International Journal of
Environmental Research and Public Health

MDPI

Article

Crash Injury Severity Prediction Using an Ordinal Classification Machine Learning Approach

Shengxue Zhu [1], Ke Wang [2,*] and Chongyi Li [2]

[1] Jiangsu Key Laboratory of Traffic and Transportation Security, Huaiyin Institute of Technology, Huaian 223003, China; zsx10316@hyit.edu.cn

[2] Key Laboratory of Road and Traffic Engineering of the State Ministry of Education, College of Transportation Engineering, Tongji University, Shanghai 201804, China; 1731304@tongji.edu.cn

[*] Correspondence: kew@tongji.edu.cn

Abstract: In many related works, nominal classification algorithms ignore the order between injury severity levels and make sub-optimal predictions. Existing ordinal classification methods suffer rank inconsistency and rank non-monotonicity. The aim of this paper is to propose an ordinal classification approach to predict traffic crash injury severity and to test its performance over existing machine learning classification methods. First, we compare the performance of the neural network, XGBoost, and SVM classifiers in injury severity prediction. Second, we utilize a severity category-combination method with oversampling to relieve the class-imbalance problem prevalent in crash data. Third, we take advantage of probability calibration and the optimal probability threshold moving to improve the prediction ability of ordinal classification. The proposed approach can satisfy the rank consistency and rank monotonicity requirement and is proved to be superior to other ordinal classification methods and nominal classification machine learning by statistical significance test. Important factors relating to injury severity are selected based on their permutation feature importance scores. We find that converting severity levels into three classes, minor injury, moderate injury, and serious injury, can substantially improve the prediction precision.

Keywords: crash severity; ordinal classification; imbalance data; machine learning; sampling

check for updates

Citation: Zhu, S.; Wang, K.; Li, C. Crash Injury Severity Prediction Using an Ordinal Classification Machine Learning Approach. *Int. J. Environ. Res. Public Health* **2021**, *18*, 11564. https://doi.org/10.3390/ijerph182111564

Academic Editors: Ediriweera Desapriya and Kazuko Okamura

Received: 30 September 2021
Accepted: 30 October 2021
Published: 3 November 2021

1. Introduction

The prediction and cause analysis of traffic crashes has always been an important topic for scholars in traffic safety. In the research of this subject, scholars often use statistical methods or machine learning methods to conduct research.

A statistical model usually specifies the mathematical relationship between explanatory variables and crash severity. Based on strict assumptions of uncertainty distribution and hypothesis tests, the statistical model can isolate the effects of explanatory variables on crash severity [1,2]. For example, Cerwick et al. [3] used the mixed logit model and the latent class multinomial logit model to predict crash severity. A large number of crash specific, temporal, roadway, vehicle, driver characteristics, and environmental factors were found significant. Haghighi et al. [4] used standard ordered logit (SOL) and Multilevel ordered logit (MOL) to analyze the effect of roadway geometric features on crash severity. However, statistical models are usually weaker in making predictions than machine learning methods. Iranitalab and Khattak [5] compared the performance of a statistical model, Multinomial Logit (MNL), with three machine learning methods including Nearest Neighbor Classification (NNC), Support Vector Machines (SVM), and Random Forests (RF) in predicting traffic crash severity, and found MNL has the worst prediction accuracy.

Machine learning models are designed to make the most accurate predictions possible. Chang et al. [6] used the Classification And Regression Tree (CART) to predict crash severity, where prediction accuracy is 90.8% for learning data and 91.7% for testing data. Abdel-Aty et al. [7] used a single-layer hidden layer Multi-layer Perceptron (MLP) to predict traffic crash severity with an average prediction accuracy of 73.5%. Delen et al. [8] used an artificial neural network (ANN) to predict the severity of a crash and improved the prediction accuracy. Alkheder et al. [9] used ANN, combined with k-means for clustering, to predict traffic crashes and then compared with probit algorithm to prove that ANN is better than probit in predicting the severity of crashes. Among many methods of crash severity prediction research, neural network methods have better performance and are more popular.

Most existing machine learning methods applied in crash severity prediction treat crash severity levels as nominal data without order information. This unrealistic simplification casts a shadow on machine learning methods' prediction ability. In general, the severity of traffic crashes is classified as fatal injury or killed, incapacitating injury, non-incapacitating, possible injury, property damage, or no injury. Moreover, the severity of the injury is ordered and increases from no injury to possible injury, to non-incapacitating, to incapacitating injury, and to fatal injury or killed. Between closely related adjacent categories (such as no injury and possible injury), there may be shared unobserved effects or correlations between their data [10]. To the authors' knowledge, in the field of crash severity study, less attention has been paid to ordinal classification machine learning, and information on natural ordering in injury severity is missed in conventional machine learning, including SVM, decision tree, and MLP. Some statistical models, such as ordered logit and ordered probit [4,11,12], can handle the ordinal severity labels. However, discrete choice models rely on statistical assumptions and pre-defined relationships between severity labels and input variables, which makes them good choices for factor analysis but restricts their prediction accuracy [13].

Gutierrez et al. [14] summarized ordinal classification machine learning algorithms developed to classify categorical variables that show a natural order between the labels. They confirmed that there is no clear winner that performs the best in all possible datasets and problem requirements. The main three categories of ordinal classification machine learning are:

(1) Cost-sensitive classification: apply cost-sensitive loss function in the evaluation of the learned system with different costs for different types of misclassification errors. For example, Riccardi et al. [15] proposed cost-sensitive AdaBoost for ordinal regression. The problem of cost-sensitive classification is how to determine the cost matrix without priori knowledge of the ordinal classification.

(2) Ordinal binary decomposition: decompose the ordinal target variable into several binary variables, which are then estimated by single or multiple models. Our new ordinal classification method falls in this category. The problem of existing ordinal binary decomposition methods is the violation of rank monotonicity or rank consistency. Related methods and their drawbacks are introduced in the method section later in more detail.

(3) Threshold model: extension of the regression model in which distances among the ordered classes are not pre-defined but estimated by finding the optimal thresholds dividing classes [16]. Li and Lin [17] proposed a general reduction framework to transform ordinal regression as a series of binary classification sub-problems and demonstrated that many threshold models and ordinal binary decomposition methods are equivalent.

The number of crash cases in each category is often imbalanced. Usually, the sample size of fatal cases is several times smaller than that of cases in other categories. With imbalanced data, traditional classification algorithms incline to the category with a large amount of data, while the category with a small amount of data is neglected [9]. Many studies merged several minority categories of injury severity into one class and converted multi-class classification problems into two-class (no injury vs. injury) classification problems [8]. Another option is to turn the multi-class classification problem into a three-class (no injury, minor injury, and fatal injury) problem [3,6]. Some studies have tried to deal with imbalanced data by under-sampling majority class examples [18] and by oversampling minority class examples [19,20] and achieved good results.

Although some scholars have tried to combine several injury severity levels into fewer categories [8], there will still be a class-imbalance problem, and the predicted results will still incline towards the category of large proportion. This paper focuses on different combinations of severity categories that can relieve imbalance and keep the model's ability to predict crash injury severity. SMOTE-NC (Synthetic Minority Oversampling Technique for Nominal and Continuous) is applied to oversample the minority class. We compare the performance of three classifiers: MLP, XGBoost, and SVM. The best classifier is combined with an ordinal binary decomposition method to handle ordinal crash severity labels.

The aim of this paper is to propose an ordinal machine learning classification approach that overcomes the ordinal nature of crash severity data and class-imbalance problems. The contributions of this presented approach include:

(1) To the authors' knowledge, this is the first paper applying ordinal classification machine learning to predict traffic crash injury severity using real-world crash data.
(2) We propose an ordinal classification machine learning method that satisfies rank monotonicity and rank consistency and takes advantage of probability calibration and the movement of optimal probability threshold to generate superior classification results compared to existing ordinal classification algorithms.
(3) We test six severity category-combination strategies and find the best three-class combination plan.

The rest of this paper is constructed as follows. The second section describes and analyzes the characteristics of crash data. The third section presents the research methods involved in this paper, including sampling, severity category-combination, machine learning, and ordinal classification. The fourth section shows the comparison and analysis of the results. The conclusions of this paper are included in the fifth section.

2. Data Description

The data were collected from the Highway Safety Information System (HSIS) for crashes that occurred in California in 2010. Variables in the crash dataset include those related to intersections, road segments, and historical traffic crashes. Several variables were dropped when the null value occurred too frequently in the dataset. The California traffic crash data contains three data files, the crash file, vehicle file, and occupant file. The crash file contains 52 variables such as time, location, crash severity, the total number of injuries, weather, etc. The vehicle file contains 42 variables such as vehicle model, whether the driver makes a phone call, whether the driver is drunk, etc. The occupant file contains ten variables such as age, gender, the severity of injury, type of collision, etc. These three data files were merged according to the crash number and the crash vehicle number. The data contains 104 variables observed in one crash, including the injury severity of the person involved. In this dataset, five crash injury severity levels are defined, namely non-injury crash (denoted as NIC), complaint of pain (denoted as COP), other visible injury (denoted as OVI), severe injury (denoted as SI), and killed (denoted as KSI). The severity level of the crash-related personnel injury is shown in Figure 1. The frequencies of severity levels in the data were 80,474 (57.20%), 41,642 (29.60%), 15,200 (10.80%), 2714 (1.93%), and 660 (0.47%), respectively.

