Article

ReMAHA–CatBoost: Addressing Imbalanced Data in Traffic Accident Prediction Tasks

Guolian Li 1,2, Yadong Wu 1,2,*, YuLong Bai 2,3 and Weihan Zhang 1,2,4

1 School of Computer Science and Engineering, Sichuan University of Science and Engineering, Yibin 644002, China; 321085406136@suse.edu.cn (G.L.); zhangweihansuse@163.com (W.Z.)
2 Sichuan Provincial Engineering Laboratory of Big Data Visual Analysis, Yibin 644002, China; 321085404428@suse.edu.cn
3 School of Automation and Information Engineering, Sichuan University of Science and Engineering, Yibin 644002, China
4 Sichuan Key Provincial Research Base of Intelligent Tourism, Zigong 643000, China
* Correspondence: selobdvat@gmail.com

Featured Application: ReMAHA–CatBoost is an advanced machine learning model designed for predicting traffic accident severity. It is constructed in two parts: ReMAHA (relief–F-based genetic algorithm with over-sampling algorithm for weighted Mahalanobis distance) and CatBoost, to offer an innovative solution in the field of imbalanced data classification. Key Features and Highlights: (1) ReMAHA Over-sampling: ReMAHA employs the Relief–F algorithm for feature selection and combines it with an innovative over-sampling technique to enhance prediction accuracy for minority classes; (2) Feature Engineering: The model leverages feature engineering to determine the significance of different attributes, enabling it to make precise predictions regarding accident severity; and (3) CatBoost Integration: ReMAHA incorporates CatBoost, a state-of-the-art gradient-boosting algorithm, to improve predictive performance by mitigating issues like overfitting and prediction bias. This paper elucidates the working principles of oversampling algorithms in machine learning tasks based on imbalanced datasets, specifically addressing how to resolve the issue of low accuracy stemming from imbalanced data at the data level. Based on the experimental results presented in this paper, it is evident that ReMAHA–CatBoost outperforms several other oversampling algorithms and models, especially on the US–Accidents traffic accident dataset characterized by an extreme class imbalance ratio of 91.40. This improved performance enhances the precision of traffic accident severity prediction.

Abstract: Using historical information from traffic accidents to predict accidents has always been an area of active exploration by researchers in the field of transportation. However, predicting only the occurrence of traffic accidents is insufficient for providing comprehensive information to relevant authorities. Therefore, further classification of predicted traffic accidents is necessary to better identify and prevent potential hazards and the escalation of accidents. Due to the significant disparity in the occurrence rates of different severity levels of traffic accidents, data imbalance becomes a critical issue. To address the challenge of predicting extremely imbalanced traffic accident events, this paper introduces a predictive framework named ReMAHA–CatBoost. To evaluate the effectiveness of ReMAHA–CatBoost, we conducted experiments on the US–Accidents traffic accident dataset, where the class label imbalance reaches up to 91.40 times. The experimental results demonstrate that the proposed model in this paper exhibits exceptional predictive performance in the domain of imbalanced traffic accident prediction.

Keywords: relief–F; imbalanced data; CatBoost; traffic accident; class imbalance
1. Introduction

Road traffic accidents pose significant economic and medical burdens globally, often leading to catastrophic family tragedies and accounting for approximately 1.3 million fatalities annually. However, as highlighted by the World Health Organization (WHO), injuries caused by traffic accidents can be prevented [1,2]. With the rapid advancement of machine learning, the field of traffic accident prediction has demonstrated tremendous potential. It not only enhances the accuracy of accident prediction but also allows for real-time response to traffic conditions, reducing accident risks. Existing research has primarily focused on whether traffic accidents will occur, often neglecting the further classification of potential accidents. Ideally, all potential traffic accidents should receive significant attention, with resources mobilized to prevent their occurrence. However, due to limited resources and the need for multi-agency co-ordination, it is often challenging to address all traffic accidents promptly. Therefore, it is essential to give additional attention to the possibility of severe traffic accidents and take necessary measures to prevent their occurrence or prepare for their management in advance.

Simultaneously, traffic accidents typically exhibit characteristics of imbalance, such as the far smaller number of accident samples compared to normal traffic samples [3], a significantly higher number of accidents in high-risk areas compared to low-risk areas [4], and an imbalance in the sample proportions of different accident types. Particularly, in traffic accidents, the number of severe events is much lower than that of common events. Machine learning models tend to learn this prior information regarding the proportions of the two sample classes in the training set, which results in an emphasis on the majority class during actual prediction (possibly leading to better accuracy for common accidents but poorer accuracy for severe accidents). Unfortunately, in previous work, the probability of occurrence of these minority class accidents is often overlooked, and especially the occurrence of severe traffic accidents should receive more attention.

Due to the presence of a considerable number of boolean features, such as points of interest (POI), streets, wind direction, weather, and other factors in traffic accident datasets, there remains an issue of feature sparsity in traffic accident prediction. To address the aforementioned issues, this paper introduces the ReMAHA–CatBoost traffic accident prediction model, aiming to classify potential traffic accidents and to resolve the problems of imbalanced sample quantities, high-dimensional, and sparse feature-induced prediction inaccuracies in traffic accidents. This model incorporates feature selection and clustering operations into the oversampling process to improve the problem of traditional genetic algorithms generating overly divergent data. Simultaneously, it addresses the issue of Mahalanobis distance calculation during the oversampling process, caused by sparse features. Additionally, the weight matrix obtained using the relief–F algorithm not only aids in Mahalanobis distance weighted computation, increasing the impact of feature importance on sampling, but also serves for original feature selection to avoid the curse of dimensionality and overfitting. In this paper, the contributions can be summarized as follows:

- We introduce ReMAHA–CatBoost, a traffic accident prediction model based on CatBoost [5]. ReMAHA–CatBoost leverages the relief–F algorithm to obtain feature importance and MAHAKIL to avoid generating data with a centralized distribution, effectively alleviating the imbalance issue in traffic accident data;
- To address the concern of Mahalanobis distance computation in MAHAKIL, which does not consider feature importance, we employ the relief–F algorithm to obtain a diagonal matrix representing feature importance and perform weighted Mahalanobis distance calculations;
- Additionally, to tackle the problem of generating overly divergent data and handling the overall large sample size in classical MAHAKIL algorithms, we utilize mini batch K-means to cluster the samples before data generation;
• Furthermore, we leverage the advantage of CatBoost in reducing gradient bias, enhancing the model’s generalization capabilities;

• We evaluate ReMAHA–CatBoost and other three sampling algorithms combined with CatBoost, as well as ReMAHA combined with four other prediction models on the US–Accidents dataset. The results demonstrate that ReMAHA–CatBoost outperforms other models in imbalanced traffic accident data, highlighting its generalization and effectiveness in the domain of traffic accidents.

The structure of this paper is outlined as follows: in Section 2, we delve into pertinent research concerning traffic accident prediction and approaches addressing data imbalance. Section 3 expounds upon the utilized dataset and delineates the architecture design of ReMAHA–CatBoost. The subsequent section, Section 4, presents a detailed analysis of the experimental results. Section 5 engages in a discussion of the research findings presented herein. Finally, Section 6 summarizes the broad body of work and real-world contributions of this study.

