of a negative binomial (NB) regression. The research showed that ANN, with a prediction accuracy of 64% for the training set and 61.4% for the test set, provides more sufficient prediction accuracy than NB, which shows 583% and 60.8%, respectively. In other words, ANN predicts more accurate numbers compared to the actual observations of the collision frequency on the road segments. Moreover, a sensitivity analysis was conducted where it was revealed that the sensitivity of ANN and the parameters of NB models are consistent, which each other. For example, both ANN and NB models show upward\downward changes when there was a change in an explanatory variable. In 2007, Xie et al. assessed the application of Bayesian neural network (BNN) to predict collision frequencies. The results of BNN were compared to predictions using back-propagation neural network (BPNN) and NB models using different scenarios of sizes for training and test set. It was concluded that overall, BNN and BPNN outperform NB both in prediction accuracy using the training data and the test set. Moreover, it is proven that BNN has higher prediction accuracy, and it acts better in terms of generalization ability meaning that it can more sufficiently deal with the overfitting problem while keeping the ability of non-linear estimation. More recently, Huang et al. (2016) developed an optimized radial-basis-function neural network (RBFNN) model to predict the collision frequency and compared the results with the NB and BPNN models. Their study showed that while RBFNN has better prediction accuracy using training and test set compared to NB and BPNN, and it significantly reduces the overfitting problem. However, significant drawbacks of fitting a neural network to a collision prediction problem are that neural networks act as a black box, and it is often time-consuming to develop them. Moreover, studies suggest that neural networks perform better when larger datasets are provided, while collision frequency datasets are usually small due to the time and effort needed for gathering collision records (Lie et al., 2008). As a result, another stream of research has explored the performance of other ML algorithms such as SVM.

In theory, SVMs are less likely to experience overfitting, and they can generalize better than ANN because SVMs are based on structural risk minimization (Suykens et al., 2002), whereas ANN is based on empirical risk minimization (Zhang and Xie, 2007). Lie et al. (2008) utilized SVM and

replicated the study done by Xie et al. (2007), where they have evaluated the application of SVM for predicting motor vehicle collisions by comparing the model with NB and BPNN models and revealed that SVM outperforms the NB model regarding prediction accuracy. In addition, SVM has less overfitting problem and provides similar, if not better, prediction performance comparing to BPNN models. A broader perspective has been adopted by Dong et al. (2015) who investigated the efficiency of SVM in collision prediction at the level of traffic analysis zones and showed that the algorithm could be considered as an alternative in regional safety modeling. They proved that SVM outperforms the Bayesian spatial model with conditional autoregressive prior (i.e., CAR), in both data fitting and predictive performance. In another study done by Singh et al. (2018), they employed SVM to develop a collision frequency prediction model for non-urban highway sections. They compared the performance of SVM with fixed-effect and random-effect negative binomial (FENB/RENB) models. The results indicate that SVM has shown better values for the selected metrics, that are correlation coefficient and root mean square error. The observed capabilities of SVM in these research studies reinforce the idea of exploring the application of ML algorithms in road safety practices. On the other hand, Olutayo and Eludire (2014) developed a collision frequency prediction model for Nigeria's arterial roads using ANN and DT. Although the purpose of this study was to determine the most contributing factors that cause collisions on the roads, they showed that decision tree performs better than a neural network with lower error, or in other words, greater accuracy, which is a higher number of correctly classified instances.

2.4 Input Variables

The choice of the input variables is an essential step of the modeling process, and which variable to choose depends on the purpose of the study as well as the data availability. Whether to include an input variable or not, it is a choice of the analyst, based on the expected degree of association with the dependent variable of interest (collision frequency). Therefore, selecting the modeling approach depends on previous judgment and knowledge about the data, prior experience in modeling, and data availability (Hauer, 2015). Although collisions are rare events, their occurrence involves the interactions of various contributing factors. To be able to rigorously select the input variables, an overview of the input variables selected for collision data modeling is required, whether the model is developed using traditional statistical modeling or machine learning.

Several studies have investigated factors that are expected to influence collisions, such as the roadway geometrical and operational variables, environmental variables, vehicle conditions, and human factors (Abdel-Aty and Radwan, 2000; Carson and Mannering, 2001; Elvik et al., 2009; Miaou and Lum, 1993; Rolison et al., 2018; Wang et al., 2013). To evaluate the variables, silva et al. (2020) grouped the variables into four major classes: human factors, road-environmental factors, vehicle-related factors, and collision characterization. They showed that all studies incorporated road-environmental factors into their modeling even though, in some cases, such as Sohn and Lee (2003), only one variable was considered. In addition to this study, only Delen et al. (2006) and Kwon et al. (2015) did not have most of the variables in their studies related to environmental conditions. The latter used vehicle-related factors which were entirely absent from the models of Alikhani et al. (2013), Das and Abdel-Aty (2010), Iranitalab and Khattak (2017), Kashani and Mohaymany (2011), Oña et al. (2011), Oña et al. (2013b), Zhang et al. (2018). Overall, human factors and collision characterization were used to the same extent in developing the models reported in the literature.

In collision frequency modeling, Silva et al. (2020) showed that the most common road-environmental factors to be used as input variables are traffic volume, segment length, horizontal alignment, shoulder width, and roadway segment. They have also shown that sex, as a human factor, is used in several instances. In addition, some variables that characterize the collisions, such as year, season, and the number of vehicles involved in collisions are also used in some studies. Table 2-3 represents studies with their most contributing factors for collision frequency modeling.

Table 2-3 Main contributing factors in the literature for collision frequency modeling

Study	Main contributing factors
Chang (2005)	Segment in military area; existence of intersections; percentage of heavy vehicles; number of lanes; traffic volume
Xie et al. (2007)	Segment length; traffic volume; lane width
Li et al. (2008)	Traffic volume; shoulder width
Çodur and Tortum (2015)	Vertical curvature; traffic volume; horizontal curvature; segment length
Zeng et al. (2016a)	Traffic volume; posted speed limit; annual precipitation; segment length; median barrier; bus stop
Zeng et al. (2016b)	Traffic volume; segment length; posted speed limit; bus stop; annual precipitation

Overall, two important conclusions emerge from the review of the literature. First, conventional statistical modeling, in most cases, is showing less prediction accuracy than ML algorithms; and second, the best ML algorithm can be selected as a trade-off between accuracy and complexity. Although studies have explored the application of ML in developing collision frequency models, the literature lacks the exploration of ML algorithms in the actual road safety procedures (i.e., RSMP). While the ability of the ML algorithm in collision frequency modeling is proved by many studies, this thesis work aims at evaluating the use of the ML algorithm in the RSMP, particularly in network screening. Therefore, SVM, DT, and RF are selected among the most common ML algorithms that are used previously in road safety studies. First, the prediction accuracies of these models are compared with the default NB statistical model; second, a sensitivity analysis is done to observe the effects of each variable on the output variable and to compare it to traditional statistical modeling; and lastly, ML models are evaluated while being employed within the RSMP.

Chapter 3

Methodology

3.1 Data Collection and Processing

The data used in this research consists 343 urban intersections of two major municipalities in Saskatchewan (i.e., Regina and Saskatoon). The time period evaluated ran from 2013 to 2018; intersections did not undergo any significant improvement or change during this time frame or in other words, no road safety engineering countermeasure implemented. However, two time periods of 3 years were considered, i.e., 2013-2015 and 2016-2018, respectively. The reason for dividing the dataset into two subsequent time period was mainly the requirement of the third objective of the study that is to compare the consistency of ML and statistical regression modeling in identifying the hotspots. In other words, if intersections of the study did not undergo any significant road safety improvement, it is expected to have consistent hotspots being identified by ML as well as the statistical regression technique. More details of the consistency check are provided in section 4.4.

Saskatchewan Government Insurance (SGI) has provided the collision data (total number of collisions for each year) and Saskatoon and Regina municipalities have provided the recorded traffic volumes in the form of average annual daily traffic (AADT) for the intersections' major and minor approaches. Also, the number of legs in the intersection and the control type (signalized or unsignalized intersection) were observed using Google Maps and Street View, respectively. Table 3-1 illustrates the summary statistics of data for this study.

Table 3-1 Summary Statistics of Data

Description	Type	Min	Max	Mean	St. Dev.
V1 (major AADT 2013-2015 - Veh/day)	continuous	2,226	69,400	17,801	12,183
V2 (minor AADT 2013-2015 - Veh/day)		708	35,967	7,931	5,806
V1 (major AADT 2016-2018 - Veh/day)		2,050	67,300	17,432	11,162
V2 (minor AADT 2016-2018 - Veh/day)		701	40,450	8,066	5,734
Y1 (total crashes/3 years 2013-2015)	discrete	0	131	22.6	23.5
Y2 (total crashes/3 years 2016-2018)		0	124	22.4	23.5
N (number of legs) *	dummy	3	4	3.75	0.44
C (control type) **		0	1	0.65	0.48

* 3 legs=0, 4 legs=1.
**unsignalized= 0, signalized=1

3.1.1 Data Normalization

Before model fitting, data normalization is an important task to avoid biases that will be created from the different measuring scales of the input variables. In ML, non-tree algorithms, such as SVM, are more prone to scaling problems. Therefore, in this study, MinMax scaling is used to normalize the input variables. In this method, all the values will be transformed to the range of [0,1] meaning that the minimum and maximum value of a feature/variable is going to be 0 and 1, respectively. The mathematical overview of this method is provided in equation 3.1.

$$x_{scaled} = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad 3.1$$

Where, x_i is the actual observation of the feature at the i[th] site. It is worth mentioning that using MinMax scaler in presence of outliers will create bias since the scaling is highly dependent on the maximum and the minimum values. Therefore, it is a crucial step to monitor and avoid reporting mistakes before normalization. For instance, if the value of a data point is relatively large in comparison with the other sites in the jurisdiction, it might be a reporting mistake and it might be excluded from the study. Excluding a site from the dataset due to suspicious value for one or more of the variables requires engineering judgment.

3.1.2 Missing Data

A challenging task when modeling collision data is to find the best way to deal with missing data. To avoid the issues that can arise from the short-term perceptions of road safety (as explained in section 1.2.2), data is usually collected for a period of time longer than one or two years and, therefore, it is very likely to have no records at some specific locations for the explanatory variables (e.g., traffic volume), especially in smaller jurisdictions, where there is less resources for completing the task. In this study, data is collected from 2013 to 2018 and as expected, some locations at some specific years are missing the traffic volume records. There are certain scenarios in the used dataset to deal with sites with missing data. Scenario A: if a site had no recorded traffic volume during the whole period of study, that site was completely omitted from the dataset. Scenario B: If a site has only one year with recorded traffic volume, the missing values for the other years are calculated using equation 3.2. In this equation, a growth factor is calculated for each year based on the overall average growth in the traffic volume, as per equation 3.3.

$$AADT_{i\pm1,j} = AADT_{i,j} \pm G_{i\pm1,j} \times AADT_{i,j} \quad 3.2$$

$$G_i = \frac{\sum_n AADT_i}{\sum_n AADT_{i-1}} \quad 3.3$$

Where, $AADT_{i,j}$ is the average annual daily traffic at site j for the year i, G_i is the average growth factor for the i year, and n is the total number of sites that has recorded AADTs for i and $i+1$ year.

To clarify, let imagine that the major AADT (also known as V_1) has record available only in 2016. To calculate all the missing AADTs of 2013 to 2018, growth factors are being used in a backward method for years 2013, 2014 and 2015 and in a forward method for years 2017 and 2018. The calculated growth factors are shown in table 3-2. Scenario C: if a site has more than one year with recorded traffic volume, a similar approach as per scenario B is done separately based on each recorded value and the average of calculated traffic volume is considered.

Table 3-2 Calculated average growth factors for the years of the study

Year	2018	2017	2016	2015	2014	2013
Growth Factor	1.90	1.55	9.55	6.8	-0.78	-1.89

3.2 Predictive Analytics

To fulfill the objectives of this study, three ML algorithms, i.e., SVM, DT, and RF are developed and compared with the NB model. In this section, these analytics are described in detail.

3.2.1 Negative Binomial Generalized Linear Regression

Negative Binomial (NB) Generalized Linear Regression (GLR) has been widely employed in the literature to model collision data and develop SPFs (Hauer, 1997, Sawalha and Sayed, 2001, 2006, Kim et al., 2007, El-Basyouny and Sayed, 2009a, Geedipally et al., 2010). GLR is introduced as opposed to the traditional linear modeling (least square approach) due to the following important shortcomings of linear regression:

- Assuming that data is always normally distributed may not be always reasonable. For instance, assuming a continuous normal distribution for count data (i.e., collision data) is not appropriate.
- Assuming that the prediction error have a constant variance for all observations (known as homoscedasticity) is not always the case. In other words, it is not unusual for data to experience an increase in the variance of residuals while the mean increases (known as heteroskedasticity).

A traditional linear model is in the form of $y_i = x_i\beta + \varepsilon_i$, where y_i is the observation (i.e., number of collisions) for the i^{th} site, x_i is the corresponding feature (i.e., road and traffic factors), β is the coefficient of the model that is to be estimated using the least square approach, and ε_i are the independent, normal random variables with a mean equal to zero and constant variance. In this model, the expected value of y_i, denoted as μ_i, is $\mu_i = x_i\beta$. In GLR, the traditional model is extended, and therefore, it applies to a wider range of data and problems. In GLR, the linear component is defined similarly to the traditional linear modeling, $\mu_i = x_i\beta$, however, a monotonic differentiable link function g describes how the expected value of μ_i is related to the linear predictor μ_i (Haur, 1997):

$$g(\mu_i) = x_i\beta \qquad 3-4$$

The response variables y_i, are independent for i = 1, 2, . . . and have a probability distribution from an exponential family. This implies that the variance of the response depends on the mean μ through a variance function V(Haur, 1997):

$$VAR(y_i) = \frac{\varphi\, V(\mu_i)}{\omega_i} \qquad 3-5$$

Where φ is a constant, known as dispersion parameter, and ω_i is a known weight for each observation. φ is either known (for example, for the binomial or Poisson distribution, $\varphi = 1$) or must be estimated.

The most common approach in the road safety analysis for GLR is the NB approach. In details, in a population of roadway sites, let Y denote the random variable describing the collision experience in different years for a site, as Poisson distributed with mean m. It is further assumed that m vary between different sites and that the exact value for a particular site is unknown and is regarded as gamma distributed. It follows that the distribution of Y in this population is NB with mean and variance (Hinde and Demetrio, 1998; Hauer, 1997):

$$E(Y) = E(m); \qquad Var\,(Y) = E(m) + \varphi \times E(m)^2 \qquad 3-6$$

where φ is the dispersion parameter of the NB distribution. The expected value of m can be modeled as an SPF with the following baseline form for intersections, for instance:

$$E(m) = a_0 \times {V_1}^{a_1} \times {V_2}^{a_2} \times e^{\sum_{j=1}^{m} b_j x_j} \qquad 3-7$$

where a_0, a_1, a_2 and b_j are model parameters, V_1 and V_2 are the annual average daily traffic (AADT) on the major and minor approaches, respectively, and x_j represents additional explanatory variables to the model (e.g., roadway and traffic features).

3.2.2 Support Vector Machine (SVM)

SVM, proposed by (Vapnik, 1995) is a supervised and non-parametric ML algorithm used for classification and regression problems. Originally, SVM has been introduced within the context of statistical learning theory and structural risk minimization (Chen et al, 2009). Given a training dataset, SVM aims at learning the so-called separative line and its boundaries to predict the outcome variable based on distances from them. Figure 3-1 describes the separative line and the boundaries.

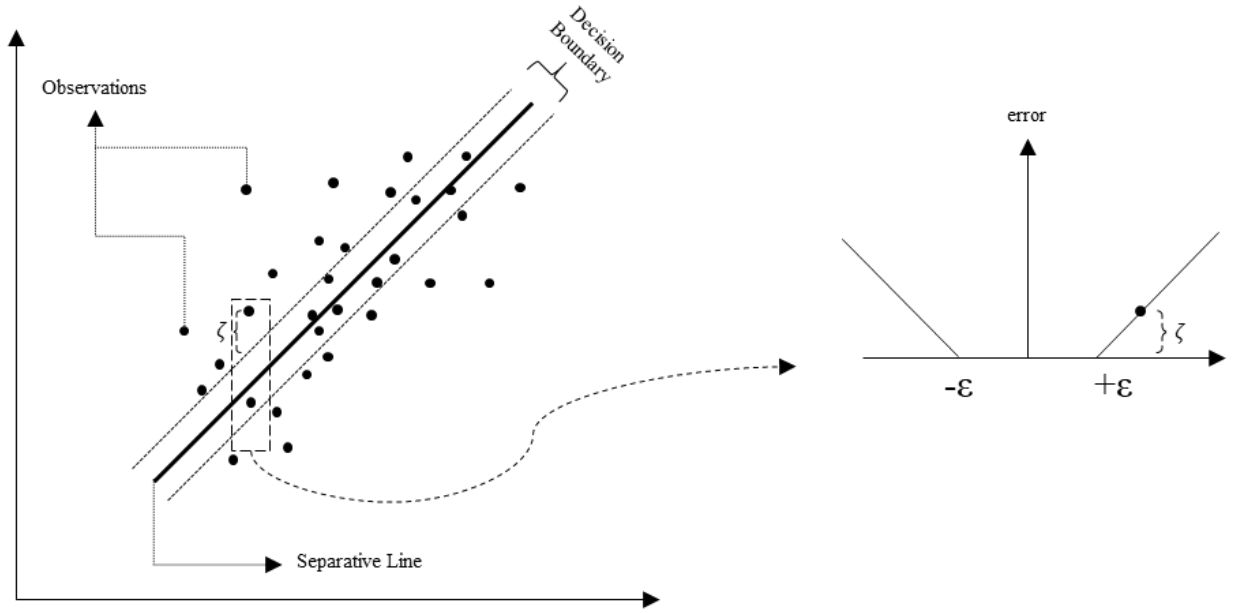


Figure 3-1 ε-insensitive linear SVM regression (adopted from Schölkopf et al., 2000)

The SVM algorithm can reveal hidden patterns in datasets that have otherwise been hidden or distorted because of non-linear phenomena such as noise or sparsity. In classification tasks, SVM requires two components: a dataset, and an algorithm for finding the best separative lane that can separate classes. As an example, in road safety applications, classes can be severity levels of collisions. The algorithm uses the training dataset to determine what vectors (observations) will best separate the target classes from the rest of the data. Then, for each data point belonging to one class, a hyperplane is created that approximately lines up with that vector. These vectors are called support vectors that are hallmarks for building the classification after training the model. By doing so, the algorithm saves computational time and minimizes data storage through these support vectors (Marsland, 2009). It is designed so that there is no confusion between different classes and so it maximizes our separation between them. Predictions are made by measuring the distance between the new points and the support vectors that were learned in training the data.