We select 17 major influencing variables in this study: occupant type, seating position, type of collision, primary collision factor, first associated factor, roadway class, ejected, object struck, the total number of vehicles involved, alcohol involved, driver's gender, driver's age, occupant's age, vehicle year, motorcycle involved, driver's safety equipment, and occupant's safety equipment. Primary collision factor is the one element that best describes the cause of the collision or, if removed, would have prevented the collision from occurring. First associated collision factor is the most important one of factors or violations that contributed, but were not the main cause of the collision. There are 14 categorical variables, except for driver's age, occupant's age, and vehicle model year. The Appendix A provides the descriptive statistics of these variables. In total, there are 140,690 crash records, out of which 139,555 remained after cleaning missing data. Samples with missing values were simply removed because the proportion of samples with missing values is relatively small. Other errors were not found in the Highway Safety Information System (HSIS) dataset.

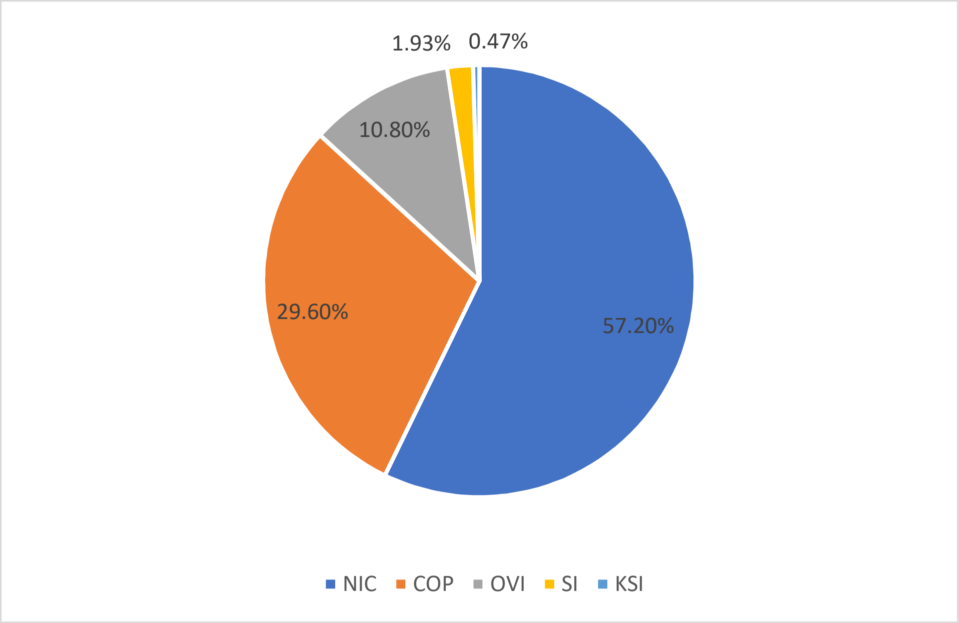


Figure 1. The proportion of crash injury severity in data. NIC—non-injury crash; COP—complaint of pain; OVI—other visible injury; SI—severe injury; KSI—killed.

3. Methodology

We summarize the research framework as a flowchart in Figure 2.

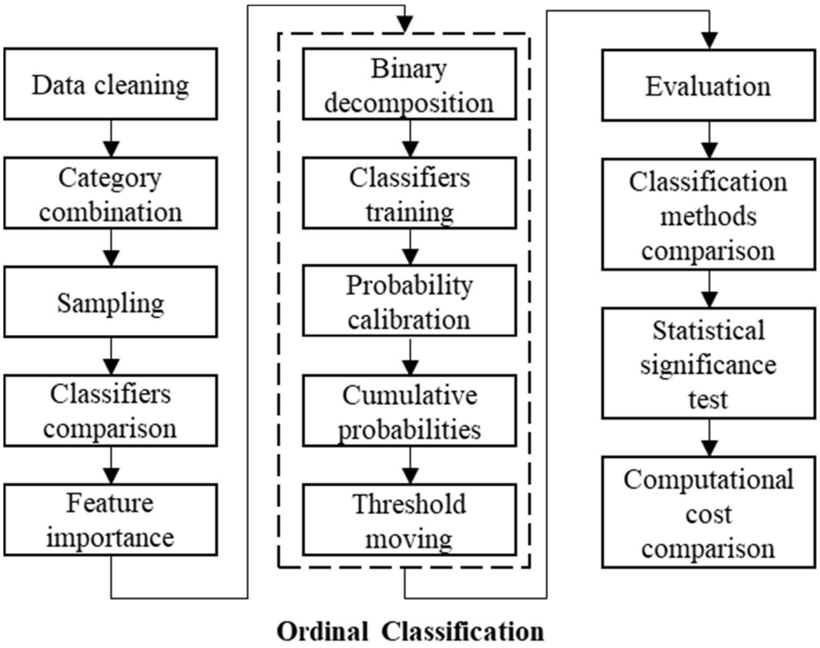


Figure 2. Research framework.

After data cleaning, we try six methods of category combination that merge crash injury severity levels into fewer categories. SMOTE-NC oversampling is applied to relieve the class-imbalance problem. We compare SVM, XGBoost, and MLP, and choose the best one as the classifier used in the ordinal classification method. The permutation feature importance is also analyzed with the chosen classifier.

The proposed ordinal classification method contains five main steps. First, the label to predict (crash injury severity) is decomposed with one-vs-all binary decomposition. Second, a multiple-output classifier or multiple single-output classifiers (chosen from SVM, XGBoost, and MLP) are trained to predict crash injury severity. Third, the predicted probabilities are calibrated to remove the bias from the classifier and data sampling. Fourth, the cumulative probabilities are calculated based on calibrated probabilities, which satisfies both the rank monotonicity and rank consistency. Fifth, the threshold moving method can help to find the optimal threshold that converts probabilities into crash injury severity predictions.

All classifiers trained in this paper are established using Python programming language with supported libraries, including keras, tensorflow, xgboost, hyperopt, and scikit-learn. The computing platform is a desktop computer with an AMD Ryzen 1700X 8-core processor and Windows 10 operating system. The proposed ordinal classification method, along with other methods, is evaluated through cross-validation. The performance of each method is compared and tested by a statistical significance test. The computational cost of the proposed method is discussed to show its computation efficiency.

3.1. Imbalanced Data Preprocessing

One of the critical characteristics of the crash dataset is that the number of crashes leading to death and severe injury is always much less than those of trivial injury. The problem of imbalanced data is prevalent in the field of traffic accident studies. For example, in our data NIC accounts for 57.20% of all crashes. This imbalance of data means that a dummy classifier that classifies all instances to NIC would still achieve an accuracy score of 57.20%. This issue would have a detrimental effect on the training process. Classifiers such as artificial neural networks, support vector machines, and decision trees are designed for balanced data with a roughly equal sample size of each class. In the case of imbalanced data, classifiers tend to overly focus on the class with the largest proportion and ignore the minority class. However, accurately predicting the minority class, SI and KSI in this case, is the main purpose of machine learning training. Existing research has tried to combine the categories of severe injuries in traffic crashes and turn multi-class problems into a binary-class problem to make predictions. However, training a binary-class classifier limits the model's ability to distinguish different levels of injury severity and therefore reduces the model's practical value.

This research performs oversampling and category combination to solve the problem of imbalanced data. We combine the five crash severity levels into three classes in order to reduce the difficulty in severity prediction. We propose all six possible ways of category combination that can convert five crash severity levels into three classes while keeping an ordinal nature (illustrated in Figure 3). Each class contains at least one of the five crash severity levels, and the severity levels are exclusive over classes. For all combinations, NIC is always included in class 1, and KSI is always included in class 3. All instances in class 3 have higher severity levels than instances in class 2, and all instances in class 2 have higher severity levels than instances in class 1. The difference between combinations is how COP, OVI, and SI are assigned into classes. As shown in Figure 4, the proportion of each class in the traffic crash data is still uneven in each combination, but the imbalance is relieved compared to the original five categories. In addition, the 3-class classification problem has more explanatory ability compared to the 2-class problem. It can be interpreted that the five severity categories are combined into three new classes: minor injury crashes, moderate injury crashes, and serious injury crashes.