2. Related Work

In the field of traffic accident research, accurate accident prediction contributes to the identification of traffic hazards, optimization of traffic system resource allocation, and timely provision of medical assistance. Currently, there have been many excellent research achievements in the domain of traffic accident prediction [6–8], and they have played a crucial role in revealing the mechanisms behind traffic accident occurrences [9]. For example, Nur et al. [10] used the least-squares method to analyze the relationship between environmental factors and accidents, identifying correlations between rainfall, temperature, wind speed, and accidents. Li et al. [11] conducted association rule mining and classification studies, revealing that physiological factors such as alcohol consumption are more likely to lead to fatalities in accidents, while the impact of natural environmental factors like weather on the fatality rate is relatively smaller. Wang et al. [12] employed the a priori algorithm based on association rules to explore influential factors in traffic accidents and investigate strong association rules among various causal factors.

Through our investigation, we found that existing research often focuses on the causes and underlying factors of accident occurrences, using this information to predict whether traffic accidents will happen. Furthermore, most studies conduct single-dimensional analyses, only allowing for shallow data analysis and making it difficult to express the spatiotemporal correlations in traffic accidents [13]. Moreover, these studies do not conduct further classification of traffic accidents. Offering only the set of conditions that may trigger a traffic accident, such coarse-grained information also has limited contributions to the traffic system.

Simultaneously, due to the phenomenon of imbalanced samples in historical traffic accident data, especially where the number of severe traffic accidents is significantly lower than that of common accidents, traditional prediction models demonstrate poor performance. Currently, there are also many outstanding works addressing the common problem of sample imbalance, which has led to improved accuracy for minority class samples in prediction models. Broadly, their approaches mainly fall into two directions: data-level and algorithm-level [14,15]. Data-level approaches include oversampling, undersampling, and hybrid sampling, while algorithm-level approaches involve ensemble learning and cost-sensitive algorithms. Specifically, they are as follows:

(1) Undersampling Methods: Undersampling methods begin with the majority class data and reduce the dataset by removing the class with a higher quantity to balance the dataset. For example, Dai et al. [16] started by eliminating duplicate samples and expanded the detection range of the Tomek-link undersampling algorithm by introducing a global re-labeling index. Wei et al. [17], focusing on data complexity, proposed an undersampling algorithm based on weighted complexity, WCP-UnderSampler, which
achieved promising results in defect prediction datasets. However, undersampling can lead to information loss in the majority class and affect classifier generalization.

(2) Oversampling Methods: Oversampling methods introduce new minority class data to create a superset of minority class samples, reducing the imbalance between data categories [18]. For instance, SMOTE is a classical oversampling algorithm for addressing imbalanced data [19]. Gao et al. [20] proposed an improved SMOTE oversampling algorithm based on ant clustering, addressing both inter-cluster and intra-cluster data imbalance. Bennin et al. [21] introduced genetic chromosome theory into the sampling domain, presenting the MAHAKIL algorithm, which ensures that the generated data inherit the features of parent data instances. It outperforms other oversampling methods on multiple datasets. However, oversampling methods can lead to data overlap and distribution marginalization issues, potentially trapping the algorithm in local optima.

(3) Mixed Sampling: Mixed sampling combines both undersampling and oversampling techniques to balance the data. For example, Wang et al. [22] synthesized minority class and majority class samples separately using a generative adversarial network (GAN) and SMOTE. They compared this dual oversampling strategy to results obtained from a single oversampling approach targeting the minority class only, and found that the dual oversampling strategy outperformed the single oversampling method. While mixed sampling can alleviate information loss in undersampling and overfitting in oversampling, research suggests that the order of executing undersampling and oversampling in mixed sampling can influence predictive accuracy [23].

(4) Ensemble Learning: Different from traditional individual learners, ensemble learning combines multiple weak learners to create a strong learner [24]. The two most typical forms of ensemble learning, based on the composition of base learners, are bagging and boosting. Navaneeth et al. [25] implemented a hybrid ensemble learning model that combines CNN with CatBoost, providing a new approach for non-invasive COVID-19 detection. Yan et al. [26] combined undersampling with ensemble learning and proposed a spatial undersampling model for local pattern learning. For ensemble methods, their purpose is to enhance overall accuracy, making them challenging to work effectively on imbalanced classification problems alone [27]. They often need to be used in conjunction with other algorithms.

(5) Cost-Sensitive Learning: A new cost-sensitive imprecise classification decision tree has been introduced, which considers error costs by weighting instances and incorporates these costs during tree construction. Serafin et al. [28] proposed the nonparametric predictive inference model to improve cost-sensitive decision trees. In addition to modifying tree generation, the fusion of active learning with cost-sensitive algorithms can effectively enhance the classification performance of imbalanced data [29]. Although combining cost-sensitive learning with classification models effectively improves predictive accuracy for classification results, determining misclassification costs still requires substantial effort.

These various types of methods each have their own characteristics and have made significant progress. Considering the randomness and diversity of traffic accident data [30], as well as the abundance of sparse features in the field of traffic accidents, applying them directly to address sample imbalance in traffic accident prediction can lead to suboptimal prediction performance. Given these considerations, this paper addresses the sample imbalance phenomenon in severe imbalanced traffic accident datasets from both a data-level and algorithm-level perspective, merging the two to enhance the robustness of predictions.

3. Methodology

This paper aims to further classify the results of traffic accident prediction and address the common issue of sample imbalance in traffic accident data, which can potentially reduce the accuracy of model predictions. To tackle this challenge, we propose a traffic accident prediction model that integrates relief–F, MAHAKIL, mini batch K-
means, and CatBoost. The primary objective is to mitigate the adverse impact of sample imbalance on model performance and more accurately identify the severity of traffic accidents.

To evaluate the effectiveness of the ReMAHA–CatBoost model in the field of traffic accident prediction, we chose the US–Accidents dataset [31,32] as the subject of our study. We initially conducted a series of data preprocessing steps, including missing values, handling duplicate values, outliers, one-hot encoding, and feature engineering. These steps provided critical data for predicting the severity of traffic accidents. Subsequently, we developed a hybrid sampling prediction model using the ReMAHA–CatBoost architecture. This model is designed to alleviate the data imbalance issue in traffic accidents and perform the classification task for the severity of traffic accidents.

At the data-level, due to the strong randomness in traffic accident data, it is challenging to generate data using traditional oversampling methods. Therefore, we initially perform clustering on minority class samples, enhancing the similarity between samples. We then use MAHAKIL to generate new minority class samples between clusters. Furthermore, due to the presence of a considerable number of boolean features in traffic accident characteristics, feature sparsity can occur. When dealing with large datasets and extremely sparse features, computing the Mahalanobis distance becomes challenging as the covariance matrix might become non-invertible, thereby impeding the use of traditional Mahalanobis distance computation.

