mathematics

MDPI

Article

GA-CatBoost-Weight Algorithm for Predicting Casualties in Terrorist Attacks: Addressing Data Imbalance and Enhancing Performance

Yuxiang He, Baisong Yang and Chiawei Chu *

Faculty of Data Science, City University of Macau, Macau 999078, China; d21092100165@cityu.edu.mo (Y.H.); d22092100266@cityu.edu.mo (B.Y.)
* Correspondence: cwchu@cityu.edu.mo

Abstract: Terrorism poses a significant threat to international peace and stability. The ability to predict potential casualties resulting from terrorist attacks, based on specific attack characteristics, is vital for protecting the safety of innocent civilians. However, conventional data sampling methods struggle to effectively address the challenge of data imbalance in textual features. To tackle this issue, we introduce a novel algorithm, GA-CatBoost-Weight, designed for predicting whether terrorist attacks will lead to casualties among innocent civilians. Our approach begins with feature selection using the RF-RFE method, followed by leveraging the CatBoost algorithm to handle diverse modal features comprehensively and to mitigate data imbalance. Additionally, we employ Genetic Algorithm (GA) to finetune hyperparameters. Experimental validation has demonstrated the superior performance of our method, achieving a sensitivity of 92.68% and an F1 score of 90.99% with fewer iterations. To the best of our knowledge, our study is the pioneering research that applies CatBoost to address the prediction of terrorist attack outcomes.

Keywords: terrorist attack prediction; feature selection; CatBoost; sample imbalance

MSC: 68T09

Citation: He, Y.; Yang, B.; Chu, C. GA-CatBoost-Weight Algorithm for Predicting Casualties in Terrorist Attacks: Addressing Data Imbalance and Enhancing Performance. *Mathematics* **2024**, *12*, 818. https://doi.org/10.3390/math12060818

Academic Editor: Florin Leon

Received: 25 February 2024
Revised: 4 March 2024
Accepted: 7 March 2024
Published: 11 March 2024

1. Introduction

Terrorism, driven by motives such as political, economic, religious, or social aims, utilizes violence and illicit methods to intimidate, coerce, and instill fear [1] and remains a grave menace to international peace and security. Since the "9/11" attacks, efforts in counter-terrorism have been significantly intensified in the European and American regions to combat the development of terrorist forces. According to data from the Global Terrorism Database, there have been over 200,000 recorded terrorist attacks between 2010 and 2020, resulting in numerous casualties and property damage. Despite a decreasing trend in terrorist attacks since 2015, the future outlook on the risk of terrorist attacks remains concerning. The outbreak of COVID-19 at the end of 2019 not only severely disrupted people's lives, health, and travel routines but also dealt a heavy blow to market economies, nurturing negative emotions among the general populace and further fueling the unfavorable trends of terrorism [2]. In early 2022, the outbreak of the Russo–Ukrainian War significantly affected the development and security of neighboring countries and regions. Under the conflicts between nations and ethnicities, ordinary people are bound to harbor feelings of hatred, exacerbating the spread and escalation of terrorism [3]. With the ongoing escalation of the Israeli–Palestinian conflict, many countries are facing internal divisions, intensified opposition sentiments, posing threats to social stability and facing significant risks of terrorist attacks [4]. While many studies indicate that the purpose of terrorism is to advance specific political objectives, the act of terrorism itself not only spreads social panic but also results in infrastructure damage and the loss of innocent lives.

Due to the complexity of counter-terrorism measures and emergency controls, it is crucial to develop effective methods to predict the consequences of terrorist attacks.

Machine learning, as a robust computational tool, plays a pivotal role in decision-making processes for analyzing casualties in terrorist attacks. Research indicates that the incorporation of textual feature information derived from terrorism datasets can significantly improve the performance of models [5]. In previous studies, textual features in terrorism datasets were typically handled separately using text vectorization techniques and then combined with other types of features for analysis of terrorist attacks through machine learning algorithms. Hence, conventional data sampling techniques face challenges in addressing data imbalance within the aforementioned methods. However, no research has proposed a method that considers data imbalance while handling textual features of terrorist attacks. To fill this gap, we present a CatBoost-based model for predicting casualties in terrorist attacks, which can forecast whether terrorist attacks pose a threat to the safety of innocent civilians. The results of this study not only assist decision-makers in adjusting and deploying appropriate emergency measures but also provide valuable information support.

In the proposed algorithm, to eliminate feature redundancy and reduce computational complexity, we conducted feature selection by combining Random Forest (RF) and Recursive Feature Elimination (RFE) methods. Leveraging the advantages of CatBoost in handling numerical, categorical, and textual features simultaneously, we trained the model using CatBoost algorithm by combining textual features with the selected features obtained through screening to improve data imbalance issues. Additionally, hyperparameter tuning was performed using a genetic algorithm. Building on the GTD dataset, we compared the performance of our method with several state-of-the-art methods such as LR, DT, RF, Adaboost, XGBoost, and LightGBM. Experimental results demonstrate the superiority of our method over the aforementioned methods. Therefore, one can conclude that our approach is an efficient and effective algorithm for analyzing the risk of casualties in terrorist attacks.