The central concept of the SVM classifier lies in the obtaining of the largest marginal space at the same time controlling the number of data points from appearing inside the margins (soft margin) or preventing their occurrence by any means inside the margins (hard margin). To maximize the

margin, the classifier needs to find the right amount of weight w and bias b in a line defined as $y = w * x + b$. The margin and weight of the vector have an indirect relationship, where a more significant margin occurs when we take a smaller amount of weight (Marsland, 2009). In using the hard margin to train the dataset, the effort is to make a function that would have a value of more than 1 for a positive x data point and less than -1 for a negative x data point. These constraints then pass through an optimization step to classify the classes correctly that in turn brings us to express the constraint in an equation for all data points as $y=\frac{1}{2}w^T\ w$ subject to $t_i(w^T\ X_i +\ b) \geq 1$, where t_i is the target variable for the ith data point.

In a soft margin classifier, where we allow an error to some extent, we use a slack variable $\zeta_i \geq 0$ quantifying the violation of margin produced by a single data point x in a dataset (Géron, 2017). In the soft margin, a balance between two options has to be found, i.e., minimize the slack variable so that the marginal violation becomes small and minimize $\frac{1}{2}w^T\ w$ to expand the margin. These trade-offs in the classifier are facilitated by hyperparameter C in which its higher value favors a small margin over higher errors. In contrast to the hard margin, the function to be minimized is given as:

$$\frac{1}{2}w^T\ w + C\sum_{i=1}^{m}\zeta^i, \qquad subject\ to \qquad t_i(w^T\ X_i +\ b) \geq 1 - \zeta^i \tag{3.8}$$

On the other hand, in the regression task in which this study used, the goal of SVM regression takes a contrary path that is the effort to fit many data points, known as support vectors, as much as possible inside the boundaries and minimize the number of data points from going out of the boundaries. The distance between the boundary line and separative line largely depends on the level of parameter ε (Suykens et al., 2002), which needs to be specified a priori. Predictions are made by measuring the distance between the new points and the support vectors that were learned in training the data. SVM extends its applicability to accommodate the non-linear task of identifying patterns and therefore, kernel function comes to play a crucial role in such applications by allowing various adjustments according to the distribution of the data (Müller and Guido, 2016). A schematic view of the kernel function used is shown on figure 3-2. There are 4 commonly used

kernel functions, and choosing the right kernel is one of the tasks in the SVM. These kernels are the Gaussian, Polynomial, Sigmoid, and Linear kernels (Géron, 2017). The impressive capability of kernel function is its performance of computation in the former low dimensional space data points despite having the data appeared on the high dimensional spaces. In this study, ε-intensive SVM is used to train the model. A description of this model is provided. For more details refer to Schölkopf et al., (2000).

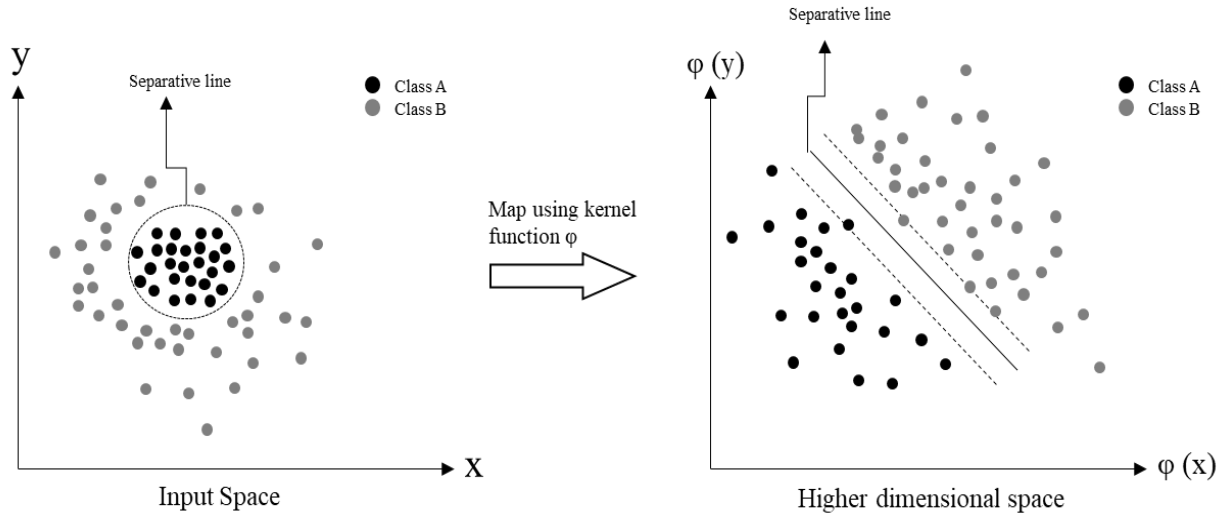


Figure 3-2 A schematic view of kernel function φ to transfer the input space into a higher dimensional space (based on Li et al., 2012)

Assume the training input is defined as vectors $x(i) \in R^1$ for i=1, ..., N, which are independent and identically distributed data with sample size N. The training output is defined as $y(i) \in R^1$ for $i = 1, \ldots, N$. The ε-SVM maps $x(i)$ into a feature space $R^h (h > 1)$ with the higher dimension using a transformer function $X_{x(i)}$ to linearize the nonlinear relationship between $x(i)$ and $y(i)$. The estimation function of $y(i)$ is:

$$y(i) = \left(w^T X_{x(i)} + b\right) \qquad 3.9$$

Where, $w \in R^h$ and $b \in R^1$ are coefficients denoting the weights and biases in the higher dimension and lower dimension spaces, respectively, used for transforming purposes. Schölkopf et al. (2000) showed that the coefficients are derived by solving the following optimization problem:

$$\text{minimize } \left(\frac{1}{2}w^T w + C\sum_{i=1}^{m}\zeta^i\right), \qquad subject\ to \qquad \left(w^T X_{x(i)} + b\right) - y_i \leq \varepsilon + \zeta^i \tag{3.10}$$

Where, $\frac{1}{2}w^T w$ is a term representing the model complexity, y_i is the observation for the i^{th} variable, ζ^i are slack variables measuring the prediction errors, and C represents a penalty variable for large and small margin violations.

One objective of this study is to predict collision frequency from the corresponding transportation predictors (roadway and traffic factors). In this case, the training data inputs (vectors) {(x1, y1),... (xn, yn)} ⊂ X × R are the roadway and traffic factors, and the output y is the number of collisions. The main task of the regression is to find the least-squares error function and convert it into an ε-insensitive error function (shown in equation 3.8) (Géron, 2017). The model is called ε-insensitive since the predictions don't rely on the number of instances found between the two marginal lines (Géron, 2017).

3.2.3 Decision Tree

Decision Tree (DT) is a predictive ML algorithm that applies to both classification and regression tasks. This algorithm is popular as its algorithm is easy to be interpreted. In other words, the simple process of learning in this algorithm along with the comprehensive, yet easy-to-follow procedure has made this algorithm popular over the recent decades. Several studies in road safety analysis have used and recommended different types of DT for modeling the collision data (Sohn and Lee,

2003; Chang and Chen, 2005; Kashani and Mohaymani, 2011; Ona et al., 2013; and Zhang et al., 2018). Some advantages and disadvantages of the DT algorithm are reported below:

Advantages:

- DT model is simply understood and can be visualized.
- Compared to other algorithms, DT requires less data processing, such as data normalization and dealing with missing data.
- The effort to fit a DT model is related logarithmically to the increase of the data points. DTs can handle numerical and categorical data and deal with multi-variable problems, where there is more than one output in the model.

Disadvantages:

- Learning a DT model can sometimes become over-complex challenging the generalization abilities of the model. To avoid this problem, hyper-parameters of the model, such as pruning (removing non-critical parts of tree for data compression), setting the maximum depth, or the minimum number of samples at each leaf can be useful.
- DTs can be highly unstable meaning that a small variation in the input data may lead to a model completely different.
- If a class is dominant in the dataset, it might affect the learning process and, therefore, it is recommended to have a dataset, which is balanced for existing classes.

Like SVM, decision trees are applicable for regression problems, but in a hierarchical manner. Meaning that queries need to be created one after another. Regression trees work differently compared to classification task of a decision tree, i.e., instead of observing the node impurity, that is a metric of success in classifying the dataset, the trees are built according to the sum of squares error (Marsland, 2009). In each node, where the query exists, predictions are made to find values instead of classes of the feature after several series of if/else questions, and after calculating the sum of squares error, the algorithm decides on changing the query or moving forward. To build the tree, it is required to calculate the value of the feature at each node for building the next leaf (Géron, 2017). The values of the feature in each leaf are derived from the mean average of the instances. These mean average values of the feature are optimal to make a split of the feature at a particular value and to minimize the sum of square errors (Marsland, 2009).

In general, the process of building the tree starts with finding the most essential features that will be situated at the top of the tree that is called the root node (Müller and Guido, 2016). In road safety studies, this usually is seen to be the traffic volume variable that has the most significant effect on the number of collisions. This feature has the most of the information for splitting the whole dataset into right and left leaf nodes that are created from a particular benchmark value of the feature. The next immediate right and left leaf nodes served to better estimation of the outcome, and the process of searching for the best split continues until each leaf constitutes a value. Figure 3-3 illustrates a schematic view of a decision process using a DT algorithm.

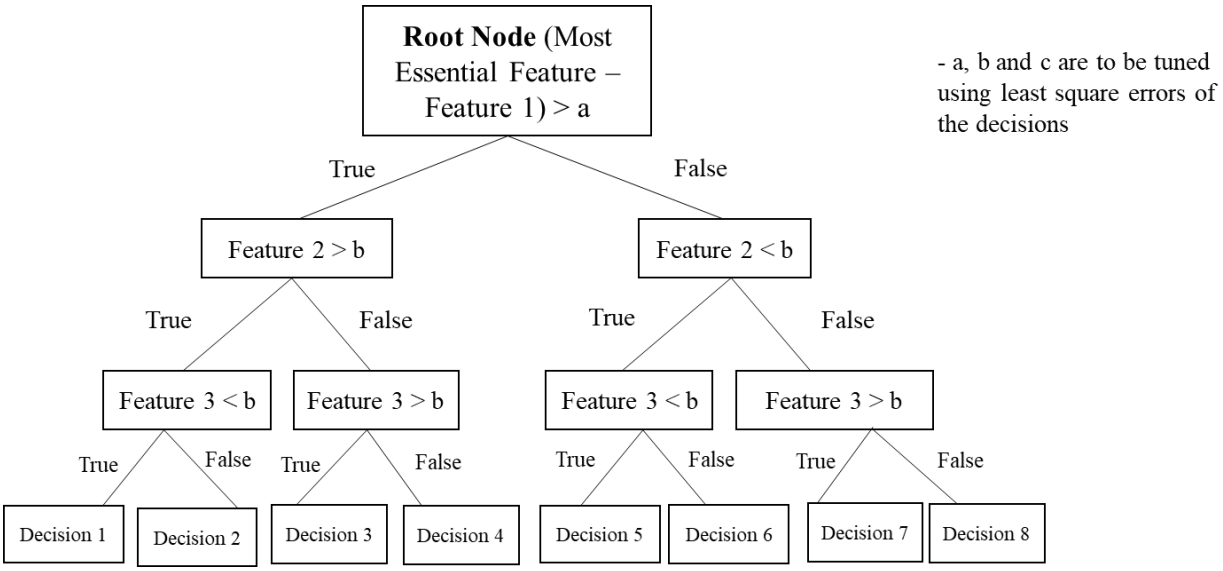


Figure 3-3 An Example of DT algorithm used for regression task

Other instances for the road safety study to be considered in lower levels of the tree are road and environmental factors. During the prediction using the test set with unseen data, the algorithm works by searching a region from top to bottom of the tree where the new points belong, and then the decision is made to categorize with the class or value that has more points. Gini impurity and entropy measures are two methods for measuring information to choose the features of the split. With its fast computation, Gini impurity by default is used as the measure of purity (Géron, 2017). The Gini impurity of a node is calculated as follows:

$$G_T = 1 - \sum_{i=1}^{n} N(i)^2 \quad 3.11$$

Where, $N(i)$ is the number of instances at the node T, which maintains purity with regards to a known estimation error $N(i)$ can be the number of instances where crash frequency that has been calculated is at an acceptable distance from the actual observation, and n is the total number of instances (i.e., intersections) that exist in the study.

3.2.4 Random Forest

Random Forest (RF) is another supervised ML algorithm implemented in this study for collision frequency modeling. It is an ensemble learning of a DT that is averaging the outputs of randomly created trees. Ensemble means the algorithms combine the predictions of many DTs and take the average result from it ($\sum$ in figure 3-4). One way of attaining randomness is by using bagging for bootstrapping the samples from the dataset (Marsland, 2009). Bagging involves using different samples of data (training data) instead of a single dataset. Unlike DTs, RF identifies the best features for splitting from randomly bagged features that give the algorithm an advantage to try different sets of features and in some cases be more optimal than the DT. Moreover, the ensemble learning of the RF algorithm solves the issues of instability in DTs. The randomness in the algorithm results in making a trade-off between a high level of bias and lower variance, which is important for building an optimal predictive model (Géron, 2017). Thus, in each node of the tree, RF selects randomly a number of features and measures the Gini impurity, a measure of impurity in each decision node, from those selected features, then selects the optimal tree. This process continues until all randomly created trees in an ensemble are considered; calculating the mean response is the final step of the algorithm (Marsland, 2009). Moreover, the problem of overfitting observed in DT is also more conveniently dealt with by RF. The random forest creates the trees by averaging the results of the trees to reduce overfitting, that is likely to happen in a single tree (Müller and Guido, 2016). Even though pruning is not needed in the RF algorithm to optimize the performance, there are plenty of hyper-parameters that can be adjusted to fit the model well in the dataset, such as minimum sample leaf, and maximum depth of the trees (Marsland, 2009). Another

characteristic of the random forest is the ability to provide feature importance by taking the mean depth of the features that are appeared in all the trees (Müller and Guido, 2016). Figure 3-4 provides an overview of the RF algorithm.

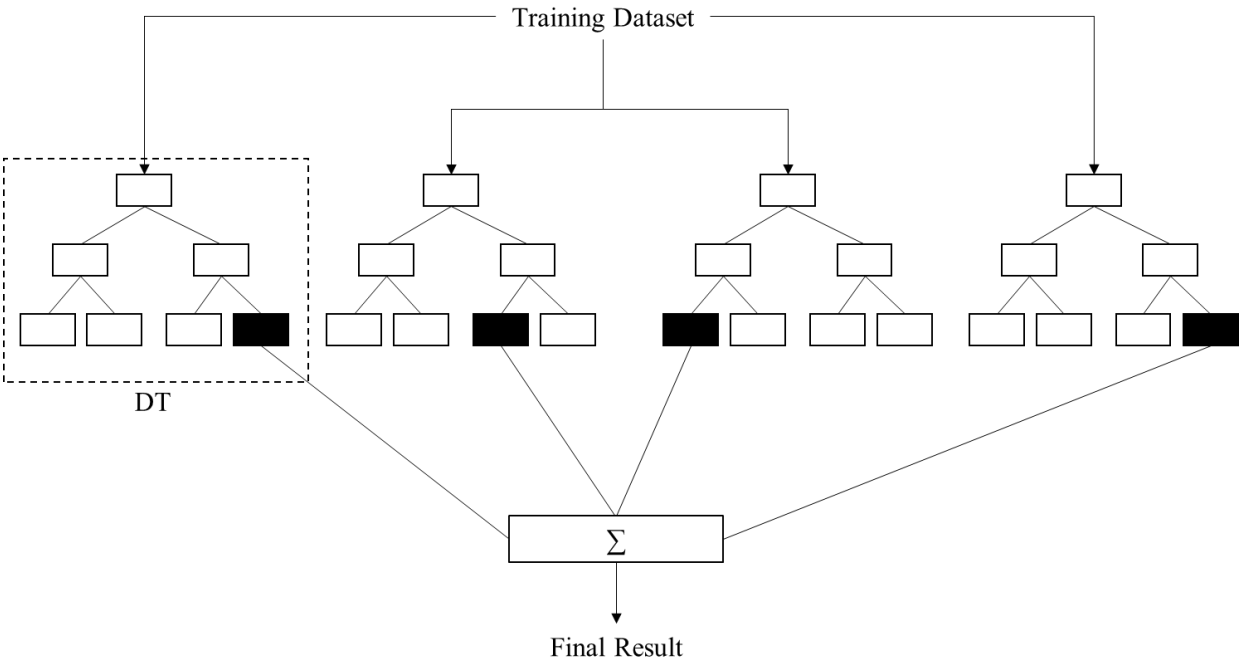


Figure 3-4 A schematic overview of the RF algorithm

3.2.5 Optimization of ML Hyper-Parameters

Due to the high degrees of freedom in fitting the data (that can lead to overfitting), optimization of the hyper-parameters that are tuned for the learning process in the ML algorithms is a necessary task. To find the best set of hyper-parameters to use in each algorithm, grid-search cross-validation (GSCV) was used to optimize the performance of ML algorithms. In summary, GSCV is a technique that looks for an array of desired values to assign to the hyper-parameters of the models and tries every possible permutation of them to find the optimal solution (Pedregosa et al., 2011). For instance, in the SVM algorithm, various types of kernel functions, such as linear, polynomial, and radial-based functions will be applied to find an optimal separation that is suitable for the dataset. On the other hand, in DT and RF algorithm, one of the hyper-parameters is the maximum depth of the tree, which is the length of the tree from the root node to the leaf nodes that will let the model avoid over-complexity. In other words, the process of creating the decision tree can go further and further with more and more decision nodes or it can be optimized to find the most economic depth that will avoid overfitted tree and reduce estimation time. Adjusting other parameters such as the minimum sample split (the minimum number of samples required before the split) and minimum leaf split (the lowest number that a leaf node must have) will also reduce the complexity of the tree that in turn prevents the likelihood of the overfitting problem. The so-called split is the decision that is to be made at each leaf for further creation of leaves in lower levels of the tree.