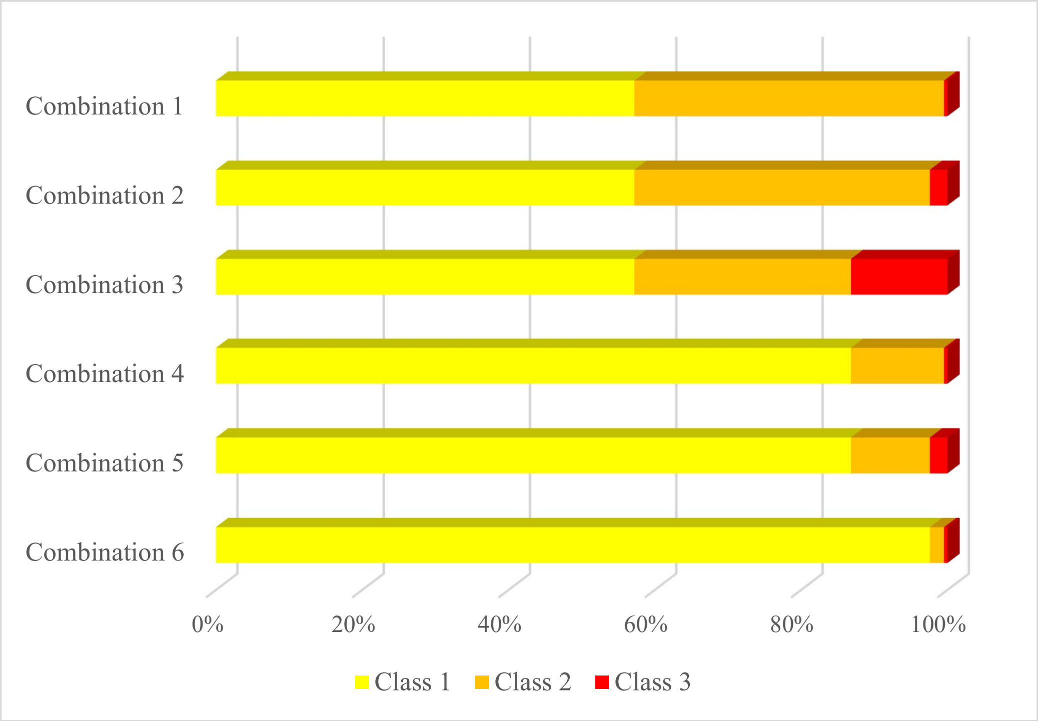
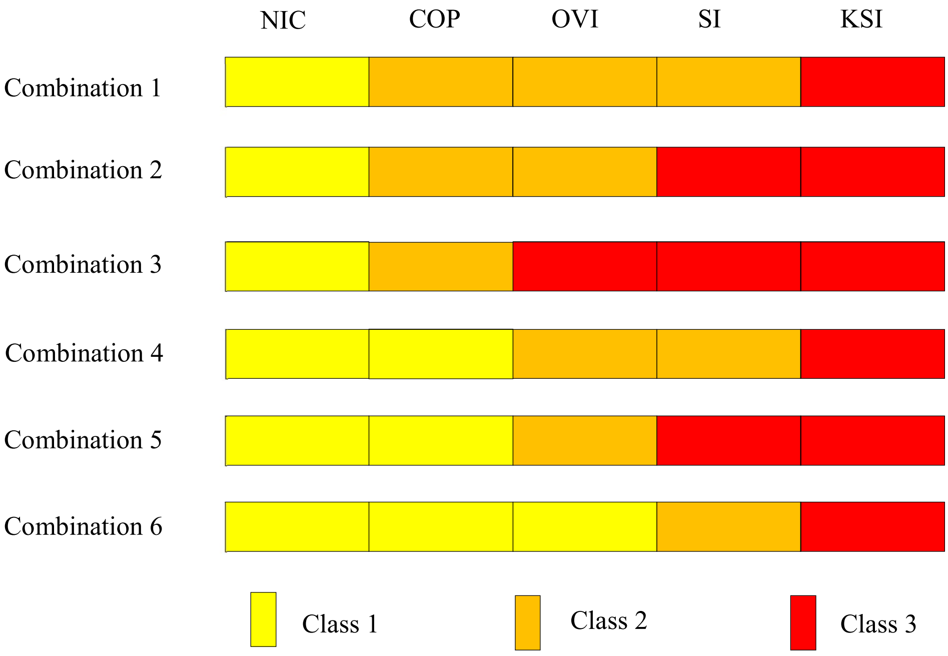


Figure 3. Six methods of 5-category combination.



	NIC	COP	OVI	SI	KSI
Combination 1	🟨	🟧	🟧	🟧	🟥
Combination 2	🟨	🟧	🟧	🟥	🟥
Combination 3	🟨	🟧	🟥	🟥	🟥
Combination 4	🟨	🟨	🟧	🟧	🟥
Combination 5	🟨	🟨	🟧	🟥	🟥
Combination 6	🟨	🟨	🟨	🟧	🟥

🟨 Class 1 🟧 Class 2 🟥 Class 3

Figure 4. The proportion of classes in each method of categorizing.

Another important task we need to complete is oversampling of the minority class. At present, there are two main sampling methods to deal with the imbalance-data problem, namely under-sampling and oversampling. Machine learning algorithms are data-hungry and require extensive data for model training, making under-sampling unpreferable, especially when the minority class sample size is small. Because the dataset is a mix of categorical and continuous features, this research uses SMOTE-NC sampling, a variation of SMOTE sampling. SMOTE-NC creates synthetic data for categorical as well as continuous features in the data set. SMOTE-NC treats categorical features differently from continuous features. For continuous features, SMOTE-NC sampling is an interpolation algorithm that looks for features between the data sample and supplements data with similar characteristics for minority class instances. By contrast, the categorical variable's value of a newly generated sample is decided by picking the most frequent category of the nearest neighbors present during the generation [21]. The proportion of each class in each combination after oversampling is shown in Figure 5. Class 1 is the majority class in all six combinations. Classes with an instance number smaller than 20% of class 1 are chosen to be oversampled. For all combinations except combination 3, class 3 is oversampled. For combination 6, class 2 is also oversampled. SMOTE-NC sampling is performed on class 3 in combination 1, 2, and 4–6, and on class 2 in combination 6. The sampling rate ensures that the minority class instance size equals one-fifth of the majority class instance size. The new proportion of classes in each combination after sampling is shown in Figure 5. Since classes 2 and 3 are more than one-fifth of class 1 in combination 3, no sample in combination 3 is oversampled.

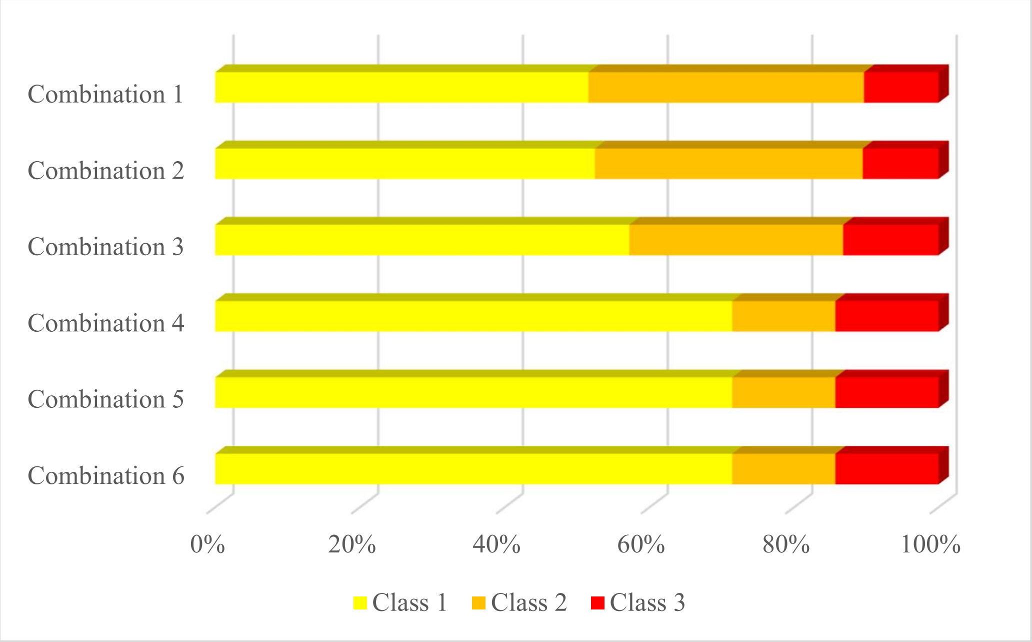


Figure 5. The proportion of classes in each method of category combination after sampling.

3.2. *Ordinal Classification*

3.2.1. Cumulative Binary Decomposition

Given a dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$ with the class label $y_i \in \{C_1, C_2, \cdots, C_K\}$ where $C_1 < C_2 < \cdots < C_K$, as expressed in (1), the cumulative binary decomposition method encodes y_i into K-1 binary labels $y_i^{(1)}, y_i^{(2)}, \cdots, y_i^{(K-1)}$ that $y_i^{(k)} = 1\{y_i > C_k\}$. The indicator function $1\{y_i > C_k\}$ is 1 when $y_i > C_k$ and 0 when $y_i \leq C_k$.

$$y = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_K \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & 0 \\ 1 & 1 & \cdots & 1 \end{bmatrix} \tag{1}$$

A single multi-output model or K-1 single-output models can be trained with the binary decomposed dataset $\left\{\mathbf{x}_i, y_i^{(k)}\right\}_{i=1}^{N}$ where $k \in \{1, 2, \cdots, K-1\}$. After training, the predicted probability of $y_i^{(k)}$ is essentially the predicted cumulative probability $\hat{p}(y_i > C_k)$. Frank's method and Cheng's method are both based on $\hat{p}(y_i > C_k)$.