We introduce the relief–F algorithm to process the features after clustering, which can eliminate near-uniform labels in the clusters and populate them after generating the data. Such processing avoids the influence of sparse features on the computation of the Mahalanobis distance. In addition, we use the obtained diagonal matrix of feature importance weights for Mahalanobis distance weighting. The purpose of this step is to optimize the traditional Mahalanobis distance which has the problem of exaggerating the role of small variables, and to improve the influence of important features on the distance.

Despite the extensive volume of traffic accident data, which is highly advantageous for subsequent predictions. However, the internal data distribution is unbalanced, which leads to the problem of overfitting easily using traditional algorithms, making the prediction results biased towards categories with more samples. In this regard, we used CatBoost for prediction at the algorithm-level, which is used in order to mitigate the overfitting problem in the traffic accident classification task. CatBoost’s ranking boost and overcoming gradient bias enhance the model’s predictive capabilities on the imbalanced traffic accident dataset. Subsequent subsections will provide detailed explanations of the mentioned methods. The overall workflow of the proposed method is outlined in Figure 1, which will be elaborated in this section.

![Figure 1. Methodology process diagram.](image-url)
3.1. Dataset

Given the extensive volume, imbalance, and sparsity of features within traffic accident data, we selected the highly representative dataset US–Accidents as the focal point of our study. It contains all the traffic accident data in the United States from 2016 to 2022 and is sourced from both Bing and MapQuest. The dataset comprises over 7 million records, making it large in terms of data quantity, and it includes a wide range of feature dimensions. Based on different feature attributes, the dataset is categorized into three types: numerical, boolean, and text, as shown in Table 1.

Table 1. Feature description.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Number of Features</th>
<th>Main Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric</td>
<td>15</td>
<td>Latitude and longitude, severity, and others</td>
</tr>
<tr>
<td>Boolean</td>
<td>13</td>
<td>Nearby points of interest and others</td>
</tr>
<tr>
<td>Text</td>
<td>33</td>
<td>Accident description, street, and others</td>
</tr>
</tbody>
</table>

The US–Accidents traffic accident dataset is imbalanced, where the accident severity levels range from 1 to 4 (Table 2). These severity levels indicate increasing degrees of severity, with 1 representing the least impact on traffic (i.e., causing short-term delays) and 4 representing a more significant impact on traffic (i.e., causing long-term delays). In terms of data quantity, there is a substantial difference between the number of events with severity levels 1 and 4 compared to severity levels 2 and 3.

The overall dataset exhibits a pronounced imbalance. Based on the imbalance ratio (IR) equation, it can be concluded that the imbalance ratio of this dataset is as high as 91.40.

\[
IR = \frac{N_{\text{major}}}{N_{\text{minor}}} \tag{1}
\]

where \( N_{\text{major}} \) represents the sample size with the highest number of categories in the dataset and \( N_{\text{minor}} \) represents the sample size with the lowest number of categories in the dataset.

Table 2. Label description.

<table>
<thead>
<tr>
<th>Accident Severity</th>
<th>Number of Classes</th>
<th>Percentage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>67,366</td>
<td>0.87</td>
</tr>
<tr>
<td>2</td>
<td>6,156,981</td>
<td>79.67</td>
</tr>
<tr>
<td>3</td>
<td>1,299,337</td>
<td>16.81</td>
</tr>
<tr>
<td>4</td>
<td>204,710</td>
<td>2.65</td>
</tr>
</tbody>
</table>

Data preprocessing stands as a crucial part throughout the entire machine learning process. In order to avoid the “noise” in the dataset from affecting the experiment, we conducted operations to handle missing values, duplicates, and outliers. Additionally, we engineered some new features. Following these data processing steps, we partitioned the dataset into two subsets, the training set and the test set, for model validation purposes.

3.1.1. Missing Value Handling

Table 3 describes the distribution of missing values in the dataset. In the experiment, missing values in natural environmental features were imputed using the median of the same weather station and the same month. Additionally, for spatial environmental features, missing values were imputed using data from geographically close locations. Some features with a significant number of missing values lost their original utility and were subsequently removed.
Table 3. Distribution of missing values (%).

<table>
<thead>
<tr>
<th>Feature</th>
<th>Missing</th>
<th>Feature</th>
<th>Missing</th>
<th>Feature</th>
<th>Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>End_Lat</td>
<td>44.03</td>
<td>Weather_Timestamp</td>
<td>1.56</td>
<td>Precipitation (in)</td>
<td>28.51</td>
</tr>
<tr>
<td>End_Lng</td>
<td>44.03</td>
<td>Temperature (F)</td>
<td>2.12</td>
<td>Weather_Condition</td>
<td>2.24</td>
</tr>
<tr>
<td>Description</td>
<td>0.00</td>
<td>Wind_Chill (F)</td>
<td>25.87</td>
<td>Sunrise_Sunset</td>
<td>0.30</td>
</tr>
<tr>
<td>Street</td>
<td>0.14</td>
<td>Humidity (%)</td>
<td>2.25</td>
<td>Civil_Twilight</td>
<td>0.30</td>
</tr>
<tr>
<td>City</td>
<td>0.00</td>
<td>Pressure (in)</td>
<td>1.82</td>
<td>Nautical_Twilight</td>
<td>0.30</td>
</tr>
<tr>
<td>Zipcode</td>
<td>0.03</td>
<td>Visibility (mi)</td>
<td>2.29</td>
<td>Astronomical_Twilight</td>
<td>0.30</td>
</tr>
<tr>
<td>Timezone</td>
<td>0.10</td>
<td>Wind_Direction</td>
<td>2.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Airport_Code</td>
<td>0.29</td>
<td>Wind_Speed (mph)</td>
<td>7.39</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.1.2. Duplicate Value Handling

From the perspective of the features in the US–Accidents traffic accident dataset, there are two sources of duplicate values in the dataset:

1. Repetitive reporting of the same traffic accident information by different reporting sources. For the same traffic accident, different reporting sources may report it, and differences in the source can also result in variations in geographic and time information. Based on the different data sources, the source field is categorized into Source 1: reported from Bing, Source 2: reported from MapQuest, and Source 3: reported from both parties;
2. Multiple reports of information caused by the same traffic accident source. This is because the determination of traffic accident types in US–Accidents is based on whether traffic delays occur in a specific geographical location, and there is a strong spatial correlation between adjacent roads. Therefore, closely located traffic congestion may have a chain reaction, leading to multiple reports of traffic congestion on closely located roads caused by the same traffic accident source.