3.2.6 Goodness-of-Fit Criteria

Three goodness-of-fit criteria were employed in this study to compare the prediction performance of ML models and SPFs: the coefficient of determination (R^2), the mean absolute deviation (MAD), and the mean square predicted error (MSE) (Oh et al., 2006).

The coefficient of determination is reported in equation 3.12:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}; \qquad 3.12$$

Where, $SS_{res}$and SS_{tot} are the residual and the total sum of squares, and are calculated as follows:

$$SS_{res} = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2; \qquad SS_{tot} = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2; \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{3.13}$$

Where, n is the number of observations, y_i is the actual value of the i^{th} observation (in this study, crash frequency), $\hat{y}_i$ is the model calculated value for the i^{th} observation and $\bar{y}$ is the average of the actual value for the dependent variable over all the observations. Other metrics are as follows:

$$MAD = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{3.14}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{3.15}$$

Chapter 4

Model Implementation and Results Analysis

4.1 Model Implementation

In this study, ML models are created using Python 3.7.4 with the help of the sci-kit-learn library and evaluation metrics (i.e., R-score, MAD, MSE) within the scientific python development environment (Spyder-IDE.org). SPFs were developed using the GENMOD procedure of SAS software where model parameters were estimated with the maximum likelihood method (SAS Institute Inc., 2018). For both time periods I (2013-2015) and II (2016-2018), 275 sites (80% of data) were employed for model development (training), and 68 sites (20% of data) were employed to assess transferability performance (testing). This proportion of sites for testing and training was chosen according to the ML literature (Lie et al., 2008; Zeng and Huang, 2014; Chen et al.,2016; and Dong et al., 2018). The resulting SPFs for both time periods using training data were reported in Table4-1.

Table 4-1 SPFs for both time periods (training data)

Parameter*	Variable	Mean	Standard Error	p-value
Time Period I (2013-2015)				
a_0	Intercept	-8.858	0.793	<0.0001
a_1	Major AADT	0.665	0.078	<0.0001
a_2	Minor AADT	0.405	0.076	<0.0001
b_1	Number of Legs	0.393	0.095	<0.0001
b_2	Control Type	0.419	0.110	0.0001
Dispersion		0.339	0.037	
Time Period II (2016-2018)				
a_0	Intercept	-10.119	0.789	<.0001
a_1	Major AADT	0.772	0.077	<.0001
a_2	Minor AADT	0.441	0.069	<.0001
b_1	Number of Legs	0.295	0.105	0.0048
b_2	Control Type	0.382	0.089	<.0001
Dispersion		0.276	0.030	

*See equation 3.7 for parameter description

All model parameters were found significant at the 95% confidence level with their p-values lower than 0.05. The goodness-of-fit indicators for fitting the NB regression, such as scaled deviance and the Pearson χ^2 values are shown for both time periods in table 4-2. As can be seen in this table, the indicators are showing sufficient values and suggesting an acceptable model, being their χ^2 lower than degrees of freedom.

Table 4-2 Goodness of fit indicators for NB model in time periods I, and II.

time period	degree of freedom (df)	scaled deviance (SD)	χ^2	SD/df	χ^2/df
I	270	302.4	245.7	1.12	0.91
II	270	320.7	260.9	1.19	0.97

For the ML algorithms, as explained before, no model can be presented as the learning process does not follow a functional form. However, as explained in the model optimization section using GSCV, an array of desired hyper-parameters is introduced. Table 4-3 and 4-4 provides the details of this array and the optimized selection of these hyperparameters by each ML algorithm for the models developed for both time periods I and II. A brief description of each hyper-parameter is provided in appendix B.

Table 4-3 Best selected hyper-parameters for ML algorithms using GSCV in time period I

Hyper-parameter	**SVM Algorithm**	**Selected Hyper-parameter**
C	600, 700, 800, 900, 1000	600
Kernel type	Linear, Polynomial, Radial-Basis Function (RBF)	RBF
Degree	2, 3, 4, 5, 6	N/A*
Gamma	Scale, Auto	scale
ε	10^{-4}, 10^{-5}, 10^{-6}	10^{-5}

Hyper-parameter	**DT**	**Selected Hyper-parameter**
Mean_samples_leaf	9, 10, 11, 12, 13	11
Max_depth	4, 5, 6, 7	6
Min_impurity_decrease	0, 1, 2	1

Hyper-parameter	**RF**	**Selected Hyper-parameter**
Mean_samples_leaf	1, 3, 5, 7, 9, 11, 13, 15	11
Max_depth	1, 3, 5, 7, 9, 11, 13	7
Bootstrap	True, False	True

*Degree does not apply to the RBF kernel function

Table 4-4 Best selected hyper-parameters for ML algorithms using GSCV in time period II

	SVM Algorithm	**Selected Hyper-parameter**
C	600, 700, 800, 900, 1000	600
Kernel type	Linear, Polynomial, Radial-Basis Function (RBF)	Linear
Degree	2, 3, 4, 5, 6	N/A*
Gamma	Scale, Auto	scale
ε	10^{-4}, 10^{-5}, 10^{-6}	10^{-4}

	DT	**Selected Hyper-parameter**
Mean_samples_leaf	9, 10, 11, 12, 13	12
Max_depth	4, 5, 6, 7	5
Min_impurity_decrease	0, 1, 2	0

	RF	**Selected Hyper-parameter**
Mean_samples_leaf	1, 3, 5, 7, 9, 11, 13, 15	9
Max_depth	1, 3, 5, 7, 9, 11, 13	5
Bootstrap	True, False	True

*Degree does not apply to the linear kernel function

The hyper-parameters of the ML models are selected based on the rationality to explore a few options while keeping the estimation time low. For example, exploring the maximum depth could be with more than 4 options in decision tree, however, only 4 options are selected for optimization in a reasonable range, that is to avoid creating too shallow or too deep trees, which will increase the likelihood of underfitting or overfitting problems, respectively.

Due to the differences in both time periods and the variations due to the selection of hyper-parameters it is recommended to use an optimization tool to come up with the best solution. For this purpose, GSCV optimization is used to select the most efficient hyperparameters from the introduced array. It is based on the project requirements to further increase the array of the hyper-parameters to select from. A rational way to select this array is when the change in the results is negligible when changing the hyper-parameters.

4.2 Prediction Performance Evaluation

As the first step in evaluating the results, the performance of the NB model is compared with the ML algorithms in terms of fitting and predicting abilities. The metrics introduced in section "3.2.5" (i.e., R^2, MSE and MAD) are calculated and compared in tables 4-5 and 4-6. It is worth mentioning that while these metrics are automatically calculated for ML algorithms using the sci-kit-learn library, the corresponding values for the NB model are calculated manually using the generated SPFs as shown in table 3-1.

Table 4-5 Performance comparison of NB and the ML algorithms in the time period I (2013-2015)

training set			
Model	$\boldsymbol{R^2}$	**MSE**	**MAD**
SVM	0. 663	191.345	8.933
DT	0.723	157.041	8.840
RF	0.702	169.208	8.654
NB	0.606	223.405	10.281
test set			
Model	$\boldsymbol{R^2}$	**MSE**	**MAD**
SVM	N/A*	212.857	9.270
DT	N/A	154.608	9.084
RF	N/A	185.903	8.774
NB	N/A	198.733	9.177

*R^2 is representing the goodness of fit and does not apply to the test set.

Table 4-6 Performance comparison of NB and the ML algorithms in the time period II (2016-2018)

training set			
Model	R^2	**MSE**	**MAD**
SVM	0.632	204.439	9.498
DT	0.734	147.613	8.276
RF	0.736	146.580	8.151
NB	0.642	198.888	9.434

test set			
Model	R^2	**MSE**	**MAD**
SVM	N/A*	164.327	8.643
DT	N/A	214.214	9.768
RF	N/A	194.420	9.275
NB	N/A	151.065	8.091

*R^2 is representing the goodness-of-fit and is not applicable to the test set.

Several observations can be made by analyzing the values of the metrics shown in Table 4-5 in the time period I. First, DT is showing a better fitting to the training dataset with its R^2 being higher , and MSE, and MAD values lower than the other models. RF and SVM holds the next positions. In this time period, the NB model is showing the poorest fit with R^2 equal to 0.606, training MSE of 223.405, and training MAD of 10.281. Regarding the training MAD values for DT and RF, one can observe a higher value for DT compared to RF, while training MSE for these two algorithms is the opposite. Knowing that MSE is the second moment of the error, shown in equation 3.15, this is suggesting that while RF is having less prediction error, its wrong predictions are relatively more distanced from the observation when comparing to DT.

Having the same analysis for time period II, RF is having a better fitting ability its R^2 being higher , and MSE, and MAD values lower than the other models . DT and NB are holding the next positions. In this time period, the SVM model is showing the poorest fit with R^2 equal to 0.632, training MSE of 204.439, and training MAD of 8.643. These differences in the ranking of models in two time periods were expected as there are variables that exist in the dataset, such as traffic volume and collision counts, and they can vary from the time period I to II. However, in terms of fitting to the training set, DT and RF are approximately showing similar fitting ability, which outperforms SVM and NB.

Looking at the test set in the time period I and comparing the values with the training set, conclusions can be made regarding the generalization ability, also known as the overfitting issue. DT still holds the first position with the lowest MSE value of 154.608, which is indeed lower than its MSE for the training set. This suggests that the DT model has sufficient generalization ability in this time period. This is also the case when looking at the NB model, with a lower MSE in the test set comparing to the training set. However, RF and SVM are experiencing higher MSE and MAD values in the test set when compared to the training set in the time period I. In time period II, however, the NB model is acting as the best model in terms of generalization ability being its test error lower than training error. Subsequently, in this time period, SVM holds the next position, with its test error still lower than the training error, but not as significant as it was in the NB model. However, DT and RF are experiencing higher MSE and MAD values in the test set when compared to the training set in time period II.

Overall, due to the natural fluctuations of traffic and collision data in the period of study, slightly different conclusions can be made between time periods. However, it seems that in both time periods, DT and RF models are acting more similarly while SVM results stand more closely to the NB model. It can be concluded that DT and RF are showing better fitting abilities, while they may still suffer from overfitting problems. Comparing the results of this study with the works of Xie et al. (2007) and Lie et al. (2008), once more it is proved that ML algorithms can fit sufficient models to the collision data that can have similar, if not better, predictions capabilities in comparison with the traditional NB regression.

4.3 Sensitivity Analysis

One of the downsides of the ML algorithms is that they lack any functional forms that will consist a model parameter for each explanatory variable (Xie et al., 2007), and therefore, making it hard to interpret the impact of each variable on the outcome. Even in DT and RF algorithms, where the model is easier to interpret, the impacts of each variable to the outcome cannot be easily understood. As a response to this critic around the use of ML algorithms in road safety analysis, a sensitivity analysis, inspired by the works of Fish and Blodgett (2003), Delen et al. (2006), and Xie et al. (2007) is conducted.

The basic procedure in conducting this analysis is that, in order to understand the effect of each explanatory variable on the outcome of the model, one needs to keep all other explanatory variables constant and change the value of the desired one within a reasonable interval. Each time that the value of the desired explanatory variable is changed, the corresponding outcome (collision frequency in this study) is recorded and, in this way, the effect of that particular explanatory variable on the outcome is obtained.

In this study, a summary statistic (mean and variance) of sensitivity analysis for 68 sites is reported. In this way, an average value for the obtained impact of each explanatory variable over the whole data set is calculated and compared with the model parameters in the NB method. The explanatory variables used in this study are average annual daily traffic on the major approach of the intersection (V_1), average annual daily traffic on the minor approach of the intersection (V_2), control type (CT), and the number of legs (Nlegs). The former two are numerical while the latter two are categorical (refer to table 3-1).

In order to compare the impacts of these variables with the NB method, the sensitivity of the model to V_1 and V_2 is considered to be related to the collision frequency with a power function. On the other hand, however, CT and Nlegs are considered to be exponentially related to the collision frequency. In other words, to be able to compare the results of sensitivity analysis for each explanatory variable with the model parameters of the NB model for the same explanatory variable, a similar form of contribution to the output of the ML model (collision frequency) is considered. To clarify, V_1 and V_2 are considered to be related to the output of the model with a power function, while CT and Nlegs are related in an exponential function as shown in equation

3.7, and therefore, a similar type of contribution is considered for ML algorithms. To obtain the sensitivity of ML algorithms to the V_1 and V_2, an array of 10 instances ranging from $0.1*V_i$ to $1.0*V_i$ is considered. This range of inputs is selected due to the fact the normalized data should remain between 0 and 1. For example, if the actual V_1 is 1000 veh/day, the analyzed interval in the sensitivity analysis is [100, 200, 300, …, 1000]. For the categorical variables (CT and Nlegs), the opposite value is given to the model. For example, if the actual intersection is signalized (with CT=1), it is changed to be unsignalized (with CT=0). Figure 4-1 is illustrating this matter in a schematic way for the first test site (7th Ave N and 33Rd St E, Saskatoon) with an actual V_1 equal to 8000 veh/day). Upon every change in the variables, the change in the output (collision frequency) is recorded.

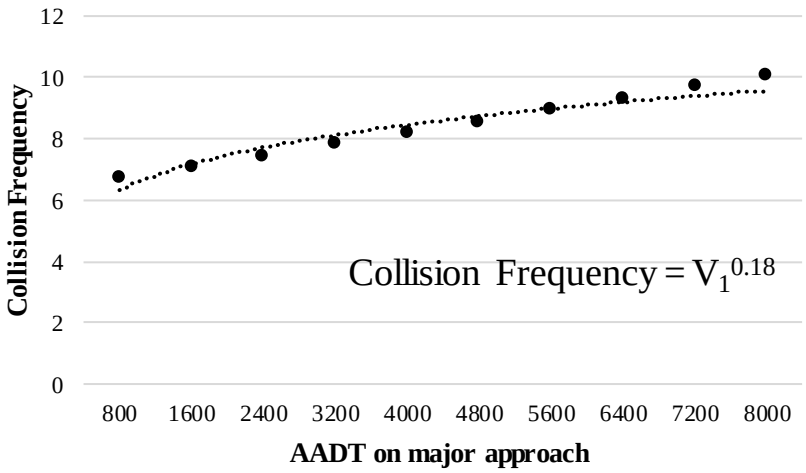


Figure 4-1 Sample relationship for sensitivity analysis of V_1 (7th Ave N and 33Rd St E, Saskatoon)

Sensitivity analysis is conducted in both time periods I and II, and the results are compared with the corresponding NB model in the same time period shown in table 3-1. Tables 4-7 and 4-8 indicate the average results of sensitivity analysis for time periods I and II, respectively. For a detailed site-by-site sensitivity analysis, refer to Appendix C.

Table 4-7 Sensitivity analysis in the time period I

Model	Sensitivity of The Output to The Explanatory Variables			
	V_1	V_2	CT	Nlegs
SVM	Mean: 0.5 Variance: 0.25	Mean: 0.39 Variance: 0.17	Mean: 0.99 Variance: 0.23	Mean: 0.94 Variance: 0.20
DT	Mean: 0.52 Variance: 0.22	Mean: 0.31 Variance: 0.20	Mean: 1 Variance: 0	Mean: 1 Variance: 0
RF	Mean: 0.48 Variance: 0.25	Mean: 0.25 Variance: 0.18	Mean: 1 Variance: 0	Mean: 1 Variance: 0
NB	Mean: 0.66 SE*: 0.08	Mean: 0.41 SE: 0.08	Mean: 1.48 SE: 0.09	Mean: 1.52 SE: 0.09

*standard error

Table 4-8 Sensitivity analysis in the time period II

Model	Sensitivity of The Output to The Explanatory Variables			
	V_1	V_2	CT	Nlegs
SVM	Mean*: 0.33 Variance: 0.33	Mean: 0.56 Variance: 0.36	Mean: 1.27 Variance: 1.20	Mean: 1.59 Variance: 1.52
DT	Mean: 0.23 Variance: 0.28	Mean: 0.14 Variance: 0.22	Mean: 2.11 Variance: 0.95	Mean: 1 Variance: 0
RF	Mean: 0.33 Variance: 0.23	Mean: 0.12 Variance: 0.17	Mean: 1.31 Variance: 0.34	Mean: 1.12 Variance: 0.09
NB	Mean: 0.77 SE**: 0.07	Mean: 0.44 SE: 0.08	Mean: 1.46 SE: 0.10	Mean: 1.35 SE: 0.09

*the mean and the variance are for the 68 recorded sensitivity of the 68 sites of the test set
**standard error

From sensitivity analysis in period I, as shown in table 4-7, the ML algorithms are influenced similarly by explanatory variables in comparison to NB model parameters. In particular, all the three ML models are showing higher sensitivity to the traffic volume of the major approach (V_1)

compared to the minor approach (V_2). However, the sensitivity of each model to these explanatory variables is different due to the different ML algorithms. On the other hand, looking at the CT and Nlegs, there is a considerable difference from model to model. While the NB model is showing that changing the CT from signalized to unsignalized is associated with higher collision frequency, ML algorithms are showing a neutral sensitivity to these two explanatory variables. This suggests that the ML algorithms are not developed around CT and Nlegs and they are mainly using MJAADT and MNAADT for their predictions. In other words, based on the interpretations of the sensitivity analysis of the ML algorithms, having a signalized or unsignalized intersection, or having a 3-leg or a 4-leg intersection are not necessarily associated with higher/lower collision frequency.

In time period II, the results are slightly different. While all the models are showing higher sensitivity to the V_1 compared to V_2, collision frequency prediction in SVM, seems to be more sensitive to the changes in the V_2. However, apart from V_1, the sensitivity of the SVM model to the V_2, CT and Nlegs, is more aligned with the parameters of NB and once again suggesting that SVM provides results more comparable to the NB while DT and RF are significantly different. In addition, in this time period, unlike one instance in DT, all the models showed to be sensitive to CT and Nlegs variations. DT model in this time period, though, is still showing insensitivity to Nlegs. Overall, it was expected that a "perfect" alignment of ML results to NB model parameters was not possible since the two techniques are based on different modeling principles. Still, this sensitivity analysis can enable practitioners to gain more insights into how the ML outcomes are affected by the existing explanatory variables.