The organization of the remaining sections of this paper is as follows: Section 2, we briefly review relevant literature. Section 3, we introduce the entire experimental process based on the CatBoost method proposed by us. Section 4, we conduct extensive numerical experiments to evaluate the performance of the proposed method. Finally, Section 5, we summarize the entire paper and analyze the advantages of our method.

2. Literature Review

Early statistical methods have been an efficient approach for analyzing terrorist attacks, with many studies conducting assessments of the consequences of terrorist attacks in real scenarios. Guo et al. constructed a risk assessment method for terrorist attacks on civil aviation airports using event trees and probabilistic risk assessment models, which evaluate the risk of various types of terrorist attacks on civil aviation airports [6]. Yang et al. utilized the FAHP-SWOT method to design a risk assessment model, analyzing various risk factors for terrorist attacks on religious sites [7]. The evaluation-based methods mentioned above allow for quantifiable analyses of specific scenarios but still suffer from high subjectivity.

With the increase in data volume and diversity of feature types, machine learning has provided algorithmic support for the analysis of consequences of terrorist attacks. Lanjun et al. aimed to select the most crucial indicators affecting the risk of terrorist attacks from various perspectives. They selected 28 indicators from several data sources and used the Random Forest algorithm to compute feature importance rankings, followed by a recursive process to obtain the most impactful set of features influencing the risk of terrorist attacks [8]. Zhang et al. considered 17 influencing factors in terrorist attack data and assessed the risk level of geographical regions in Southeast Asia using an improved location recommendation algorithm [9]. Feng et al. proposed a novel RP-GA-XGBoost algorithm to predict whether terrorist attacks would result in casualties [10].

While the aforementioned methods are single-model approaches, the complexity and quantity of data have led to the increasing advantages of hybrid models in analyzing ter-

rorist attack issues. Shafiq et al. introduced a hybrid classifier to predict the type of attack in terrorist events, incorporating K-nearest neighbors, Naive Bayes, and decision trees [11]. Meng et al. proposed a hybrid classifier framework for terrorist attack prediction, comprising SVM, K-nearest neighbors (KNN), Bagging, and C4.5, and optimized the weights of individual classifiers using a genetic algorithm to enhance prediction accuracy [12].

Most current research on terrorist attacks focuses primarily on numerical and categorical features, with relatively limited studies exploring the inclusion of textual features to enhance model performance. Mohammed Abdalsalamde et al.'s study demonstrated that by using text representation techniques to process textual features and combining them with other types of features, the performance of predictive models for terrorist attack types can be improved [5]. However, they did not perform data cleaning on the text to remove redundant information and utilizing text representation techniques added processing steps and extra computational overhead to the model. Therefore, in future research, it may be beneficial to further investigate how to effectively handle textual features, optimize model performance, and reduce unnecessary computational costs.

If different terrorist attacks with varying levels of risk are treated equally in the context of terrorist attacks, it may lead decision-makers to make misjudgments and result in the waste of resources. Therefore, the issue of data imbalance is a crucial factor influencing terrorist attack models. Varun Teja Gundabathula et al. proposed using machine learning models to predict terrorist groups based on historical data and employing data sampling techniques to improve the accuracy of classification models [13]. Fahad Ali Khan and colleagues utilized the Particle Swarm Optimization (PSO) algorithm to determine the optimal weight distribution for Random Forest and Extreme Gradient Boosting Machine in accurately predicting whether terrorist activities would result in casualties. To address the issue of class imbalance, they applied the Synthetic Minority Oversampling Technique (SMOTE) to handle imbalanced data [14]. The aforementioned studies alleviate data imbalance issues through data sampling techniques. However, due to the structural nature of text data, these data sampling techniques are not suitable for directly processing textual information. Although text data can be first vectorized before data sampling, the conversion may result in the loss of some semantic information. Additionally, data sampling techniques are difficult to generate text information based on context. Currently, there is no research considering data sampling for terrorist attack texts, and text vectorization techniques independent of predictive models do not enhance model performance as effectively as end-to-end models.

Comparing with the above-mentioned studies, the main contributions of this article are as follows:

1. First, we propose an innocent civilian casualties prediction model named GA-CatBoost-Weight for terrorist attacks based on CatBoost. CatBoost is capable of directly handling numerical, categorical, and textual features, and its powerful computational capability has been widely applied in various fields. However, to our knowledge, there has been no research applying CatBoost to the issue of terrorist attacks. Therefore, we use the CatBoost algorithm combined with some strategies to enhance algorithm performance to predict whether terrorist attacks will result in casualties;
2. Secondly, we employ RF-RFE for feature selection. High-dimensional features not only increase the computational cost of models but also affect model performance. In this paper, we combine RF and RFE to reduce feature redundancy and effectively decrease computational costs. This method obtains importance scores for each feature using RF, reduces feature numbers based on feature importance ranking to generate a model performance curve, and obtains the optimal feature subset based on the trend of the curve;
3. Thirdly, we conduct hyperparameter tuning for CatBoost. In the case of data imbalance where traditional data sampling techniques struggle to handle textual information, we propose using CatBoost's built-in parameters to improve the data imbalance issue in terrorist attack scenarios. Instead of additional processing at the data level,

we address the data imbalance issue from the model perspective in an end-to-end manner to prevent the loss of excessive semantic information in textual features. GA is an excellent hyperparameter optimization algorithm that is not commonly combined with CatBoost for tuning. Hence, in this study, we choose genetic algorithm to effectively enhance the performance of our innocent civilian casualties prediction model for terrorist attacks.