Given the two mentioned reasons, to ensure minimal and accurate duplicate data in the experiment’s dataset, while also avoiding data leakage due to duplicate values, we excluded the 2016 data that had overall fewer records and a substantial disparity in data volume between different months. We retained data from 2017 to 2022 for the experiment. Subsequently, we screened the remaining dataset of over six million entries for duplicate values, filtering out entries with a time interval smaller than 10 min and a geographical distance less than 250 m, assessed by the geographical location formula illustrated in Equation (2). Here, $\phi_a$ and $\phi_b$ represent the longitudes of points a and b, respectively, while $\lambda_a$ and $\lambda_b$ denote the latitudes of points a and b, respectively.

$$
\text{dist}(a,b) = 2\arcsin\left(\sqrt{\sin^2\left(\frac{\phi_a - \phi_b}{2}\right) + \cos(\phi_a) \cos(\phi_b) \sin^2\left(\frac{\lambda_a - \lambda_b}{2}\right)}\right)
$$

To provide a more detailed demonstration of the process of eliminating duplicate data based on time and distance, we will illustrate it through the following example (Table 4):

Table 4. Example of Duplicate Data Processing.

<table>
<thead>
<tr>
<th>Start_Time</th>
<th>Start_Lat</th>
<th>Start_Lng</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>23 August 2019 19:00:21</td>
<td>33.775450</td>
</tr>
<tr>
<td>X2</td>
<td>23 August 2019 19:15:40</td>
<td>33.992460</td>
</tr>
<tr>
<td>X3</td>
<td>23 August 2019 19:09:30</td>
<td>32.766960</td>
</tr>
<tr>
<td>X4</td>
<td>23 August 2019 19:02:37</td>
<td>33.773290</td>
</tr>
</tbody>
</table>

Assuming that the above four data are subjected to repeat value judgment, the data pairs $(X_1, X_2)$, $(X_2, X_3)$, and $(X_2, X_3)$ with time interval less than 10 min are firstly identified. Then, the longitude and latitude values of the above data pairs are substituted into
Equation (2) for distance judgment, which yields dist(X_i, X_3) = 129,461.41 m, dist(X_i, X_4) = 240.38 m, and dist(X_3, X_4) = 179,192.03 m. Therefore, we believe that the duplicated point may be between X_i and X_4, and that X_i occurs earlier than X_4, therefore, X_i is the real place where the accident originated.

In this section, we screened and deleted duplicate values from 2017 to 2022, and deleted duplicate value data with time intervals of less than 10 min and geographic locations of less than 250 m to ensure the purity of the data and to avoid data leakage.

3.1.3. Handling Outliers

In addition, there are some anomalies in the dataset due to malfunctioning environmental testing instruments and other reasons. If they are not dealt with, they will cause the model’s predictions to be skewed. In order to provide a visual representation of the original data, this experiment conducted a feature analysis. Table 5 presents descriptive statistics for some of the features in the original data.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Q1</th>
<th>Q3</th>
<th>Upper</th>
<th>Lower</th>
<th>Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature (F)</td>
<td>49.5</td>
<td>76.0</td>
<td>115.75</td>
<td>9.75</td>
<td>203.0</td>
<td>-89.015</td>
</tr>
<tr>
<td>Humidity (%)</td>
<td>48.0</td>
<td>84.0</td>
<td>138.0</td>
<td>-6.0</td>
<td>100.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Pressure (in)</td>
<td>29.35</td>
<td>30.03</td>
<td>31.05</td>
<td>28.33</td>
<td>58.63</td>
<td>0.0</td>
</tr>
<tr>
<td>Wind_Speed (mph)</td>
<td>4.6</td>
<td>10.4</td>
<td>19.1</td>
<td>-4.10</td>
<td>1087.0</td>
<td>0.033</td>
</tr>
</tbody>
</table>

The Table 5 includes the upper quartile (Q1), lower quartile (Q3), lower bound (lower), upper bound (upper), maximum (max), and minimum (min) for numerical features such as temperature (F), humidity (%), pressure (in), and wind speed (mph). The interquartile range (IQR) is calculated as:

\[ IQR = Q3 - Q1 \]  
\[ lower = Q1 - 1.5 \times IQR \]  
\[ upper = Q3 + 1.5 \times IQR \]

The maximum and minimum values represent the maximum and minimum occurrences of a particular feature in the original dataset. If the maximum value is greater than the upper bound or the minimum value is lower than the lower bound, it indicates the presence of outliers in that feature (e.g., Wind_Speed (mph)). And when the maximum value is less than the upper bound and the minimum value is greater than the lower bound, it signifies the absence of outliers in that feature (e.g., humidity (%)). It is worth noting that the maximum and minimum values of some environmental features far exceed the most extreme values in a normal natural environment. Since the upper and lower bounds of the IQR can effectively represent the distribution of the majority of the data [33,34]. To ensure that the data are closer to real environmental conditions, in this experiment, data that fall significantly outside the interquartile range are removed.

3.1.4. Feature Engineering

In order to fully incorporate text features into the model building process, the original text features were one-hot encoded. Here are some examples of the original text features (Table 6):
Table 6. Example of Text Features.

<table>
<thead>
<tr>
<th>Street</th>
<th>Wind_Direction</th>
<th>Weather_Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex.1</td>
<td>I-580 W</td>
<td>Calm</td>
</tr>
<tr>
<td>Ex.2</td>
<td>N Main St</td>
<td>WSW</td>
</tr>
<tr>
<td>Ex.3</td>
<td>Bayview Ave</td>
<td>S</td>
</tr>
</tbody>
</table>

The processed textual features are shown below (Table 7):

Table 7. Example of Processed Textual Features.

<table>
<thead>
<tr>
<th>Rd</th>
<th>Dr</th>
<th>St</th>
<th>Ave</th>
<th>Hwy</th>
<th>WindE</th>
<th>WindN</th>
<th>WindS</th>
<th>WindW</th>
<th>Clear</th>
<th>Cloud</th>
<th>Rain</th>
<th>Snow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Ex.2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ex.3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

3.2. The ReMAHA Algorithm

This study introduces a novel model for predicting the severity of unbalanced traffic accidents (ReMAHA–CatBoost). It aims to address the issue of poor prediction performance caused by data imbalance in the traffic accidents domain. At the data level, we propose a new oversampling algorithm called ReMAHA, which effectively prevents overfitting in the generated data.

The ReMAHA model consists of three main components: feature weight calculation, inter-cluster weighted Mahalanobis distance ranking, and genetic algorithm for generating new data pairwise. First, we use the relief–F algorithm to extract significant features, alleviating the issue of excessively sparse features. Through the relief–F algorithm, we calculate the importance weights of features and represent them as a diagonal matrix. Next, we use a clustering algorithm to cluster minority class samples, aiming to ensure that the data do not become too scattered during the generation of new data pairwise using a genetic algorithm. Once the data are clustered into clusters, we apply weighted Mahalanobis distance ranking to the data between different clusters. Finally, we employ a genetic algorithm to generate new data pairwise, addressing the problems of data overlap and excessive concentration that may arise from oversampling. The flow of the ReMAHA model is shown in Figure 2.

![Figure 2. Flowchart of the ReMAHA algorithm.](image-url)
3.2.1. Feature Weight Calculation

In this study, we first employed the relief–F algorithm for feature selection and obtained a weight matrix used to assess feature importance. The choice of using the relief–F algorithm is driven by the following reasons: due to the sparsity of features and the large volume of traffic accident data, both of which result in issues with the covariance matrix being non-invertible, using the relief–F algorithm to filter out these overly sparse and uninformative features can alleviate such problems.

