8-2021

Severity Analysis of Large Truck Crashes- Comparision Between the Regression Modeling Methods with Machine Learning Methods.

Jinli Liu

SEVERITY ANALYSIS OF LARGE TRUCK CRASHES

- COMPARISION BETWEEN THE REGRESSION MODELING METHODS WITH MACHINE LEARNING METHODS

THESIS

Presented in Partial Fulfillment of the Requirements for

the Master of Science Degree in the Graduate School

of Texas Southern University

By

Jinli Liu, M.S.

Texas Southern University

2021

Approved By

Dr. Yi Qi

Chairperson, Thesis Committee

Dr. Gregory H. Maddox

Dean, The Graduate School

Approved By:

Dr. Yi Qi	**03/34/2021**
Yi Qi, Ph.D., Chairperson of Thesis Committee	Date
Dr. Fengxiang Qiao	**03/34/2021**
Fengxiang Qiao, Ph.D., Committee Member	Date
Dr. Mehdi Azimi	**03/34/2021**
Mehdi Azimi, Ph.D., Committee Member	Date
Dr. Yunjiao Wang	**03/34/2021**
Yunjiao Wang, Graduate School Representative	Date

ABSTRACT

According to the Texas Department of Transportation's Texas Motor Vehicle Crash Statistics, Texas has had the highest number of severe crashes involving large trucks in the US.

As defined by the US Department of Transportation, a large truck is any vehicle with a gross vehicle weight rating greater than 10,000 pounds. Generally, it requires more time and much more space for large trucks to accelerating, slowing down, and stopping. Also, there will be large blind spots when large trucks make wide turns. Therefore, if an unexpected traffic situation comes upon, It would be more difficult for large trucks to take evasive actions than regular vehicles to avoid a collision.

Due to their large size and heavy weight, large truck crashes often result in huge economic and social costs. Predicting the severity level of a reported large truck crash with unknown severity or of the severity of crashes that may be expected to occur sometime in the future is useful. It can help to prevent the crash from happening or help rescue teams and hospitals provide proper medical care as fast as possible. To identify the appropriate modeling approaches for predicting the severity of large truck crash, in this research, four representative classification tree-based ML models (e.g., Extreme Gradient Boosting tree (XGBoost), Adaptive Boosting tree(AdaBoost), Random Forest (RF), Gradient Boost Decision Tree (GBDT)), two non-tree-based ML models (e.g., the Support Vector Machines (SVM), k-Nearest Neighbors (kNN)), and LR model were selected. The results indicated that the GBDT model performs best among all of seven models.

Keywords: Large Truck Crash, Crash Severity Prediction, and Machine Learning Methods

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ML:	Machine Learning
CART:	Classification and Regression Tree
CRIS:	Crash Records Information System
LR:	Logistic Regression
RF:	Random Forest model
AdaBoost:	Adaptive Boost tree
GBDT:	Gradient Boost Decision Tree
XGBoost:	Extreme Gradient Boost tree
SVM:	Support Vector Machine
k-NN:	k-Nearest neighbor
TxDOT:	Texas Department of Transportation

VITA

2012-2016…….....B.S, Zhejiang Normal University, Zhejiang, P. R. China

2016-2019…….....M.S, Zhejiang Normal University, Zhejiang, P. R. China

2019-2021……Graduate Research Assistant and Graduate Teach Assistant, Texas Southern University, Houston, Texas

Major Field……Transportation Planning and Management

ACKNOWLEDGEMENT

First, I would like to thank my advisor Dr. Yi Qi for the continuous support to my academic and research studies in TSU. Dr. Qi is a highly respected advisor with excellent professional experiences and great accomplishments. Working side-by-side with her was a great experience, which helped me to achieve my goal of completing the master's study at TSU.

I would like to express my sincere appreciation to my thesis committee members: Dr. Fengxiang Qiao, Dr. Mehdi Azimi, Dr. Yunjiao Wang.

I would also like to thank Qun Zhao for her support and helpful suggestions in my study life at TSU. I would like to thank my family and my friends for their support throughout my life.

CHAPTER 1

INTRODUCTION

1.1 Background of Research

Improving traffic safety is one of the most serious topics for transportation engineers and politicians of any country. In the United States, large trucks, as a significant means of freight transportation, play a major role in the transportation system. According to the definition given by the U.S. Department of Transportation, a large truck is any vehicle with a gross weight rating greater than 10,000 pounds. Due to their size and weight, the operation of large tuck is often more difficult than passenger vehicles. Besides, there would be blind spots when large trucks making wide turns. All thesis conditions indicate that crashes involving large trucks often result in fatal injuries, severe property damage, and economic and social costs. Texas has had the highest number of fatal crashes involving large trucks in the U.S. since 1994 (Zhao et al, 2018). According to the Federal Motor Carrier Safety Administration, fatal crashes involving large trucks continue increasing, from 2016 to 2018, fatal crashes involving large trucks increased about 5.7 percent.

Typically, depending on the number of vehicles involved, crashes can be categorized into two types: single-vehicle and multi-vehicle large truck crash. Under the above two categories, there are more specific crash types: rollover, jackknifing, head-on, and rear-end crash, and so on. There is a wide spectrum of risk factors contributing to crashes. Generally, the contributing factors can be categorized into driver-related contributing factors, roadway-related contributing factors, vehicle-related contributing factors, environmental-related contributing factors, and so on. In this study, all types of crashes were considered in the prediction of the severity of crashes considering a wide spectrum of contributing factors.

According to Highway Safety Manual, crash severity can be used for establishing the level of injury caused by a crash costs. KABCO scale is frequently used by law enforcement for classifying injuries. The definition of KABCO scale is:

K- Fatal injury;

A- Incapacitating injury;

B- Non-incapacitating injury;

C- Possible injury;

O- No injury;

In crash severity prediction researches, the response classes, also known as outcome classes can be categorized into two, such as AK level (A is the incapacitating crash, and K is the fatal crash) crashes and non-AK level crashes, three, four, or five (Fiorentini & Losa, 2020). The response classes are usually determined by the research objectives and data quality. Previous studies showed that it is relatively difficult to accurately predict five levels of severity than three levels of severity, in order to more accurately predict the severity range of a crash, therefore, in this study, the response classes (output class) will be categorized into three levels: accidents with Property Damage Only (PDO), Slight Injuries (SLIG), and accidents with Killed or Severe Injuries (KSEV), the detailed information concerning the datasets will be presented in Section 3.2.

From a methodological perspective, a wide spectrum of modeling approaches has been adopted in the crash severity prediction. Both traditional regression models and the Machine Learning (ML) based have been applied for the crash severity analysis. These two types of modeling approaches have their advantages and limitations. The regression models have equations that explicitly link the independent variables (risk factors in this study) to the dependent variable (crash severity level), thereby it has a good capability in analyzing the impacts of independent

variables. However, they have difficulties in detecting and interpreting complex or high-order interactions among independent variables (Su et.al, 2008). Some ML-based methods like neural networks have been known for their strong prediction capabilities. However, they have been criticized for operating like a black box and unable to explicitly explain the impacts of independent variables on the dependent variables (Yu & Abdel-Aty, 2013). In recent years, the classification tree-based Machine Learning (ML), Support Vector Machine (SVM), and k-Nearest Neighbor (k-NN) methods have been widely employed for crash severity prediction. There is a lack of studies on comparing the performance of different types of models including ML models and traditional regression models. Besides, even though a lot of modeling approaches have been adopted in crash severity prediction, few of them focus on large truck crashes. Therefore, it is important to find out that the performance of different models in predicting the severity levels of large truck crashes and provide some guidance for its modeling approaches.

In order to develop a reliable prediction model, some attention has been paid to the selection of sample datasets for training or fitting classifiers. Some researchers believe that a training sample with skewed class distribution tends to make classifiers be overwhelmed by the majority classes and overlook the minority one (Kotsiantis et al. 2006). On the contrary, some researchers suggest that it is important to select a sample that has the same class distribution as the original population rather than ensuring the classes are balanced. Indeed, in crash severity prediction problem, the number of instances relating to AK level crash are generally far fewer than the number of instances relating to Property Damage Only (PDO) or non-AK level crash. Since relatively little attention has been paid to the data-imbalance issue in large truck crash prediction, and the effects of different resampling methods to different modeling approaches are still not clear. In this study, three resampling techniques, random undersampling, oversampling, and mix

sampling will be used to preprocess the original training dataset to testify the effects of resampling in model prediction performance.

Predicting the severity level of a reported crash with unknown severity is useful. It can help rescue teams and hospitals provide proper medical care as fast as possible.

1.2 Research Objective

The introduction provided the background to define the objectives of this research, this research has two main objectives: 1) to testify the effects of class balancing techniques in model prediction performance using three resampling techniques: random undersampling, oversampling, and mix sampling; 2) comparison of the performance of four classification tree-based ML models (Extreme Gradient Boosting tree(XGBoost), Adaptive Boosting tree(AdaBoost), Random Forest (RF), Gradient Boost Decision Tree (GBDT)), two non-tree-based ML models (Support Vector Machines (SVM), and k-Nearest Neighbors (kNN)), and the traditional Logistic Regression model (LR) in crash severity prediction;

The findings of this study can help to predict the severity of a reported truck crash with unknown severity. It can help rescue teams and hospitals provide proper medical care as fast as possible.

1.3 Outline of the Study

This thesis is comprised of five chapters. The first chapter provides a background of the problems, the research objectives, and the outlines of the study. The second chapter presents studies of the existing research on large truck crash severity prediction, different modeling approaches for crash severity prediction, and critical issues in developing crash severity models. The third chapter describes the data used in this study, introducing the Extreme Gradient Boosting tree (XGBoost), Adaptive Boosting tree (AdaBoost), Random Forest (RF), Gradient Boost

Decision Tree (GBDT)), Support Vector Machines (SVM), and k-Nearest Neighbors (kNN)), and the logistic regression model in details, introducing the oversampling, undersampling, and mix in detail, and describes the prediction evaluation measures. Then, the fourth chapter compares and discusses the results of the model prediction performances, and discusses the effects of data balancing techniques. Finally, the fifth chapter provides the study conclusions and recommendations for future research.

CHAPTER 2

LITERATURE REVIEW

The literature review contains three perspectives to establish the context for the proposed research. First, the existing study focuses relating to crash severity prediction will be presented. Secondly, different types of models used to develop crash severity prediction models will be introduced, including the regression models and the machine learning models. Thirdly, critical issues in developing the crash severity prediction models are discussed. Finally, a summary of the existing studies will be discussed.

2.1 Crash Severity Prediction

Crash severity prediction falls into the scope of crash severity analysis, has the distinct advantage of including driver-related contributing factors and individual crash characteristics into severity analysis. The analyzed topics related to crash severity analysis are manifold. Previous studies have investigated the factors that affect the severity of crashes with a variety of focuses. Some studies focused on truck crashes, or passenger car crashes, and others focused on bicycle crashes. Some studies focused on certain types of crashes, such as rear-end crashes or rollover crashes, and others focused on certain locations where the crashes occurred. Besides, The dependent variables of existing crash severity models are typically either a binary response outcome (e.g., injury or non-injury, AK or non-AK) or a multiple response outcomes (e.g., three responses, four responses, and five responses). For research objectives, some researchers aim at investigating the factors that contribute to the severity of crashes, while others aim at predicting the severity of a crash. Overall, crash severity prediction is a promising research topic in the traffic

safety field. It can help to predict the severity that may be expected to occur for a crash, which helps rescue teams and hospitals provide proper medical care as fast as possible.