3. Materials and Methods

In this paper, we have developed a casualty prediction algorithm for terrorist attacks based on CatBoost to predict whether terrorist attacks will result in casualties. The overall framework of the proposed method consists of four stages, as shown in Figure 1. Firstly, we preprocess the missing values, features, and labels in the dataset. Secondly, we conduct feature selection using the RF-RFE method. Subsequently, we employ GA to optimize the hyperparameters of CatBoost. Finally, we evaluate the trained model on the test set using evaluation metrics. The following sections provide detailed information on the primary steps.

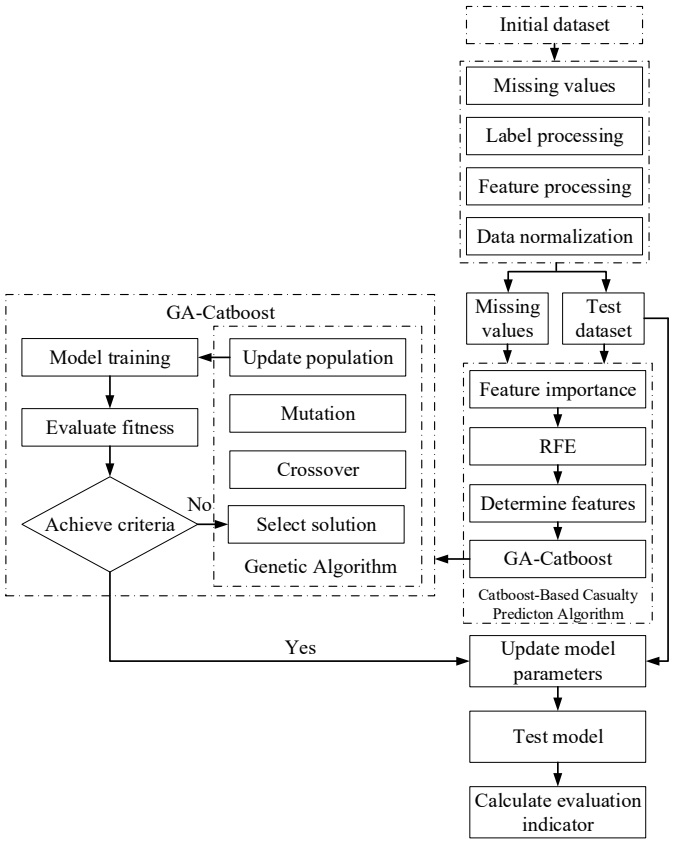

Figure 1. A flowchart of the proposed method.

3.1. Data Preprocessing

The data used in this study are from the Global Terrorism Database (GTD) maintained by the National Consortium for the Study of Terrorism and Responses to Terrorism (START) (https://www.start.umd.edu/gtd) (accessed on 28 August 2022). The database collects information on global terrorist events from 1970 to 2020, including data from sources such as news, books, and legal documents. It consists of 135 features and over 200,000 samples.

To ensure the quality and reliability of the data, we first removed features with missing values exceeding 70%. We then further eliminated irrelevant features for predictions such as event number, event summary, event source, and similar. We also directly eliminated event records with outliers. The data was then normalized using the Min-Max scaling

method to scale the data to the [0, 1] interval [15]. The transformation formula is defined as follows:

$$x'(k) = \frac{x(k) - \min x(k)}{\max x(k) - \min x(k)} \quad (1)$$

After preprocessing, we have obtained 98,508 samples and 24 features. For the selected event records, we use "nkill" to represent the number of fatalities in the event and "nwound" to represent the number of wounded individuals in the event. An attack event is considered to have resulted in casualties only when the sum of fatalities and injuries is greater than 0. Since the focus is on civilian casualties in each attack event, casualties caused by terrorists themselves, such as in suicide attacks or due to being killed, are excluded from the event. We define "risk" as a binary label related to the casualty situation. Among the 98,508 samples, there are 33,499 samples where no casualties occurred and 65,009 samples where casualties occurred.

3.2. Feature Processing

The Global Terrorism Database (GTD) contains numerical, categorical, and text features. In order to fully leverage the performance of the model, feature preprocessing is necessary.

While some studies on terrorist attacks have shown that time variable features have less importance in the analysis of terrorist attacks, it is important to consider that the consequences of terrorist attacks may not solely depend on a single time variable but may be related to specific dates within a week. Therefore, for the specific time variable features "iyear", "imonth", and "iday" that represent the occurrence of events, these three individual time variable features are transformed into a new time variable feature "date" representing the day of the week, defined within the interval [1,7]. This can provide additional context related to the day of the week.

The text feature "summary" provides a brief description of terrorist events, including specific details such as the time, location, individuals involved, and the method of the attack. Current studies on terrorist attacks that consider text features typically extract information from text through text vectorization techniques. With the advantage of CatBoost's text processing technology, it can directly compute text features without the need for additional natural language processing (NLP) steps. However, in this experiment, the "summary" feature contains information about the date of the attack and the consequences of the attack, which conflicts with the prediction target of this study. Therefore, it needs to be removed.