Relief–F is an algorithm designed for multi-class problems to calculate feature weights based on the correlation between features and class labels. Its primary objective is to calculate the weights between features, thus determining which features are more influential for subsequent classification tasks [35]. Given a feature A, its weight at the t-th iteration is denoted as \( W_t(A) \). In any \( t + 1 \) iteration, the relief–F algorithm updates the weighting of the feature vector as follows:

\[
W_{t+1}(A) = W_t(A) + \frac{1}{T} \sum_{k} \frac{p(c)}{1 - p(Class(X_i))} \times \sum_{i \in (X_i, NM_i)} \frac{\text{diff}(A, X_i, NH_i)}{k}
\]

where \( T \) represents the number of iterations, \( k \) is the number of nearest neighbors chosen, \( p(c) \) represents the prior probability of class \( c \), and \( Class(X_i) \) indicates the class to which \( X_i \) belongs.

In a given dataset, the relief–F algorithm operates as follows: first, a sample \( X_i \) is randomly selected, and then the nearest same-class sample point \( NH_i \) and the nearest different-class sample point \( NM_i \) are found in the dataset. Here, the function \( \text{diff}(A, X_i, X_j) \) is used to calculate the distance between sample points \( X_i \) and \( X_j \) in feature \( A \), where \( \max(A) \) and \( \min(A) \) denote the maximum and minimum values of feature \( A \).

For numerical features, when feature \( A \) is numerical:

\[
\text{diff}(A, X_i, NH_i) = \frac{[X_i, NH_i]}{\max(A) - \min(A)}
\]

For discrete features, when feature \( A \) is discrete:

\[
\text{diff}(A, X_i, NH_i) = \begin{cases} 0 & X_i = NH_i \\ 1 & X_i \neq NH_i \end{cases}
\]

After \( T \) iterations, each feature will obtain a feature weight \( w_i \), forming a feature weight matrix \( W = [w_1, w_2, ..., w_m] \). The larger \( w_i \) is, the better the feature is for classification.

3.2.2. Inter-cluster Weighted Mahalanobis Distance Sorting

Directly generating new data using a genetic algorithm can lead to excessive divergence in the generated data, potentially causing a loss of original features [36]. Given the large volume of traffic accident data, this study first employs mini batch K-means to cluster minority class sample data. This process groups data with similar features into clusters, which are then used for inter-cluster sorting of the sample data.

Additionally, traditional Mahalanobis distance, although addressing the issue of dimension measurement not resolved by Euclidean distance, neglects the varying influence of features on labels. Applying feature weighting can lead to more precise distance calculations between sample points. Here, it is assumed that the sample for calculating the distance originates from the same cluster, with a cluster center represented as \( C = (c_1, c_2, ..., c_n) \), and the point for Mahalanobis distance calculation as \( X = (x_1, x_2, ..., x_n) \). Following that, the feature importance diagonal matrix \( B \) is generated using the relief–F algorithm. The weighted Mahalanobis distance is then represented as follows:
\[ D_M(X, C) = \sqrt{(X - C)^T B \Sigma^{-1} B^T (X - C)} \]  

(9)

\[ \Sigma\] represents the covariance matrix, and the formula for calculating the covariance is as follows:

\[ \text{cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \]  

(10)

3.2.3. Genetic Algorithm for Generating New Data

The genetic theory of chromosome inheritance posits that during the formation of chromosomes, each parent contributes an equal half of the total genes to their offspring [37]. The new offspring inherits 50% of their genes from each parent, making them similar to their parents while also possessing unique traits. In this study, sample data are treated as chromosomes, and new data are constructed from these samples. This approach ensures that the newly generated data retain common characteristics while also exhibiting the uniqueness of the original two datasets. Using genetic algorithms guarantees the diversity of generated data and effectively addresses the issues of marginalization of data distribution and data overlap that traditional oversampling methods encounter.

The formula for generating data using genetic algorithms, as represented in Equation (11), incorporates an influence factor \( \beta \) to preserve the uniqueness of samples. In this study, we set \( \beta \) to 0.5 to a greater extent:

\[ x = \beta a + (1 - \beta) b \]  

(11)

The letters \( a \) and \( b \) represent the parent samples used for generating data, while ‘\( x \)’ represents the newly generated sample. This means that, in this experiment, since \( \beta \) is equal to 0.5, both parent samples have the same effect on the newly generated sample.

3.2.4. ReMAHA

The ReMAHA algorithm, formed by combining the above-mentioned methods, can be summarized into three main steps:

Step 1: Utilize the relief–F feature selection algorithm on the entire dataset to assess the influence of features on accident severity and acquire a weight diagonal matrix;

Step 2: Perform clustering on the minority class samples in the dataset, ensuring that data with similar features are clustered together in the same clusters. Calculate the distance from each point to the cluster center within each cluster and rank them based on the Mahalanobis distance weighted using the weight diagonal matrix;

Step 3: Generate new data using genetic algorithms with \( \beta \) equal to 0.5 based on the sorted distances between data points, and then re-sort the newly generated data. To prevent excessive dispersion, iterate up to 5 times at most for each cluster.

The schematic of ReMAHA-generated data is shown in Figure 3 and the algorithm is shown in Algorithm 1.
Algorithm 1: ReMAHA

Input: Imbalanced dataset $D$, the number of clusters $K$.
Output: Oversampling dataset $D_M$

1: Split $D$ into a minority class $D^+$ and the largest majority class $D^-$
2. $b = \text{Relief-F Feature Selection} (D)$
3. $B = \text{diag}(b)$
4: Initialize dataset $D_M$
5: Calculate each cluster $D^+_i$ through $\text{Mini\_Batch\_Kmeans} (D^+)$
6: FOR $i$ in 1, 2, ..., $K$ do:
7: \hspace{1em} $T = \text{len}\left(D^+_i\right)/\text{len}\left(D^+\right)$
8: \hspace{1em} iter = 0
9: \hspace{1em} WHILE $T > 0$ and iter < 5:
10: \hspace{2em} $D^+_\text{sorted} = \text{sorted\_by\_weightedMAHA} (D^+; B)$
11: \hspace{2em} Initialize $D^+\text{temp} = \{\}$
12: \hspace{2em} iter++
13: \hspace{2em} FOR $x_1, x_2$ in $D^+\text{sorted}$:
14: \hspace{3em} $x_{\text{new}} = \text{Generate\_by\_GA} (x_1, x_2)$
15: \hspace{3em} IF $T > 0$:
16: \hspace{4em} $D^+\text{temp}.\text{append} (x_{\text{new}})$
17: \hspace{3em} ELSE BREAK
18: \hspace{2em} $T--$
19: \hspace{2em} END FOR
20: \hspace{2em} $D^+\text{temp}.\text{append} (D^+_i)$
21: \hspace{2em} END WHILE
22: \hspace{2em} $D_M.\text{append} (D^+_i)$
23: END FOR
24: RETURN $D_M$

3.3. CatBoost

On the algorithm level, we have chosen to employ CatBoost, a variant of gradient-boosting algorithms [38,39]. Similar to other gradient-boosting algorithms, CatBoost aims to iteratively learn from errors by combining multiple weak learners.