4.4 Validation of ML Algorithms in Network Screening

In this section, the performance of ML algorithms used within the RSMP was compared to the NB model. To do this, data was collected for sites with no major road improvements during the period of study (2013-2018). Then, the time period was divided into time periods I (2013-2015) and II (2016-2018). Due to the fact that no major road safety improvements were installed during this period of time, sites that are identified as hotspots in time period I were expected to remain hotspots in time period II as well. A method consistency expressed by the consistency test (Tc) proposed in

the literature (Cheng and Washington, 2008; Lan and Persaud, 2011; Sacchi et al, 2015) was employed. The consistency test involves two ranked lists of the same dataset in subsequent time periods i and i+1 being compared and evaluated using the following evaluation criteria:

$$Tc = \{k_1, k_1, \dots, k_{n\delta}\}_{j,i} \cap \{k_1, k_1, \dots, k_{n\delta}\}_{j,i+1} \qquad 4-1$$

where k_i is the i^{th} ranked site identified as a hotspot, n is the total number of sites in the dataset, δ is the threshold of identified high-risk sites (e.g., δ=0.1 corresponds with the top 10% of sites identified as hotspots), and j represents the ranking method(s) being compared.

To conduct a comparison between the list of hotspots, network screening has been carried out in both time periods using NB and ML models. The results were divided into two parts. First, each method was used for ranking the sites in both time periods and the consistency of them in identification of similar hotspots is studied. This step is called within methodology consistency check. In step two, however, the consistency of ML algorithms and the NB model in identifying the hotspots was studied in each time period separately. This step is called across methodology consistency check. The findings of the two analyses assist with proving the ability of ML algorithms to be used within the actual practices of RSMP.

The performance measures selected in this study are "excess predicted average crash frequency using the SPF", "excess average crash frequency with EB adjustment", and "excess expected average crash frequency with EB adjustment", also known as "potential for safety improvement (PSI)". The details on how to calculate each of these performance measures are provided in equations 4.2 to 4.4. For the NB regression model, all the three performance measures are available, however, for comparing the ML algorithms together we are only relying on the outcomes of the first performance measure by substituting the SPF results with the ML results. This is because there is no such thing as a dispersion parameter in the ML algorithms that can be used for the two latter performance measures. In short, the following terms will represent each performance measure:

- y-NB or y-ML (i.e., SVM, DT, RF) for the excess predicted average crash frequency using the SPF or ML:

$$(y - NB)_i = y_{observed_i} - E(m)_i \quad 4.2$$

- EB for the expected average crash frequency with EB adjustment:

$$EB = w \times E(m)_i + (1 - w) \times y_{observed_i} \quad 4.3$$

- PSI for the excess expected average crash frequency with EB adjustment

$$PSI = EB - E(m)_i \quad 4.4$$

Where, $y_{observed_i}$ is the observed collision frequency, and $E(m)_i$ is the predicted collision frequency at the i^{th} site.

It is worth mentioning that the collision prediction models are the same as the models developed using the training set for time periods I and II in previous sections that are shown in tables 4-2, 4-3, and 4-4. To perform the consistency analysis, network screening was conducted for the 68 sites of the test set.

4.4.1 Within Methodology Consistency Check

Within methodology consistency check reveals useful information about the ranking consistency of the performance measures that are based on the NB model in identifying the hotspots in two time periods (network screening). This is done by using the obtained ranking consistency in the NB-based performance measure and comparing it with the ranking consistency of the ML-based performance measures. It is being called "within method consistency" as the consistency of ranking is first being overviewed in each modeling technique (i.e., NB-based ranking and ML-based ranking) separately, and then compared with each other to understand the adequacy of ML algorithms in providing the consistent ranking. Figures 4-2 and 4-3 describe the results of NB-based and ML-based consistency checks, respectively.

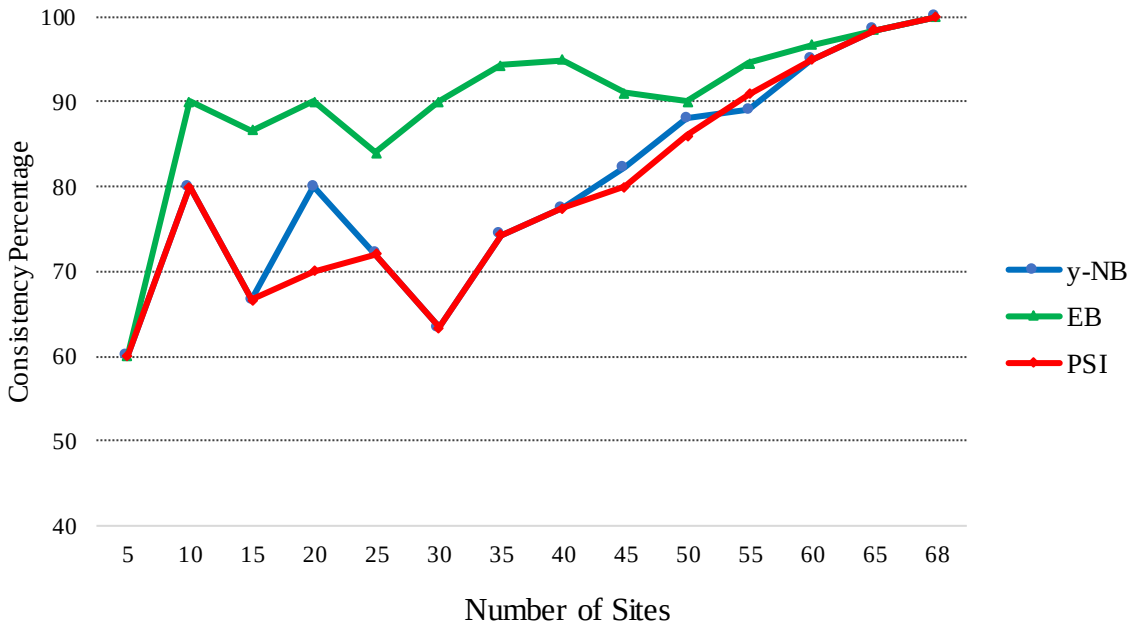


Figure 4-2 Within methodology consistency check in ranking of hotspots using NB model

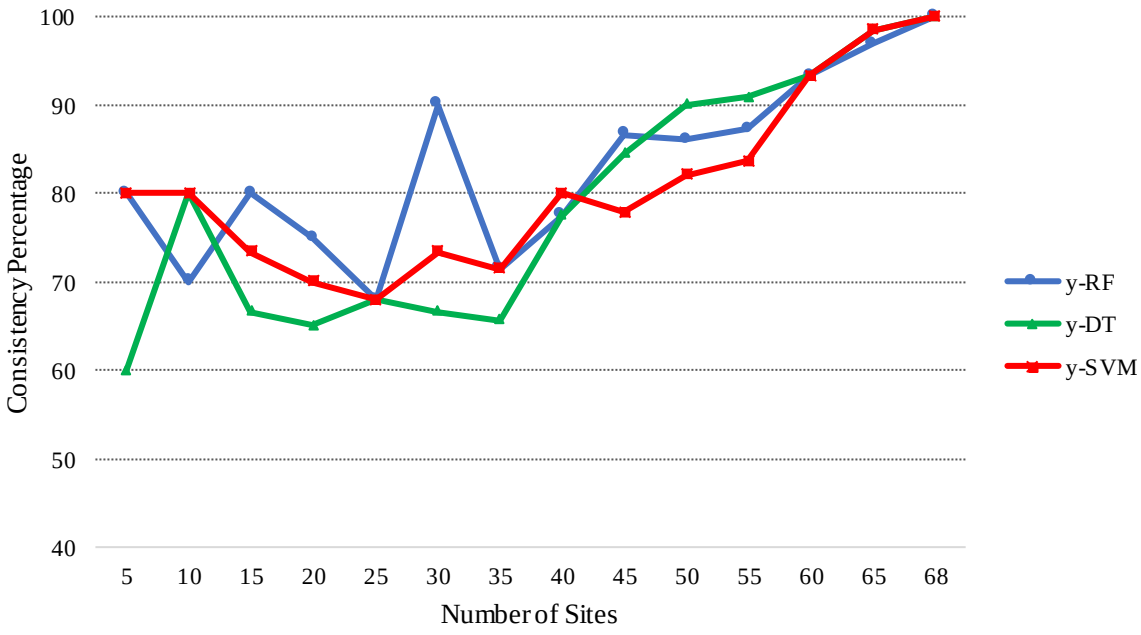
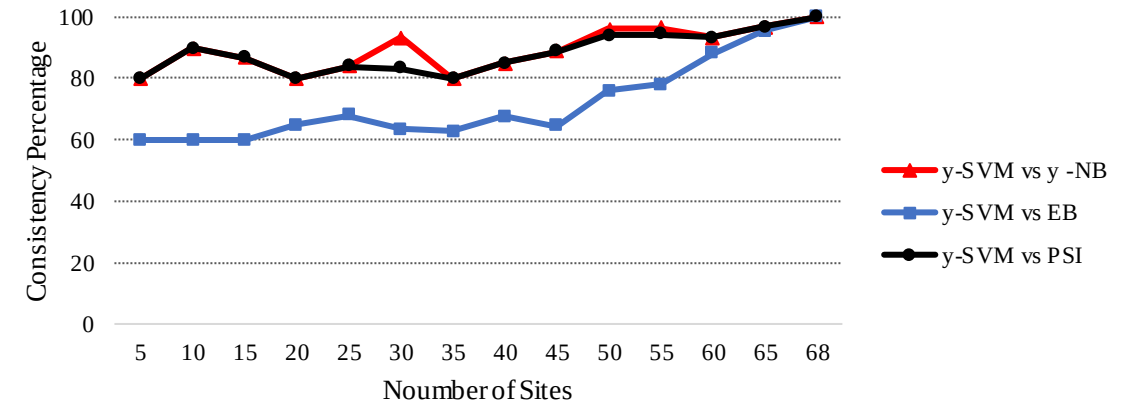


Figure 4-3 Within methodology consistency check in the ranking of hotspots using ML algorithms

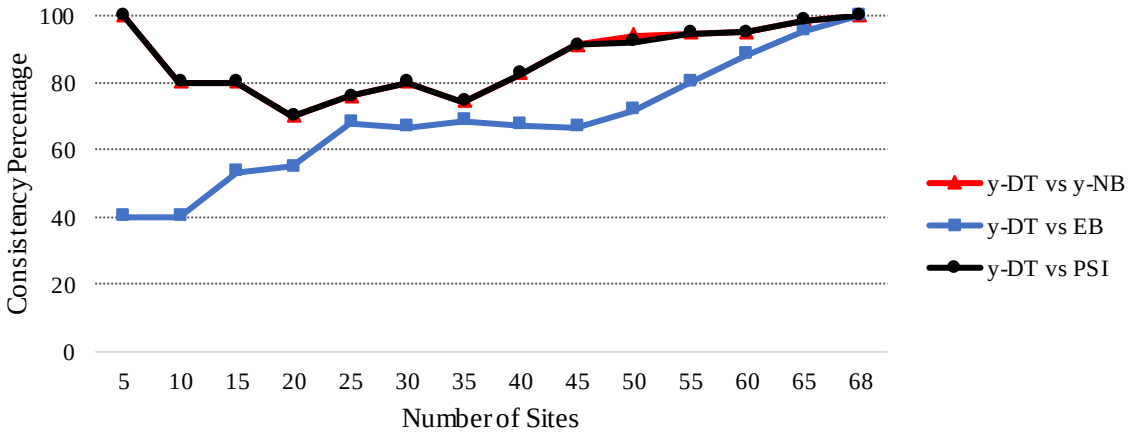
Overall, both SPF-based and ML-based performance measures showed ranking consistency above 60% in the two time periods of the study. Among SPF-based measures, Figure 4-2 shows that the excess average crash frequency with EB adjustment (EB) outperformed y-NB and PSI methods, being its consistency always beyond or equal to these two latter measures. The EB method shows a 60% consistency in identifying the top 5 hotspots and a 90% consistency for the top 10 hotspots. Similarly, y-NB and the PSI method are showing a 60% consistency for the top 5 hotspots, while their consistency is at 80% for the top 10. Among ML-based measures, Figure 4-3 shows that the results are mixed and there is no single algorithm outperforming others; for the first 5 hotspots, y-SVM and y-RF appear to be superior to y-DT with a consistency of 80%. When 10 hotspots are considered, y-DT and y-SVM consistencies are equal to 80% and above y-RF that standing at 70%. Overall, the ML-based measure consistency appears to be comparable to the SPF-based measures shown in Figure 4-2 and in particular to y-NB. These results can also be compared to the findings of other consistency analyses available in the literature. In 2008, Cheng and Washington (Cheng & Washington, 2008) conducted a consistency check using four performance measures (i.e., observed collision frequency, collision rate, EB and PSI) and reported, as highest rate, 56% consistency for the EB method for the top 10% hazardous sites. Lan and Persaud (2011) explored both the EB and the FB method and observed a consistency (sensitivity) of up to 60% for the top 10 hotspot locations. More recently, Sacchi and others (Sacchi et al., 2015) developed a multivariate FB performance metric to identify hotspots and reported a consistency above 70% for the top 10 hotspots. Therefore, ML-based performance methods appear to provide similar consistency, being their value is always above 60%.

4.4.2 Across Methodology Consistency Check

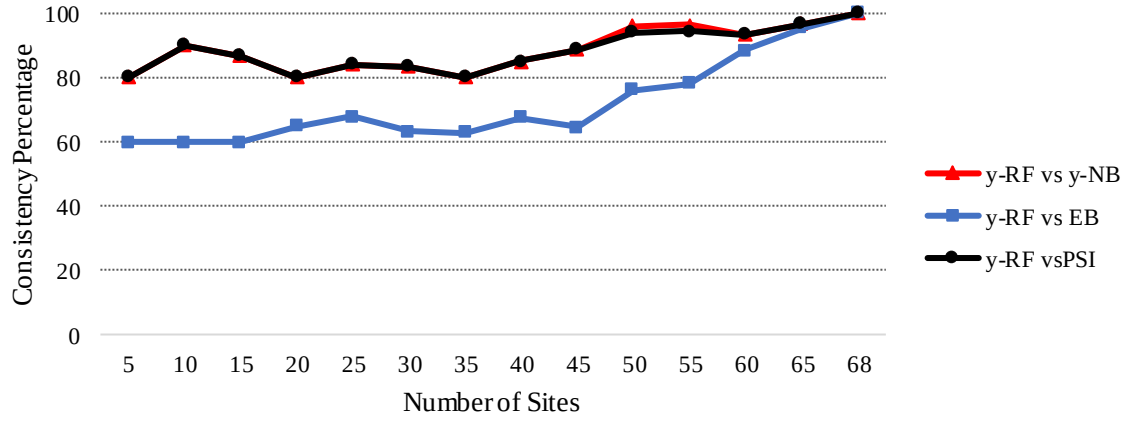
In this last analysis, a ranking consistency test of collision-prone locations across performance measures (SPF-based versus ML-based) in both time periods was conducted. This can be done by using different ranking methods (j) in the two components of equation 4.1. The results of the analysis are reported in Figure 4-4 and 4-5 for time period I and time period I, respectively.



(a)

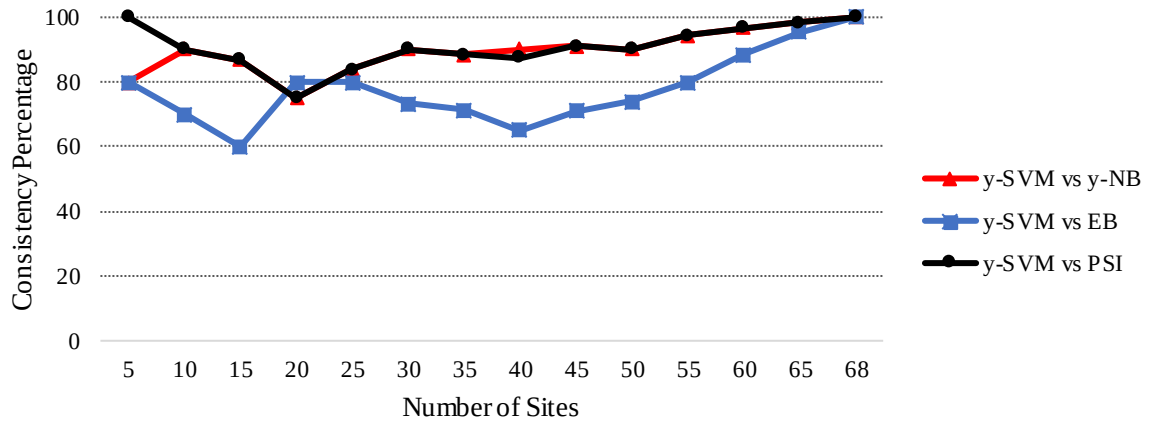


(b)

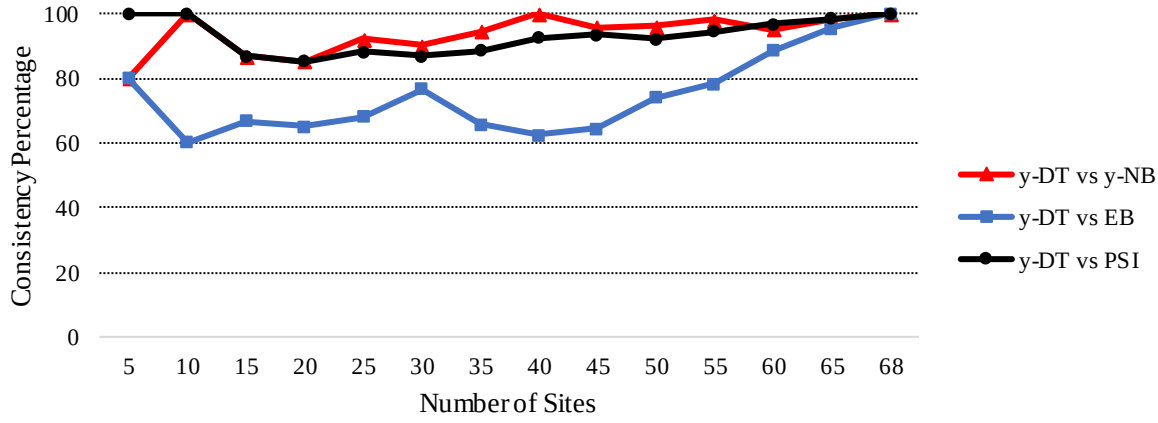


(c)