2.2 Methodologies for Crash Severity Prediction

From a methodological perspective, a wide spectrum of modeling approaches has been adopted in the crash severity analysis. Both traditional regression models (such as logistic regression model) and the ML-based methods (such as random forest, adaptive boosting, gradient boost decision tree, extreme gradient boost tree, and support vector machine) have been applied for the crash severity analysis. The traditional regression models and ML-based methods have their own advantages and limitations. These models' capabilities in predict the severity level of large truck crashes need to be investigated.

2.2.1 Regression Models

For the traditional regression models, logistic regression models and ordered probit models have been widely used for crash severity analysis. For example, Chang and Mannering (1999) used nested logit models to analyze the severity of injuries for both truck-involved crashes and non-truck-involved crashes. Khattak et al. (2003) used ordered probit models to identify the contributing factors, and the focus of their study was the large truck rollover crashes.

Zhu and Srinivasan (2011) examined the factors that contributed to the severity of large truck crashes using the regression model basing on a dataset with a thousand crashes extracted from the Large Truck Crash Causation Study (LTCCS) from April 2001 and December 2003.

Dissanayake and Roy (2014) conducted a crash severity analysis of single-vehicle crashes and run-off-road crashes. A binary logistic regression model was selected to perform the analysis. The model comprised of 72,181 crash records extracted from the Kansas Accident Reporting System database from 1999 to 2008. The results indicate that factors that significantly associated

with the severity of ROR crash included 1) driver-related factors; 2) road-related variables; 3) environment-related factors; 4) vehicle-related factors; and 5) trees and ditches as fixed types of objects.

Besides the traditional logit models and ordered probit models, some advanced regression modeling techniques have been explored by previous studies. For example, Xie et al. (2009) conducted a motor vehicle crash injury severity analysis using Bayesian ordered probit (BOP) models. Pahukula et al. (2015) utilized random parameter logit models to examine the impacts of time of day on the injury severity of large truck-involved crashes. Al-Bdairi and Hernandez (2017) used an ordered random parameter probit model to analyze the injury severity of large truck-involved run-off-road crashes in Oregon. Ahmed et al. (2018) explored the contributing factors to the large truck-involved crashes on rural highways in Wyoming using Bayesian binary logit models.

2.2.2 Machine Learning Models

For ML-based techniques, the techniques applied to crash severity prediction include classification tree-based models, neural networks, and support vector machine models. In recent years, the classification tree-based ML methods have been widely employed for crash risk prediction and identification of contributing factors. A classification tree-based ML method decides which crash risk factors should be chosen as the decision nodes and which features can provide more information or reduce more uncertainty about the severity of traffic crashes based on information gain and entropy. Chang and Chien (2013) used the classification and regression tree (CART) method to examine the impacts of the driver and vehicle-related factors on the severity of injuries in large truck crashes. The importance of factors was analyzed according to the

structure of the developed classification tree. The results showed that drunk-driving is the most significant factor that contributes to the severity of injuries in large truck crashes on the freeways.

Yu and Abdel-Aty (2014) focused on developing crash severity analysis models by first selecting the most important variables associated with the severe crash occurrence using the random forest (RF) method. Then, three different types of models (fixed-parameter logit model, support vector machine model, and random parameter logit model) were developed to analyze crash injury severity. In some other studies on crash severity analysis, the RF method was also used for preselecting the independent variables for the regression models.

Zeng and Huang (2014) proposed a convex combination (CC) algorithm to train a neural network (NN) model for crash injury severity prediction and a modified NN pruning for function approximation (N2PFA) algorithm to optimize the NN structure. According to the results of this study, the CC algorithm outperforms the traditional back-propagation algorithm both in convergence ability and training speed.

Iranitalab and Khattak (2017) compared the performance of four statistical and machine learning methods including Multinomial Logit (MNL), Nearest Neighbor Classification (NNC), Support Vector Machines (SVM), and Random Forests (RF), in predicting traffic crash severity. In this study, the effects of data clustering methods including K-means Clustering (KC) and Latent Class Clustering (LCC) were also investigated. The analysis used reported two-vehicle crash data from Nebraska from 2012 to 2015. The correct prediction rates and the proposed approach showed that the NNC had the best prediction performance in all levels of crashes and especially in more severe crashes. Data clustering did not affect the prediction performance of SVM, but KC improved the prediction performance of MNL, NNC, and RF, while LCC caused improvement of MNL and RF but weakened the performance of NNC.

Tang et al. (2019) proposed a two-layer stacking framework to predict crash injury severity. The first layer integrates the advantages of three base classification methods: RF, AdaBoost, and GBDT. The second layer completes the classification of crash injury severity based on a logistic regression model.

2.3 Imbalanced versus Balanced Training Datasets

In order to develop a reliable prediction model, some attention has been paid to the selection of appropriate sample datasets for training or fitting models. As we know, high imbalance datasets often occur in practical applications. In such cases, standard machine learning classifiers tend to be overwhelmed by the majority classes and overlook the minority ones (Kotsiantis et al. 2006). The effects of class imbalance have attracted more and more attention in recent years. A number of solutions to the class-imbalance problem were previously proposed both at the data and algorithmic levels. At the data level, these solutions include many different forms of resampling to preprocess the data in order to balance datasets. At the algorithmic level, solutions include create new algorithms or modify existing ones. Compared with the algorithmic level approach, the data level approach (preprocessing approach) seems to be the more straightforward approach that has greater promise to overcome the class-imbalance problem (Thammasiri et al., 2014). For the data level approach, there are three resampling techniques affirmed to handle imbalanced datasets: oversampling techniques, undersampling techniques, and mixed techniques. Oversampling concerns techniques that balance the number of instances between classes through increase the number of minority classes until the dataset is balanced. Conversely, undersampling concerns the techniques to balance classes by reducing the number of instances from the majority class. Finally, mixed concerns the techniques that combine the above two techniques, integrating oversampling of minority class with undersampling majority class.

Chawla et al. (2002) proposed the synthetic minority oversampling (SMOTE) and compared the effects of different resampling approaches. SMOTE was tested on a variety of datasets, with varying degrees of imbalance and varying amounts of data in the training dataset. The results indicated that SMOTE approach can improve the accuracy of classifiers for a minority class. The combination of SMOTE and undersampling performed better than plain undersampling. The combination of SMOTE and undersampling also performed better, based on AUC score. The definition of AUC score can be found in Section 3.4.

García et al. (2020) investigated and illustrated the effects of the resampling methods on the inner structure of a data set by exploiting local neighborhood information, identifying the sample types in both classes and analyzing their distribution in each resampled set. Experimental results indicated that the resampling methods that produce the highest proportion of safe samples (safe if at least 4 neighbors are from the same class) and the lowest proportion of unsafe samples correspond to those with the highest overall performance. This paper also explained why oversampling has been reported to be usually more efficient than undersampling.

Algorithm level methods involve specific solutions dedicated to improving a given classifier. Within the algorithm level approaches, ensembles are quite often applied. Ensemble learning is a machine learning paradigm where multiple models (often called “weak learners”) are trained to solve the same problem and combined to get better results. Most of these ensembles are based on known strategies from bagging and boosting. Bagging, which often considers homogeneous weak learners, learns them independently from each other in parallel and combines them following some kind of deterministic averaging process. Boosting, which often considers homogeneous weak learners, learns them sequentially in a very adaptative way (a base model depends on the previous ones) and combines them following a deterministic strategy. Bootstrap

aggregating is one of the most famous “bagging” approaches that aim at producing an ensemble model that is more robust than the individual models composing it. Currently, there are various existing extensions of bagging and a lot of related works indicated the good performance of bagging extensions versus the other ensembles (Anyfantis et al., 2008).

Błaszczyński and Stefanowski (2015) proposed Neighbourhood Balanced Bagging, where sampling probabilities of examples were modified according to the class distribution in their neighborhood. Two of its versions were considered: the first one keeping a larger size of bootstrap samples by hybrid oversampling and the other reducing this size with stronger undersampling. The results showed that the first version is significantly better than existing oversampling bagging extensions while the other version is competitive to Roughly Balanced Bagging. Besides, they demonstrated that detecting types of minority examples depending on their neighborhood may help explain why some ensembles work better for imbalanced data than others.

In recent years, several studies were related to crash severity analysis with data balancing techniques. For example, Mujalli et al. (2016) used three different data balancing techniques: undersampling, oversampling, and a mix technique that combines both to balance the traffic accident data collected on urban and suburban roads in Jordan from 2009 to 2011. Then, different Bayes classifier models were developed based on the imbalanced and balanced datasets. The results indicated that using the balanced data sets, especially those created using oversampling techniques, with Bayesian networks improved classifying a traffic accident according to its severity and reduced the misclassification of killed and severe injuries instances.

Jeong et al. (2018) used five classification learning models (Logistic regression, Decision tree, Neural network, Gradient boosting model, and Naïve Bayes classifier) to classify the levels of injury severity and the classification performance was improved by two training-testing

methods including Bootstrap aggregation and majority voting. To account for the imbalanced classes, under-sampling and over-sampling were used. The results showed that the effect of treatments for the imbalanced data was maximized when under-sampling was combined with bagging.

Schlögl et al. (2019) conducted a comparison of statistical learning methods for deriving determining factors of accident occurrence from an imbalanced high resolution dataset. A series of statistical learning techniques (including all four types of logistic regression, tree-based ensemble methods, the BRNN, and the Pegasos SVM) were compared with respect to their predictive performance. A combination of synthetic minority oversampling and maximum dissimilarity undersampling was used to balance the training dataset. Findings substantiated that a trade-off between accuracy and sensitivity was inherent to imbalanced classification problems. Results also showed satisfying performance of tree-based methods which exhibit accuracies between 75% and 90% while exhibiting sensitivities between 30% and 50%.

Rivera et al. (2020) assessed five classification algorithms: Classification and Regression Tree (CART), Naïve Bayes, kNN, Random Forest, and Support Vector Machine (SVM)on a class-imbalanced benchmark; this challenging issue was dealt with via five sampling algorithms: synthetic minority oversampling technique (SMOTE), borderline SMOTE, adaptive synthetic sampling, random oversampling, and random undersampling. The results indicated that the imbalance between both classes (the class was binarized as ‘traffic accident’ and ‘not traffic accident’) negatively affected the performance of both classifiers. Besides, random oversampling obtained the most encouraging results among the sampling algorithms tested.

Abou Elassad et al. (2020) designed an ensemble fusion framework founded on the use of various base classifiers that operate on fused features and a Meta classifier that learns from base

classifiers' results to acquire more performant crash predictions. In this study, a resampling-based scheme, including Bagging and Boosting, was conducted to generate diversity in learner combinations comprising Bayesian Learners (BL), k-Nearest Neighbors (kNN), Support Vector Machine (SVM), and Multilayer Perceptron (MLP). Then, to ensure that the proposed framework provides powerful and stable decisions, an imbalance-learning strategy was adopted using the Synthetic Minority Oversampling Technique (SMOTE) to address the class imbalance problem as crash events usually occur in rare instances. The findings showed that Boosting depicted the highest performance within the fusion scheme and can accomplish a maximum of 93.66% F1 score and 94.81% G-mean with Naïve Bayes, Bayesian Networks, k-NN, and SVM with MLP as the Meta-classifier. The definition of performance measures can be found in Section 3.4.