In gradient-boost models, a commonly used node-splitting method is greedy target-based statics, which utilizes the average label value as the node-splitting criterion. However, when extreme values exist in the dataset or when there is a discrepancy between the data distribution of the training and test sets, using the mean value to split nodes can lead to conditional shift, resulting in decreased prediction accuracy. CatBoost mitigates...
conditional shift by introducing a prior distribution term, represented by $p$ and a weight coefficient, represented by $a$. The formula is as follows:

$$x'_{ik} = \frac{\sum^{p-1}_{i=1} x_{\sigma_{i,s}} = x_{\sigma_{i,s}}} {\sum^{p-1}_{i=1} x_{\sigma_{i,s}} = x_{\sigma_{i,s}}} + a \times p$$

(12)

Furthermore, since gradient-boost models use the same dataset for training in each iteration, it may lead to prediction shift concerning the gradient and the true distribution[40]. CatBoost addresses this issue by employing a ranking boost approach, training a separate model $M$ for each sample. Consequently, the current model $M$ is not trained using the current sample. CatBoost’s handling of conditional and prediction shifts makes it more suitable for situations involving imbalanced data.

4. Experiments

When evaluating the performance of the proposed models, we addressed the following research questions:

(1) Does varying the number of clusters affect the generation of new data?

(2) How does the quality of the newly generated data using our proposed method compare with other sampling techniques?

(3) How does this method perform on traffic accident datasets compared to other baseline methods?

4.1. Experimental Setup

The experimental platform utilized in this study includes a system with 64 GB of RAM, an 11th Gen Intel (R) Core (TM) i7-11700K CPU clocked at 3.60 GHz, an NVIDIA GeForce RTX 3070 Ti GPU, all running on CentOS 7.9.

4.2. Evaluation Metrics

The confusion matrix is the most commonly used method for evaluating the performance of classification problems. The definition of a confusion matrix is shown in Table 8.

<table>
<thead>
<tr>
<th>Predicate Positive</th>
<th>Predicate Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positive</td>
<td>TP</td>
</tr>
<tr>
<td>True Negative</td>
<td>FN</td>
</tr>
<tr>
<td>False Positive</td>
<td>FP</td>
</tr>
<tr>
<td>False Negative</td>
<td>TN</td>
</tr>
</tbody>
</table>

Based on the definition of the confusion matrix, various other data classification evaluation metrics have been derived.

Precision represents the proportion of correctly predicted samples for a specific class to the total number of samples predicted as that class:

$$\text{Precision} = \frac{TP}{TP + FP}$$

(13)

Recall represents the proportion of correctly predicted samples for a specific class to the total number of samples for that class:

$$\text{Recall} = \frac{TP}{TP + FN}$$

(14)
Typically, precision and recall are trade-offs, and in many cases, it is necessary to consider both precision and recall simultaneously. The $F_\beta$ score is used to provide a weighted harmonic mean of precision and recall [41]. Its calculation is as follows:

$$F_\beta = \frac{1}{C} \sum_{i=1}^{C} \left( \frac{1 + \beta^2}{\beta^2} \right) \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

(15)

The F1–Score is the most commonly used $F_\beta$ score, where $\beta$ is equal to 1. It balances precision and recall equally.

4.3. The Impact of Clustering on Generated Data

Considering that at the algorithm level, the proposed model in this paper involves the selection of the number of clusters. To demonstrate the effectiveness of clustering in traffic accident prediction, we conducted experiments to investigate the impact of varying cluster numbers on the results. Given that oversampling algorithms primarily affect the minority class in imbalanced datasets, this experiment particularly focused on observing the effect of cluster numbers on labels 1 and 4 in the dataset.

Clustering is one of the key steps of the model proposed in this paper, and Figure 4 depicts the relationship between the number of clusters and the F1–Score. In this experiment, experiments were conducted from the number of clusters being 1 to 9. Considering the problem of the relatively large amount of data, we choose mini batch K-means as the clustering method.

Figure 4. The impact of cluster variations on F1–Score graph.

In Figure 4, the relationship between the number of clusters and the F1–Score variation is illustrated. When the number of clusters is set to 1 (i.e., the entire category forms a single cluster without clustering), the performance is relatively poor compared to cases where clustering is applied. Within a smaller range, the F1–Score improves as the number of clusters increases. However, when the number of clusters becomes excessive, it may lead to highly consistent features within clusters, potentially resulting in unstable quality in the generated data.

4.4. Quality Comparison of Data Generated by Sampling Algorithms

In the previous section, we assessed the impact of internal parameters on model performance. In this section, we compare ReMAHA with different sampling algorithms. At the data level, we compared three oversampling algorithms, namely, SMOTE [19],
ADASYN [42], and random oversampling, with the proposed ReMAHA. The parameters used in the sampling models during the experiments are as follows (Table 9):

**Table 9. Sampling Parameter Settings.**

<table>
<thead>
<tr>
<th>Models</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMOTE</td>
<td>sampling_strategy = 'auto', k_neighbors = 5</td>
</tr>
<tr>
<td>ReMAHA</td>
<td>k = 4</td>
</tr>
<tr>
<td>ADASYN</td>
<td>sampling_strategy = 'auto', n_neighbors = 5</td>
</tr>
<tr>
<td>ROS</td>
<td>sampling_strategy = 'auto'</td>
</tr>
</tbody>
</table>

We used three oversampling algorithms, SMOTE, ADASYN, and random oversampler, and compared them with the proposed ReMAHA method. The predictive models in all cases were CatBoost. The experimental results are shown in Table 10.

**Table 10. Performance of each sampling model (%).**

<table>
<thead>
<tr>
<th>Class</th>
<th>Metric</th>
<th>SMOTE</th>
<th>ADASYN</th>
<th>ROS</th>
<th>ReMAHA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity 1</td>
<td>Precision</td>
<td>52.91</td>
<td>43.05</td>
<td>29.21</td>
<td>71.41</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>85.12</td>
<td>81.96</td>
<td>97.46</td>
<td>75.60</td>
</tr>
<tr>
<td></td>
<td>F1-Score</td>
<td>65.26</td>
<td>56.45</td>
<td>44.95</td>
<td>73.44</td>
</tr>
<tr>
<td>Severity 2</td>
<td>Precision</td>
<td>93.26</td>
<td>91.65</td>
<td>90.81</td>
<td>91.51</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>93.82</td>
<td>91.34</td>
<td>93.54</td>
<td>94.57</td>
</tr>
<tr>
<td></td>
<td>F1-Score</td>
<td>93.54</td>
<td>91.49</td>
<td>92.15</td>
<td>93.02</td>
</tr>
<tr>
<td>Severity 3</td>
<td>Precision</td>
<td>78.01</td>
<td>69.02</td>
<td>73.93</td>
<td>73.43</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>69.96</td>
<td>61.52</td>
<td>57.90</td>
<td>64.63</td>
</tr>
<tr>
<td></td>
<td>F1-Score</td>
<td>73.76</td>
<td>65.06</td>
<td>64.94</td>
<td>68.75</td>
</tr>
<tr>
<td>Severity 4</td>
<td>Precision</td>
<td>40.28</td>
<td>33.47</td>
<td>73.05</td>
<td>76.31</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>53.90</td>
<td>53.46</td>
<td>33.56</td>
<td>53.23</td>
</tr>
<tr>
<td></td>
<td>F1-Score</td>
<td>46.11</td>
<td>41.17</td>
<td>45.99</td>
<td>62.71</td>
</tr>
</tbody>
</table>