Frank's method [22] first calculate the probability of each class based on (2), and the predicted class is given by (3).

$$\hat{p}(y_i = C_k) = \begin{cases} 1 - \hat{p}(y_i > C_k), k = 1 \\ \hat{p}(y_i > C_{k-1}) - \hat{p}(y_i > C_k), k \in \{2, \cdots, K-1\} \\ \hat{p}(y_i > C_{k-1}), k = K \end{cases} \tag{2}$$

$$\hat{y}_i = \underset{k}{\operatorname{argmax}}[\hat{p}(y_i = C_k)] \tag{3}$$

Cheng's method [23], by contrast, predicts class labels by (4).

$$\hat{y}_i = \sum_{k=1}^{K-1} 1\{\hat{p}(y_i > C_k) > 0.5\} + 1 \tag{4}$$

3.2.2. One-vs-All Binary Decomposition

Different from cumulative binary decomposition, one-vs-all binary decomposition method encodes y_i into K binary labels $y_i^{(1)}, y_i^{(2)}, \cdots, y_i^{(K)}$ that $y_i^{(k)} = 1\{y_i = C_k\}$, which is expressed in (5).

$$y = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_K \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{5}$$

A single multi-output model can be trained to predict $\hat{p}(y_i = C_k)$ and guarantee that

$$\sum_{k=1}^{K} \hat{p}(y_i = C_k) = 1 \tag{6}$$

Beckham and Pal [24] propose a method that

$$\hat{y}_i = \sum_{k=1}^{K} \beta_k \cdot \hat{p}(y_i = C_k) \tag{7}$$

where β_k is to be determined. One option is setting $\beta_k = k$; another is to calculate β_k by optimizing the following objective function (8). We denote these two options as Beckham1 and Beckham2.

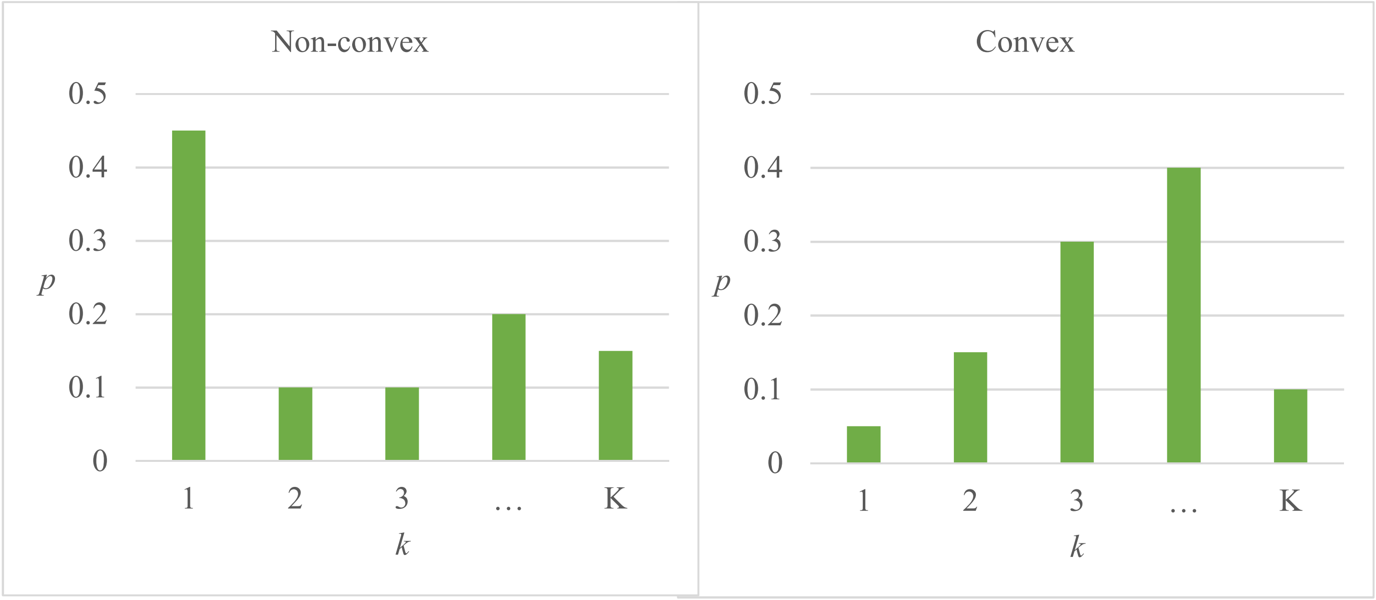
$$\max_{\beta_k} \sum_{i=1}^{N} \left[y_i - \sum_{k=1}^{K} \beta_k \cdot \hat{p}(y_i = C_k) \right]^2 \tag{8}$$

3.2.3. Existing Drawbacks

Both cumulative binary decomposition and one-vs-all binary decomposition have drawbacks.

Rank monotonicity requires that $\hat{p}(y_i > C_k) \leq \hat{p}(y_i > C_j)$ for any $k > j$. However, predicted cumulative probabilities based on cumulative binary decomposition do not guarantee rank monotonicity [25] since cumulative probabilities $\hat{p}(y_i > C_k)$ and $\hat{p}(y_i > C_j)$ are predicted independently. If $\hat{p}(y_i > C_k) > \hat{p}(y_i > C_j)$ for any $k > j$, then $\hat{p}(C_k \geq y_i > C_j) < 0$, which is unrealistic and hurts model performance.

Predicted class probabilities based on one-vs-all binary decomposition do not guarantee rank consistency, which requires $p_i(k) = \hat{p}(y_i = C_k)$ to have a convex shape, illustrated in Figure 6.



(**a**) (**b**)

Figure 6. Illustration of rank inconsistency and rank consistency. (**a**) rank inconsistency; (**b**) rank consistency.

3.2.4. Proposed Method

We summarize all methods in Table 1. Frank's method and Cheng's method are both based on cumulative binary decomposition. Frank's method converts predicted cumulative probabilities into class probabilities, while Cheng's method applies cumulative probabilities directly to predict class labels without knowing class probabilities. Beckham1 and Beckham2 are based on one-vs-all binary decomposition and class probabilities.

Table 1. Summary of ordinal classification methods.

Class Prediction Based on	Cumulative Binary Decomposition	One-vs-All Binary Decomposition
Class Probability	Frank's method [22]	Beckham1 and Beckham2 [24]
Cumulative Probability	Cheng's method [23]	This paper

We propose a new method that uses one-vs-all binary decomposition and K single output models to predict class probabilities. The main difference between our method and Beckham1/Beckham2 is that we convert predicted class probabilities into cumulative probabilities:

$$\hat{p}(y_i > C_k) = \sum_{j=k+1}^{K} \hat{p}\left(y_i = C_j\right) \tag{9}$$

The advantage of this method is that it generates predicted cumulative probabilities satisfying rank monotonicity. Then the class label is determined as follows:

$$\hat{y}_i = \sum_{k=1}^{K-1} 1\{\hat{p}(y_i > C_k) > T_k\} + 1 \tag{10}$$

where T_k is the optimal threshold of $y^{(k)}$. Some machine learning algorithms, such as tree-based learning, usually generate biased probabilities. Probabilities can also be distorted because of data imbalance and sampling [26]. Therefore, we find the F1-maximizing T_k with validation data instead of simply setting T_k to 0.5.

Since $\hat{p}\left(y_i = C_j\right)$ in (6) could be biased, the bias is delivered to $\hat{p}(y_i > C_k)$ and causes inaccurate estimation of $\hat{y}_i$ in (7). We perform probability calibration by isotonic regression to remove bias in $\hat{p}(y_i = C_j)$. There are two main methods to calibrate probability: Platt scaling and isotonic regression. Platt [27] introduced Platt scaling, which trains a logistic regression to map the original output to the real class probability. Isotonic regression is a non-parametric approach introduced by Zadrozny and Elkan [28,29]. Isotonic regression is preferable to Platt scaling when the sample size is large enough. Isotonic regression fits a piecewise constant non-decreasing function, where predicted probabilities or scores in each bin are assigned the same calibrated probability that is monotonically increasing over bins. More formally,

$$\begin{array}{cc} \min\limits_{\boldsymbol{\theta},\mathbf{a}} & \sum\limits_{m=1}^{M} \sum\limits_{i=1}^{N} 1(a_m \leq \hat{p}_i < a_{m+1})(\theta_m - y_i)^2 \\ \text{subject to} & 0 = a_1 \leq a_2 \leq \cdots \leq a_{M+1} = 1,\ \theta_1 \leq \theta_2 \leq \ldots \leq \theta_M \end{array} \tag{11}$$

where M is the number of bins, $a_1, a_2, \cdots, a_{M+1}$ are the interval boundaries, $\theta_1, \theta_2, \ldots, \theta_M$ are the corresponding calibrated probabilities for that falls in each bin.

3.3. Machine Learning Algorithms

We test the performance of three classifiers, Multi-Layer Perceptron (MLP), eXtreme Gradient Boosting (XGBoost), and Support Vector Machine (SVM), on 5-category classification. The winner of three candidates is kept for a 3-class problem and other analysis tasks after.