Abou Elassad et al. (2020) developed a proactive decision support system for predicting traffic crash events. Modeling approaches that rely on Random Forest, Support Vector Machine, and Multilayer Perceptron machine learning techniques were applied to establish efficient crash predictions. This study also compared different data balancing techniques in improving the predictive performance through three balancing techniques: oversampling, undersampling, and synthetic minority over-sampling (SMOTE). The highest performances were acquired using SMOTE strategy as MLP achieved a 94.5% precision, 94.2% f1-score, 93.7% AUC and 95.3% recall, while SVM achieved a 91.5% g-mean. A more detailed explanation of these performance measures can be found in Section 3.4.

2.4 Summary

From the various literature references mentioned above, two aspects of conclusions were reached. First, various modeling approaches have been used to predict crash severity, both traditional regression models and ML-based methods. Among these models, the Logistic

Regression model was among one of the most frequently used regression models. Besides, classification tree-based ML models (e.g., Extreme Gradient Boosting tree(XGBoost), Adaptive Boosting tree(AdaBoost), Random Forest (RF), Gradient Boost Decision Tree (GBDT)), and the Support Vector Machines (SVM), k-Nearest Neighbors (kNN) are ones of the most popular ML techniques that have been used for crash severity prediction. However, there is a lack of studies on comparing the performance of different types of models including ML models and traditional regression models. Moreover, few studies have considered the tree-based ML models as a group and compare them with other equally popular ML method. Several questions remain open and need further exploration. Therefore, this study aims to compare the predictive performances for crash injury severity analysis between six machine learning models and one logistic regression model.

Secondly, although a wide variety of modeling approaches have been adopted to study injury severity of truck-involved crashes, relatively little attention has been paid to the data-imbalance issue, and the effects of different data balancing methods on different modeling approaches are still not clear. To fill this gap, three most commonly used resampling techniques, random undersampling, oversampling, and mix sampling will be used to preprocess the original training dataset to testify the effects of resampling in model prediction performance.

CHAPTER 3

METHODOLOGY

This chapter is to present the overall study design procedure of the research. To accomplish the research objectives, e.g. predict the severity level of the large truck crash, this study is designed in four aspects: 1) study approaches, 2) data description, 3) methodology for severity level prediction, and, 4) prediction evaluation measures.

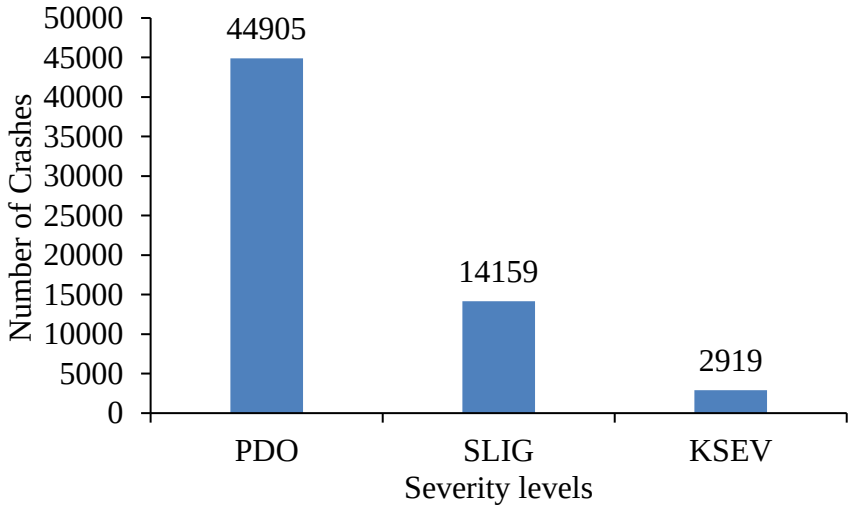
3.1 Data

3.1.1 Data Description

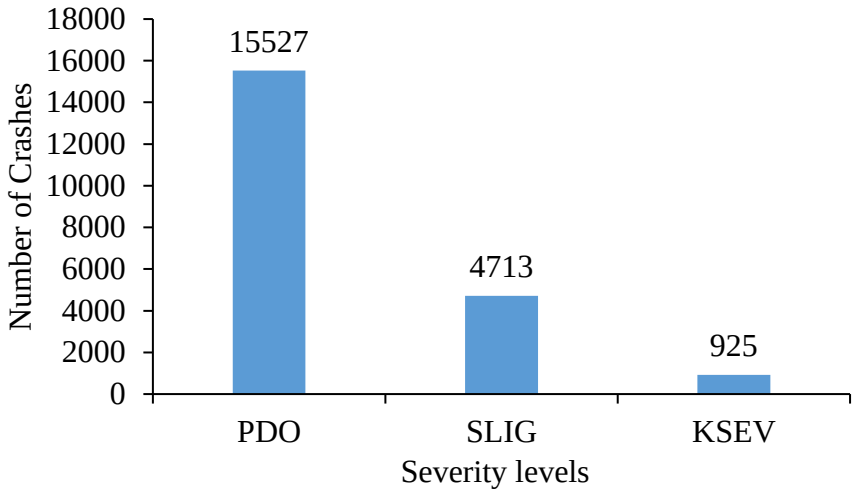
A large and comprehensive truck crash dataset used in this research was developed based on the crash records collected from the Texas Crash Records Information System (CRIS). It contained the truck crash records of the entire state of Texas from 2016 to 2019. A large truck, as defined by the U.S. Department of Transportation, is any truck with a gross weight rating greater than 10,000 pounds. Different types of crashes were indiscriminately collected for this study. In the CRIS, each record has more than 170 attributes, including information about the drivers, vehicles, characteristics of the crashes, roadway conditions, and environmental conditions. The attributes used in this research will be carefully selected from over 170 attributes of the large truck crash data based on their categories, their correlations between each other, their relationship to the dependent variable, and the quality of the data. The detailed information of attribute selection, known as independent variables selection will be discussed in the following section. Finally, the dataset will be divided into a training dataset, which covers the years 2016 to 2018, and a dedicated test dataset, which covers the year 2019 for evaluation purposes.

3.1.2 Dependent and Independent Variables

The dependent variable in this analysis was the severity level of large truck crashes. It was categorized into three levels: accidents with Property Damage Only (PDO) (y = 0), Slight Injuries (SLIG) (y = 1), and accidents with Killed or Severe Injuries (KSEV) (y = 2). In the training dataset, as shown in Figure 2. (a), there were 72.45% of PDO level crashes, 22.84% of SLIG level crashes and 4.71% of KSEV level crashes. In the testing dataset, as shown in Figure 2. (b), there were 73.36% of PDO level crashes, 22.27% of SLIG level crashes and 4.37% of KSEV level crashes. As we can see, three levels of severity distribution of the testing dataset are highly consistent with the training dataset. Besides, a class distribution with an imbalance ratio less than 1.5 can be considered to be balanced (Fernández et al. 2008), in this consideration, the training and testing datasets are imbalanced.



(a) Training dataset



(b) Testing dataset

Figure 1. Distribution of Large Truck Crash Injury Severity in Training and Testing dataset

The independent variables were carefully selected from over 170 attributes of the large truck crash data based on their categories, their correlations between each other, their relationship to the dependent variable, and the quality of the data.

At first, different types of variables related to the roadway, environment, and driver's characteristics were derived and classified into different categories. Then, the correlations between these variables were analyzed. Some of these variables were highly correlated. For example, road surface conditions (dry, wet, and ice-covered) and weather characteristics (clear, rain, and snow) were highly correlated factors. To avoid the collinearity problem, the weather characteristic factors were kept in the model, while the surface-condition factors were removed, since the weather characteristic factors were more correlated to dependent variable than the surface-condition factors. In addition, most of the independent variables were categorical variables, and they were all converted to the dummy variables, as shown in Table 1. It can be seen that the variables in the same category were highly correlated. Taking the "Lighting Conditions" category as an example, the lighting conditions included the "daylight", "dark no light", "dawn", "dark light", and "dusk", which was a complete list, and the lighting condition of a crash must be one of these five conditions.

Thus, if we included all these dummy variables, their sum would be equal to 1. To avoid the dummy variable trap, one baseline variable was identified for each category and was excluded from the model (Greene, 2000). Furthermore, some factors did not have a very clear causal relationship with the dependent variable. For example, the factor "number of lanes blocked by the crash" was not the cause of a severe crash but was determined simultaneously with the crash severity level when a crash occurred. Therefore, this type of variable should also be removed from the model to avoid the endogeneity problem (Duncan et al., 2004). Finally, by carefully examining all the factors in different categories, only 40 independent variables were finally selected, as listed in Table 1, and the distributions of variables are presented in Table 2. After deleting the crash records that contained missing information, the final dataset contained records of 83,148 large truck crashes.

Table 1
Variables and Descriptions

Traffic Control		***Weather Characteristics***	
none	1 if no traffic control, 0 otherwise (Baseline)	clear	1 if clear, 0 otherwise(Baseline)
stopsign	1 if traffic control is stop sign, 0 otherwise	rain	1 if raining, 0 otherwise
signallight	1 if traffic control is signal light, 0 otherwise	snow	1 if snowing, 0 otherwise
yieldsign	1 if traffic control is yield sign, 0 otherwise	blowing	1 if blowing sand, 0 otherwise
flashinglight	1 if traffic control is flashing light, 0 otherwise	fog	1 if fog, 0 otherwise
markedlane	1 if traffic control is markedlane, 0 otherwise	sleet	1 if sleet, 0 otherwise
signal camera	1 if traffic control is signal camera, 0 otherwise	severcrosswinds	1 if severe crosswinds, 0 otherwise
Light Characteristics		***Median Type***	
daylight	1 if incident occurred when daylight, 0 otherwise(Baseline)	mediannone	1 if no median, 0 otherwise(Baseline)
dawn	1 if incident occurred when dark not lighted, 0 otherwise	unprotected	1 if median type is unprotected, 0 otherwise
darknolight	1 if incident occurred when dawn, 0 otherwise	positivebarrier	1 if median type is positive barrier, 0 otherwise
darklight	1 if incident occurred when dark but lighted, 0 otherwise	onewaypair	1 if median type is one-way pair, 0 otherwise
dusk	1 if incident occurred when dusk, 0 otherwise	curbed	1 if median type is curbed, 0 otherwise
Roadway Functional System		***Road Alignment***	
rintersatehighway	1 if rural interstate highway, 0 otherwise (Baseline)	strailevel	1 if road alignment is straight level, 0 otherwise(Baseline)
uinterstatehighway	1 if urban interstate highway, 0 otherwise	straigrade	1 if road alignment is straight grade, 0 otherwise
rprincipalarterial	1 if rural principle arterial, 0 otherwise	straihillcrest	1 if road alignment is straight hillcrest, 0 otherwise
uotherprincipalarterial	1 if urban other principle arterial, 0 otherwise	curlevel	1 if road alignment is curve level, 0 otherwise
uminorarterial	1 if urban minor arterial, 0 otherwise	curgrade	1 if road alignment is curve grade, 0 otherwise
rminorarterial	1 if rural minor arterial, 0 otherwise	curhillcrest	1 if road alignment is curve hillcrest, 0 otherwise
Location of First Harmful Event		***Base Type***	
onroad	1 if crash occurred on road, 0 otherwise(Baseline)	soil	1 if base type is soil, 0 otherwise(Baseline)
onshoulder	1 if crash occurred on shoulder, 0 otherwise	granular	1 if base type is granular, 0 otherwise
onmedian	1 if crash occurred on median, 0 otherwise	asphalt	1 if base type is asphalt, 0 otherwise
offroad	1 if crash occurred off road, 0 otherwise	concrete	1 if base type is concrete, 0 otherwise
Shoulder Type Left		***Curb Type Left***	
shoulderlnone	1 if no left shoulder, 0 otherwise(Baseline)	curblnone	1 if no left curb, 0 otherwise(Baseline)
shoulderleft	1 if left shoulder exists, 0 otherwise	curbleft	1 if left curb exists, 0 otherwise
Shoulder Type Right		***Curb Type Right***	
shoulderrnone	1 if no right shoulder, 0 otherwise(Baseline)	curbrnone	1 if no right curb, 0 otherwise(Baseline)
shoulderright	1 if right shoulder exists, 0 otherwise	curbright	1 if right curb exists, 0 otherwise
Road Type		***Crash Contributing Factors***	
2 lane, 2 way	1 if road type is 2 lane, 2 way, 0 otherwise(Baseline)	fatigue	1 if driver under influence of fatigue, 0 otherwise
4 ormore,divided	1 if road type is 4 ormore,divided, 0 otherwise	drug	1 if driver under influence of drug, 0 otherwise
4 or more,undivided	1 if road type is 4 ormore,undivided, 0 otherwise	alcohol	1 if driver under influence of alcohol, 0 otherwise
Lane Width and Shoulder Width		***Numerial variables***	
Lanewid	The width of travel lanes in feet	Adt_Adj_Curnt_Amt	Adjusted average daily traffic for the current year for crashes located on the road