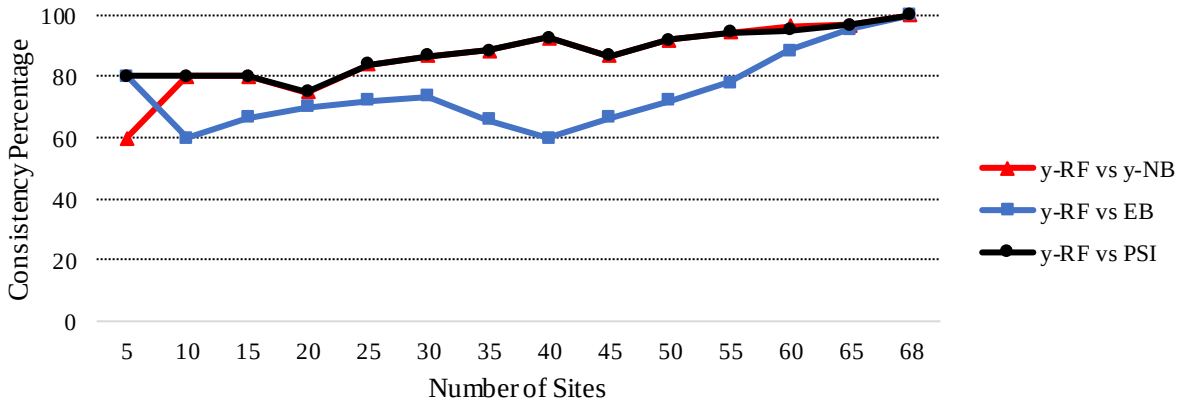
Figure 4-4 Across methodology consistency check using NB and ML algorithms in time period I



(a)



(b)



(c)

Figure 4-5 Across methodology consistency check using NB and ML algorithms in time period II

In time period I, ML-based performance measures showed a consistency with PSI and y-NB approximately equal to 80% or above when more hotspots were added (Figure 4-4a, 4-4b, and 4-4c). However, the consistency with the EB method was found to be lower: approximately 60% for the first 10 hotspots for y-SVM and y-RF, and approximately 40% for the first 10 hotspots for y-DT. Similar conclusions can be inferred also for time period II, but the consistency with the EB method was found to be higher: in all cases, a consistency above 60% was recorded (see Figure 5-5a, 5-5b, and 5-5c). It is important to note that, similar conclusions compared to time period I cannot be expected also in time period II as the models developed in both time periods are independent and therefore, different from each other.

Another comparison can be made within the ML algorithms in this section to understand their differences in providing consistent rankings compared to the NB-based performance measures. In time period I, the DT model is having 100% consistency with the y-NB and the PSI method in identifying the top 5 hotspots, and is superior to SVM and RF with both having 80% consistency with the y-NB and the PSI method in identifying the top 5 hotspots. However, as the number of hotspots included in the consistency test increase, an over 80% consistency with the y-NB and the PSI method is steadily seen for SVM and RF, while DT consistency has dropped lower than 80% at 20, 25 and 35 sites (figure 4b). A conclusion can be drawn that due to the nature of the DT model for being sensitive to certain values in decision nodes, the model may be less consistent with the NB-based rankings when larger number of sites are considered. This idea is also supported in time period II, where y-DT is 100% consistent in both 5 and 10 sites level with the y-NB and the PSI, while SVM is 100% and 90% consistent for 5 and 10 sites, respectively, and RF is 80% consistent for both 5 and 10 sites.

Overall, ML-based metrics analyzed in this study appear to be acceptable and reliable measures for network screening exercises, comparable to SPF-based measures. As expected, their consistency is more similar to the excess (predicted or expected) average crash frequency rather than the expected crash frequency with the EB adjustment alone.

Chapter 5

Conclusions

5.1 Summary of Findings

The RSMP is the approach suggested by the HSM for addressing safety issues in a road network. The first step of the RSMP is network screening that consists in ranking and identifying the most hazardous locations. Traditionally, statistical models have been used for modeling collision data and screening a network. More recently with the advent of computing technologies, ML algorithms have being more widely explored and compared with traditional modeling techniques. In this study, three well-known ML regression models were used and their performance was compared with the traditional NB framework for collision frequency modeling. Moreover, the ML algorithms were validated in the first step of the RSMP. To accomplish these objectives, a 6-year study period was analyzed for a group of urban intersections.

The results of this study revealed that DT and RF learning algorithms outperformed traditional NB techniques and SVM learning algorithm, in terms of fitting, with higher R^2 and lower prediction errors (MSE and MAD). On the other hand, NB and SVM models showed lower overfitting problems (higher generalization capabilities) when transferring the models from training to test set. These findings supported the fact that ML can be used to model collision data with similar, if not better, performance than traditional statistical modeling. A disadvantage of ML algorithms is that no interpretable parameters for the explanatory variables are provided as they work as "black boxes". Therefore, a sensitivity analysis was conducted in this thesis work in both time periods and the effect of each explanatory variable on the output (collision frequency) was analyzed. In time period I, SVM, DT, and RF show an average "parameter" of 0.5, 0.52, and 0.48 for the MJAADT while the NB model is showing 0.66. The MNAADT shoed an average of 0.39, 0.31, and 0.25 for SVM, DT, and R, respectively, and 0.41 for the NB model. It is worth mentioning that the collision frequency is related to the traffic volume with a power function and these values for MJAADT and MNAADT were estimated as power coefficients. For the CT and Nlegs, however, the ML models seem to have less influence collision frequency while the NB model

showed a more significant effect. In time period II, while DT, RF, and NB models showed higher sensitivity to MJAADT, the SVM model was more sensitive to MNAADT. This part of the study proved that ML had roughly similar sensitivity to the explanatory factors in terms of directions and magnitudes although they seemed to experience different contributions from the explanatory variables.

Finally, ML algorithms were used to perform the network screening in both time periods and the consistency of hotspots identified in each time period is investigated. The results indicate that similar to the NB model, ML models were providing more than 60% consistency in identifying the hotspots in both time periods. Among the ML algorithms, SVM has the highest best performance with 80% consistency in identifying the top five hotspots. However, increasing the number of sites in the consistency check makes the RF model superior. Using the across methodology consistency check, it is proved that the ML algorithms and the NB model are pretty consistent in identifying the similar hotspots for both time periods of the study, which validated the use of ML in the actual practices of road safety, such as RSMP.

5.2 Research Implications

The RSMP is a systematic approach to identify the most hazardous locations and assign budgets in an efficient way to improve the safety conditions at transportation facilities. This study demonstrated that ML algorithms can be employed in place of the statistical regression models that are used traditionally within the RSMP to model the collision data and identify the most hazardous locations.

ML algorithms provide road safety practitioners with predictive tools that require fewer pre-assumptions and bring more modeling flexibility in comparison with the NB modeling as the current default approach for developing SPFs. Although ML algorithms shows similar prediction errors to traditional techniques, they are different tools providing predictions through a learning process from the actual observations of the study. Like any other modeling technique, ML algorithms also have some drawbacks. First, the learning process is limited to the domain of the input variables meaning that biased predictions can be estimated if a developed predictive model is used for out-of-the-domain observations. For instance, an observation of zero circulating traffic

volume at an intersection has no real value in a dataset and, it is obvious that this would correspond to zero collisions. However, ML algorithms are not designed to account for this scenario in their predictive models, and might provide inaccurate predictions when used outside the boundary that they are being trained from. (i.e., predicting collisions when volume is equal to zero). This is not an issue with traditional SPFs, since their pre-set function is defined in a way that zero collisions are predicted for zero traffic volume. Second, ML algorithms cannot be easily transferred to other jurisdictions, where data is not available. More research is required to study ways to transfer ML algorithms to other jurisdictions. One important factor to take into consideration is that ML is being learned from a set of specific explanatory variables and transferring the developed model to another jurisdiction that does not include similar explanatory variables is not recommended. In other words, considering the modeling nature of ML algorithms, the model might change significantly if one explanatory variable is removed or added.

Regarding model fitting and generalization abilities, it was shown that ML algorithms used in this study (SVM, DT, and RF) are providing similar and, sometimes, better results than traditional models. It was observed that in terms of fitting, regression trees (DT and RF) are able to fit data better and provide less prediction errors. On the other side, it was observed that SVM performs closer to the NB model in terms of fitting and errors. However, the generalization abilities of regression trees were observed to be lower than the SVM and the NB models. The sensitivity analysis results show that the SVM and the NB models are similarly affected by the explanatory variables in most cases. Some insensitive responses were observed from DT and RF when changing the values of CT and Nlegs suggesting that the regression trees' predictions were mainly affected by the changes in the traffic volume. Using sensitivity analysis, research should be conducted to observe the impact of different explanatory variables on the outcome.

Overall, ML algorithms can be used within the RSMP for performing network screening. As a current practice, network screening in the HSM is based on statistical SPFs, which have some important limitations. The findings of this study support the use of ML in the actual practices of the RSMP that is currently based on statistical regression techniques.

5.3 Limitations and Future Work

While the results of this thesis work prove the efficacy of ML algorithms in collision data modeling and road safety analysis, these algorithms have several limitations. First, the performance of the ML algorithms highly depends on the learning procedure which contains functional mapping and parameter selection, or in other words, the selection of ML model is a critical step. Second, several performance measures introduced by the HSM, which are the most recommended performance measures, are based on the statistical parameters, such as the dispersion parameter of the NB model, which brings a limitation towards the use of ML algorithms in place of the statistical modeling. Third, although sensitivity analysis may increase the interpretability of ML algorithms and the impact of each explanatory variable on the outcome, these models are still difficult to interpret compared to the NB models and it may be a barrier towards their use. Lastly, the method used for dealing with the missing data can have an adverse effect on the performance of the models and one ML models results can be poor due to the lack of sufficient data.

With the findings of this study, it is positively evaluated that ML techniques can be used within the RSMP. However, many sections in HSM, including RSMP, are still based on a rigid implementation of traditional statistical modeling, such as, before-after studies to evaluate the effectiveness of the countermeasures, crash modification factors and their calculation, and more. For future studies, it is important to explore new ideas that can use the non-parametric models such as ML algorithms and look for ways to incorporate it with the parametric estimations in the HSM, such as the EB performance measure. In addition, it is recommended to conduct studies that will explore the use of ML algorithms in other steps of the RSMP, such as before-after studies that require collision data modeling.

References

AASHTO. 2010. Highway Safety Manual. 1st ed.. American Association of State Highway and Transportation Officials, Washington, D.C.

Abbas, K.A., 2004. Traffic safety assessment and development of predictive models for accidents on rural roads in Egypt. Accident Analysis and Prevention 36 (2), 149–163.

Abdel-Aty, M.A., Abdelwahab, H.T., 2004. Predicting injury severity levels in traffic crashes: a modelling comparison. Journal of Transportation Engineering 130(2), 204-210.

Abdel-Aty, M.A., Radwan, A.E., 2000. Modelling traffic crash occurrence and involvement. Accident Accident Analysis & Prevention 32(5), 633-642.

Abdelwahab, H.T., Abdel-Aty, M.A., 2002. Artificial Neural Networks and Logit Models for Traffic Safety Analysis of Toll Plazas. Transportation Research Record 1784, 115–125.

Aguero-Valverde, J., Jovanis, P.P., 2008. Analysis of road crash frequency with spatial models. Transportation Research Record 2061, 55–63.

Aguero-Valverde, J., Jovanis, P.P., 2009. Bayesian multivariate poisson log-normal models for crash severity modeling and site ranking. Paper Presented at the 88th Annual Meeting of the Transportation Research Board, Washington, DC.

Alikhani, M., Nedaie, A., Ahmadvand, A., 2013. Presentation of clustering-classification 537 heuristic method for improvement accuracy in classification of severity of road crashes in Iran. Safety Science 60, 142–150.

Amiri, A.M., Sadri, A., Nadimi, N., et al., 2020. A comparison between artificial neural network and hybrid intelligent genetic algorithm in predicting the severity of fixed object crashes among elderly drivers. Accident Analysis & Prevention 138, 1-10.

Amoros, E., Martin, J.L., Laumon, B., 2003. Comparison of road crashes incidence and severity between some French counties. Accident Analysis and Prevention 35 (4), 537–547.

Anastasopoulos, P.C., Mannering, F.L., 2009. A note on modeling vehicle accident frequencies with random-parameters count models. Accident Analysis and Prevention 41 (1), 153–159.

Aptel, I., Salmi, L.R., Masson, F., Bourdet, A., Henrion, G., Erny, P., 1999. Road accident statistics: discrepancies between police and hospital data on a French island. Accident Analysis and Prevention 31 (1), 101–108.

Ayodele, T. O., 2010. Types of Machine Learning Algorithms, New Advances in Machine Learning, Yagang Zhang, IntechOpen. Available from: https://www.intechopen. com/chapters/10694

Bijleveld, F.D., 2005. The covariance between the number of accidents and the number of victims in multivariate analysis of accident related outcomes. Accident Analysis and Prevention 37 (4), 591–600.

Bonneson, J.A., McCoy, P., 1993. Estimation of safety at two-way stop-controlled intersections on rural roads. Transportation Research Record 1401, 83–89.

Bonneson, J.A., Pratt, M.P., 2008. Procedure for developing accident modification factors from cross-sectional data. Transportation Research Record 2083, 40–48.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2, 121–167.

Cafiso, S., Di Silvestro, G., Persaud, B., Begum, M.A., 2010. Revisiting the variability of the dispersion parameter of safety performance functions using data for two-lane rural roads. 89th Annual Meeting of the Transportation Research Board, Washington, DC. (Preprint No. Paper 10-3572).

Caliendo, C., Guida, M., Parisi, A., 2007. A crash-prediction model for multilane roads. Accident Analysis and Prevention 39 (4), 657–670.

Cameron, A.C., Trivedi, P.K., 1998. Regression Analysis of Count Data. Cambridge University Press, Cambridge, UK.

Carson, J., Mannering, F., 2001. The effect of ice warning signs on accident frequencies and severities. Accident Analysis and Prevention 33 (1), 99–109.

Chang, L., 2005. Analysis of freeway crash frequencies: negative binomial regression versus artificial neural network. Safety Science 43, 541-557.

Chang, L., Chen, W., 2005. Data mining of tree-based models to analyze freeway crash frequency. Journal of Safety Research 36(4), 365-375.

Chang, L., Wang, H., 2006. Analysis of traffic injury severity: an application of non-parametric classification tree techniques. Accident Analysis & Prevention 38, 1019-1027.

Cheng, W. and Washington, S., (2008). New Criteria for Evaluating Methods of Identifying Hot Spots. Transportation Research Record: Journal of the Transportation Research Board, Transportation Research Board 2083, 76–85.

Chen H., Wang W., Zuylen H. V., 2009. Construct support vector machine ensemble to detect traffic incident, Expert Systems with Applications 36 (8), 10976-10986.

Çodur, M.Y., Tortum, A., 2015. An artificial neural network model for highway crash prediction: a case study of Erzurum, Turkey. PROMET-Traffic & Transportation 27(3), 217-225.

Daniels, S., Brijs, T., Nuyts, E., Wets, G., 2010. Explaining variation in safety performance of roundabouts. Accident Analysis and Prevention 42(2):393-402.

Das, A., Abdel-Aty, M., 2010. A genetic programming approach to explore the 564 crash severity on multi-lane roads. Accident Analysis & Prevention 42(2), 548-557.

Delen, D., Sharda, R., Bessonov, M., 2006. Identifying significant predictors of injury severity in traffic crashes using a series of artificial neural networks. Accident Analysis & Prevention 38(3), 434-444.

Devroye, L., Gyorfi, L., Krzyzak, A., et al., 1994. On the strong universal consistency of nearest neighbor regression function estimates. Annals of Statistics 22, 1371–1385.

Dong, N., Huang, H., and Zheng, L. 2015. Support vector machine in crash prediction at the level of traffic analysis zones: Assessing the spatial proximity effects. Accident Analysis & Prevention 82, 192–198.

El-Basyouny, K., Sayed, T., 2006. Comparison of two negative binomial regression techniques in developing accident prediction models. Transportation Research Record 1950, 9–16.

El-Basyouny, K., Sayed, T., 2009a. Collision prediction models using multivariate Poisson-lognormal regression. Accident Analysis and Prevention 41 (4), 820–828.

El-Basyouny, K., Sayed, T., 2009b. Accident prediction models with random corridor parameters. Accident Analysis and Prevention 41 (5), 1118–1123.

Elvik, R., Vaa, T., Høye, A., et al., 2009. The Handbook of Road Safety Measures. Emerald Group Publishing Limited, Bingley.

Fish, K.E., Blodgett, J.G., 2003.A visual method for determining variable importance in an artificial neural network model: an empirical benchmark study. J. Target. Meas. Anal. Market. 11 (3), 244–254.

Flahaut, B., Mouchart, M., San Martin, E., Thomas, I., 2003. The local spatial autocorrelation and the kernel method for identifying black zones: a comparative approach. Accident Analysis and Prevention 35 (6), 991–1004.

Geedipally, S.R., and Lord, D., 2010. Investigating the effect of modeling single-vehicle and multi-vehicle crashes separately on confidence intervals of Poisson-gamma models. Accident Analysis and Prevention 42.

Géron, A., 2017. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 1st ed., O'Reilly Media.

Global status report on road safety, 2018. World Health Organization.

Guo, F., Wang, X., Abdel-Aty, M., 2010. Modeling signalized intersection safety with corridor spatial correlations. Accident Analysis and Prevention 42 (1), 84–92.

Halekoh, U., Højsgaard, S., Yan, J., 2006. The R Package geepack for generalized estimating equations. Journal of Statistical Software 15 (2), 1–11.

Hauer, E., 1997. Observational Before–After Studies in Road Safety. Pergamon Press, Elsevier Science Ltd., Oxford, United Kingdom.