To avoid introducing high subjectivity during feature selection, common feature selection methods include filter, embedded, and wrapper methods [16]. However, many studies on terrorist attacks often subjectively determine the number of features [12,17]. In order to effectively eliminate redundant features, reduce computational costs, and manage feature dimensions, this study utilizes a method combining RF with RFE to determine a subset of features.

Random Forest can be considered as an ensemble learning model consisting of multiple decision trees, widely used for classification and regression analysis. Due to its inherent randomness, Random Forest retains some sample data through random sampling with replacement during training, known as Out of Bag (OOB) data. These OOB samples are often used to measure the importance of features [18]. In essence, Random Forest calculates the corresponding Out of Bag score (OOB score) through the OOB data. If adding noise to a particular feature significantly decreases the OOB score, it indicates that this feature is important for influencing the outcome.

In this experiment, the importance of features is evaluated based on the OOB score of Random Forest, which helps in ranking the features based on their importance. The OOB scores are used to assess the importance of features, as shown in Figure 2, where the horizontal axis represents the score of each feature and the vertical axis represents the feature names.

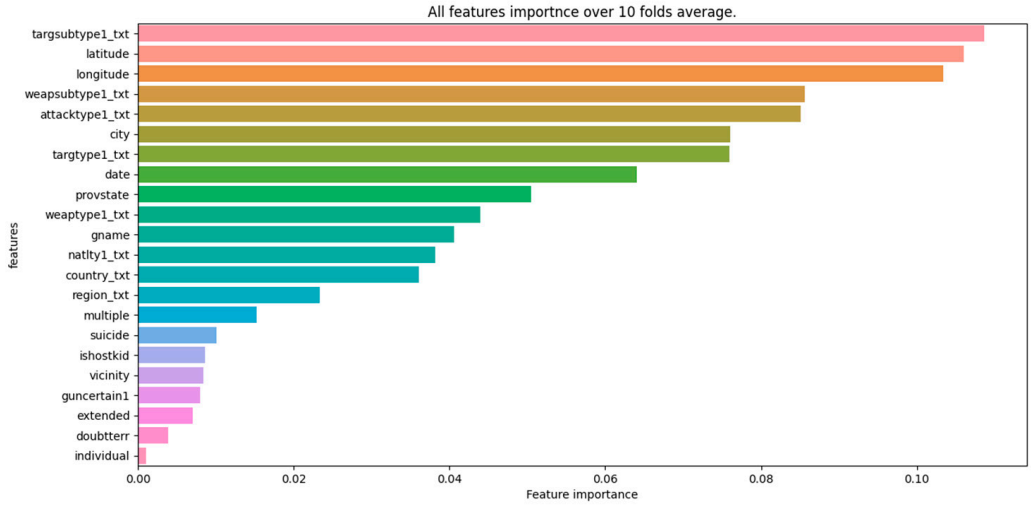

Figure 2. Feature importance ranking.

After obtaining the importance ranking of the original features, in order to objectively determine the number of features, we utilize RFE to analyze the performance of RF with different subsets of features. RFE works by iteratively constructing different subsets of features based on their importance, starting with the most important features determined earlier. This process involves incrementally adding individual features in order of importance to form various feature subsets. Using RF, the performance of the model is validated with different feature subsets, and the optimal feature subset is identified based on the trend of performance changes.

Figure 3 illustrates the model performance curve based on RF, where the horizontal axis represents the number of features, and the vertical axis represents the model evaluation metric. By analyzing this curve, we can determine the optimal number of features that maximize the model's performance. Based on Figure 3, we observe that by continuously adding features based on their importance ranking, the ROC, accuracy, F1, and sensitivity model evaluation curves continue to increase until reaching a peak when the feature subset contains eight features. After that, the model performance starts to decline and reaches a balance. While there is some incremental improvement afterwards, it is not significant. The precision model evaluation curve reaches its peak when the number of features reaches seven.

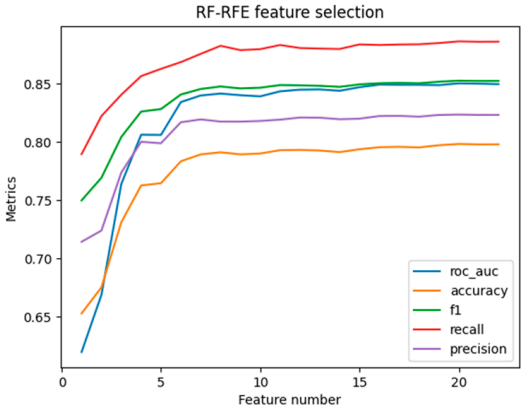

Figure 3. RF performance curve.

Therefore, we have selected the top 8 features according to their importance ranking as the feature subset for our method: 'city', 'latitude', 'longitude', 'attacktype1_txt', 'targtype1_txt', 'targsubtype1_txt', 'weapsubtype1_txt', 'date'. At this point, the model not only exhibits high reliability but also significantly reduces computational cost. It is worth

noting that our custom time variable feature "date" is also included in the selected feature subset, validating the rationality of the features we constructed.