MLP is a pass-forward artificial neuron network that maps an n-dimensional input vector to an m-dimensional output vector. It has many successful applications in classification tasks such as MNIST handwriting digit number recognition by transforming high dimensional input of related elements to low dimensional discriminative representation. Several researchers have attempted to utilize deep learning frameworks to model potential factors that may lead to different injury severity levels [30,31]. They input the randomly shuffled dataset directly into the network to capture the feature of all input factors of a particular crash.

Back-propagation multi-layer perceptron (BP-MLP) applies weighted input from every previous layer to a non-linear function, evaluates the difference between network output and actual label, and optimizes the parameters in the network using optimizers such sophistic gradient decedent (SGD) or Adam optimizer. Thus, the characteristic of a particular MLP model can be defined by its depth, non-linear function of each layer, loss function, and optimizer. This research utilizes two hidden layers in the network. The numbers of neurons are 64 and 10, respectively. Both layers use a rectifier linear unit (ReLU) as the activation function. We then feed the mapped 3-dimension output learned representation vector to a softmax layer to compute the final predicted class and predict probabilities for the input variables.

Among the 17 variables used to predict injury severity, there are 14 categorical variables, except driver's age, occupant's age, and vehicle model year. When training a neural network, one-hot encoding is more appropriate for categorical data where no specific numerical relationship exists between categories. This involves representing each categorical variable with a group of binary vectors that has one {0,1} code for each unique variable value. However, One-hot encoding dramatically increases the dimension of data. For example, if a {0,1} code is used to represent every numerical code of object1, then the one-hot encoding will create 99 more dimensions than numerical encoding. One-hot encoding also leads to sparse data space, making it challenging to optimize neural networks.

XGBoost is a Gradient Tree Boosting-based algorithm that has been proven to be a powerful classifier. The advantage of XGBoost in this study is that decision tree-based machine learning has no issues with the numerical encoding of categorical variables. Moreover, XGBoost requires much less training time than neural network and often produce remarkable prediction results in crash-related studies [32–34].

SVM has been and still is a widely used classifier. Many studies of traffic crash injury prediction have applied SVM as a benchmark classifier [30,35,36]. Therefore, it is used as such in this study.

To extract the maximum performance out of classifiers, we need hyperparameter tuning to determine the optimal combination of hyperparameters. Hyperopt is one of the most popular hyperparameter tuning packages and implements the Tree of Parzen Estimators (TPE) algorithm to search the optimal value of hyperparameters efficiently in a search space described by the user [37]. We apply Hyperopt in this paper to optimize the main hyperparameters of MLP, XGBoost, and SVM. The hyperparameters of MLP include the number of layers/neurons, activation function in each layer, optimizer, learning rate, number of epochs, and batch size. The hyperparameters of XGBoost include the number of estimators, learning rate, maximum depth, subsample ratio, etc. The hyperparameters of SVM are the C parameter and gamma.

3.4. Cross-Validation and Evaluation Metrics

We use stratified 10-fold cross-validation to evaluate the classification algorithm's performance. Stratified 10-fold cross-validation divides the 139,555 records randomly into ten equal-sized subsets. Each subset has the same proportion of each class as the total dataset. At each time, eight subsets are used for sampling (if required) and training, and one subset is used for probability calibration and threshold optimization (only for the ordinal classification method proposed in this paper). The last subset is used to test the performance of the trained model. This process rotates through each subset, and the average precision, recall, and F1 score of each class represent the algorithm's performance.

3.5. Statistical Significance Test

Machine learning algorithms are commonly evaluated using k-fold cross-validation, and their evaluation metrics, such as mean accuracy scores, are compared directly. Statistically significance tests are designed to test whether the difference between evaluation metrics is statistically significant or the result of a statistical fluke. The null hypothesis is that metric scores observed from two algorithms were drawn from the same distribution. If this assumption is rejected, it suggests that the difference in metric scores is statistically significant. Otherwise, the two algorithms' performances are statistically equal.

K-fold cross-validated paired Student's *t*-test is the most used statistical test for machine learning algorithms comparison. However, the calculation of the t-statistic in the test is misleading since the metric scores in each sample are not independent [38]. In k-fold cross-validation, a given observation will be used in the training dataset k-1 times. This means that the estimated metric scores are dependent.

Dietterich [38] recommended a resampling method called 5 × 2 cross-validation that involves five repeats of 2-fold cross-validation. Two-fold cross-validation can ensure that each observation appears only in the train or test dataset once. A paired Student's *t*-test is used on the results.

$$t = \frac{\mu}{\sqrt{\frac{1}{5}\sum_{i=1}^{5}\left(\left(\Delta_i^{(1)} - \mu\right)^2 + \left(\Delta_i^{(2)} - \mu\right)^2\right)}} \tag{12}$$

where:

$\Delta_i^{(1)}$ is the scores difference of two algorithms for the first fold of the *i*-th 2-fold cross-validation;

$\Delta_i^{(2)}$ is the scores difference of two algorithms for the second fold of the *i*-th 2-fold cross-validation;

$\mu = \frac{\Delta_1^{(1)} + \Delta_1^{(2)}}{2}$ is the mean of scores difference for the first 2-fold cross-validation.

Under the null hypothesis that two algorithms are statistically equal, *t* is assumed to follow a Student's t-distribution with 5 degrees of freedom. If *t* stays close enough to 0, then the null hypothesis is satisfied. The threshold is 2.571 at the 95% confidence level. 5 × 2 cross-validation is used in this paper to compare algorithms' performance.

4. Results

4.1. Comparison of Classifiers

The result of the 5-category classification problem is listed in Table 2. We compare each classifier's precision rate and find that XGBoost has the highest precision rate in COP, SI, and KSI categories. MLP only outperforms other classifiers in category OVI. The gap between the performance of XGBoost and MLP may be caused by the data characteristic that most variables are categorical. Therefore, we utilize XGBoost as the only classifier used for analysis in the following sections.

Table 2. Precisions of five-category classification problem.

Classifier	Precision (%)					Macro-Average
	NIC	COP	OVI	SI	KSI	
MLP	99.1	56.4	36.5	0.4	1.9	38.9
XGBoost	99.3	64.9	23.4	1.9	4.5	38.8
SVM	100	64.7	12.7	0	0	35.5

The performance of SVM relies on marginal data that lies near the separating hyperplane. SVM yields poor performance when fed with data with ambiguous distinction or imbalanced class. Several modifications to the SVM kernel function and preprocessing methods have been used to improve SVM's capability to distinguish minority samples. This paper uses radial basis function (RBF) as SVM's kernel function and achieves significant improvement on minority class prediction compared to other kernel functions. Although SVM can identify 100% NIC instances, it is still the worst classifier and fails to identify any SI and KSI instances.

In general, the precision of injury severity decreases as the severity level rises. More than 99% of NIC cases can be correctly classified regardless of the classifier used. For COP, the precision is about 60%, but the precision of SI and KSI decreases dramatically to almost 0. As discussed in the Introduction, the poor performance on serious injury crashes is caused by class-imbalance.

4.2. Comparison of Category-Combination and Sampling

In order to overcome the shortcomings of the five-category problem, this research proposes six ways of category-combination and generates a 3-class problem through category-combination. As shown in Table 3, it is clear that the macro-average precision rate is improved for most combinations except combination 6. However, it can be seen that the precision rate of class 3 (KSI only) is still very low in combinations 1 and 4 because KSI is not combined with others to relieve the imbalance.

Table 3. Precisions of three-class classification problem for six combinations before SMOTE-NC sampling.

Combination	Precision (%)			Macro-Average
	Class 1	Class 2	Class 3	
1	98.0	73.7	0.0	57.2
2	98.4	73.1	4.8	58.8
3	99.3	62.6	33.6	65.2
4	97.7	34.6	0.0	44.1
5	98.3	25.0	6.9	43.4
6	99.9	1.1	6.1	35.7

In Table 4, after preprocessing by SMOTE-NC, the precision rate of each class is further improved. Among all combinations, combination 3 has the highest F1 score of 45.0% for class 3, but it combines OVI with SI and KSI and loses the ability to predict serious injury crashes. Combination 5 has the second-highest F1 score of 24.5% for class 3, but its F1 score for class 2 is only 19.4%. Combination 2 achieves acceptable F1 scores of 90.1%, 78.3%, and 24.1% for classes 1, 2, and 3, respectively. Moreover, combination 2 groups COP and OVI into class 2 and groups SI and KSI into class 3, which is a reasonable combination strategy. The three classes can be considered as minor injury crashes, moderate injury crashes, and serious injury crashes. Therefore, we apply combination 2 to convert 5-category into a 3-class classification problem.

Table 4. Performance of three-class classification problem for six combinations after SMOTE-NC sampling.