Hauer, E., 2004. Statistical road safety modelling. Transportation Research Record 1897, 81-87.

Hauer, E. 2015. The Art of Regression Modeling in Road Safety. Springer International Publishing, New York, NY.

Hauer, E., Hakkert, A.S., 1988. Extent and some implications of incomplete accident reporting. Transportation Research Record 1185, 1–10.

Hauer, E., Ng, J.C.N., Lovell, J., 1988. Estimation of safety at signalized intersections. Transportation Research Record 1185, 48–61.

Haykin, S., 2009. Neural Networks and Learning Machines, third ed. Prentice Hall, New York.

Hinde, J., and Demétrio C.G.B., 1998. Overdispersion: Models and estimation. Computational Statistics & Data Analysis 27 (2), 151-170.

Hirst, W.M., Mountain, L.J., Maher, M.J., 2004. Sources of error in road safety scheme evaluation: a method to deal with outdated accident prediction models. Accident Analysis and Prevention 36 (5), 717–727.

Huang, H., Zeng, Q., Pei, X., Wong, S.C., and Xu, P., 2016. Predicting crash frequency using an optimised radial basis function neural network model. Transportmetrica Transport Science 12, 330–345.

Iranitalab, A., Khattak, A., 2017. Comparison of four statistical and machine learning methods for crash severity prediction. Accident Analysis & Prevention 108, 27-36.

James, H.F., 1991. Under-reporting of road traffic accidents. Traffic Engineering and Control 32 (12), 574–583.

Johansson, P., 1996. Speed limitation and motorway casualties: a time series count data regression approach. Accident Analysis and Prevention 28 (1), 73–87.

Jones, B., Janssen, L., Mannering, F., 1991. Analysis of the frequency and duration of freeway accidents in Seattle. Accident Analysis and Prevention 23 (2), 239–255.

Joshua, S.C., Garber, N.J., 1990. Estimating truck accident rate and involvements using linear and Poisson regression models. Transportation Planning and Technology 15 (1), 41–58.

Jovanis, P.P., Chang, H.L., 1986. Modeling the relationship of accidents to miles traveled. Transportation Research Record 1068, 42–51.

Kashani, A.T., Mohaymany, A.S., 2011. Analysis of the traffic injury severity on two-lane, two-way rural roads based on classification tree models. Safety Science 49, 1314-1320.

Kidando, E., Moses, R., Ozguzen, E.E., et al., 2019. Incorporating travel time reliability 592 in predicting the likelihood of severe crashes on arterial highways using non-parametric random-

effect regression. Journal of Traffic and Transportation Engineering (English Edition) 6(5), 470-481.

Kim, D., Washington, S., 2006. The significance of endogeneity problems in crash models: an examination of left-turn lanes in intersection crash models. Accident Analysis and Prevention 38 (6), 1094–1100.

Kim, D.-G., Lee, Y., Washington, S., and Choi, K., 2007. Modeling crash outcome probabilities at rural intersections: application of hierarchical binomial logistic models. Accident Analysis & Prevention 39 (1), 125–134.

Kwon, O.H., Rhee, W., Yoon, Y., 2015. Application of classification algorithms for analysis of road safety risk factor dependencies. Accident Analysis & Prevention 75, 1-15.

Lan, B., & Persaud, B. (2011). Fully Bayesian approach to investigate and evaluate ranking criteria for black spot identification. Transportation research record, 2237(1), 117-125.

Lee, J., Mannering, F., 2002. Impact of roadside features on the frequency and severity of run-off-roadway accidents: an empirical analysis. Accident Analysis and Prevention 34 (2), 149–161.

Li, H., Graham, D.J., Majumdar, A., 2012. The effects of congestion charging on road traffic casualties: a causal analysis using difference-in-difference estimation. Accident Analysis & Prevention 49, 366-377.

Li, X., Lord, D., Zhang, Y., 2009. Development of accident modification factors for rural frontage road segments in Texas using results from generalized additive models. Working Paper, Zachry Department of Civil Engineering, Texas A&M University, College Station, TX.

Li, X., Lord, D., Zhang, Y., et al., 2008. Predicting motor vehicle crashes using support vector machine models. Accident Analysis & Prevention 40(4), 1611-1618.

Lord, D., 2006. Modeling motor vehicle crashes using Poisson-gamma models: examining the effects of low sample mean values and small sample size on the Estimation of the fixed dispersion parameter. Accident Analysis and Prevention 38 (4), 751–766.

Lord, D., Geedipally, S.R., Guikema, S., 2010. Extension of the application of Conway–Maxwell–Poisson models: analyzing traffic crash data exhibiting underdispersion. Risk Analysis (8):1268-76.

Lord, D., Guikema, S., Geedipally, S.R., 2008. Application of the Conway–Maxwell–Poisson generalized linear model for analyzing motor vehicle crashes. Accident Analysis and Prevention 40 (3), 1123–1134.

Lord, D., Mahlawat, M., 2009. Examining the application of aggregated and disaggregated Poisson-gamma models subjected to low sample mean bias. Transportation Research Record 2136, 1–10.

Lord, D., Manar, A., Vizioli, A., 2005a. Modeling crash-flow-density and crash-flow-v/c ratio for rural and urban freeway segments. Accident Analysis and Prevention 37 (1), 185–199.

Lord, D., Mannering, F., 2010. The statistical analysis of crash-frequency data: a review and assessment of methodological alternatives. Transportation Research Part A: Policy and Practice 44(5), 291–305.

Lord, D., Miranda-Moreno, L.F., 2008. Effects of low sample mean values and small sample size on the estimation of the fixed dispersion parameter of Poisson-gamma models for modeling motor vehicle crashes: a Bayesian perspective. Safety Science 46 (5), 751–770.

Lord, D., Persaud, B.N., 2000. Accident prediction models with and without trend: application of the generalized estimating equations procedure. Transportation Research Record 1717, 102–108.

Lord, D., Washington, S.P., Ivan, J.N., 2007. Further notes on the application of zero inflated models in highway safety. Accident Analysis and Prevention 39 (1), 53–57.

Ma, J., Kockelman, K.M., 2006. Bayesian multivariate Poisson regression for models of injury count by severity. Transportation Research Record 1950, 24–34.

Ma, J., Kockelman, K.M., Damien, P., 2008. A multivariate Poisson-lognormal regression model for prediction of crash counts by severity, using Bayesian methods. Accident Analysis and Prevention 40 (3), 964–975.

MacNab, Y.C., 2004. Bayesian spatial and ecological models for small-area crash and injury analysis. Accident Analysis and Prevention 36 (6), 1019–1028. Mahalel, D., 1986. A note on accident risk. Transportation Research Record 1068, 85–89.

Maher, M.J., Summersgill, I., 1996. A comprehensive methodology for the fitting predictive accident models. Accident Analysis and Prevention 28 (3), 281–296.

Malyshkina, N., Mannering, F., 2010. Zero-state Markov switching count-data models: an empirical assessment. Accident Analysis and Prevention 42 (1), 122–130.

Marsland, S., 2009. Machine learning: an algorithmic perspective, 2nd ed., New York, United States of America, CRC PRESS.

Miaou, S.-P., 1994. The relationship between truck accidents and geometric design of road sections: Poisson versus negative binomial regressions. Accident Analysis and Prevention 26 (4), 471–482.

Miaou, S.-P., Bligh, R.P., Lord, D., 2005. Developing median barrier installation guidelines: a benefit/cost analysis using Texas data. Transportation Research Record 1904, 3–19.

Miaou, S.-P., Lord, D., 2003. Modeling traffic crash-flow relationships for intersections: dispersion parameter, functional form, and Bayes versus Empirical Bayes. Transportation Research Record 1840, 31–40.

Miaou, S.-P., Lum, H., 1993. Modeling vehicle accidents and highway geometric design relationships. Accident Analysis and Prevention 25 (6), 689–709.

Miaou, S.-P., Song, J.J., 2005. Bayesian ranking of sites for engineering safety improvements: decision parameter, treatability concept, statistical criterion and spatial dependence. Accident Analysis and Prevention 37 (4), 699–720.

Mountain, L., Maher, M.J., Fawaz, B., 1998. The influence of trend on estimates of accidents at junctions. Accident Analysis and Prevention 30 (5), 641–649.

Müller, A.C., and Guido, S., 2016. Introduction to Machine Learning with Python: A Guide for Data Scientists 1st ed., United States of America, O'Reilly Media.

Mussone, L., Ferrari, A., and Oneta, M., 1999. An analysis of urban collisions using an artificial intelligence model. Accident Analysis & Prevention 31, 705–718.

Nodari, C.T., Lindau, L.A., 2007. Proactive method for safety evaluation of two-lane rural highway segments. Advances in Transportation Studies 11, 51-61.

Noland, R.B., Quddus, M.A., 2004. A spatially disaggregated analysis of road casualties in England. Accident Analysis and Prevention 36 (6), 973–984.

Oh, J., Washington, S.P., Nam, D., 2006. Accident prediction model for railway-highway interfaces. Accident Analysis and Prevention 38 (2), 346–356.

Olutayo,V.A, and Eludire, A.A, 2014. Traffic Accident Analysis Using Decision Trees and Neural Networks. International Journal of Information Technology and Computer Science 6, 22–28.

Oña, J., López, G., Mujalli, R.O., et al., 2013b. Analysis of traffic acidentes on rural highways using Latent Class Clustering and Bayesian Networks. Accident Analysis & Prevention 51, 1-10.

Oña, J., Mujalli, R.O., Calvo, F.J., 2011. Analysis of traffic crash injury severity on Spanish rural highways using Bayesian networks. Accident Analysis & Prevention 43(1), 402-411.

Park, E.-S., Lord, D., 2007. Multivariate Poisson-lognormal models for jointly modeling crash frequency by severity. Transportation Research Record 2019, 1–6.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

Persaud, B.P., 1994. Accident prediction models for rural roads. Canadian Journal of Civil Engineering 21 (4), 547–554.

Persaud, B.P., Nguyen, T., 1998. Disaggregate safety performance models for signalized intersections on Ontario provincial roads. Transportation Research Record 1635, 113–120.

Qin, X., Ivan, J.N., Ravishankar, N., 2004. Selecting exposure measures in crash rate prediction for two-lane highway segments. Accident Analysis and Prevention 36 (2), 183–191.

Raybaut, P. (2009). Spyder-documentation. Available Online at: Spyder-IDE.org.

Rolison, J.J., Regev, S., Moutari, S., et al., 2018. What are the factors that contribute to road crashes? An assessment of lawenforcement views, ordinary drivers' opinions, and road crash records. Accident Analysis & Prevention 115, 11-24.

Sacchi, E., Sayed, T., & El-Basyouny, K. (2015). Multivariate full Bayesian hot spot identification and ranking: New technique. Transportation research record, 2515(1), 1-9.

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513, USA.

Savolainen, P., Mannering, F., Lord, D., et al., 2011. The statistical analysis of crash-injury severities: a review and assessment of methodological alternatives. Accident Analysis & Prevention 43(5), 1666-1676.

Sawalha, Z. and Sayed, T., 2001. Evaluating Safety of Urban Arterial Roadways. Journal of Transportation Engineering-ASCE – Transportation Engineering, American Society of Civil Engineers.

Sawalha, Z. and Sayed, T., 2006. Traffic accident modeling: some statistical issues. Canadian Journal of Civil Engineering. 33(9), 1115-1124.

Schölkopf, B., Smola, A.J., Williamson, R.C., and Bartlett, P.L., 2000. New Support Vector Algorithms. Neural Compute 12 (5), 1207–1245.

Schölkopf, B., Smola, AJ., 2002. Learning with kernels: support vector machines, regularization, optimization, and beyond. Cambridge, MA, USA:MIT Press.

Shankar, V.N., Albin, R.B., Milton, J.C., Mannering, F.L., 1998. Evaluating median cross-over likelihoods with clustered accident counts: an empirical inquiry using random effects negative binomial model. Transportation Research Record 1635, 44–48.

Silva, P. B., Andrade, M., Ferreira, S., 2020. Machine learning applied to road safety modeling: A systematic literature review, Journal of Traffic and Transportation Engineering (English Edition) 7 (6), 775-790.

Singh, G., Sachdeva, S. N., and Pal, M., 2018. Support vector machine model for prediction of accidents on non-urban sections of highways. Proceedings of the Institution of Civil Engineers - Transport, 171(5), 253–263.

Sittikariya, S., Shankar, V., 2009. Modeling Heterogeneity: Traffic Accidents, vol. 80. VDM-Verlag.

Smola, A.J., Scholkopf, B., 2004. A tutorial on support vector regression. Statistics and Computing 14, 199-222.

Sohn, S., Lee, S., 2003. Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic crash in Korea. Safety Science 41(1), 1–14.

Song, J.J., Ghosh, M., Miaou, S., Mallick, B., 2006. Bayesian multivariate spatial models for roadway traffic crash mapping. Journal of Multivariate Analysis 97 (1), 246–273.

Suykens, J.A.K., Gestel, T.V., Brabanter, J.D., Moor, B.D., and Vanderwalle, J., 2002. Least Squares Support Vector Machines. World Scientific Publishing Co. Pte. Ltd., Singapore.

Trafalis T. B. , Robin C. Gilbert, 2006. Robust classification and regression using support vector machines, European Journal of Operational Research 173 (3), 893-909.

Ulfarsson, G.F., Shankar, V.N., 2003. An accident count model based on multi-year cross-sectional roadway data with serial correlation. Transportation Research Record 1840, 193–197.

Üstün, B., Melssena, W.J., Oudenhuijzenb, M., et al., 2005. Determination of optimal 648 support vector regression parameters by genetic algorithms and simplex optimization. Analytica Chimica Acta 544(1–2), 292–305.

Vapnik, V. 1995. Support vector machine. Machine Learning 20, 273–297.

Villiers, J., Barnard, E., 1993. Back propagation neural nets with one and two hidden layers. IEEE Transactions on Neural Networks 4(1), 136–141.

Wahab, L., Jiang, H., 2019. A comparative study on machine learning based algorithms for prediction of motorcycle crash severity. PLoS One 14(4), 1-17.

Wang, C., Quddus, M.A., Ison, S., 2009. The effects of area-wide road speed and curvature on traffic casualties in England. Journal of Transport Geography 17 (5), 385–395.

Wang, C., Quddus, M.A., Ison, S.G., 2013. The effect of traffic and road characteristics on road safety: a reviewand future research direction. Safety Science 57, 264-275.

Wang, X., Abdel-Aty, M., 2006. Temporal and spatial analyses of rear-end crashes at signalized intersections. Accident Analysis and Prevention 38 (6), 1137– 1150.

Washington, S.P., Karlaftis, M.G., Mannering, F.L., 2003. Statistical and Econometric Methods for Transportation Data Analysis. Chapman Hall/CRC, Boca Raton, FL.

Washington, S.P., Karlaftis, M.G., Mannering, F.L., 2010. Statistical and Econometric Methods for Transportation Data Analysis, second ed. Chapman Hall/ CRC, Boca Raton, FL.

Xie, Y., Lord, D., Zhang, Y., 2007. Predicting motor vehicle collisions using Bayesian neural networks: an empirical analysis. Accident Analysis & Prevention 39(5), 922-933.

Xie, Y., Zhang, Y., 2008. Crash frequency analysis with generalized additive models. Transportation Research Record 2061, 39–45.

Zeng, Q., Huang, H., 2014. A stable and optimized neural network model for crash injury severity prediction. Accident Analysis & Prevention 73, 351–358.

Zeng, Q., Huang, H., Pei, X., et al., 2016a. Rule extraction from an optimized neural network for traffic crash frequency modelling. Accident Analysis & Prevention 97, 87–95.

Zeng, Q., Huang, H., Pei, X., et al., 2016b. Modelling nonlinear relationship between crash frequency by severity and contributing factors by neural networks. Analytic Methods in Crash Research 10, 12–25.

Zhang, J., Li, Z., Pu, Z., et al., 2018. Comparing Prediction Performance for Crash Injury Severity Among Various Machine Learning and Statistical Methods. IEEE Access 6, 60079-60087.

Zhang, Y., and Xie, Y., 2007. Forecasting of Short-Term Freeway Volume with v-Support Vector Machines. Transportation Research Record: Journal of the Transportation Research Board, No. 2024, National Research Council, Washington, DC, 92–99.

Appendix A

A.1 Performance Measures

The Highway Safety Manual (HSM), offers variety of performance measures that can be used within the network screening process. The selection of performance measure in a network screening project is dependent upon data availability. While more data are required for the some of the performance measures, HSM also offers more simple methods in case proper data were not available. To achieve the objectives of this research study, three performance measures are selected:

1. Excess predicted average collision frequency using SPFs
2. Expected average collision frequency with EB Adjustment, and
3. Excess expected average collision frequency with EB Adjustment – also known as Potential for Safety Improvement (PSI)

Tables A-1 and A-2 present the performance measures offered by HSM, and provide useful information on whether they account for RTM or not, whether a performance threshold is being calculated or not, and the data needs for using each of them (HSM, 2010).