3.3. Hyperparameter Tuning Method Based on CatBoost

The CatBoost algorithm is a new gradient boosting algorithm that improves model performance by having each decision tree learn from the previous tree and influence the next tree. In traditional Gradient Boosting Decision Tree (GBDT) algorithms, the ensemble of weak classifiers for each round is used as the final result. Representing a decision tree as $T(\cdot)$, the model can be expressed as follows:

$$f_D(x) = \sum_{d=1}^{D} T(x; \Theta_d) \tag{2}$$

Here, D represents the number of decision trees, and Θ_d denotes the parameters of each decision tree. Each decision tree continuously minimizes the empirical risk of the parameters Θ_d by training on the residuals from the previous round's decision tree to obtain:

$$\Theta_d = \underbrace{\operatorname{argmin}}_{\Theta_d} \sum_{s=1}^{S} L(y_i, f_{d-1}(x_i) + T(x; \Theta_d)) \tag{3}$$

Here, s represents the number of parameters x_i, and y_i represents the target value that needs to be fitted. During the process of estimating residuals, GBDT mainly uses the negative gradient of the loss function to iteratively fit each decision tree in every round. The overall basic workflow of the algorithm is as follows:

1. Initialize the first decision tree:

$$f_0(x) = \operatorname{argmin} \sum_{s=1}^{S} L(y_i, \mathrm{c}) \tag{4}$$

where $L(y_i, \mathrm{c})$ is the loss function, and c is the initialized constant value.

2. For each iteration d = 1, 2, . . ., D:

   (a) Compute the negative gradient of the loss function to fit the residual values in the current iteration of the model:

$$r_{di} = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{d-1}(x)} \tag{5}$$

   (b) In the leaf node region of the current iteration $R_{mj}, j = 1, 2, \ldots, J$ of the model, fit a decision tree for r_{di} (using CART regression tree as an example) using (x_i, y_{it}). Here, t represents the index of y_i. Calculate the optimal value within the leaf node region:

$$C_{dj} = \operatorname{argmin} \sum_{x_i \epsilon R_{mj}} L(y_i, f_{m-1}(x_i) + \mathrm{c}) \tag{6}$$

   (c) Update the model:

$$f_m(x_i) = f_{m-1}(x_i) + \sum_{j=1}^{j} c_{dj} I(x_i \in R_{mj}) \tag{7}$$

3. Output the final strong learner $f_M(x)$.

CatBoost has made several improvements over traditional GBDT, with key characteristics in the following areas:

1. Feature handling: CatBoost introduces the Ordered Target Statistic method to handle categorical features. This method sorts each category feature value based on its relationship with the target variable and performs corresponding statistical calculations. This technique can be used to encode category features, helping the model better understand the meaning of category features. The formula is as follows:

$$x_{\sigma_p,z} = \frac{\sum_{n=1}^{p-1} \left[x_{\sigma_n,z} = x_{\sigma_p,z}\right] Y_{\sigma_r} + ap}{\sum_{n=1}^{p-1} \left[x_{\sigma_n,z} = x_{p,z}\right] + a} \tag{8}$$

Here, $\left(x_{\sigma_p,z}, Y_{\sigma_p}\right)$ represents the sample representation of example σ_p in the sample sequence σ. p is the prior; a is the weight; z is the category to which the sample belongs.

For text features, CatBoost first maps text features to a fixed-length feature vector through feature hashing, compressing different text features into vectors of the same length. Subsequently, CatBoost processes the feature vectors obtained from feature hashing by combining them. This combination enables the model to capture relationships and interactions between features more effectively. This processing method can effectively process text features, making CatBoost more convenient and efficient in handling text data. In addition, CatBoost provides various additional methods to handle text features, including converting text features to numerical features using category encoding, representing text features as bag-of-words models such as word frequency or TF-IDF, transforming text features into vector representations of fixed length as text embeddings, and extracting n-gram features based on text features for feature derivation. Depending on the specific problem and characteristics of the data, appropriate methods can be chosen for feature processing and model training. In this study, we use the default handling method.

During the second split of trees, CatBoost combines features within the tree to enrich the feature dimensions of the model, thereby further learning the nonlinear relationships between features.

2. Addressing Gradient Bias: Traditional GBDT methods estimate gradients using the same dataset for model training, which can lead to cumulative bias and overfitting due to incomplete consistency in data distribution. To address this issue, CatBoost introduces the Ordered Boosting method. The approach involves first shuffling the sample data. For each sequence σ, t models $M_1, \ldots, M_t$ are trained, where t represents the number of samples. Each model $M_q(q = 1, 2, \ldots, t)$ is trained using data preceding the current sample sequence.
3. Symmetric Trees. Compared to conventional decision trees, CatBoost uses a lower-degree symmetric tree structure, which has the following characteristics:
    (a) Symmetric Splitting: In contrast to traditional decision tree algorithms that split nodes based on a single optimal feature dimension, the symmetric tree in CatBoost splits nodes based on two feature dimensions simultaneously. This symmetric splitting allows for more effective utilization of relationships and interactions between features, enhancing the model's training efficiency and generalization capability;
    (b) Feature Interaction: In a symmetric tree, decision tree nodes at the same level consider the mutual influence of multiple features simultaneously. This feature interaction helps the model capture feature interactions better, enhancing accuracy and robustness.