Shldr_Width_Left	The width of left shoulder in feet	Crash_Speed_Limit	Speed Limit
Shldr_Width_Right	The width of right shoulder in feet	Trk_Aadt_Pct	Adjusted average daily traffic percent for trucks for crashes located on the road
		Nbr_Of_Lane	Number of lanes, not including turning and climbing lanes, for crashes located on the road

Table 2
Distribution of the Variables

Variable	Crash Injury Severity			Total	Percent	Variable	Crash Injury Severity			Total	Percent
	PDO	SLIG	KSEV				PDO	SLIG	KSEV		
Traffic Control						***Weather Characteristics***					
none	6587	1819	275	8681	10.44%	clear	43087	13219	2764	59070	71.04%
stopsign	2679	915	273	3867	4.65%	rain	6333	1978	332	8643	10.39%
signallight	7271	2081	253	9605	11.55%	snow	133	26	5	164	0.20%
yieldsign	938	262	28	1228	1.48%	blowing	36	13	8	57	0.07%
flashinglight	283	96	32	411	0.49%	fog	422	188	90	700	0.84%
markedlane	31653	10224	1872	43749	52.62%	sleet	153	35	9	197	0.24%
signal camera	116	33	5	154	0.19%	severcrosswinds	159	40	11	210	0.25%
Light Characteristics						***Median Type***					
daylight	45662	13980	2326	61968	74.53%	mediannone	17147	5482	1639	24268	29.19%
dawn	828	276	84	1188	1.43%	unprotected	5882	1818	368	8068	9.70%
darknolight	6667	2249	894	9810	11.80%	positivebarrier	11128	3634	720	15482	18.62%
darklight	6534	2150	480	9164	11.02%	onewaypair	103	18	1	122	0.15%
dusk	448	139	39	626	0.75%	curbed	675	220	27	922	1.11%
Roadway Functional System						***Road Alignment***					
uinterstatehighway	21967	6639	829	29435	35.40%	strailevel	46507	14148	2729	63384	76.23%
rprincipalarterial	5766	1958	733	8457	10.17%	straigrade	6265	2161	516	8942	10.75%
uotherprincipalarterial	17158	5611	769	23538	28.31%	straihillcrest	1799	741	150	2690	3.24%
uminorarterial	2348	680	139	3167	3.81%	curlevel	3304	999	265	4568	5.49%
rminorarterial	2853	963	438	4254	5.12%	curgrade	1947	652	152	2751	3.31%
rintersatehighway	6567	1769	538	8874	10.67%	curhillcrest	449	121	26	596	0.72%
Location of First Harmful Event						***Base Type***					
onroad	52128	16415	3226	71769	86.31%	soil	372	133	42	547	0.66%
onshoulder	764	190	130	1084	1.30%	granular	34451	10964	2561	47976	57.70%
onmedian	1873	641	115	2629	3.16%	asphalt	788	223	50	1061	1.28%
offroad	5653	1608	373	7634	9.18%	concrete	24821	7534	1191	33546	40.34%
Shoulder Type Left						***Curb Type Left***					
shoulderlnone	4725	1254	586	6565	8.39%	curblnone	3211	1162	197	4570	27.43%
shoulderleft	51941	16290	3459	71690	91.61%	curbleft	9225	2518	348	12091	72.57%
Shoulder Type Right						***Curb Type Right***					
shoulderrnone	5813	1964	755	8532	10.19%	curbrnone	3754	1239	234	5227	29.12%
shoulderright	54458	17180	3573	75211	89.81%	curbright	9721	2636	368	12725	70.88%

Road Type						***Crash Contributing Factors***					
2 lane, 2 way	9890	3310	1185	14385	17.30%	fatigue	804	386	129	1319	1.59%
4 ormore,divided	43114	13338	2202	58654	70.54%	drug	100	84	88	272	0.33%
4 or more,undivided	7355	2189	454	9998	12.02%	alcohol	348	235	167	750	0.90%

3.2 Study Design Approaches

This research was designed to predict the severity level of the large truck crash based on the comparison of different classification models. For this purpose, a historical crash data analysis was conducted. At first, historical crash records for the entire texas state from 2016 to 2019 were extracted from Texas Crash Record Information System (CRIS); After that, the final dataset used in the study was determined by carefully conducting variable selection, data cleaning, and data preprocessing based on the originally extracted dataset; Then the cleaned dataset was divided into a dedicated training dataset (contains records from the year 2016 to 2018), and a dedicated testing dataset (contains records of the year 2019). As shown in Figure 1, the cleaned dataset is named the training dataset, three resampling techniques including random undersampling, oversampling, and mix sampling were used in the training dataset to create correspondingly three balanced datasets. Add up with the original dataset, which is kept the same as the training dataset, a total of four datasets were used to develop different prediction models. Since seven classifiers were selected in this study, combining with the four datasets, a total of twenty-eight prediction models were developed. In this way, the effects of class balancing techniques in model prediction performance were tested. Finally, the final best performance of four classification tree-based ML models (XGBoost, AdaBoost, RF, GBDT), two non-tree-based ML models(SVM, k-NN), and LR in crash severity prediction can be compared.

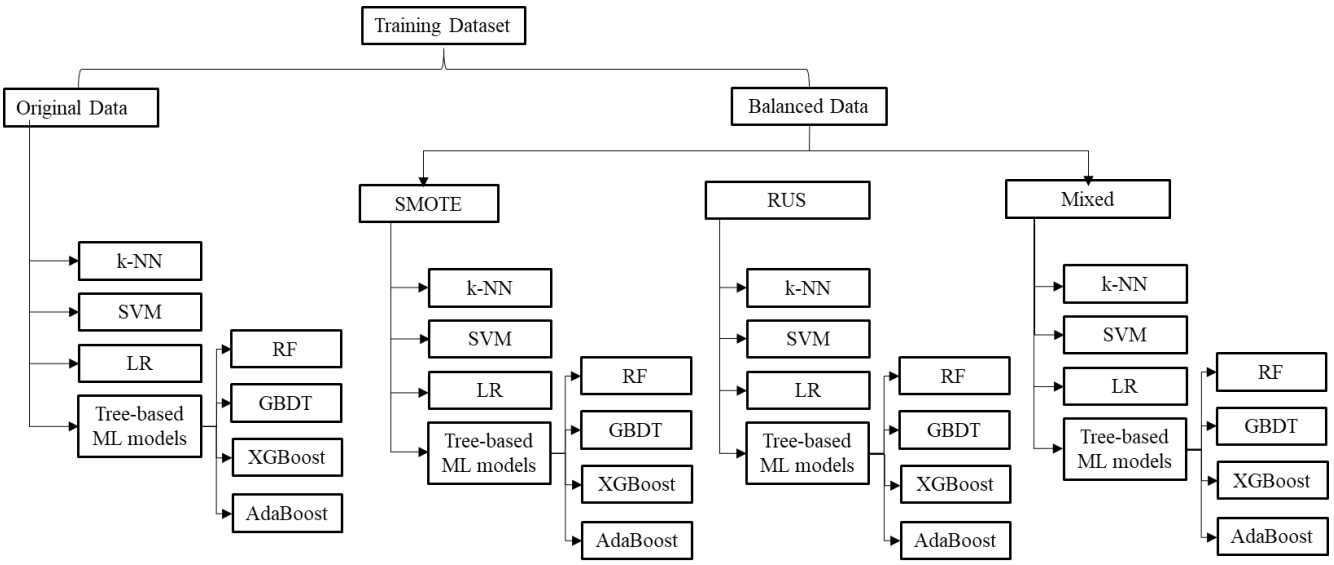


Figure 2. Study Procedure

3.3 Methodology

As mentioned in the literature review, a variety of traditional regression methods have been applied to predict crash severity, including the traditional regression models, such as the logistic regression model (or multinomial logit model), ordered logit model, and so on. The logistic regression model is a widely used regression model for the severity prediction of crash injury. Many studies have found that the logistic regression model could achieve a closer estimation of the crash probabilities to the observations (Iranitalab and Khattak, 2017; Zhang et al., 2018). Thus, in this study, the logistic regression model was chosen to compare with the ML models. Besides, the classification tree-based ML methods have been widely employed for crash risk prediction and identification of contributing factors (Jiang et al., 2016, Lu et al., 2020, and Zhou et al., 2020). Note that classification tree-based algorithms usually fall into the scope of ensemble learning-a machine learning paradigm where multiple models (often called “weak learners”) are trained to solve the same problem and combined to get better results. Ensembles are often recognized as the algorithm-level approaches to handle the classification problem for the class-imbalance dataset. Some researches suggest that ensemble algorithms work better for imbalanced data than others

(Błaszczyński and Stefanowski, 2015). Besides, the Support Vector Machines (SVM), k-Nearest Neighbors (kNN) are also among the most popular non-tree-based ML techniques that have been widely selected for crash severity prediction (Rivera et al., 2020; Abou Elassad et al., 2020). Therefore, there is a need to find out if the classification tree-based methods can effectively and correctly predict crash severity than non-tree-based ML models and what is the prediction difference between these ML models and the logistic regression model. In this study, four representative classification tree-based ML models (e.g., Extreme Gradient Boosting tree (XGBoost), Adaptive Boosting tree(AdaBoost), Random Forest (RF), Gradient Boost Decision Tree (GBDT)), two non-tree-based ML models (e.g., the Support Vector Machines (SVM), k-Nearest Neighbors (kNN)), and LR model were selected for developing models for crash severity prediction.