Table A-1 Performance Measures (Adopted from HSM, 2010)

Performance Measure	Accounts for RTM Bias	Method Estimates a Performance Threshold
Average Crash Frequency	No	No
Crash Rate	No	No
Equivalent Property Damage Only (EPDO) Average Crash Frequency	No	No

Relative Severity Index	No	Yes
Critical Rate	Considers data variance but does not account for RTM bias	Yes
Excess Predicted Average Crash Frequency Using Method of Moments	Considers data variance but does not account for RTM bias	Yes
Level of Service of Safety	Considers data variance but does not account for RTM bias	Expected average crash frequency plus/minus 1.5 standard deviations
Excess Predicted Average Crash Frequency using Safety Performance Functions (SPFs)	No	Predicted average crash frequency at the site
Probability of Specific Crash Types Exceeding Threshold Proportion	Considers data variance; not effected by RTM Bias	Yes
Excess Proportion of Specific Crash Types	Considers data variance; not effected by RTM Bias	Yes
Expected Average Crash Frequency with EB Adjustment	Yes	Expected average crash frequency at the site
Equivalent Property Damage Only (EPDO) Average Crash Frequency with EB Adjustment	Yes	Expected average crash frequency at the site
Excess Expected Average Crash Frequency with EB Adjustment	Yes	Expected average crash frequency per year at the site

Table A-2 Summary of Data Needs for Performance Measures (From HSM, 2010)

Performance Measure	Crash Data	Roadway Information for Categorization	Traffic Volume[1]	Calibrated Safety Performance Function and Overdispersion Parameter	Other
Average Crash Frequency	X	X			
Crash Rate	X	X	X		
Equivalent Property Damage Only (EPDO) Average Crash Frequency	X	X			EPDO Weighting Factors
Relative Severity Index	X	X			Relative Severity Indices
Critical Rate	X	X	X		
Excess Predicted Average Crash Frequency Using Method of Moments	X	X	X		
Level of Service of Safety	X	X	X	X	
Excess Predicted Average Crash Frequency using Safety Performance Functions (SPFs)	X	X	X	X	
Probability of Specific Crash Types Exceeding Threshold Proportion	X	X			

Excess Proportion of Specific Crash Types	X	X			
Expected Average Crash Frequency with EB Adjustment	X	X	X	X	
Equivalent Property Damage Only (EPDO) Average Crash Frequency with EB Adjustment	X	X	X	X	EPDO Weighting Factors
Excess Expected Average Crash Frequency with EB Adjustment	X	X	X	X	

[1]Average Annual Daily Traffic (AADT), Average Daily Traffic (ADT), or peak hour volumes.

[2]Traffic volume is needed to apply Method of Moments to establish the reference populations based on ranges of traffic volumes as well as site geometric characteristics.

A.2 Appendix References

AASHTO, 2010. Highway Safety Manual. 1st ed. American Association of State Highway and Transportation Officials, Washington, D.C.

Appendix B

As explained in section 3.2.4, Grid-Search Cross-Validation (GSCV) method is used to optimize the selection of the hyper-parameters in the ML algorithms. In the following, a brief description of each hyper-parameter is provided . The utilized arrays of hyper-parameters values and the selected value is presented in table 4-3.

B.1 Hyper-parameters of SVM:

C: this hyper-parameter is used as a penalty constant for the complexity of the model.

Kernel type (linear, polynomial, or RBF): The choice of kernel function

Degree: is the degree of polynomial kernel function (if selected by the model)

Gamma: a coefficient to be used in "rbf", "polynomial" and some other types of kernel function not used in this study, such as "sigmoid".

Epsilon (ε): The error margin allowed for finding the decision boundaries of the SVM separative line (see figure 3-1)

B.2 Hyper-parameters of DT:

Mean_samples_leaf: The minimum number of samples required to split an internal node. If not met, the node will remain as the last node.

Max_depth: The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

Min_impurity_decrease: A node will be split if this split induces a decrease of the impurity greater than or equal to this value

B.3 Hyper-parameters of RF:

Mean_samples_leaf: The minimum number of samples required to split an internal node. If not met, the node will remain as the last node.

Max_depth: The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

Bootstrap (True, False): Whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.

It should be noted that the model hyperparameters of SVM, DT and RF are not limited to the ones used in this study. Although optimizing for all the hyperparameters will sometimes make the models more accurate, due to insignificance of the change in the results comparing to the additional processing time that was being added, only the hyperparameters introduced are optimized in this study.

B.4 Appendix References:

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3), 1–27.

Platt, John, (2000). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. Adv. Large Margin Classifiers 10.

Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.J. (1984). Classification And Regression Trees (1st ed.). Routledge.

Breiman, L. (2001). Random Forests. Machine Learning 45, 5–32.

Appendix C

A sensitivity analysis is conducted in two time periods of the study using the data in the test set. In table C – 0, the test intersections are assigned with the site IDs. To simplify the reporting in the consecutive tables, only the corresponding site ID is used.

Table C – 0 Test data – intersections of the test set

Site ID	City	Major Approach	Minor Approach
1	Saskatoon	Sherwood Dr	Dorothy St
2	Regina	Quebec Ave	36th St E
3	Regina	14th St E	Arlington Ave
4	Regina	Saskatchewan Dr	Broad St
5	Regina	Lewvan Dr	Regina Ave
6	Regina	Truesdale Dr	Prince of Wales Dr
7	Saskatoon	8th St E	Broadway Ave
8	Regina	Gordon Rd	Lockwood Rd
9	Saskatoon	Rochdale Blvd	Lakeridge Rd
10	Saskatoon	E Arens Rd	University Park Dr
11	Saskatoon	Assinboine Ave E (Wascana Gate N)	Prince of Wales Dr
12	Saskatoon	Wascana Gate E	Wascana View Dr
13	Saskatoon	Main St	Preston Ave
14	Saskatoon	4 Ave	Albert St (CanAm Hwy)
15	Saskatoon	Louise St	Arlington Ave
16	Saskatoon	Woodhams Dr	Renfrew Crescent (Buckingham Dr)
17	Regina	14th St E	Acadia Dr
18	Regina	Saskatchewan Dr	Elphinstone St

19	Saskatoon	Ave W N	Richardson Rd
20	Saskatoon	College Ave	Albert St (CanAm Hwy)
21	Saskatoon	6 Ave N	Broad St
22	Saskatoon	Broadway Ave	Winnipeg St
23	Regina	Victoria Ave	Elphinstone St
24	Saskatoon	Confederation Dr	Laurier Dr
25	Regina	Ross Ave E	Park St
26	Saskatoon	12 Ave	Winnipeg St
27	Regina	8th St E	Cumberland Ave S
28	Saskatoon	Lenore Dr	Primrose Dr
29	Regina	Hill Ave	Montague St
30	Regina	Taylor St E	Louise Ave
31	Regina	19th St E	2nd Ave S
32	Saskatoon	1st Ave S	20th St E
33	Saskatoon	Ruth St E	Cumberland Ave S
34	Saskatoon	8th St E / W	Lorne Ave
35	Regina	Kingsmere Blvd	Waterbury Rd / Brightsand Way
36	Regina	Central Ave	Somers Rd
37	Regina	Circle Dr E	Millar Ave / Venture Cres
38	Regina	College Ave	Winnipeg St
39	Regina	Rochdale Blvd	Pasqua St (N)
40	Regina	33rd St E	Quebec Ave
41	Regina	7th Ave	Pasqua St
42	Regina	Montague St	Regina Ave
43	Regina	22nd St E	1st Ave S / N
44	Saskatoon	Toronto St	Ross Ave

45	Saskatoon	Wilson Cres	Cumberland Ave S
46	Saskatoon	Preston Ave S	Hunter Rd / Cornish Rd
47	Saskatoon	University Park Dr	Quance St
48	Regina	Albert St (CanAm Hwy)	Parliament Ave
49	Regina	33rd St W	Northumberland Ave / Catherwood Ave
50	Regina	Wascana Pkwy	23 Ave (Lakeshore Dr)
51	Regina	Victoria Ave	Broad St
52	Saskatoon	45th St W / Airport Rd	Airport Dr
53	Regina	Lorne Ave	Ruth St E / W
54	Saskatoon	Lewvan Dr	Gordon Rd
55	Regina	Marquis Dr E	Faithful Ave
56	Regina	Mikkelson Dr (2nd Ave)	Campbell St
57	Regina	Arcola Ave	Wascana View Dr
58	Regina	13th Ave	Pasqua St
59	Regina	9th Ave N	Broad St
60	Regina	Confederation Dr	Massey Dr
61	Saskatoon	Kerr Rd	Cowley Rd / Chotem Cres
62	Regina	Albert St (CanAm Hwy)	23 Ave
63	Saskatoon	4th Ave S	19th St E
64	Regina	Central Ave	105th St E / W
65	Regina	Gordon Rd	Harvard Way
66	Regina	Assinboine Ave E	Edinburgh Dr
67	Regina	Prince of Wales Dr	Haughton Rd
68	Regina	Green Ridge Gate	Woodland Grove Dr

C.1 Results of Sensitivity Analysis in Time Period I

Table C-1 Results of sensitivity analysis of SVM algorithm in time period I

Site ID	V_1	V_2	CT	Nlegs
1	0.93	0.081	1.1415	0.8760
2	0.4798	N/A*	0.9384	1.0656
3	0.3816	0.4202	1.0932	1.0932
4	0.5571	0.4776	0.7480	0.7480
5	0.5896	0.474	1.0571	1.0571
6	0.4011	0.2304	0.8875	0.8875
7	0.4353	0.4639	0.9436	0.9436
8	0.4202	0.4228	0.9146	0.9146
9	0.458	0.2598	0.8724	0.8724
10	0.3242	0.3967	0.8583	0.8583
11	1.2571	N/A	1.3887	0.7201
12	N/A	0.5669	-0.1425	-0.1425
13	0.9512	N/A	1.0463	0.9557
14	0.5396	0.6177	1.1807	1.1807
15	1.0128	N/A	1.2556	0.7964
16	0.6904	N/A	0.8295	1.2056
17	0.2564	0.5421	1.1206	1.1206
18	0.4283	0.5163	0.9736	0.9736
19	0.2569	0.627	1.1123	1.1123
20	0.5491	0.0552	0.7099	0.7099
21	0.4233	0.3724	0.8951	0.8951
22	0.3795	0.3486	0.8749	0.8749

23	0.2967	0.3259	0.8375	0.8375
24	0.4001	0.5596	0.9830	0.9830
25	0.3774	0.6287	1.0048	1.0048
26	0.4168	0.2436	0.8816	0.8816
27	0.4286	0.4952	0.9596	0.9596
28	0.3027	0.4859	0.8726	0.8726
29	0.7879	N/A	1.0710	0.9337
30	0.1256	0.0868	1.6097	1.6097
31	0.3077	0.184	0.9301	0.9301
32	0.2377	0.3744	0.8007	0.8007
33	0.345	N/A	1.6595	0.6026
34	0.3853	0.309	0.8731	0.8731
35	0.3191	N/A	0.9083	1.1010
36	0.4767	0.3487	1.0558	1.0558
37	0.6634	0.4485	1.1324	1.1324
38	0.3732	0.401	0.8851	0.8851
39	0.5344	0.6314	1.1650	1.1650
40	0.3345	0.2567	0.8702	0.8702
41	0.6837	N/A	1.1841	0.8445
42	0.2554	0.3718	0.8124	0.8124
43	0.3406	0.3884	0.8654	0.8654
44	0.7111	0.1506	1.0451	1.0451
45	0.3898	N/A	1.5318	0.6528
46	0.4705	0.2477	0.8666	0.8666
47	0.3597	0.6144	0.9808	0.9808
48	0.4538	0.6532	1.0980	1.0980

49	0.4311	0.1771	0.8965	0.8965
50	0.8557	N/A	0.9468	1.0561
51	0.5231	0.6234	1.0907	1.0907
52	0.3148	0.2468	0.8696	0.8696
53	0.227	0.3296	0.7970	0.7970
54	0.4316	0.6662	1.0060	1.0060
55	N/A	N/A	0.7788	1.2840
56	0.7757	N/A	1.2917	0.7742
57	0.3816	0.3538	0.8766	0.8766
58	0.8648	N/A	0.9564	1.0456
59	0.3059	0.5162	0.8875	0.8875
60	1.389	0.0234	0.9450	0.9450
61	0.275	0.4649	0.9520	1.0504
62	0.6309	0.5292	1.1003	1.1003
63	0.2383	0.5292	1.0639	1.0639
64	0.3149	0.173	0.9474	0.9474
65	0.5161	0.2627	0.8565	0.8565
66	0.9049	N/A	1.1998	0.8335
67	0.7786	0.1322	1.0446	1.0446
68	0.3285	0.4358	1.1170	1.1170

* N/A: faulty results due to falling beyond the training domain

Table C-2 Results of sensitivity analysis of DT algorithm in time period I

Site ID	V_1	V_2	CT	Nlegs
1	0.3351	0.1647	1	1
2	N/A	N/A	1	1
3	N/A	0.1647	1	1
4	0.0736	N/A	1	1
5	0.7967	0.1647	1	1
6	0.6488	N/A	1	1
7	0.9259	0.1647	1	1
8	0.864	N/A	1	1
9	0.6207	0.1647	1	1
10	0.7098	N/A	1	1
11	0.3351	0.1647	1	1
12	N/A	N/A	1	1
13	0.6488	0.1647	1	1
14	N/A	N/A	1	1
15	0.1795	0.1647	1	1
16	0.2993	N/A	1	1
17	N/A	0.1647	1	1
18	N/A	N/A	1	1
19	N/A	0.1647	1	1
20	0.6449	N/A	1	1
21	0.6207	0.1647	1	1
22	0.6488	N/A	1	1
23	0.5626	0.1647	1	1
24	N/A	N/A	1	1

25	N/A	0.1647	1	1
26	0.6488	N/A	1	1
27	N/A	0.1647	1	1
28	N/A	N/A	1	1
29	0.1158	0.1647	1	1
30	N/A	N/A	1	1
31	0.464	0.1647	1	1
32	0.3797	N/A	1	1
33	N/A	0.1647	1	1
34	0.6488	N/A	1	1
35	N/A	0.1647	1	1
36	N/A	N/A	1	1
37	0.5667	0.1647	1	1
38	0.7352	N/A	1	1
39	0.0736	0.1647	1	1
40	0.5626	N/A	1	1
41	N/A	0.1647	1	1
42	0.5258	N/A	1	1
43	0.7098	0.1647	1	1
44	0.6207	N/A	1	1
45	N/A	0.1647	1	1
46	0.6207	N/A	1	1
47	N/A	0.1647	1	1
48	N/A	N/A	1	1
49	0.4003	0.1647	1	1
50	0.6488	N/A	1	1

51	0.067	0.1647	1	1
52	0.5626	N/A	1	1
53	0.1795	0.1647	1	1
54	N/A	N/A	1	1
55	N/A	0.1647	1	1
56	N/A	N/A	1	1
57	0.6488	0.1647	1	1
58	0.6488	N/A	1	1
59	N/A	0.1647	1	1
60	0.4003	N/A	1	1
61	N/A	0.1647	1	1
62	0.804	N/A	1	1
63	0.5258	0.1647	1	1
64	0.2993	N/A	1	1
65	0.5274	0.1647	1	1
66	N/A	N/A	1	1
67	0.4003	0.1647	1	1
68	0.3351	N/A	1	1

Table C-3 Results of sensitivity analysis of RF algorithm in time period I

Site ID	V_1	V_2	CT	Nlegs
1	0.596	0.1203	1	1
2	0	0.0274	1	1
3	0.2587	0.1203	1	1
4	0.2262	0.0274	1	1
5	0.5036	0.1203	1	1
6	0.833	0.0274	1	1
7	0.434	0.1203	1	1
8	0.5644	0.0274	1	1
9	0.649	0.1203	1	1
10	0.4636	0.0274	1	1
11	0.3389	0.1203	1	1
12	N/A	0.0274	1	1
13	0.9168	0.1203	1	1
14	0.1804	0.0274	1	1
15	0.4769	0.1203	1	1
16	0.6151	0.0274	1	1
17	0.2481	0.1203	1	1
18	0.13	0.0274	1	1
19	N/A	0.1203	1	1
20	0.7135	0.0274	1	1
21	0.7655	0.1203	1	1
22	0.9071	0.0274	1	1
23	0.786	0.1203	1	1
24	0.0306	0.0274	1	1

25	N/A	0.1203	1	1
26	0.7813	0.0274	1	1
27	0.2479	0.1203	1	1
28	0.0957	0.0274	1	1
29	0.3865	0.1203	1	1
30	N/A	0.0274	1	1
31	0.6327	0.1203	1	1
32	0.2391	0.0274	1	1
33	N/A	0.1203	1	1
34	0.9125	0.0274	1	1
35	N/A	0.1203	1	1
36	0.2511	0.0274	1	1
37	0.3645	0.1203	1	1
38	0.4994	0.0274	1	1
39	0.2212	0.1203	1	1
40	0.837	0.0274	1	1
41	0.1674	0.1203	1	1
42	0.28	0.0274	1	1
43	0.4561	0.1203	1	1
44	0.6763	0.0274	1	1
45	N/A	0.1203	1	1
46	0.6471	0.0274	1	1
47	N/A	0.1203	1	1
48	N/A	0.0274	1	1
49	0.6552	0.1203	1	1
50	0.8251	0.0274	1	1

51	0.2115	0.1203	1	1
52	0.7702	0.0274	1	1
53	0.3547	0.1203	1	1
54	N/A	0.0274	1	1
55	0.3148	0.1203	1	1
56	0.1718	0.0274	1	1
57	0.7884	0.1203	1	1
58	0.8148	0.0274	1	1
59	0.0958	0.1203	1	1
60	0.6091	0.0274	1	1
61	N/A	0.1203	1	1
62	0.4794	0.0274	1	1
63	0.2986	0.1203	1	1
64	0.6198	0.0274	1	1
65	0.6264	0.1203	1	1
66	0.384	0.0274	1	1
67	0.7051	0.1203	1	1
68	0.3307	0.0274	1	1

C.2 Results of Sensitivity Analysis in Time Period II

Table C-4 Results of sensitivity analysis of SVM algorithm in time period II

Site ID	V_1	V_2	CT	Nlegs
1	0.1801	0.6868	1.1099	1.5946
2	1.0878	1.0065	1.5578	N/A*
3	0.3035	1.1594	1.1851	1.6287
4	0.118	0.5086	1.0137	1.0482
5	0.2747	0.2826	1.0288	1.1049
6	0.2204	0.3784	1.0604	1.2395
7	0.2162	0.3583	1.0356	1.1319
8	0.2401	0.3438	1.0523	1.2028
9	0.3882	0.209	1.0543	1.2111
10	0.1914	0.4256	1.0626	1.2499
11	0.1265	0.6968	1.0702	1.3124
12	1.8579	N/A	4.4628	12.7598
13	0.2997	0.3547	1.0688	1.3055
14	0.1218	0.5211	1.0280	1.1021
15	0.1474	0.6561	1.0771	1.3540
16	0.3688	0.4652	1.1539	2.0951
17	0.2019	1.2358	1.1295	1.4395
18	0.1596	0.4494	1.0351	1.1303
19	0.2128	1.0557	1.1991	1.6760
20	0.9378	0.0378	1.0631	1.2504
21	0.2031	0.3897	1.0472	1.1806
22	0.1474	0.5122	1.0634	1.2538