Combination	Group	Class 1	Class 2	Class 3	Macro-Average
1	Precision (%)	97.8	71.0	22.7	63.9
	Recall (%)	83.5	95.6	7.2	62.1
	F1 (%)	90.1	81.5	11.0	60.9
2	Precision (%)	98.0	67.8	27.5	64.4
	Recall (%)	83.3	92.6	21.4	65.8
	F1 (%)	90.1	78.3	24.1	64.2
3	Precision (%)	99.3	62.6	33.6	65.2
	Recall (%)	82.4	75.4	68.2	75.4
	F1 (%)	90.1	68.4	45.0	67.8
4	Precision (%)	97.1	23.1	36.4	52.2
	Recall (%)	91.2	66.4	5.6	54.4
	F1 (%)	94.1	34.3	9.7	46.0
5	Precision (%)	97.1	11.8	36.7	48.5
	Recall (%)	90.7	54.8	18.4	54.6
	F1 (%)	93.8	19.4	24.5	45.9
6	Precision (%)	97.0	21.6	27.3	48.6
	Recall (%)	98.4	18.6	8.7	41.9
	F1 (%)	97.7	20.0	13.2	43.6

4.3. Feature Importance

After category-combination and SMOTE-NC sampling, the classification model is more efficient in predicting the severity of crash injuries than the original five-category problem. We analyze the permutation feature importance of the classification models, as shown in Table 5. In combination 1-3, occupant type has the most significant impact on the injury severity to crash-injured individuals. Ejected from vehicle has the second-highest importance in combinations 3 and 5. In combination 1 and 2, the number of vehicles involved in the crash has the second-greatest impact on the severity of injuries to crash-injured individuals. Vehicle model year is also an important feature in combinations 3, 4, and 5. Based on repeated permutation feature importance calculation, we find that the differences between the most and less important input features are statistically significant.

It is worth noting that in each combination, features with high importance are basically the same. They are ejected from vehicle, number of vehicles, occupant type, and vehicle model year. As shown in Appendix A, ejected from vehicle and occupant type are highly related to the severity of the injury. The driver’s injury severity is more considerable when the number of vehicles involved is one or two. The proportion of cases in which drivers were ejected from vehicles is not particularly large, accounting for only 2.45% of the data. In cases where drivers were ejected from vehicles, the proportion of fatal and severe injuries is high, accounting for 27–29%. In cases where drivers were not ejected from vehicles, the ratio of fatal and severe injury is only 1.7%.

Table 5. Permutation feature importance for six combinations.

Variable	Definition	Combination					
		1	2	3	4	5	6
occupant	Occupant type	0.13	0.25	0.19	0.07	0.07	0.04
seat	Seating position	0.06	0.06	0.07	0.05	0.06	0.06
collision	Collision type	0.06	0.06	0.07	0.07	0.07	0.07
factor	First associated factor	0.05	0.04	0.05	0.06	0.06	0.06
cause	Primary collision cause	0.05	0.03	0.04	0.06	0.05	0.05
road	Roadway class	0.05	0.03	0.04	0.06	0.05	0.07
eject	Ejected from	0.10	0.09	0.11	0.02	0.11	0.09
object	First object struck	0.06	0.06	0.05	0.05	0.07	0.07
vehicles	number of vehicles	0.11	0.10	0.06	0.07	0.08	0.12
alcohol	alcohol involved	0.01	0.01	0.01	0.05	0.01	0.02
gender	driver's gender	0.02	0.01	0.01	0.06	0.01	0.02
drv_safe	Driver safety equipment	0.05	0.06	0.06	0.10	0.07	0.06
occ_age	Occupant's age	0.06	0.04	0.04	0.01	0.06	0.07
drv_age	Driver's age	0.04	0.03	0.03	0.01	0.04	0.04
motor	Motorcycle involved	0.02	0.01	0.01	0.11	0.01	0.02
occ_safe	Occupant safety equipment	0.06	0.06	0.06	0.06	0.07	0.06
veh_year	Vehicle model year	0.08	0.07	0.11	0.11	0.11	0.08

4.4. Comparison of Ordinal Classifications

In total, we test the performance of 5 ordinal classification methods on injury severity data, including Frank's method, Cheng's method, Beckham1, Beckham2, and the method proposed by this paper. We also compared the results of ordinal classification methods with nominal classification to prove ordinal classification's advantage. In each method, XGBoost is used as the basic classifier.

In Section 1, we explained why ordinal classification methods are better than nominal classification when the labels are ordinal. In Section 3, we interpreted the drawbacks of ordinal classification benchmarks used in this paper and why our proposed ordinal classification method is more advanced theoretically. We believe that our proposed method should outperform other ordinal classification methods, which outperform nominal classification. Most results shown in Table 6 are consistent with our expectations.

Table 6. Performance of six classification methods.

Method	Evaluation Metric	Class			Macro-Average
		1	2	3	
Nominal classification	Precision (%)	98.0	67.8	27.5	64.4
	Recall (%)	83.3	92.6	21.4	65.8
	F1 (%)	90.1	78.3	24.1	64.2
Beckham1	Precision (%)	92.9	73.6	22.7	63.0
	Recall (%)	85.6	83.9	21.8	63.8
	F1 (%)	89.1	78.4	22.3	63.2
Beckham2	Precision (%)	95.6	75.7	0.00	57.1
	Recall (%)	84.4	86.9	0.00	57.1
	F1 (%)	89.7	80.9	0.00	56.9
Frank	Precision (%)	97.6	68.2	31.6	65.8
	Recall (%)	83.6	92.1	23.6	66.4
	F1 (%)	90.1	78.4	27.0	65.1
Cheng	Precision (%)	96.0	70.3	29.2	65.2
	Recall (%)	84.4	88.9	24.1	65.8
	F1 (%)	89.8	78.5	26.4	64.9
This paper	Precision (%)	94.0	68.9	41.2	68.0
	Recall (%)	85.2	86.4	21.3	64.3
	F1 (%)	89.4	76.7	28.1	64.7

The ordinal classification method proposed in this paper achieves the highest precision rate for class 3, at 41.2%. The corresponding recall rate is acceptable, at 21.3%. The F1 score is 28.1%, which is also the highest among all methods. In the meantime, the proposed approach can still get high F1 scores for classes 1 and 2. This method also has the highest macro-average precision and the third-highest macro-average F1 score.

As expected, Frank's method and Cheng's method have the second highest and third highest F1 score for class 3. Cheng's method gets almost the same F1 scores for classes 1 and 2 as Frank's method. This shows that Frank's and Cheng's methods are superior to the traditional nominal classification method, although rank monotonicity is not satisfied.

Surprisingly, Beckham1 and Beckham2 perform worse than nominal classification. Beckham1′s F1 scores for class 2 and 3 are smaller than these of nominal classification. Beckham2 cannot even predict any cases in class 3, resulting in a 0% F1 score for class 3. A possible reason is that Beckham's method is adversely impacted by the class-imbalance issue. Beckham's method relies on the estimation of β_k, which could be biased if the numbers of cases in classes are not equal-sized.

5 × 2 cross-validation and paired Student's *t*-test are used to test whether the ordinal classification method proposed in this paper is statistically better than other methods. As shown in Table 7, we compare this paper's method (method A) and other methods (method B) by testing whether their accuracy scores are from the same distribution. All p-values are smaller than 0.01, indicating that performance differences are statistically significant, and this paper's method is superior to the nominal classification method and other ordinal classification methods.

Table 7. Paired Student's *t*-test result.

Method A	Method B	*t*	*p*-Value
This paper	Nominal classification	24.99	0.000
	Beckham1	15.82	0.000
	Beckham2	12.68	0.000
	Frank	4.89	0.005
	Cheng	16.58	0.000

The computational cost of the proposed method is not much higher than nominal classification and other ordinal classification methods. The main computational cost of classification, either categorical or ordinal, is training k or k-1 single output XGBoost classifiers, which takes 21.8 s on the computing platform. Compared to other methods, the extra computational work of the proposed method is probability calibration and threshold moving, which costs about 4.5 s and are much faster than XGBoost training. Therefore, the proposed method can improve model performance with minimal extra computational cost.

Figures 7 and 8 present the predicted probabilities before and after calibration, respectively. The probability plot is a standard way to check how predicted probabilities fit empirical probabilities. Take class 1, for example, in which all samples are binned into groups based on their predicted probabilities of class 1. For each bin, we calculate the percentage of samples that are actually in class 1 (fraction of positives). The horizontal axis of Figures 7 and 8 are the mean predicted probabilities of each bin, and the vertical axis is the corresponding fraction of positives. Perfectly calibrated probabilities should have the mean predicted probability equal to the fraction of positives in each bin and should form a diagonal line in the probability plot.

Before calibration, the predicted probabilities of class 1 are very close to being perfectly calibrated. The predicted probabilities of class 2 are slightly underestimated. For example, when the predicted probability of class 2 is around 60%, the actual fraction of positive cases is 80%. This underestimation bias could be caused by XGBoost itself since decision tree-based classifiers do not generate calibrated probabilities. The predicted probabilities of class 3 are obviously overestimated since the probability plot is below the perfectly calibrated line. This problem is due to class imbalance and oversampling, which distorts the class distribution in the original data. Therefore, if the biased and uncalibrated probabilities of classes 1, 2, and 3 are directly used to calculate the cumulative probabilities, the cumulative probabilities will also be wrong and unreliable.