Figure 5 shows that the addition of oversampling improves the predictive performance, with a significant increase in the number of predictions for minority class samples. This is mainly reflected in the improvement of recall, where higher recall indicates better identification of minority class samples. However, as recall increases, some precision values decrease. This is because with the increase in minority class samples predicted, the precision of predictions may decrease. As shown in Table 10, ReMAHA achieves a precision of 71.41 and 76.31, recall of 75.60 and 53.23, and an F1–Score of 73.44 and 62.71 for labels 1 and 4, respectively, with oversampling performance relatively superior to other models.
Figure 5. Predictive performance of different oversampling models.

Figure 6a illustrates the ranking of feature importance in the original data. Figure 7 shows the effect of the oversampling algorithms ReMAHA, SMOTE, ADASYN, and ROS on the correlation of the data, respectively. It is noticeable that, compared to the original data, ReMAHA and ROS demonstrate the closest effects. However, ROS directly duplicates existing data, leading to a sensitive identification of certain classes during prediction, while yielding poor identification for other classes.

Figure 6. Performance of the original data: (a) feature correlation; and (b) confusion matrix.
Figure 7. Feature correlation: (a) ReMAHA; (b) SMOTE; (c) ADASYN; and (d) ROS.

Figure 6b demonstrates the confusion matrix before sampling, while Figure 8 presents the confusion matrices for various oversampling algorithms. Manifested in the confusion matrices, SMOTE, ROS, and ADASYN all exhibited varying degrees of overfitting, primarily by misclassifying label 2 and label 3 as label 1 or label 4. This occurrence is primarily attributed to insufficient data quality in the generated samples. For instance, the process of generating new data might not account for correlations or result in excessive overlap among generated data, rendering correct categorization challenging.

4.5. Predictive Model Comparison Experiment

To evaluate the effectiveness of ReMAHA–CatBoost, we conducted experiments involving eight different models (Table 11). These eight models were divided into two groups for comparison: prediction model comparison experiments, and prediction + sampling (hybrid) model comparison experiments.
At the algorithm level, to ensure the accuracy of the experiments, we chose boosting algorithms, including AdaBoost [43], GBDT [44], XGBoost [45], LightGBM [46], and CatBoost.

Table 11. Predictive Model Parameter Settings.

<table>
<thead>
<tr>
<th>Models</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>CatBoost</td>
<td>iterations = 500, learning_rate = 0.03, depth = 6</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>n_estimators = 50, learning_rate = 1.0</td>
</tr>
<tr>
<td>GBDT</td>
<td>learning_rate = 0.1, n_estimators = 100, max_depth = 3</td>
</tr>
<tr>
<td>XGBoost</td>
<td>learning_rate = 0.3, max_depth = 3</td>
</tr>
<tr>
<td>LightGBM</td>
<td>num_leaves = 31, learning_rate = 0.1, max_depth = -1, n_estimators = 100</td>
</tr>
</tbody>
</table>

To ensure fairness in the experiments, default parameters were used for all models. First, we compared the prediction models without any sampling. US–Accidents Table 12 shows the results of the five algorithms on the unsampled U.S. accident dataset. It can be observed that AdaBoost performs significantly worse on the minority class label 1 compared to the other models. While it achieves a high recall for the minority class label 4, the low F1–Score indicates that AdaBoost identifies more instances of label 4 but with poor precision. This is because adaptive models focus more on misclassified samples in each iteration, and with the highly imbalanced dataset, the adaptive model tends to favor the majority class during prediction.

Table 12. Performance of each prediction model (%).

<table>
<thead>
<tr>
<th>Class</th>
<th>Metric</th>
<th>CatBoost</th>
<th>AdaBoost</th>
<th>GBDT</th>
<th>XGBoost</th>
<th>LightGBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity 1</td>
<td>Precision</td>
<td>71.82</td>
<td>53.71</td>
<td>64.99</td>
<td>72.70</td>
<td>69.25</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>73.27</td>
<td>13.09</td>
<td>44.52</td>
<td>65.37</td>
<td>61.55</td>
</tr>
<tr>
<td></td>
<td>F1–Score</td>
<td>72.53</td>
<td>21.05</td>
<td>52.84</td>
<td>68.84</td>
<td>65.17</td>
</tr>
<tr>
<td>Severity 2</td>
<td>Precision</td>
<td>91.02</td>
<td>84.04</td>
<td>86.52</td>
<td>90.63</td>
<td>89.64</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>94.96</td>
<td>95.34</td>
<td>95.78</td>
<td>94.91</td>
<td>94.93</td>
</tr>
<tr>
<td></td>
<td>F1–Score</td>
<td>92.95</td>
<td>89.33</td>
<td>90.91</td>
<td>92.73</td>
<td>92.21</td>
</tr>
<tr>
<td>Severity 3</td>
<td>Precision</td>
<td>73.92</td>
<td>58.32</td>
<td>68.94</td>
<td>72.70</td>
<td>71.42</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>62.82</td>
<td>29.19</td>
<td>41.91</td>
<td>61.27</td>
<td>56.58</td>
</tr>
<tr>
<td></td>
<td>F1–Score</td>
<td>67.92</td>
<td>38.90</td>
<td>52.13</td>
<td>66.50</td>
<td>63.14</td>
</tr>
<tr>
<td>Severity 4</td>
<td>Precision</td>
<td>73.97</td>
<td>77.58</td>
<td>80.56</td>
<td>77.17</td>
<td>75.63</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>44.35</td>
<td>37.46</td>
<td>31.12</td>
<td>44.08</td>
<td>41.89</td>
</tr>
<tr>
<td></td>
<td>F1–Score</td>
<td>55.46</td>
<td>50.53</td>
<td>44.89</td>
<td>56.11</td>
<td>53.91</td>
</tr>
</tbody>
</table>

CatBoost, XGBoost, and LightGBM, as emerging variants of gradient-boosting algorithms, have similar performance. CatBoost achieves an F1–Score of 72.53 for class label 1 on this dataset, slightly higher than the other two. When considering both minority class labels, CatBoost performs better overall. CatBoost’s superior performance in identifying minority classes in this experiment is mainly attributed to its built-in symmetric trees used as split nodes, making it more suitable for imbalanced datasets.