3.3.1 Testing of Different Resampling Techniques

High imbalance often occurs in practical applications where the minority one is often rare but important. Take the traffic accident datasets as an example, the instances of fatal crashes often much fewer than the PDO crash. Some researchers believe that in such cases classifiers tend to be overwhelmed by the majority classes and overlook the minority ones(Kotsiantis et al. 2006). To be more specific, classifiers tend to produce high predictive performance over the majority class, but poor predictive performance over the minority class. A number of solutions to the class-imbalance problem were previously proposed both at the data and algorithmic levels. Compared with the algorithmic level approach, the data level approach (preprocessing approach) seems to be the more straightforward approach (Thammasiri et al., 2014). In this consideration, resampling approaches are extensively studied to diminish the class imbalance problem before developing classification models (García et al., 2020). Resampling techniques are essentially data

preprocessing methods that aim to balance different classes (Thammasiri et al., 2014). According to the results of the literature review, some researchers suggest resampling is an effective approach in improving the prediction performances of minority crashes (Mujalli et al, 2016; Fiorentini et al., 2020), while others suggest that when the distribution of the classes in the population is known, the user should choose a sample that has the same distribution as the population to ensure optimal performance ((Oommen et al., 2010).

Therefore, to testify the effectiveness of sampling balancing techniques in detecting the severity level of the large truck crash, three commonly used resampling approaches were selected to balancing the datasets: Synthetic minority oversampling technique (SMOTE), Random undersampling (RUS), and mixed techniques.

- Synthetic minority oversampling technique (SMOTE): a heuristic method that creates synthetic instances of the minority class using the k-Nearest Neighbors approach within a bootstrapping procedure until the dataset is balanced. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen (Chawla et al., 2002). Moreover, SMOTE can be used for handling both continuous and categorical features.
- Random undersampling (RUS): a non-heuristic method that aims to balance the class distribution by randomly eliminating the number of instances of the majority class until the dataset is balanced. The major disadvantage of RUS is that it can delete potentially useful instances that could be important for data analysis. (Kotsiantis et al., 2006).

- The mixed techniques: this method combines both SMOTE and RUS techniques. In this method, the instance number of minority class is increased while the instance number of majority class are discarded until the number of instances of each class is the same, while the dataset size remains the same as the original dataset size (Witten and Frank, 2005).

These three resampling techniques are performed in the program python, the package "imbalanced-learn" is used.

3.3.2 Regression Model

3.3.2.1 The Logistic Regression model (LR)

The logistic regression model is the most widely used discrete choice model (Train, 2009) and has a long history of use in crash severity analysis literature. When the logistic regression is multinomial. Multinomial logistic regression is used for the multi-class response variables. In a multinomial logit model of crash injury severity outcomes, the propensity of crash *i* towards severity category *k* is represented by severity propensity function, T_{ki}, as shown in Equation (1) (Kim et al., 2008).

$$T_{ki} = \alpha_k + \boldsymbol{\beta}_k \mathbf{X}_{ki} + \varepsilon_{ki} \quad (1)$$

Where, α_k is a constant parameter for crash severity category *k*; $\boldsymbol{\beta}_k$ is a vector of the parameters for crash severity category *k*; *k*=1, 2, …, K (K=3 in the paper) representing all the three severity levels: Property Damage Only (PDO), Slight Injuries (SLIG), and accidents with Killed or Severe Injuries (KSEV); $\mathbf{X}_{ki}$ represents a vector of independent variables (risk contributing factors) affecting the crash severity for *i* at severity category *k*; ε_{ki} is a random error term that accounts for unobserved effects following the Type I generalized extreme value (i.e., Gumbel) distribution; *i*=1, 2, …, n where n is the total number of crash events included in the model.

If $P_i(k)$ is the probability of accident *i* ending in crash severity category *k*, then:

$$P_i(k) = \frac{e^{(\alpha_k + \boldsymbol{\beta}_k \mathbf{X}_{ki})}}{\sum_{\forall k} e^{(\alpha_k + \boldsymbol{\beta}_k \mathbf{X}_{ki})}} \quad (2)$$

In this study, the python interface to Logistic Regression, available through package Logistic Regression from sklearn is used.

3.3.3 Machine Learning Models

3.3.3.1 Random Forest (RF)

In Random Forests (RF) method, each tree in the ensemble is built from a sample drawn with replacement from the training set. The method combines Brieman's bagging idea and Ho's "random subspace method" to construct a collection of decision trees with various sub-sample of the dataset (Breiman, 2001; Ho, 1995). A predetermined number of classification trees are generated from the bootstrap sample and combined to give a final prediction. In this study, the model combines classifiers by averaging their probabilistic prediction, instead of letting each classifier vote for a single class. The performance of an RF can be improved by minimizing the bias of each tree and the correlations among trees. To minimize the bias, each tree should be grown to maximum depth based on Gini index (Breiman, 1984).

In this study, the input samples for RF are represented as $x = \{[x_{i1}, x_{i2}, \ldots, x_{in}], y_i\}$, *i*=1,2,3…, *m* and *m* indicates the number of crash samples, *n* is the number of independent variables. The values of dependent variable y (*y*=0,1,or 2) correspond to different levels of crash severity. The output is the probability of a single sample belongs to different severity levels. The RF algorithm includes three basic calculation processes: sample set selection (bootstrap samples), decision tree generation and decision tree combination.

The python interface to RF, available through package RandomForestClassifier from sklearn is used.

3.3.3.2 Adaptive Boosting (AdaBoost)

The basic idea of the AdaBoost algorithm is to combine a sequence of weak learners through a weighted majority vote (or sum) to make classifications. It repeatedly updated the data by taking the previous weak learners' mistakes into account. The basic steps of this algorithm can be explained as follows (Chen, 2015).

Given a classification training data set $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$, a strong classifier $C(x)$ generated by the following steps:

Initialization of the weight value distribution of the training data,

$$W_1 = (w_{11}, \cdots, w_{1i}, \cdots, w_{1N},),\ w_{1i} = \frac{1}{N},\ i = 1,2, \cdots, N,\ m=1,2,\cdots,M \text{(m is the times of iteration)} \quad (3)$$

Using the training data set has the weight distribution W_m to learn, get the basic classification $C_m(x)$ according to the Gini indexes of different influencing factors k

The classification error rate of $C_m(x)$ is calculated as follows

$$e_m = P(C_m(x) \neq y_i) = \sum_{i=1}^{N} w_{mi}\, I\big(C_m(x)\big) \neq y_i \quad (4)$$

Calculation the "amount of say", a_m of $C_m(x)$ according to its classification error e_m

$$a_m = \frac{1}{2}\log\frac{1-e_m}{e_m} \quad (5)$$

Update the weight distribution based on the calculated "amount of say", a_m

$$W_m = \big(w_{m+1,1}, \cdots, w_{m+1,i}, \cdots, w_{m+1,N}\big) \quad (6)$$

$$w_{m+1,i} = \frac{w_{mi}}{z_m}\exp\big(-a_m y_i C_m(x_i)\big) \quad (7)$$

where, Z_m is normalization factor which could make the sum of W_m equal to 1.

Calculate the weighted sum of all the classifiers

$$f(x) = \sum_{m=1}^{M} a_m C_m(x) \quad (8)$$

The final strong classifier can be expressed as

$$C(x) = signf(x) = sign(\sum_{m=1}^{M} a_m C_m(x)) \quad (9)$$

The python interface to AdaBoost, available through package AdaBoostClassifier from sklearn is used.

3.3.3.3 Gradient Boosting Decision Tree (GBDT)

GBDT is a generalization of boosting to arbitrary differentiable loss functions. The motivation is to combine several weak models to produce a powerful ensembl. Assume that F(x) is an approximation function of the dependent variable y based on a set of independent variables x. F(x) can be expressed as $F(x) = \sum_{m=1}^{M} \gamma_m h_m(x)$, where $h_m(x)$ are the basis functions, which are usually called weak learners in the context of boosting. The loss function can be defined as, $L\big(y, F(x)\big) = \log\big(1 + e^{-yF(x)}\big)$.

Similar to other boosting algorithms, GBDT builds the additive model in a greedy fashion:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \quad (10)$$

where the newly added tree h_m tries to minimize the loss L, given the previous ensemble $F_{m-1}(x)$:

$$h_m = \underset{h}{\arg\min} \sum_{i=1}^{n} L\big(y_i, F_{m-1}(x_i) + h(x_i)\big) \quad (11)$$

The initial model F_0 is problem-specific; for the least-squares regression, one usually chooses the mean of the target values.

Gradient boosting attempts to solve this minimization problem numerically via steepest descent:

$$F_m(x) = F_{m-1}(x) - \gamma_m \sum_{i=1}^{n} \nabla_F L(y_i, F_{m-1}(x_i)) \quad (12)$$

where the step length γ_m is chosen using the line search:

$$\gamma_m = \underset{\gamma}{\arg\min} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) - \gamma \frac{\partial L\big(y_i, F_{m-1}(x_i)\big)}{\partial F_{m-1}(x_i)}\right) \quad (13)$$

3.3.3.4 Extreme Gradient Boosting (XGBoost)

The Extreme Gradient Boosting (XGBoost) is a variant of the gradient boosted regression trees. The algorithm is based on the boosting idea, which combines an ensemble of weak learners into a single strong model through iteratively improving the ensemble learner. XGBoost relies on training new models to the gradient of the loss function. Due to a number of optimizations- simplifying the objective functions but maintaining the optimal computational speed, XGBoost is a very fast and efficient tree boosting algorithm (Chen and Guestrin, 2016).

The processes of additive learning in XGBoost are explained below. The first learner is fitted based on the whole space of input data, then according to the residuals of the first learner, a second learner is then fitted for tackling the drawbacks of the first weak learner. The ultimate prediction of the model is obtained by the sum of the prediction of each learner. The general function for the prediction at step *t* is presented as follows:

$$f_i^{(t)} = \sum_{k=1}^{n} f_k\,(x_i) = f_i^{t-1} + f_t(x_i) \tag{14}$$

Where $f_t(x_i)$ is the learner at step t, f_i^{t-1} and $f_i^{(t)}$ are the predictions at step *t-1* and *t*, and x_i is the input variable.

To preventing over-fitting issue without compromising the computational speed, the analytic expression below is used to evaluate the "goodness" of the model from the original function:

$$Obj^{(t)} = \sum_{k=1}^{n} l(\vec{y}_i, y_i) + \sum_{k=1}^{t} \Omega(f_t) \tag{15}$$

Where *l* is the loss function, *n* is the number of observations used and Ω is the regulation term and defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\|\vec{w}\|^2 \tag{16}$$

where $\vec{w}$ is the vector of scores in the leaves, λ is the regularization parameter, and γ is the minimum loss needed to further partition the leaf node.

The python interface to XGBoost, available through package xgboost is used.