These features of the symmetric tree enhance CatBoost's understanding of feature interactions, thereby improving the model's generalization capability and reducing overfitting to noise and irrelevant features in training data. Additionally, the classification method of the symmetric tree during predictions eliminates the need for traversal from the root node; instead, it can be implemented simply through array indexing, reducing computation during prediction and improving prediction speed.

CatBoost is composed of multiple hyperparameters, each controlling different functions, which makes hyperparameter optimization extremely complex. Firstly, there are interactions and influences between hyperparameters; changing one hyperparameter may affect the optimal values of other hyperparameters, making the optimization process complex and challenging. Secondly, a larger hyperparameter search space requires trying more different combinations of hyperparameter values, leading to high computational costs. Additionally, the large number and wide range of hyperparameters create a high-dimensional search space that demands significant time and computational resources. Lastly, excessive or frequent searching may result in overfitting on the training set, compromising the model's generalization ability. Therefore, hyperparameter optimization is seen as a complex and challenging task.

Genetic algorithms simulate the theory of survival of the fittest, allowing the initial population to evolve towards better solutions and eventually converge to the most suitable individual for the environment [19]. In genetic algorithms, chromosomes are typically represented as binary strings in the solution space, with a fitness function indicating the quality of individuals or solutions. Genetic operators typically include selection, mutation, and crossover. The selection operator involves selecting good individuals from the old population with a certain probability to form a new population. The mutation operator helps prevent the algorithm from getting stuck in local optimal solutions during optimization. The crossover operator randomly selects two individuals' chromosomes for exchange and recombination to create new individuals. Genetic algorithms have good robustness and simplicity, which can make CatBoost more stable and efficient. In this study, the Uniform Crossover genetic operator is used to enhance the algorithm's search capability, while model performance evaluation metrics are used as the fitness function.

The pseudo of GA-CatBoost is shown in Algorithm 1.

Algorithm 1. GA-CatBoost hyperparameters tuning mechanism.

Input: cross-validation fold K, mutation type MT, fitness function Func, crossover type CT, total iterations I, Dataset D, crossover probability C, mutation probability M
Output: The optimal hyperparameter values for CatBoost
1. Initialize i to 0
2. Initialize population randomly
3. Execute the following loop until i < I:
4. For each solution in the population do
5. Extract hyperparameters for CatBoost from solution
6. Split D into K parts, one part as testing set and the rest as training set
7. For each fold from 1 to K do
8. Training CatBoost on the training set
9. Predict values using CatBoost on the testing set
10. Calculate fitness value based on Func
11. End for
12. Compare and select the optimal model performance parameters
13. End for
14. Select solutions using roulette wheel selection
15. Apply crossover on the selected solutions with CT and C
16. Mutate of the new solutions with MT and M
17. Generate the new population
18. End loop
19. Return the optimal hyperparameter values of CatBoost

Due to the imbalance in the number of samples between the category of personnel deaths and other categories in the dataset, there is a problem of data imbalance. This imbalance may lead to classification models that overly rely on limited data samples, resulting in overfitting and reduced accuracy and robustness of the model. In the classification of casualties in terrorist attacks, misclassification of major casualty events can lead to unreasonable resource allocation and decision-making errors, resulting in significant costs. Since

the dataset involves textual features, traditional sampling methods at the data level may not be suitable for sampling textual information.

Therefore, our method addresses the issue of data imbalance by leveraging the built-in parameter of CatBoost. We need to optimize key hyperparameters in CatBoost, including learning rate, depth, l2_leaf_reg, min_data_in_leaf, and max_ctr_complexity. These hyperparameters affect the performance and generalization ability of the CatBoost model during training. Additionally, by setting the "auto_class_weights" parameter to "Balanced", we enable CatBoost to automatically handle the issue of data imbalance. In other words, by ensuring that the "auto_class_weights" parameter is set to "Balanced", CatBoost optimizes the aforementioned five parameters. Specifically, CatBoost achieves data balancing by comprehensively adjusting the frequency of each class in the training data, the gradient of the loss function on samples in each iteration, and the splitting of tree nodes. Our method utilizes the parameter tuning functionality of genetic algorithms, setting the crossover probability of the genetic algorithm to 0.6, and the mutation probability to 0.01. The tuning range for each hyperparameter of CatBoost is as shown in Table 1.

Table 1. Experimental hyperparameters tuning range.

Hyperparameter Name	Interval	Explain
learning_rate	[0.01, 1]	Weight of each step
depth	[1, 16]	Limiting the maximum depth of the tree model
l2_leaf_reg	[0, 10]	Penalizing the model complexity.
min_data_in_leaf	[1, 1000]	Making the model more robust.
max_ctr_complexity	[1, 10]	Controlling the complexity of feature combinations.
auto_class_weights	Balanced	Automatically adapt to the data imbalance issue.