Table 13 displays the performance of hybrid models. We combined ReMAHA with five predictive algorithms to assess the impact of sampling models on predictive models.

Table 13. Performance of each hybrid model (%).

<table>
<thead>
<tr>
<th>Class</th>
<th>Metric</th>
<th>CatBoost</th>
<th>AdaBoost</th>
<th>GBDT</th>
<th>XGBoost</th>
<th>LightGBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity 1</td>
<td>Precision</td>
<td>71.41</td>
<td>53.96</td>
<td>65.26</td>
<td>74.40</td>
<td>69.25</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>75.60</td>
<td>13.65</td>
<td>44.82</td>
<td>69.02</td>
<td>61.19</td>
</tr>
<tr>
<td></td>
<td>F1–Score</td>
<td>73.44</td>
<td>21.78</td>
<td>53.15</td>
<td>71.61</td>
<td>65.15</td>
</tr>
</tbody>
</table>
Severity 2  
Precision 91.51 84.55 87.06 91.66 89.69 
Recall 94.57 95.38 95.53 94.88 95.07 
F1-Score 93.02 89.64 91.10 93.24 92.30

Severity 3  
Precision 73.43 60.42 68.92 74.29 71.92 
Recall 64.63 31.49 44.23 66.55 55.96 
F1-Score 68.75 41.40 53.88 70.20 62.94

Severity 4  
Precision 76.31 80.68 84.32 78.17 78.93 
Recall 53.23 44.31 39.95 44.64 51.17 
F1-Score 62.71 57.20 54.21 56.83 62.08

As shown in Figure 9, the experimental results show that all model predictions using ReMAHA oversampling mixing are better than the unmixed model. With the introduction of ReMAHA oversampling, the models have shown notable improvements in F1-Score, ranging from 0.3% to 9.32%, especially in the severity levels 1 and 4. While the addition of oversampling increases the number of predictions for minority class samples, it can lead to a decrease in precision if misclassified. However, ReMAHA has demonstrated improvements in F1-Score, precision, and recall for minority class categories 1 and 4. From the perspective of algorithms and sampling, the combined ReMAHA and CatBoost show the most stable predictive performance. Particularly, the F1-Score and recall for minority classes 1 and 4 are the highest, indicating that this hybrid model accurately identifies minority class samples.

![Figure 9](image-url)

**Figure 9.** Performance on different models: (a) performance on the prediction model; and (b) performance on the hybrid model.
To highlight the contrast before and after sampling, we computed the differences between the models using various prediction algorithms (CatBoost, AdaBoost, GBDT, XGBoost, and LightGBM) after employing the ReMAHA sampling and those used before, as depicted in Figure 10. Although after processing with the ReMAHA oversampling algorithm, the weights of minority class samples have increased, potentially leading to a decrease in precision while enhancing recall. Overall, upon incorporating ReMAHA oversampling, the majority of performance metrics for prediction models have significantly improved, especially for the label 4.

Figure 10. Comparison of effects before and after sampling.

5. Discussion

ReMAHA–CatBoost’s comparisons at different levels confirm its excellent performance in oversampling and prediction. These experimental results indicate that ReMAHA–CatBoost is effective in recognizing different severity levels of traffic accidents. At the same time, we did not consume too much extra computer resources in the process of generating data, in the ReMAHA oversampling algorithm to generate new data, consumed 2594.3 MB of computer memory, less than the consumption of SMOTE, ROS, and ADASYN algorithms.

It is worth noting that the addition of oversampling algorithms can increase the number of predictions for minority class samples. However, with increased data generation, the risks also grow. If the generated data are too concentrated, they will not yield significant performance improvement, while overly dispersed generated data can blur the boundaries between different categories. ReMAHA, as proposed in this paper, balance the generated data well through the process of clustering and data generation.

Due to CatBoost’s improvements in handling prediction shift and conditional shift, it reduces overfitting. Combined with the characteristics of symmetric trees, it performs exceptionally well on datasets like traffic accidents, which are characterized by sparse features and extreme class imbalance. Recent research has already demonstrated the effectiveness of hybrid approaches combining sampling and prediction models [47]. Long [48] proposed a hybrid sampling model combined with the GNN model, validating the effectiveness of hybrid samplers on public datasets. Wang et al. [49] introduced the MatFind model, which used the K-nearest neighbors algorithm to balance the dataset and employed SVM for prediction, leading to improved performance in miRNA identification. These successful cases suggest that hybrid models combining sampling and prediction are feasible when dealing with imbalanced and large datasets.

However, based on existing work, it is clear that the problems caused by data imbalance in machine learning tasks are far from being completely resolved. Additionally, data imbalance is a common phenomenon in real-life datasets, and the current achievements are not enough to fully trust the models. Particularly, the dataset used in this study, the
US–Accidents traffic accident dataset, has a class data difference of 91.40 times, such a massive difference leads to many oversampling algorithms and prediction models not achieving ideal results. Therefore, future work will continue to focus on improving the credibility of models and enhancing their generalization capabilities. Future work will primarily focus on the following three aspects: exploring correlations among data within the same category from multiple dimensions to generate more representative data; leveraging additional traffic accident data to enhance the model’s generalization ability; and further refining feature engineering to improve the model’s computational efficiency.

6. Conclusions

In this study, we proposed the ReMAHA–CatBoost model tailored for the imbalanced domain of traffic accident prediction. Our key design comprised three parts: utilizing the relief–F algorithm for feature selection and acquiring feature weight matrices in the feature selection step; integrating the feature weight matrices into the distance calculation process of sample points to enhance the reliability of oversampled data; and, finally, training the CatBoost model using the oversampled dataset.

These designs ensured the model’s capability to identify different features’ impact on accident severity while enhancing the recognition of minority class samples. Experimental results demonstrated the effectiveness of ReMAHA–CatBoost in predicting accident severity, especially its superior performance in identifying minority class samples compared to other oversampling algorithms (SMOTE, ADASYN, and ROS). ReMAHA–CatBoost enables us to accurately classify potential traffic accidents, assisting traffic management authorities in efficiently allocating limited resources for potential severe accidents. Moreover, the proposed model aids relevant authorities in taking necessary preventive measures to mitigate the economic and healthcare burdens caused by traffic accidents.

7. Patents

Patent application number: CN202211527542.5.

Author Contributions: Conceptualization, G.L. and Y.W.; methodology, G.L.; validation, G.L., Y.W., and Y.B.; formal analysis, W.Z.; investigation, G.L.; data curation, Y.B.; writing—original draft preparation, G.L.; writing—review and editing, W.Z.; visualization, Y.B.; supervision, Y.W.; project administration, Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Sichuan University of Science and Engineering Graduate Student Innovation Fund (y2022179), the Sichuan Provincial Science and Technology Department Project (2023YFG0307) and the Sichuan Province Intelligent Tourism Research Base Project (ZHYJ22-02).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This study used the US–Accidents dataset, which is publicly available datasets. They can be found at US Accidents (2016–2023) (kaggle.com) (accessed on 31 October 2023).

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

References