3.3.3.5 Support Vector Machine (SVM)

Developed by Vapnik (Vapnik, 2013), a support vector machine (SVM) is a supervised binary linear classifier that can be used to solve a classification problem by constructing hyperplanes in a way that the resulting gaps between classes exhibit margins that are as large as possible (Cristianini and Shawe-Taylor, 2000; Vapnik, 2000, 1998). Let us consider a training set represented by$\{(x_i, d_i)\}_{i=1}^{N}$,where x_i is the *n*-dimensional dependent variables and d_i represents the independent variable d_i=1 represents the positive group and the independent variable d_i=-1 represents the negative group. SVM maps each point x_i from the input space *n* to the feature space *H* by means of the mapping function $\Phi(\vec{x}_i)$ and finds a linear decision surface to separate the negative data points from the positive ones in the feature space. The linear decision surface is defined as

$$\vec{\mathrm{w}} \cdot \Phi(\vec{x}_i) + b=0 \tag{17}$$

$$\text{s.t. } d_i(\vec{\mathrm{w}} \cdot \Phi(\vec{x}_i) + b) \geq 1 \tag{18}$$

where the $\vec{\mathrm{w}}$ is a vector perpendicular to the decision surface and *b* is a decision surface bias. In order to maximize the margin of separation between the classes ($\frac{2}{\|\vec{w}\|}$ or equivalent to minimize $\frac{1}{2}\|\vec{w}\|^2$), SVM constructs a unique decision surface by applying Lagrange multiplier and transforming it into the following dual problem:

$$\min_{\lambda} \left(\frac{1}{2}\sum_{j,k=1}^{N} \lambda_j\, \lambda_k y_j y_k K\left(\vec{x}_j, \vec{x}_k\right) - \sum_{j=1}^{N} \lambda_i\right) \tag{19}$$

$$\text{Subject to } \sum_{i=1}^{N} \lambda_i\, y_i = 0 \; and \; 0 \leq \lambda_i \leq C$$

Where $\lambda = (\lambda_1, \cdots, \lambda_N)$ is the Lagrange multiplier, C is a constant parameter that determines the tradeoff between the maximum margin and minimum classification error. $K(.,.)$ is denoted as $K(\vec{x}_j, \vec{x}_k) = \Phi(\vec{x}_j) \cdot \Phi(\vec{x}_k)$, which is the so-called kernel function. By using kernel function, SVM does not need to know explicitly the mapping function $\Phi(\vec{x}_i)$; it is sufficient only to know the dot product between the mapping of two data points. Having determined the optimum Lagrange multiplier, the optimum solution for the vector $\overrightarrow{\mathrm{w}}$ is given by:

$$\overrightarrow{\mathrm{w}} = \sum_{j=1}^{N} \lambda_j \, y_j \Phi(\vec{x}_j) \tag{20}$$

Then SVM is able to classify any input $\vec{x}$ using the function:

$$f(\vec{x}) = sign(\vec{w} \cdot \Phi(\vec{x}_i) + b) = sign\left(\sum_{j=1}^{N} \lambda_j \, y_j K(\vec{x}_j, \vec{x}_k) + b\right) \tag{21}$$

The python interface to SVM, available through package SVM from sklearn is used.

3.3.3.6 k-Nearest neighbor (k-NN)

k-Nearest neighbor (k-NN) classifier is conventional non-parametric classifier (Cover and Hart 1967). Instances are represented by some feature vectors as a point in the feature space. To classify one instance, the k-NN classifier calculates the distances between the point and points in the training data set. In this study, the Euclidean distance is used to measure the distance. Then, it assigns the point to the class among its k nearest neighbours (where k is an integer). Figure 3 illustrates this concept. where * represents the new data point. If k = 3, the point belongs to class A; if k = 5, the point belong to class B.

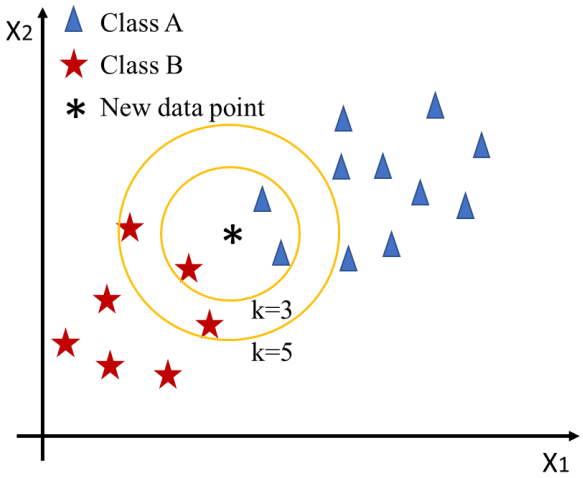


Figure 3. k-Nearest Neighbor (k-NN) Classifier

To measure the distance between points A and B in a feature space, various distance functions have been used in the literature, in which the Euclidean distance function is the most widely used one. Let A and B are represented by feature vectors A = (x_1, x_2, . . . , x_m) and B = (y_1, y_2, . . . , y_m), where m is the dimensionality of the feature space. To calculate the distance between A and B, the normalized Euclidean metric is generally used by

$$dist(A,B) = \sqrt{\frac{\sum_{i=1}^{m}(x_i - y_i)^2}{m}} \tag{22}$$

The python interface to k-NN, available through package Nearest Neighbors from sklearn is used.

3.4 Prediction Evaluation Measures

In imbalanced learning, there were mainly two types of evaluation measures. One is the threshold-based measures, like sensitivity, precision, specificity, and F-measure, which means these measures rely on one specific threshold. The other is non-threshold-based measures, like ROC-AUC and PR-AUC (Fernández et al. 2018), which will be explained in the following paragraph. In this study, the dependent variable was categorized into three levels: PDO, SLIG, and

KSEV, based on this class division condition, the detailed description of these threshold-based metrics are summarized in Table 3

Table 3

Threshold-based Evaluation Metrics

Metric	Formula
Sensitivity or recall	$R(PDO/SLIG/KSEV) = \frac{\#\ of\ crashes\ correctly\ predicted\ as\ PDO/SLIG/KSEV}{Total\ actual\ PDO/SLIG/KSEV\ crashes}$
Precision	$P(PDO/SLIG/KSEV) = \frac{\#\ of\ crashes\ correctly\ predicted\ as\ PDO/SLIG/KSEV}{Total\ predicted\ PDO/SLIG/KSEV\ crashes}$
F1-score	$F(PDO/SLIG/KSEV) = \frac{2 * P(PDO/SLIG/KSEV) * R(PDO/SLIG/KSEV)}{P(PDO/SLIG/KSEV) + R(PDO/SLIG/KSEV)}$
Specificity	$S(PDO) = \frac{\#\ of\ crashes\ correctly\ predicted\ as\ SLIG\ and\ KSEV}{Total\ predicted\ SLIG\ and\ KSEV\ crashes}$ $S(SLIG) = \frac{\#\ of\ crashes\ correctly\ predicted\ as\ PDO\ and\ KSEV}{Total\ predicted\ PDO\ and\ KSEV\ crashes}$ $S(KSEV) = \frac{\#\ of\ crashes\ correctly\ predicted\ as\ PDO\ and\ SLIG}{Total\ \text{predicted}\ PDO\ and\ SLIG\ crashes}$

As it is often the case that accident severity datasets are typically imbalanced, thus there is usually a trade-off between Sensitivity and Specificity (Jeong et al. 2018). For example, in a two-level crash severity classification problem where the instances of the non-AK crash is much more than the instances of AK crash (non-AK crash takes 99% of all instances). There is one possible extreme situation that the model classifies all accidents to be non-AK, such a model would have a very high recall rate, while it exhibits a very low specificity rate. This is often the case, when the training dataset is rebalanced using resampling techniques, the specificity rate can be improved while the recall rate will be compromised. Besides, all these metrics are decided by one threshold,

which means these metrics cannot present an overall performance, thus result in failing to be informative in reality.

While, ROC-AUC, which is calculated in function of the threshold metrics (Rivera et al. 2020). The ROC-Receiver Operating Characteristic-is the curve formed when the transversal axis represents the false positive rate (1-specificity), and the longitudinal axis represents the true positive rate (sensitivity) for different cut-off points. ROC is a probability distribution, and its area under the curve (AUC) represents the degree of separability between classes. With a maximum of ROC-AUC value close to 1 describing that the classifier has an excellent performance in separating classes, and a value close to 0.5 describing a valueless test. PR-AUC. Like the ROC curve, the PR (Precision-Recall) curve is a plot of the precision (y-axis) and the recall (x-axis) for different probability thresholds. ROC-AUC does not place more emphasis on one class over the other, so it is not biased against the minority class (Kotsiantis et al, 2006). Besides, some researchers suggest that for the evaluation of probabilistic models, ROC-AUC is recommended to evaluate the separability between the classes (Oommen et al., 2010). Therefore, in this study, ROC-AUC is selected as the evaluation measure for prediction performance in classifying large truck crash severity, in the following part, ROC-AUC will be simplified to be AUC.

CHAPTER 4

RESULTS ANALYSIS

In this chapter, the effects of resampling techniques are tested first. Then, the final results of different prediction models are presented and discussed. Below, we present the experiments conducted to estimate (a) the effects of sampling balancing techniques, (b) the performance of the classifiers for identifying crash severities.

All the models are programmed in Python (version 3.7), using scikit-learn 0.22, imbalanced-learn 0.5, XGBoost 1.4.0, pandas 0.25.3, matplotlib 3.1.2, numpy 1.17.4.

4.1 Imbalanced versus Balanced Training Datasets

An analysis of challenging real-world classification problems still reveals difficulties in finding accurate classifiers. One of the sources of these difficulties is class imbalance in data, where at least one of the target classes contains a much smaller number of examples than the other classes. This section aims to investigate the effects of sample balancing techniques in model's prediction ability. The possible resulting differences between balanced and imbalanced (original) datasets are measured by applying eight different prediction models - LR, OP, RF, AdaBoost, GBDT, XGBoost, SVM, and k-NN and their resulting performance evaluation measures. To achieve this goal, three balanced datasets were created based on the original imbalanced datasets using three sampling strategies namely RUS, SMOTE, and Mixed.

The original training dataset contained 62,066 accidents in which the severity distribution was: 44,905 PDO crashes and 14,159 SLIG crashes, 2,919 KSEV crashes, and in which the dependent variable was predominantly imbalanced. To deal with the imbalanced dataset problem, three new balanced data sets were developed using three different resample techniques: RUS,

SMOTE, and Mixed. Table 4 shows the total number of instances in all the datasets used and their distribution amongst different severity levels.

As shown in Table 4, when the RUS undersampling technique was used, the dataset was reduced to the size of the minority class, in this case to KSEV class (2,925 instances for KSEV as shown in Table 4). While when using SMOTE oversampling, the number of instances in the resulting dataset was increased to the size of the majority class (44,905 instances for PDO class). Finally, in the mixed sampling, the resulting dataset preserved the original number of instances (61,983 accidents), the instance of the majority class was reduced to 20,661 and the instance of minority class was increased to 20,66.