4. Results

4.1. Model Training

After data preprocessing, we compared our method with traditional machine learning methods in terms of model performance. In this experiment, the feature selection part using RF-RFE with three-fold cross-validation to obtain intermediate results, while the main machine learning algorithms employed ten-fold cross-validation to prevent overfitting and better evaluate the model's generalization ability. The iteration number for CatBoost was set at 150 for the main experiment, with some comparative experiments using 50 iterations. The final results were evaluated using metrics such as accuracy, sensitivity, precision, F1 score, and AUC. The experimental setup included an Intel Core i7 processor @2.80 GHz, 16 GB of memory, and Windows 10 operating system. Python environment was constructed using Anaconda, and coding was performed using Python third-party libraries such as numpy and pandas.

4.2. Comparison among CatBoost and Other Classifcation Methods

We analyzed the proposed method and compared its model performance with other commonly used machine learning methods. When data preprocessing and feature selection were the same, we comprehensively analyzed the performance of the models from different categorical feature processing and different machine learning methods. Table 2 summarizes the experimental results of LR, Adaboost, DT, RF, XGBoost, LightGBM, and CatBoost. Since only LightGBM and CatBoost have the ability to handle categorical features, the input features for these two algorithms do not require further processing, while the input features for the remaining algorithms are processed using LabelEncoder. We can see that CatBoost without text features performs similarly to LightGBM, but after adding text features, all evaluation metrics show significant improvement. After further tuning with GA, the performance of the GA-CatBoost model further improved. Building upon the GA-CatBoost model, we addressed the data imbalance issue by setting data balance parameters, and our proposed model outperformed other algorithms in terms of accuracy (87.87), sensitivity (92.68), precision (89.35), F1 score (90.99), and AUC (85.59). Additionally, we found that the

performance of Adaboost and RF surpassed that of LR and DT. Adaboost and RF belong to ensemble learning algorithms, indicating that ensemble learning methods outperform single learning classifiers.

Table 2. Performance comparison of different classification methods.

Feature Processing	Training Models	AUC	Accuracy	F1	Sensitivity	Precision
LabelEncoder	Logistics regression	61.54	70.93	80.48	90.74	72.31
	Adaboost	72.62	78.55	84.87	91.07	79.46
	Decision tree	74.98	77.25	82.65	82.04	83.27
	Random forest	79.22	82.65	87.26	89.90	84.76
	XGBoost	78.56	82.52	87.29	90.87	83.98
Built-in category processing	LightGBM	78.87	82.60	87.29	90.48	84.32
	CatBoost	78.24	82.24	87.09	90.69	87.37
	CatBoost(text)	84.06	86.79	90.25	92.56	88.06
	GA-CatBoost	85.50	87.77	90.91	92.59	89.29
	GA-CatBoost-weight	85.59	87.87	90.99	92.68	89.35

4.3. Comparison between Training Models with Different Fitness Evaluations

After obtaining the optimal feature subset through RFE feature selection, we evaluated the impact of different fitness functions on the model results, as GA requires specifying a specific fitness function. We introduced the optimal fitness function into CatBoost. Table 3 displays the model performance with different fitness functions.

Table 3. Experimental results of GA-CatBoost with different fitness functions.

Training Model	AUC	Accuracy	F1	Sensitivity	Precision
GA-CatBoost-Accuracy	84.92	87.39	90.65	92.59	88.80
GA-CatBoost-Sensitivity	84.92	87.39	90.65	92.59	88.80
GA-CatBoost-Precision	84.91	87.38	90.65	92.59	88.78
GA-CatBoost-F1	84.92	87.39	90.65	92.59	88.80
GA-CatBoost-AUC	84.91	87.38	90.65	92.59	88.78

We can observe that the model performance is very similar under different fitness functions, with GA-CatBoost showing the best performance in accuracy, sensitivity, and F1 score functions. However, we are more focused on the performance of sensitivity, as from the definition of sensitivity, we understand that sensitivity refers to the proportion of correctly predicting positive samples among all actual positive samples, indicating the model's ability to identify all terrorist attacks with casualties. In the context of terrorist attacks, the model may accidentally predict events with no casualties as events with casualties, as decision-makers may prepare for the worst-case scenario by allocating sufficient resources. However, even if a terrorist attack with casualties is not accurately predicted, it could lead to severe consequences due to insufficient preparedness for terrorist attacks. Therefore, in this study, we selected sensitivity as the fitness function for GA.

4.4. Comparison between Different Hyperparameter Tuning Methods

In this section, we analyzed the model performance comparison of our hyperparameter optimization method with other commonly used hyperparameter optimization methods. Table 4 lists the comparison of the results of CatBoost after parameter tuning using manual hyperparameter adjustment, grid search algorithm, random search algorithm, Bayesian hyperparameter optimization algorithm, and genetic hyperparameter optimization algorithm.

Table 4. Results of different hyperparameter tuning methods.