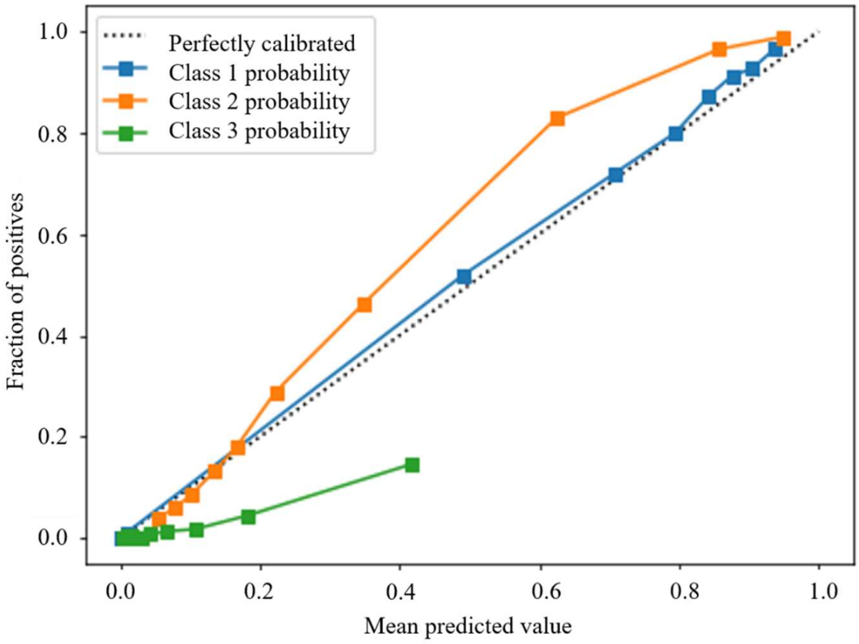


Figure 7. Probability plot of 3 classes before calibration.

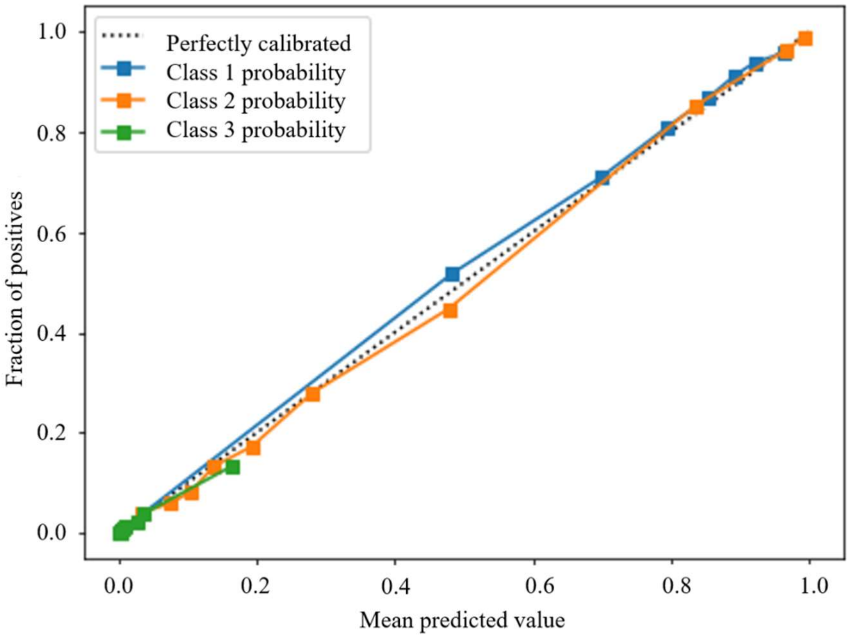


Figure 8. Probability plot of 3 classes after calibration.

After isotonic regression, all predicted probabilities are perfectly calibrated, as shown in Figure 8.

The thresholds T_k in (7) are determined by finding the F1-maximizing thresholds for the validation data. The optimal thresholds found for the data used in this paper are T_1 = 0.43 and T_2 = 0.33. Since 0.5 is the default threshold used in many studies and algorithms, we compare the optimal thresholds to the default threshold in Table 8. The default threshold leads to significantly worse results than the optimal thresholds. For the default threshold, the precision rate of class 3 is 15.8%, much smaller than 41.2% of the optimal thresholds, and the F1 score of class 3 is also smaller than that of the optimal thresholds.

Table 8. Performance of proposed method with different thresholds.

Method	Evaluation Metric	Class 1	Class 2	Class 3	Macro-Average
$T_1 = 0.5$ $T_2 = 0.5$	Precision (%)	86.4	81.5	15.8	61.2
	Recall (%)	88.2	77.4	27.3	64.3
	F1 (%)	87.3	79.4	20.0	62.2
$T_1 = 0.43$ $T_2 = 0.33$	Precision (%)	94.0	68.9	41.2	68.0
	Recall (%)	85.2	86.4	21.3	64.3
	F1 (%)	89.4	76.7	28.1	64.7

5. Conclusions

This research proposed an ordinal classification machine learning method to improve the prediction of imbalanced traffic crash injury severity. SMOTE-NC oversampling and category-combination are applied to relieve the class imbalance problem. XGBoost, SVM, and multi-layer perceptron machine learning are utilized to predict the injury severity of traffic crashes. Based on the analysis results, the effects of ejected from vehicle, number of vehicles involved, occupant type, and vehicle model year on the severity of traffic crashes are found to be important. The experimental results suggest that the proposed ordinal classification method provides better prediction results than other existing ordinal classification methods and traditional nominal classification, especially in minority classes. It was shown from the results that probability calibration and optimal thresholds are helpful in injury severity prediction.

Future efforts should focus on the following aspects: (1) establish a more comprehensive ordinal classification that combines cost-sensitivity with the ordinal classification method proposed in this paper. (2) try to solve the 5-category classification problem without combining any two or more categories.

Author Contributions: Conceptualization, S.Z. and K.W.; Formal analysis, S.Z. and C.L.; Funding acquisition, S.Z.; Methodology, K.W.; Supervision, K.W.; Validation, S.Z. and C.L.; Visualization, C.L.; Writing–original draft, S.Z. and K.W.; Writing–review & editing, S.Z. and K.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Foundation of Jiangsu Key Laboratory of Traffic and Transportation Security (No. TTS2018-03), Jiangsu Technology Innovation&C-V2X Academy Open Topics (No. KC-V2X2021002), and Natural Science Fund for Colleges and Universities in Jiangsu Province (No. 18KJA580001).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data can be requested to download at https://www.hsisinfo.org/ (accessed on 1 November 2021).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Input variables and counts of different crash injury severity cases.

No.	Variables	Code	Value	Severity				
				NIC	COP	OVI	SI	KSI
1	occupant: occupant type	1	Driver	0	27,964	10,989	1904	461
		2	Passenger	80,474	13,674	4209	809	199
		4	Bicyclist	0	0	2	0	0
		5	Other	0	4	0	1	0
2	seat: seating position	0	Other occupants	1022	81	37	16	3
		1	Driver	42,623	34,360	12,728	2220	581
		2~6	Passengers	33,206	6849	2301	449	68
		7	Station wagon rear	1402	164	61	15	1
		8	Truck/van rear	1121	102	44	8	2
		9	Position unknown	1100	86	29	6	5
3	collision: type of collision	10	Head-on	883	1045	588	277	106
		11	Sideswipe	17,943	4244	1717	271	51
		12	Rear end	43,690	23,685	3632	472	95
		13	Broadside	5460	4436	1563	317	62
		14	Hit object	9572	6093	4962	832	232
		15	Overturned	1531	1829	2520	502	105
		16	Auto-pedestrian	192	16	22	1	0
		17	Other	981	187	147	37	9
4	factor: first associated factor	10	Vehicle code violation	3338	1981	2762	671	186
		14	Vision obscurement	100	63	39	1	1
		15	Inattention	1438	747	462	68	4
		16	Stop and go traffic	5056	2358	301	39	5
		17	Enter/leave ramp	2163	994	310	32	7
		18	Previous collision	928	493	154	37	10
		19	Unfamiliar with road	119	35	36	11	1
		20	Defect vehicle equipment	259	158	132	20	6
		21	Uninvolved Vehicle	612	345	190	25	0
		22	Other	327	198	135	27	10
		23	None apparent	65,694	33,904	10,586	1765	427
		24	Runaway vehicle	100	49	14	0	0
5	cause: primary collision factor	1	Under influence of alcohol	3594	2044	2478	648	146
		2	Following too closely	3990	1382	476	77	17
		3	Failure to yield	3197	2162	695	139	27
		4	Improper turn	9711	6417	4527	741	227
		5	Speeding	41,831	23,529	4710	681	137
		6	Other violations	18,151	6108	2314	428	106
6	road: roadway class	1	Urban freeways	58,265	28,482	9088	1273	255
		2	Urban freeways < 4 lanes	259	91	49	9	1
		3	Urban two-lane roads	1398	969	289	73	17
		4	Urban multilane divided non-freeways	4133	3376	810	121	21
		5~11	Others	16,419	8724	4964	1238	366
7	eject: ejected from	0	Not ejected	79,883	40,309	13,382	1969	400
		1	Fully ejected	24	760	1500	658	223
		2	Partially ejected	4	68	134	52	33
		3	Unknown	563	505	184	35	4