Table 4

Number of Instances in Original and Balanced Training Datasets

Datasets		Total	PDO	SLIG	KSEV
Original dataset		61,983	44,905	14,159	2,919
Balanced datasets	SMOTE	134,715	44,905	44,905	44,905
	RUS	8,757	2,919	2,919	2,919
	Mixed	61,983	20,661	20,661	20,661

Seven classifiers described in Chapter 5 were used to build different models. For each training dataset (original, SMOTE, RUS, and mix), seven models were developed. Firstly, each training dataset was used to train the model, and then the testing dataset was used to test the model's prediction performance. All the parameters for each model were optimized separately through the function GridSearchCV from scikit-learn until the best AUC score was reached. The testing results of classifiers developed from balanced datasets were then compared with those developed from the original dataset. In order to perform this comparison, the results of the AUC used to compare the models developed from different datasets are summarized in Table 5. The comparison is based on the performance measures of AUC. An AUC value close to 1 indicates that the classifier has excellent performance when separating classes, and a value close to 0.5

indicates that the classifier cannot discriminate classes correctly. With respect to the results obtained by the testing set, the following findings were extracted:

1) For the tree-based classifier (XGBoost, AdaBoost, RF, and GBDT), the overall results indicate that the original dataset works better in predicting all three levels of severity when compared to the balanced datasets. Look at the XGBoost classifier first, the highest prediction performance of XGBoost classifier is evaluated as 0.59 for PDO level crash, which means the original dataset performed better than the balanced datasets in terms of its ability to classify the PDO level crash. Besides, the best models for SLIG and KSEV level crash were also obtained using the original dataset. The above results suggested that original datasets performed better than the other three balanced datasets in all three levels of crash severity prediction. Furthermore, it indicates that using the original dataset, meaning dataset with original population, to train the XGBoost model, the model will produce better prediction performance than using the datasets with revised sample population. Similar results are obtained for GBDT and RF classifiers. For the AdaBoost classifier, the highest prediction performance is evaluated as 0.58 for PDO level crash using the original dataset. And better prediction performance of SLIG level crash is achieved by trained in the original dataset and rebalanced dataset based on SMOTE technique. As for KSEV level crash, better performance is achieved by in the original dataset and rebalanced dataset obtained through the RUS technique. Consistent results can be found in Liu's research, where some experimental study was conducted showing that ensembles specialized for class imbalance should work better than an approach consisting of first pre-processing data and then using ensembles (Liu et al., 2013).

2) For non-tree-based classifiers (k-NN and SVM), the original dataset also works better. Look at the k-NN classifier, the original dataset works better than the balanced datasets in

SLIG and KSEV level prediction. Only the SMOTE dataset produced a relatively better PDO level prediction than the original dataset. The overall results indicate that the original dataset works better in predicting most levels of severity when compared to the balanced datasets. Similar results are obtained for the SVM classifier. Some recent studies on class imbalances have shown that the global imbalanced ratio between classes is not a problem itself. For some data sets with high imbalance ratio, the minority class can still be sufficiently recognized even by standard classifiers. The degradation of classification performance is often linked to other difficulty factors related to data distribution, such as decomposition of the minority class into many rare sub-concepts playing a role of small disjuncts (Ting1994; Weiss and Hirsh2000), the effect of too strong overlapping between the classes (Garcia et al. 2007), or the presence of too many minority examples inside the majority class regions (Napierala et al. 2010).

3) For LR classifier, using the balanced dataset technique to train the model showed an improvement in prediction performances in contrast to using the original dataset, and the balanced dataset acquired from SMOTE approach showed the best performance. A similar result was found in Salas-Eljatib's research, where the data balancing technique was proved to improve the prediction capability of the LR model (Salas-Eljatib et al., 2018).

Table 5

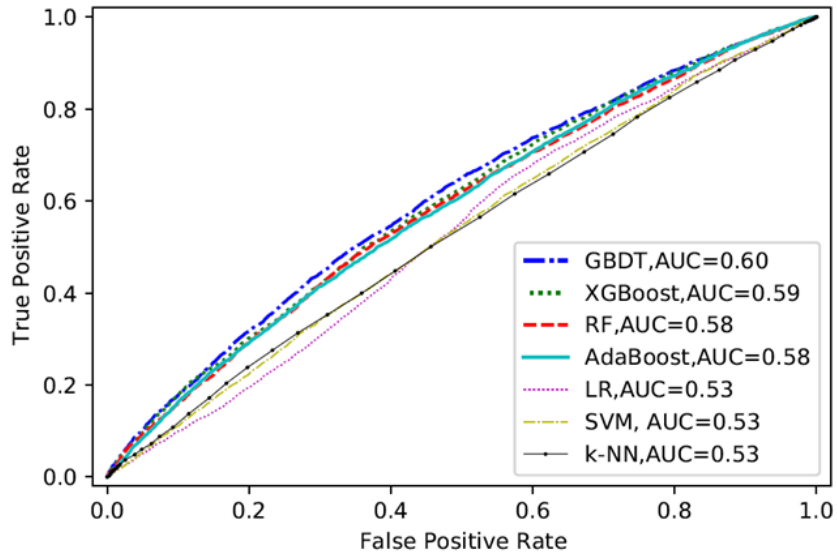
Overview of AUC Using Different Datasets

Severity levels	Datasets	AUC						
		XGBoost	GBDT	RF	AdaBoost	k-NN	SVM	LR
PDO	Original	**0.59**	**0.60**	**0.58**	**0.58**	0.53	**0.53**	0.47
	SMOTE	0.57	0.57	0.57	0.55	**0.55**	0.51	**0.53**
	RUS	0.57	0.58	0.55	0.57	0.51	**0.53**	**0.53**
	Mix	0.53	0.53	0.55	0.53	0.52	0.50	0.52
SLIG	Original	**0.57**	**0.58**	**0.56**	**0.51**	**0.53**	**0.54**	0.48
	SMOTE	0.55	0.55	0.52	**0.51**	0.52	0.51	**0.51**
	RUS	0.50	0.51	0.51	0.50	0.51	0.52	0.50
	Mix	0.52	052	0.53	0.49	0.50	0.50	**0.51**
KSEV	Original	**0.72**	**0.72**	**0.70**	**0.71**	**0.62**	0.51	0.52
	SMOTE	0.70	0.69	**0.70**	0.67	0.61	0.50	**0.66**
	RUS	0.71	0.72	**0.70**	**0.71**	**0.62**	0.51	**0.66**
	Mix	0.63	0.62	0.67	0.63	0.57	**0.55**	**0.66**

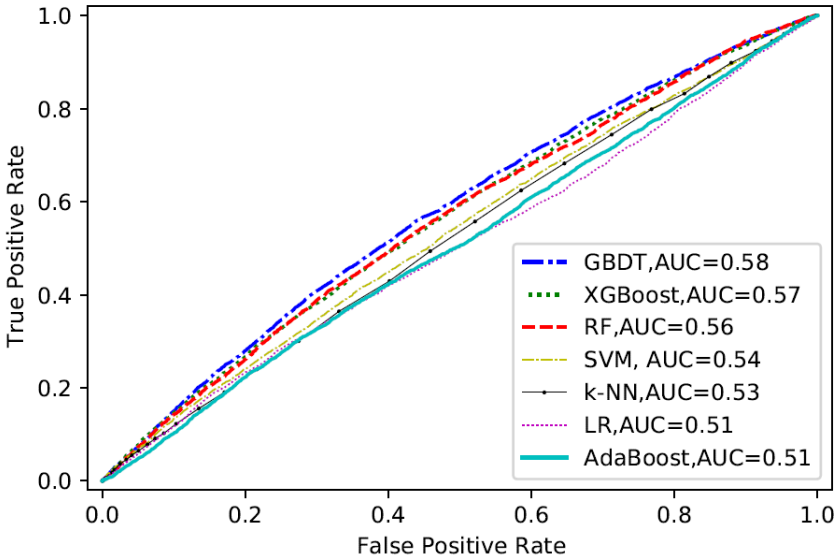
Based on the above results, since there is no improvement achieved by resampling the training dataset for ML-based models, the original dataset was finally chosen to develop the prediction models for ML-based models. As for the LR model, the data balancing technique showed a prediction improvement in all of the three severity levels, and SMOTE sampling, which obtained the most encouraging results among the sampling algorithms tested, was selected to build the final model.

4.2 Regression versus Machine Learning models

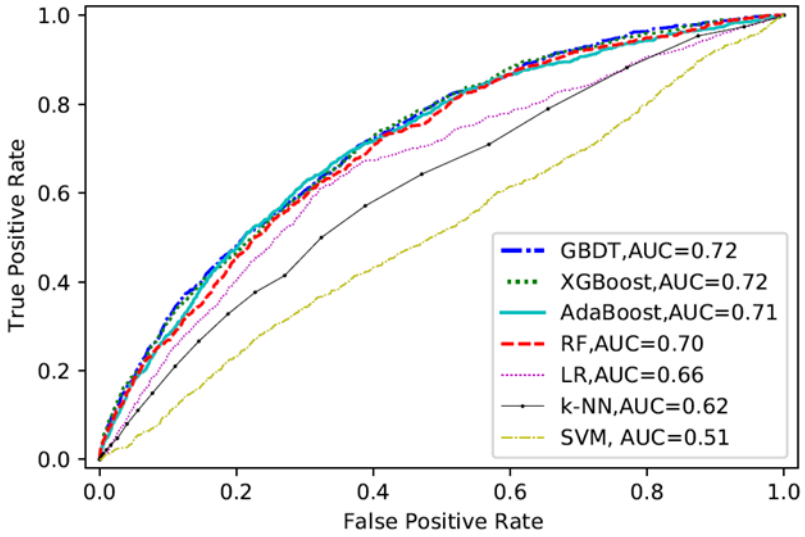
Based on the above results, the final dataset is selected for each model. To make a detailed comparison of the final seven models, Figure 4 presents ROC curves of different severity levels. A ROC-AUC value close to 1 indicates that the classifier has excellent performance when separating classes, and a value close to 0.5 indicates that the classifier cannot discriminate classes correctly.



a) ROC curves of PDO level crash



b) ROC curves of SLIG level crash



c) ROC curves of KSEV level crash

Figure 4. Comparison of Prediction Performance of Different Models

As shown in Figure 4.a) the pattern of these curves indicates that there are two groups, one group consists of XGBoost, AdaBoost, RF, and GBDT, the other group consists of SVM, k-NN, and LR. It means there is a significant difference between these two groups while within these groups, the prediction performance of classifiers are similar. Besides, one of the groups comprised of classification tree-based ML models (XGBoost, AdaBoost, RF, GBDT) is relatively above the none-tree-based ML models (SVM and k-NN) and LR. It indicated that for PDO severity level prediction, the prediction performance between the four tree-based ML models are similar, the prediction performance between SVM, k-NN, and LR models are also similar, and overall, the prediction performance of four tree-based ML models are better. And GBDT is relatively above all the curves.

Similar to Figure 4.a), there are two groups of curves in Figure 4.b). The distance between these two groups are closer than that in Figure 4.a) and one of the tree-based ML method (AdaBoost) showed the relatively low performance, the other three tree-based algorithms (XGBoost, RF and GBDT) still performed well. Still, GBDT showed the best results.

As shown in Figure 4.c). the classification tree-based ML curves (XGBoost, AdaBoost, RF, GBDT) are highly overlapped and above the other three curves. The other three curves are highly separated, and SVM showed the weakest performance.