23	0.1629	0.4832	1.0666	1.2689
24	0.1537	0.4531	1.0292	1.1065
25	0.1211	0.5219	1.0275	1.1002
26	0.1914	0.427	1.0637	1.2553
27	0.2	0.3789	1.0332	1.1224
28	0.1298	0.5175	1.0385	1.1441
29	0.1318	0.7833	1.0936	1.4649
30	0.1352	0.9831	1.3022	4.7200
31	0.2836	0.3337	1.0971	1.4288
32	0.1197	0.5796	1.0623	1.2489
33	0.1215	1.0008	1.3110	N/A
34	0.256	0.336	1.0646	1.2591
35	N/A	N/A	-0.2745	N/A
36	1.1538	1.1936	1.4654	2.5817
37	0.3592	0.2045	1.0211	1.0750
38	0.1809	0.4222	1.0443	1.1684
39	0.1543	0.4337	1.0150	1.0530
40	0.2252	0.3789	1.0676	1.2735
41	0.2391	0.9975	1.2062	3.3322
42	0.1063	0.6265	1.0666	1.2696
43	0.1635	0.4549	1.0453	1.1727
44	1.2794	0.3101	1.1088	1.3704
45	0.1401	1.4305	1.2963	N/A
46	0.3544	0.2323	1.0531	1.2059
47	0.1587	0.4427	1.0285	1.1040
48	0.1677	0.4194	1.0229	1.0821

49	0.6027	0.1281	1.0920	1.3990
50	0.344	0.2609	1.0858	1.4116
51	0.1446	0.4563	1.0179	1.0637
52	0.1508	0.4965	1.0573	1.2257
53	0.1152	0.621	1.0795	1.3337
54	0.1144	0.5253	1.0190	1.0676
55	0.6543	1.0011	1.3842	1.9412
56	0.1569	0.891	1.1329	1.8208
57	0.284	0.2904	1.0482	1.1847
58	0.3173	0.3348	1.0684	1.3032
59	0.1073	0.577	1.0393	1.1476
60	1.0774	0.0211	1.1961	1.6686
61	N/A	N/A	10.3463	N/A
62	0.2621	0.3011	1.0346	1.1279
63	0.1424	1.1341	1.0601	1.2040
64	0.2374	0.3969	1.1023	1.4595
65	0.4212	0.1897	1.0564	1.2207
66	0.3918	0.3167	1.0959	1.4843
67	1.0415	0.3846	1.1155	1.3933
68	0.3106	2.12	1.2134	1.7248

* N/A: faulty results due to falling beyond the training domain

Table C-5 Results of sensitivity analysis of DT algorithm in time period II

Site ID	V_1	V_2	CT	Nlegs
1	0.5896	-0.182	1.1064	1*
2	0.1533	N/S**	1.5456	1
3	0.4805	N/S	3.1752	1
4	0.2181	0.5553	1.0000	1
5	0.3449	0.2111	1.0000	1
6	N/S	0.1163	2.2728	1
7	0.1422	0.3614	3.1799	1
8	0.055	0.3134	3.1799	1
9	0.1725	0.1	3.1799	1
10	N/S	0.141	2.2728	1
11	0.6563	-0.246	2.2728	1
12	N/S	N/S	3.9394	1
13	0.6483	-0.182	2.2728	1
14	N/S	0.4921	1.0000	1
15	0.6556	-0.246	2.2728	1
16	0.8778	N/S	0.7195	1
17	0.4805	N/S	3.1752	1
18	0.1027	0.3458	3.1799	1
19	0.1533	N/S	3.1752	1
20	0.4457	N/S	1.0000	1
21	0.055	0.3489	3.1799	1
22	N/S	0.3697	2.2728	1
23	N/S	0.3697	2.2728	1
24	N/S	0.452	1.0000	1

25	N/S	0.4921	1.0000	1
26	N/S	0.1163	2.2728	1
27	0.1725	0.3614	3.1799	1
28	N/S	0.1555	2.2728	1
29	0.4805	N/S	3.1752	1
30	N/S	N/S	3.9394	1
31	0.073	N/S	1.7272	1
32	N/S	0.4116	2.2728	1
33	N/S	N/S	3.9394	1
34	N/S	0.084	2.2728	1
35	N/S	N/S	3.9394	1
36	0.4805	N/S	1.5456	1
37	0.3659	0.2111	1.0000	1
38	N/S	0.157	2.2728	1
39	0.2645	0.6082	1.0000	1
40	N/S	0.1163	2.2728	1
41	0.2861	N/S	1.5456	1
42	N/S	0.4116	2.2728	1
43	N/S	0.1626	2.2728	1
44	0.8835	N/S	1.1232	1
45	N/S	N/S	3.9394	1
46	0.1422	0.1867	3.1799	1
47	N/S	0.452	1.0000	1
48	0.0844	0.525	1.0000	1
49	0.2546	N/S	1.1232	1
50	0.6483	-0.07	1.7272	1

51	0.1575	0.6082	1.0000	1
52	N/S	0.157	2.2728	1
53	N/S	0.3049	2.2728	1
54	N/S	0.5614	1.0000	1
55	N/S	N/S	1.5456	1
56	0.3962	N/S	1.5456	1
57	0.1422	0.3134	3.1799	1
58	0.6397	-0.182	2.2728	1
59	N/S	0.1555	2.2728	1
60	0.7254	N/S	1.1232	1
61	N/S	N/S	3.9394	1
62	0.2485	0.203	1.0000	1
63	0.7017	-0.244	2.2728	1
64	N/S	N/S	1.1064	1
65	0.2637	N/S	1.7272	1
66	0.9031	N/S	1.1232	1
67	0.8835	N/S	1.1232	1
68	0.3962	N/S	1.5456	1

* when 1, model shows insensitivity

** N/S: not sensitive

Table C-6 Results of sensitivity analysis of RF algorithm in time period II

Site ID	V_1	V_2	CT	Nlegs
1	0.5579	-0.021	1.1689	1.3233
2	0.1306	N/S*	1.3175	1.0596
3	0.4417	0.0029	1.3553	1.1548
4	0.1877	0.5451	1.0000**	1.0000
5	0.4695	0.2017	1.0000	1.0000
6	0.3185	0.0841	1.2920	1.1600
7	0.3306	0.2415	1.1033	1.0182
8	0.3327	0.1021	1.1475	1.0517
9	0.3327	0.0354	1.1425	1.0545
10	0.4749	0.1009	1.3832	1.1639
11	0.3194	-0.029	1.4616	1.3414
12	N/S	N/S	2.3039	1.0742
13	0.7889	-0.017	1.3042	1.2321
14	0.0075	0.4706	1.0000	1.0000
15	0.6717	-0.03	1.4599	1.3610
16	0.698	N/S	1.0736	1.2819
17	0.4442	-0.022	1.7593	1.2006
18	0.3028	0.2649	1.1033	1.0182
19	0.135	0.0198	1.6469	1.0565
20	0.5329	N/S	1.0000	1.0109
21	0.3183	0.2139	1.1356	1.0166
22	0.2243	0.1356	1.3556	1.1559
23	0.3129	0.1105	1.4280	1.1592
24	0.015	0.3871	1.0000	1.0000

25	0.0075	0.4899	1.0000	1.0000
26	0.3339	0.0992	1.3832	1.1639
27	0.3308	0.2605	1.1033	1.0182
28	0.2533	0.2073	1.2692	1.1352
29	0.4501	-0.016	1.6522	1.1537
30	N/S	N/S	2.2138	1.0322
31	0.4375	N/S	1.0822	1.1992
32	0.1892	0.1563	1.4504	1.1629
33	N/S	N/S	2.2138	1.0742
34	0.3255	0.072	1.3129	1.1552
35	N/S	N/S	2.2138	1.0742
36	0.4512	N/S	1.1620	1.1915
37	0.4383	0.1899	1.0000	1.0000
38	0.255	0.2162	1.2617	1.0816
39	0.2307	0.5517	1.0000	1.0000
40	0.325	0.0778	1.3700	1.1502
41	0.2439	N/S	1.3175	1.0596
42	0.1572	0.1769	1.5111	1.1770
43	0.2547	0.1863	1.2692	1.1352
44	0.8135	N/S	1.0841	1.0680
45	N/S	N/S	2.2138	1.0742
46	0.3007	0.0518	1.1425	1.0545
47	0.0194	0.3966	1.0000	1.0000
48	0.0841	0.4965	1.0000	1.0000
49	0.4791	N/S	1.0787	1.0627
50	0.8042	-0.0004	1.0650	1.2308

51	0.1386	0.5772	1.0000	1.0000
52	0.2354	0.1431	1.4224	1.1602
53	0.1733	0.1364	1.5265	1.1565
54	0.0145	0.5973	1.0000	1.0000
55	0.158	N/S	1.1620	1.1069
56	0.3467	0.0165	1.3263	1.0551
57	0.3332	0.0797	1.1323	1.0509
58	0.7877	-0.018	1.1922	1.2403
59	0.2657	0.2464	1.3266	1.1340
60	0.6876	N/S	1.0841	1.0680
61	N/S	N/S	2.2138	1.0742
62	0.3844	0.2155	1.0000	1.0000
63	0.3804	0.0427	1.4620	1.2171
64	0.4046	0.0026	1.0701	1.1873
65	0.4707	0.0159	1.1074	1.0566
66	0.8089	N/S	1.0235	1.2183
67	0.7807	N/S	1.0637	1.1200
68	0.3491	0.0132	1.3785	1.0565

* N/S: not sensitive

** when 1, model shows insensitivity

Appendix D

Software Codes

For developing the negative binomial (NB) SPF, SAS software version 3.8, University Edition (SAS Institute Inc 2018) was used, which allows developing SPFs using generalized linear regression (GLR). The NB method was used within the GLR in order to model the collision data. The code for this model is as follows:

```
Proc genmode data=work.import;
MODEL Collisions = LNMJAADT LNMNAADT CT Nlegs/dist=NEGBIN;
Run;
```

For developing the ML algorithms and conducting the sensitivity analysis, Python 3.7.4 is used (The Scientific Python Development Environment 2019), which allows developing ML algorithms using the sci-kit learn library. The code is as follows:

A: importing libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import MinMaxScaler
```

B: reading the dataset as pandas dataframe

```
data = pd.read_csv(r"Datase.csv")
```

C: checking the existence of missing data

```
data.isnull().sum()
```

D: recalling specific functions from the imported libraries

```
from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
```

E: separate the explanatory variables (X) from the outcome y

```
X = data.drop('Collisions', axis = 1)
y = data['Collisions']
y = y.values.reshape(-1,1)
```

F: normalizing the data

```
n_scaler = MinMaxScaler()
X = n_scaler.fit_transform(X.astype(np.float))
```

G: Splitting the data into train and test sets. Randomization is being made before importing the dataset

```
X_train=X[0:275]

X_test = X[275:343]

y_train=y[0:275]

y_test = y[275:343]
```

Note: the following section is only used for sensitivity analysis

H: creating specific datasets for sensitivity analysis

```
X_test1 = X_test.astype(np.float)

X_test2 = X_test.astype(np.float)

X_test3 = X_test.astype(np.float)

X_test4 = X_test.astype(np.float)

X_test5 = X_test.astype(np.float)

X_test6 = X_test.astype(np.float)

X_test7 = X_test.astype(np.float)

X_test8 = X_test.astype(np.float)

X_test9 = X_test.astype(np.float)

X_test10 = X_test.astype(np.float)


X_test1[:,1] *= 0.1

X_test2[:,1] *= 0.2

X_test3[:,1] *= 0.3
```

```
X_test4[:,1] *= 0.4

X_test5[:,1] *= 0.5

X_test6[:,1] *= 0.6

X_test7[:,1] *= 0.7

X_test8[:,1] *= 0.8

X_test9[:,1] *= 0.9

X_test10[:,1] *= 1
```

I: optimization of hyperparameters using GSCV

```
from sklearn.model_selection import GridSearchCV
```

I–1: creating functions for optimizing the ML and printing the selected hyperparameters

```
def print_best_params(gd_model):
    param_dict = gd_model.best_estimator_.get_params()
    model_str = str(gd_model.estimator).split('(')[0]
    print("\n*** {} Best Parameters ***".format(model_str))
    for k in param_dict:
        print("{}: {}".format(k, param_dict[k]))
    print()
```

I-1-1: SVR parameter grid

```
param_grid_svr = dict(kernel=[ 'linear', 'poly', 'rbf'], degree=[2, 3, 4, 5, 6], gamma =
['scale', 'auto'],  C=[600, 700, 800, 900, 1000], epsilon=[0.0001, 0.00001, 0.000001])

svr = GridSearchCV(SVR(), param_grid=param_grid_svr, cv=5, verbose=False, refit =
True)
```

I-1-2: Regression Tree parameter grid

```
param_grid_dt = dict(min_samples_leaf=np.arange(1, 16, 1, int), max_depth =
np.arange(1,10,1, int),  min_impurity_decrease = [0, 1, 2, 3, 4, 5])

dt =  GridSearchCV(DecisionTreeRegressor(random_state=0),
param_grid=param_grid_dt, cv=5,  verbose=False)
```

I-1-3: Radom Forest Regressor parameter grid

```
param_grid_rf = dict(n_estimators=[20], max_depth=np.arange(1, 13, 2),
min_samples_split=[2], min_samples_leaf= np.arange(1, 15, 2, int), bootstrap=[True,
False], oob_score=[False, ])

forest = GridSearchCV(RandomForestRegressor(random_state=0),
param_grid=param_grid_rf, cv=5, verbose=False)
```

J: fitting the SVR model to the training set

```
model = svr

model = model.fit(X_train, y_train.ravel())
```

K: fitting the Regression Tree model to the training set

```
model2 = dt
model2 = dt.fit(X_train, y_train.ravel())
```

L: fitting the Radom Forest Regressor model to the training set

```
model3 = forest
model3 = model3.fit(X_train, y_train.ravel())
```

M: prediction using SVR

```
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)
print ('SVR \n', y_test_pred)
print ('-'*100)
```

N: prediction using Regression Tree

```
y_train_pred2 = model2.predict(X_train)
y_test_pred2 = model2.predict(X_test)
print ('Regression Tree \n', y_test_pred2)
print ('-'*100)
```

O: prediction using RandomForestRegressor

```
y_train_pred3 = model3.predict(X_train)

y_test_pred3 = model3.predict(X_test)

print ('RandomForestRegressor \n', y_test_pred3)

print ('-'*100)
```

P: recalling accuracy metrics and measures

```
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
```

Q: calculating the measures of effectiveness for SVR and printing the values

```
print('the measures of effectiveness for SVR model: \n')

print_best_params(model)

print ('\n'*2)

r2score_train = r2_score(y_train, y_train_pred)

mse_train = mean_squared_error(y_train, y_train_pred)

mae_train = mean_absolute_error (y_train, y_train_pred)

mse_test = mean_squared_error(y_test, y_test_pred)

mae_test = mean_absolute_error (y_test, y_test_pred)

print('R2 score for training set is equal to: ', r2score_train)

print ('\n'*2)

print('Mean Squared error for training set is equal to: ', mse_train)

print('Mean Squared error for test set is equal to: ', mse_test)

print ('\n'*2)
```

```
print('Mean Absolute error for training set is equal to: ', mae_train)

print('Mean Absolute error for test set is equal to: ', mae_test)

print ('\n'*2)

print ('-'*100)
```

R: **calculating the measures of effectiveness for Regression Tree and printing the values**

```
print('the measures of effectiveness for Regression Tree model: \n')

print_best_params(model2)

print ('\n'*2)

r2score_train2 = r2_score(y_train, y_train_pred2)

mse_train2 = mean_squared_error(y_train, y_train_pred2)

mae_train2 = mean_absolute_error (y_train, y_train_pred2)

mse_test2 = mean_squared_error(y_test, y_test_pred2)

mae_test2 = mean_absolute_error (y_test, y_test_pred2)

print('R2 score for training set is equal to: ', r2score_train2)

print ('\n'*2)

print('Mean Squared error for training set is equal to: ', mse_train2)

print('Mean Squared error for test set is equal to: ', mse_test2)

print ('\n'*2)

print('Mean Absolute error for training set is equal to: ', mae_train2)

print('Mean Absolute error for test set is equal to: ', mae_test2)

print ('\n'*2)

print ('-'*100)
```

S: calculating the measures of effectiveness for Random Forest Regressor and printing the values

```
print('the measures of effectiveness for RandomForestRegressor model: \n')

print_best_params(model3)

print ('\n'*2)

r2score_train3 = r2_score(y_train, y_train_pred3)

mse_train3 = mean_squared_error(y_train, y_train_pred3)

mae_train3 = mean_absolute_error (y_train, y_train_pred3)

mse_test3 = mean_squared_error(y_test, y_test_pred3)

mae_test3 = mean_absolute_error (y_test, y_test_pred3)

print('R2 score for training set is equal to: ', r2score_train3)

print ('\n'*2)

print('Mean Squared error for training set is equal to: ', mse_train3)

print('Mean Squared error for test set is equal to: ', mse_test3)

print ('\n'*2)

print('Mean Absolute error for training set is equal to: ', mae_train3)

print('Mean Absolute error for test set is equal to: ', mae_test3)


data.describe()


data.mad()
```

T: conducting the sensitivity analysis

```
import xlsxwriter as xls

workbook = xls.Workbook('sensitivity1.xlsx')  **sensitivity1 is the file for MJAADT sensitivity

worksheet = workbook.add_worksheet()

worksheet2 = workbook.add_worksheet()

worksheet3 = workbook.add_worksheet()


array = [X_test1,X_test2,X_test3,X_test4,X_test5,X_test6,X_test7,X_test8,X_test9,X_test10]

k=0

for i in array:

row = 0
```

U: prediction using SVR

```
y_test_pred = model.predict(i)

for col, data in enumerate(y_test_pred):

    worksheet.write(row, col, data)
```

V: prediction using Regression Tree

```
y_test_pred2 = model2.predict(i)

for col2, data2 in enumerate(y_test_pred2):

    worksheet2.write(row, col2, data2)
```

W: prediction using RandomForestRegressor

```
y_test_pred3 = model3.predict(i)

for col3, data3 in enumerate(y_test_pred3):

    worksheet3.write(row, col3, data3)

k=k+1


workbook.close()
```