Table A1. *Cont.*

No.	Variables	Code	Value	Severity				
				NIC	COP	OVI	SI	KSI
8	object: first object struck	1~7	Bridge structure	281	229	181	44	19
		10~15	Pole or sign post	1573	1003	828	138	26
		16~24, 27, 30	Concrete barrier	5646	4761	3840	611	176
		25~26	Water /drainage ditch	122	116	116	28	9
		28~29, 40	Plants	279	223	227	69	12
		41~43	Temporary barricades	1416	296	286	53	18
		44~46	Overturned/crash-cushion	993	1213	1864	377	62
		51	Call box	53	23	13	6	2
		98~99	Unkown or no object involved	486	191	225	42	18
		100	Vehicle	69,495	33,449	7517	1339	317
9	vehicles: number of vehicles	1	Total vehicle number = 1	10,087	7272	6971	1240	304
		2	Total vehicle number = 2	56,147	24,906	6116	1159	268
		3	Total vehicle number > 2	14,240	9464	2113	315	88
10	alcohol: alcohol involved	1	Yes	6327	3183	1209	214	32
		2	No	74,147	38,459	13,991	2500	628
11	motor: motorcycle involved	1	Yes	78,400	40,480	14,780	2635	642
		2	No	1517	738	292	56	15
12	drv_gender: driver's gender	1	Female	32,269	19,975	5481	792	185
		2	Male	48,205	21,667	9719	1922	475
		Total		80,474	41,642	15,200	2714	660

References

1. Farid, A.; Ksaibati, K. Modeling two-lane highway passing-related crashes using mixed ordinal probit regression. *J. Transp. Eng. Part A. Syst.* **2020**, *146*, 04020092. [CrossRef]
2. Rezapour, M.; Wulff, S.S.; Molan, A.M.; Ksaibati, K. Application of Bayesian ordinal logistic model for identification of factors to traffic barrier crashes: Considering roadway classification. *Transp. Lett.* **2021**, *13*, 308–314. [CrossRef]
3. Cerwick, D.M.; Gkritza, K.; Shaheed, M.S.; Hans, Z. A comparison of the mixed logit and latent class methods for crash severity analysis. *Anal. Methods Accid. Res.* **2014**, *3–4*, 11–27. [CrossRef]
4. Haghighi, N.; Liu, X.C.; Zhang, G.; Porter, R.J. Impact of roadway geometric features on crash severity on rural two-lane highways. *Accid. Anal. Prev.* **2018**, *111*, 34–42. [CrossRef] [PubMed]
5. Iranitalab, A.; Khattak, A. Comparison of four statistical and machine learning methods for crash severity prediction. *Accid. Anal. Prev.* **2017**, *108*, 27–36. [CrossRef] [PubMed]
6. Chang, L.-Y.; Wang, H.-W. Analysis of traffic injury severity: An application of non-parametric classification tree techniques. *Accid. Anal. Prev.* **2006**, *38*, 1019–1027. [CrossRef]
7. Abdel-Aty, M.A.; Abdelwahab, H.T. Predicting injury severity levels in traffic crashes: A modeling comparison. *J. Transp. Eng.* **2004**, *130*, 204–210. [CrossRef]
8. Delen, D.; Sharda, R.; Bessonov, M. Identifying significant predictors of injury severity in traffic accidents using a series of artificial neural networks. *Accid. Anal. Prev.* **2006**, *38*, 434–444. [CrossRef]
9. Alkheder, S.; Taamneh, M.; Taamneh, S. Severity prediction of traffic accident using an artificial neural network. *J. Forecast.* **2017**, *36*, 100–108. [CrossRef]
10. Savolainen, P.T.; Mannering, F.L.; Lord, D.; Quddus, M.A. The statistical analysis of highway crash-injury severities: A review and assessment of methodological alternatives. *Accid. Anal. Prev.* **2011**, *43*, 1666–1676.
11. Yasmin, S.; Eluru, N.; Ukkusuri, S.V. Alternative ordered response frameworks for examining pedestrian injury severity in New York City. *J. Transp. Saf. Secur.* **2014**, *6*, 275–300. [CrossRef]
12. Taylor, S.G.; Russo, B.J.; James, E. A comparative analysis of factors affecting the frequency and severity of freight-involved and non-freight crashes on a major freight corridor freeway. *Transp. Res. Rec.* **2018**, *2672*, 49–62. [CrossRef]
13. Chang, L.-Y.; Chien, J.-T. Analysis of driver injury severity in truck-involved accidents using a non-parametric classification tree model. *Saf. Sci.* **2013**, *51*, 17–22. [CrossRef]
14. Gutierrez, P.A.; Perez-Ortiz, M.; Sanchez-Monedero, J.; Fernandez-Navarro, F.; Hervas-Martinez, C. Ordinal regression methods: Survey and experimental study. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 127–146. [CrossRef]

15. Riccardi, A.; Fernández-Navarro, F.; Carloni, S. Cost-sensitive AdaBoost algorithm for ordinal regression based on extreme learning machine. *IEEE Trans. Cybern.* **2014**, *44*, 1898–1909. [CrossRef] [PubMed]
16. Verwaeren, J.; Waegeman, W.; Baets, B.D. Learning partial ordinal class memberships with kernel-based proportional odds models. *Comput. Stat. Data Anal.* **2012**, *56*, 928–942. [CrossRef]
17. Niu, Z.; Zhou, M.; Wang, L.; Gao, X.; Hua, G. Ordinal regression with multiple output CNN for age estimation. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4920–4928.
18. Jeong, H.; Jang, Y.; Bowman, P.J.; Masoud, N. Classification of motor vehicle crash injury severity: A hybrid approach for imbalanced data. *Accid. Anal. Prev.* **2018**, *120*, 250–261. [CrossRef]
19. Basso, F.; Basso, L.J.; Bravo, F.; Pezoa, R. Real-time crash prediction in an urban expressway using disaggregated data. *Transp. Res. Part C Emerging Technol.* **2018**, *86*, 202–219. [CrossRef]
20. Drosou, K.; Georgiou, S.; Koukouvinos, C.; Stylianou, S. Support vector machines classification on class imbalanced data: A case study with real medical data. *J. Data Sci.* **2014**, *12*, 727–754. [CrossRef]
21. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
22. Frank, E.; Hall, M. A simple approach to ordinal classification. In Proceedings of the 2001 European Conference on Machine Learning, Freiburg, Germany, 5–7 September 2001; pp. 145–156.
23. Cheng, J.; Wang, Z.; Pollastri, G. A neural network approach to ordinal regression. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 1279–1284.
24. Beckham, C.; Pal, C. A simple squared-error reformulation for ordinal classification. *arXiv* **2020**, arXiv:1612.00775.
25. Cao, W.; Mirjalili, V.; Raschka, S. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognit. Lett.* **2020**, *140*, 325–331. [CrossRef]
26. Collell, G.; Prelec, D.; Patil, K.R. A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multi-class imbalanced data. *Neurocomputing* **2018**, *275*, 330–340. [CrossRef]
27. Platt, J.C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*; MIT Press: Cambridge, MA, USA, 1999; pp. 61–74.
28. Zadrozny, B.; Elkan, C. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In Proceedings of the Eighteenth International Conference on Machine Learning, San Francisco, CA, USA, 28 June–1 July 2001; pp. 609–616.
29. Zadrozny, B.; Elkan, C. Transforming classifier scores into accurate multi-class probability estimates. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 694–699.
30. Sarkar, S.; Vinay, S.; Raj, R.; Maiti, J.; Mitra, P. Application of optimized machine learning techniques for prediction of occupational accidents. *Comput. Oper. Res.* **2019**, *106*, 210–224. [CrossRef]
31. Rahim, M.A.; Hassan, H.M. A deep learning based traffic crash severity prediction framework. *Accid. Anal. Prev.* **2021**, *154*, 106090. [CrossRef] [PubMed]
32. Yang, C.; Chen, M.; Yuan, Q. The application of XGBoost and SHAP to examining the factors in freight truck-related crashes: An exploratory analysis. *Accid. Anal. Prev.* **2021**, *158*, 106153. [CrossRef] [PubMed]
33. Guo, M.; Yuan, Z.; Janson, B.; Peng, Y.; Yang, Y.; Wang, W. Older pedestrian traffic crashes severity analysis based on an emerging machine learning XGBoost. *Sustainability* **2021**, *13*, 926. [CrossRef]
34. Parsa, A.B.; Movahedi, A.; Taghipour, H.; Derrible, S.; Mohammadian, A. Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. *Accid. Anal. Prev.* **2020**, *136*, 105405. [CrossRef] [PubMed]
35. Chen, C.; Zhang, G.; Qian, Z.; Tarefder, R.A.; Tian, Z. Investigating driver injury severity patterns in rollover crashes using support vector machine models. *Accid. Anal. Prev.* **2016**, *90*, 128–139. [CrossRef] [PubMed]
36. Li, X.; Lord, D.; Zhang, Y.; Xie, Y. Predicting motor vehicle crashes using support vector machine models. *Accid. Anal. Prev.* **2008**, *40*, 1611–1618. [CrossRef]
37. Bergstra, J.; Yamins, D.; Cox, D.D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 115–123.
38. Dietterich, T.G. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.* **1998**, *10*, 1895–1923. [CrossRef] [PubMed]