Overall, all these models relative are good at predicting KSEV level crash, except for SVM, which performs better at predicting SLIG level crash. Besides, the area under the ROC curves (AUC) of the GBDT model is greater than those of the other six models, which indicates that the GBDT model has better prediction performance than the other models. Finally, classification tree-based ML models (XGBoost, AdaBoost, RF, GBDT) are relatively above the none tree-based ML models(SVM, k-NN) and LR at all of the three levels.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

This research was designed to predict the severity level of the large truck crash based on the comparison of different classification models (XGBoost, AdaBoost, RF, GBDT, SVM, k-NN, and LR). For this purpose, a historical crash records for the entire texas state from 2016 to 2019 were extracted from Texas Crash Record Information System (CRIS). In order to determine the appropriate training dataset for each model, three sampling strategies namely RUS, SMOTE, and Mixed are employed to test the effects of data balancing techniques. The following are the key findings of the study, along with some corresponding recommendations:

- XGBoost, GBDT, RF, AdaBoost are classification tree-based ML classifiers. For these four classifiers, the original dataset works better in predicting all three levels of severity when compared to the balanced datasets. For two non-tree-based classifiers(k-NN and SVM), the original dataset also works better, while balancing technique can realize prediction improvement for a certain level of severity.
- For the LR classifier, using the balanced dataset to train the model showed an improvement in prediction performance when compared to the employing of the original dataset, and the balanced dataset acquired from SMOTE approach showed the most promising results.
- All these models are good at predicting KSEV level crash, except for SVM, which performs better at predicting SLIG level crash. The GBDT model performs best among all of the seven models.

- Finally, classification tree-based ML models (XGBoost, AdaBoost, RF, GBDT) perform relatively better than the none tree-based ML models(SVM, k-NN) and LR at all three severity levels.

Overall, the results of this study can help to predict the severity of a reported crash with unknown severity or of the severity of crashes that may be expected to occur sometime in the future. Besides, the modeling procedure can provide some insight into the selection and development of classifiers for large truck crash severity prediction.

More studies concerning the modeling effectiveness analysis of the mixed logit model and the ordered probit model will be conducted to make a full understanding of characteristics of different traditional models. Besides, it is also worth attention that the resampling techniques used in this research is limited, the results of resampling may not be applicable to all kinds of resampling approaches. Furthermore, the author will put more emphasis on how to improve the prediction results by using more advanced parameter optimization strategies and by employing more efficient data cleaning methods.

REFERENCES

1. Abou Elassad, Z. E., Mousannif, H., & Al Moatassime, H. (2020). A proactive decision support system for predicting traffic crash events: A critical analysis of imbalanced class distribution. *Knowledge-Based Systems*, *205*, 106314.
2. Abou Elassad, Z. E., Mousannif, H., & Al Moatassime, H. (2020). A real-time crash prediction fusion framework: An imbalance-aware strategy for collision avoidance systems. *Transportation research part C: emerging technologies*, *118*, 102708.
3. Anyfantis, D., Karagiannopoulos, M., Kotsiantis, S., & Pintelas, P. (2008). Creating ensembles of classifiers by distributing an imbalance dataset to reach balance in each resulting training set. In Proceedings of the IEEE DHMS Conference.
4. Błaszczyński, J., & Stefanowski, J. (2015). Neighbourhood sampling in bagging for imbalanced data. *Neurocomputing*, *150*, 529-542.
5. Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, *49*(2), 1-50.
6. Chang, L. Y., & Chien, J. T. (2013). Analysis of driver injury severity in truck-involved accidents using a non-parametric classification tree model. *Safety science*, *51*(1), 17-22.
7. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.
8. Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, *20*(2), 215-232.

9. Cui, Z., & Gong, G. (2018). The effect of machine learning regression algorithms and sample size on individualized behavioral prediction with functional connectivity features. *Neuroimage*, *178*, 622-637.
10. Dissanayake, S., & Roy, U. (2014). Crash severity analysis of single vehicle run-off-road crashes. *Journal of Transportation Technologies*, *2014*.
11. Duncan, G. J., Magnuson, K. A., & Ludwig, J. (2004). The endogeneity problem in developmental studies. *Research in human development*, *1*(1-2), 59-80.
12. Fernández, A., García, S., del Jesus, M. J., & Herrera, F. (2008). A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems*, *159*(18), 2378-2398.
13. Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets* (Vol. 11). Berlin: Springer.
14. Fiorentini, N., & Losa, M. (2020). Handling imbalanced data in road crash severity prediction by machine learning algorithms. *Infrastructures*, *5*(7), 61.
15. García, V., Sánchez, J. S., Marqués, A. I., Florencia, R., & Rivera, G. (2020). Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data. *Expert Systems with Applications*, *158*, 113026.
16. García, V., Sánchez, J. S., Marqués, A. I., Florencia, R., & Rivera, G. (2020). Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data. *Expert Systems with Applications*, *158*, 113026.
17. García, V., Sánchez, J., & Mollineda, R. (2007, November). An empirical study of the behavior of classifiers on imbalanced and overlapped data sets. In *Iberoamerican Congress on Pattern Recognition* (pp. 397-406). Springer, Berlin, Heidelberg.

18. Greene, W. H. (2000). Econometric analysis 4th edition. *International edition, New Jersey: Prentice Hall*, 201-215.

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

20. Jeong, H., Jang, Y., Bowman, P. J., & Masoud, N. (2018). Classification of motor vehicle crash injury severity: A hybrid approach for imbalanced data. *Accident Analysis & Prevention*, *120*, 250-261.

21. Jeong, H., Jang, Y., Bowman, P. J., & Masoud, N. (2018). Classification of motor vehicle crash injury severity: A hybrid approach for imbalanced data. *Accident Analysis & Prevention*, *120*, 250-261.

22. Kim, J. K., Ulfarsson, G. F., Shankar, V. N., & Kim, S. (2008). Age and pedestrian injury severity in motor-vehicle crashes: A heteroskedastic logit analysis. *Accident Analysis & Prevention*, *40*(5), 1695-1702.

23. Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, *30*(1), 25-36.

24. Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, *30*(1), 25-36.

25. Liu, X. Y., & Zhou, Z. H. (2013). Ensemble methods for class imbalance learning. *H. He, & Y. Ma, Imbalanced Learning: Foundations, Algorithms, and Applications*, 61-82.

26. Lu, P., Zheng, Z., Ren, Y., Zhou, X., Keramati, A., Tolliver, D., & Huang, Y. (2020). A gradient boosting crash prediction approach for highway-rail grade crossing crash analysis. *Journal of advanced transportation*, *2020*.

27. Moghaddam, Davoud Davoudi, et al. "The effect of sample size on different machine learning models for groundwater potential mapping in mountain bedrock aquifers." *Catena* 187 (2020): 104421.

28. Mujalli, R. O., López, G., & Garach, L. (2016). Bayes classifiers for imbalanced traffic accidents datasets. *Accident Analysis & Prevention*, *88*, 37-51.

29. Mujalli, R. O., López, G., & Garach, L. (2016). Bayes classifiers for imbalanced traffic accidents datasets. *Accident Analysis & Prevention*, *88*, 37-51.

30. Napierala, K., & Stefanowski, J. (2012, March). Identification of different types of minority class examples in imbalanced data. In *International Conference on Hybrid Artificial Intelligence Systems* (pp. 139-150). Springer, Berlin, Heidelberg.

31. Oommen, T., Baise, L. G., & Vogel, R. M. (2011). Sampling bias and class imbalance in maximum-likelihood logistic regression. *Mathematical Geosciences*, *43*(1), 99-120.

32. Pahukula, J., Hernandez, S., & Unnikrishnan, A. (2015). A time of day analysis of crashes involving large trucks in urban areas. *Accident Analysis & Prevention*, *75*, 155-163.

33. Rivera, G., Florencia, R., García, V., Ruiz, A., & Sánchez-Solís, J. P. (2020). News classification for identifying traffic incident points in a spanish-speaking country: A real-world case study of class imbalance learning. *Applied Sciences*, *10*(18), 6253.

34. Rivera, G., Florencia, R., García, V., Ruiz, A., & Sánchez-Solís, J. P. (2020). News Classification for Identifying Traffic Incident Points in a Spanish-Speaking Country: A Real-World Case Study of Class Imbalance Learning. *Applied Sciences*, *10*(18), 6253.

35. Salas-Eljatib, C., Fuentes-Ramirez, A., Gregoire, T. G., Altamirano, A., & Yaitul, V. (2018). A study on the effects of unbalanced data when fitting logistic regression models in ecology. *Ecological Indicators*, *85*, 502-508.

36. Schlögl, M., Stütz, R., Laaha, G., & Melcher, M. (2019). A comparison of statistical learning methods for deriving determining factors of accident occurrence from an imbalanced high resolution dataset. *Accident Analysis & Prevention*, *127*, 134-149.

37. Stockwell, David RB, and A. Townsend Peterson. "Effects of sample size on accuracy of species distribution models." *Ecological modelling* 148.1 (2002): 1-13.

38. Su, X.; Zhou, T.; Yan, X.; Fan, J.; Yang, S. Interaction trees with censored survival data. *Int. J. Biostat*. 2008, 4, 1–28.

39. Tang, J., Liang, J., Han, C., Li, Z., and Huang, H. (2019). Crash injury severity analysis using a two-layer stacking framework. *Accident Analysis and Prevention*, 122, 226-238.

40. Thammasiri, D., Delen, D., Meesad, P., & Kasap, N. (2014). A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition. *Expert Systems with Applications*, *41*(2), 321-330.

41. Thammasiri, D., Delen, D., Meesad, P., & Kasap, N. (2014). A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition. *Expert Systems with Applications*, *41*(2), 321-330.

42. Ting, K. M. (1994). The problem of small disjuncts: its remedy in decision trees. In *PROCEEDINGS OF THE BIENNIAL CONFERENCE-CANADIAN SOCIETY FOR COMPUTATIONAL STUDIES OF INTELLIGENCE* (pp. 91-98). CANADIAN INFORMATION PROCESSING SOCIETY.

43. Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge university press.

44. Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

45. Weiss, G. M., & Hirsh, H. (2000). A quantitative study of small disjuncts. *AAAI/IAAI*, *2000*, 665-670.

46. Ye, Fan, and Dominique Lord. "Comparing three commonly used crash severity models on sample size requirements: Multinomial logit, ordered probit and mixed logit models." *Analytic methods in accident research* 1 (2014): 72-85.

47. Yu, R., & Abdel-Aty, M. (2014). Analyzing crash injury severity for a mountainous freeway incorporating real-time traffic and weather data. *Safety science*, *63*, 50-56.

48. Yu, R.; Abdel-Aty, M. Utilizing support vector machine in real-time crash risk evaluation. *Accid. Anal. Prev*. 2013, 51, 252–259.

49. Zhang, J., Li, Z., Pu, Z., & Xu, C. (2018). Comparing prediction performance for crash injury severity among various machine learning and statistical methods. *IEEE Access*, *6*, 60079-60087.

50. Zhao, Q., Goodman, T., Azimi, M., & Qi, Y. (2018). Roadway-related truck crash risk analysis: case studies in Texas. *Transportation research record*, *2672*(34), 20-28.

51. Zhu, X., & Srinivasan, S. (2011). A comprehensive analysis of factors influencing the injury severity of large-truck crashes. *Accident Analysis & Prevention*, *43*(1), 49-57.