Training Model	AUC	Accuracy	F1	Sensitivity	Precision
Manual	78.52	82.22	87.00	90.04	84.16
Grid search	84.64	87.13	90.46	92.38	88.62
Random search	81.82	84.93	88.91	91.49	86.48
Bayesian	83.81	86.51	90.03	92.19	87.96
Genetic algorithm	84.92	87.39	90.65	92.59	88.80

Through comparison, we can draw the following conclusions. Firstly, manual hyperparameter optimization methods not only incur significant human costs but also make it difficult to manually grasp the regularities of hyperparameters, hence resulting in the poorest model performance. Secondly, the grid search algorithm outperforms the random search algorithm and the Bayesian hyperparameter optimization algorithm. This is because the grid search algorithm traverses candidate hyperparameter combinations in a polling manner, which allows for testing the performance of more hyperparameter combinations. However, the computation cost increases with the number of hyperparameters. Therefore, the efficiency of the grid search algorithm is lower compared to the random search algorithm and the Bayesian hyperparameter optimization algorithm. Additionally, when the search iterations of the random search algorithm and Bayesian hyperparameter optimization algorithm are sufficiently large, the performance results of these two optimization algorithms tend to approach the results of the grid search algorithm. Lastly, the results of genetic hyperparameter optimization algorithm are superior to other hyperparameter optimization algorithms, indicating that the genetic algorithm is a more effective hyperparameter optimization algorithm, particularly for CatBoost.

5. Conclusions

Rapid and effective assessment of the potential severe consequences of terrorist attacks can provide valuable information support for decision-makers to formulate emergency measures and counter-terrorism plans. This paper proposes a terrorist attack casualty prediction algorithm named GA-CatBoost-Weight based on CatBoost, aiming to predict whether a terrorist attack will cause harm to innocent civilians.

In the proposed algorithm, to address the data imbalance issue of traditional data sampling methods in handling textual information, the performance of the model is further improved on balanced data basis through the inherent functions and superior feature analysis capabilities of CatBoost. Additionally, genetic algorithm is utilized to optimize various parameters of CatBoost. The algorithm is evaluated on a terrorist attack dataset, demonstrating superior performance compared to several commonly used machine learning methods.

Using machine learning methods to predict the consequences of terrorist attacks can effectively reduce the harm caused by attacks. This study is the first to use the CatBoost algorithm for predicting terrorist attacks, which holds multiple significances. Firstly, CatBoost can directly handle textual information without the need for intermediate steps such as text vectorization, preserving semantic information of the text and reducing time complexity. Secondly, traditional data sampling struggles to handle data imbalance issues with textual features directly, while CatBoost can address sample balance for various modal data through its inherent hyperparameters. Lastly, CatBoost can achieve superior model performance with a small number of iterations, paving the way for new research avenues.

Author Contributions: Conceptualization, Y.H.; methodology, Y.H. and B.Y.; software, Y.H. and B.Y.; validation, Y.H. and B.Y.; formal analysis, Y.H.; investigation, Y.H.; resources, Y.H.; data curation, Y.H.; writing—original draft preparation, Y.H.; writing—review and editing, C.C.; supervision, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The open-source data used in this study comes from the Global Terrorism Database, which can be obtained from https://www.start.umd.edu/gtd (accessed on 28 August 2022).

Acknowledgments: The authors declare any support not covered by the author's contribution or funding section.

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

References

1. LaFree, G.; Dugan, L. Introducing the Global Terrorism Database. *Terror. Political Violence* **2007**, *19*, 181–204. [CrossRef]
2. Li, W.; Guo, L. The Impact of COVID-19 on International Terrorism and Its Counter-measures. *Glob. Gov.* **2021**, *3*, 65–77+157.
3. Han, X. Kremlin Drone Attack Raises Concerns of Escalating Russia-Ukraine Conflict. *Guangming Dly.* **2023**, *008*. [CrossRef]
4. Lu, Y. Intensification of Israel-Palestine Conflict Exacerbates Social Division, Europe Faces High Risk of Major Terrorist Attacks. *Lib. Dly.* **2023**, *007*.
5. Abdalsalam, M.; Li, C.; Dahou, A.; Noor, S. A Study of the Effects of Textual Features on Prediction of Terrorism Attacks in GTD Dataset. *Eng. Lett.* **2021**, *29*, 416–443.
6. Guo, X.; Wu, W.; Xiao, Z. Civil aviation airport terrorism risk assessment model based on event tree and PRA. *Appl. Res. Comput.* **2017**, *34*, 1809–1811.
7. Yang, Y. Research on the Risk Assessment and Prevention of Terrorist Attacks in Religious Site Based on FAHP-SWOT. *J. Hunan Police Acad.* **2019**, *31*, 99–106.
8. Luo, L.; Qi, C. An analysis of the crucial indicators impacting the risk of terrorist attacks: A predictive perspective. *Saf. Sci.* **2021**, *144*, 105442. [CrossRef]
9. Zhang, D.; Qian, L.; Mao, B.; Huang, C.; Huang, B.; Si, Y. A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost. *IEEE Access* **2018**, *6*, 21020–21031. [CrossRef]
10. Feng, Y.; Wang, D.; Yin, Y.; Li, Z.; Hu, Z. An XGBoost-based casualty prediction method for terrorist attacks. *Complex Intell. Syst.* **2020**, *6*, 721–740. [CrossRef]
11. Shafiq, S.; Haider Butt, W.; Qamar, U. Attack type prediction using hybrid classifier. In *Advanced Data Mining and Applications, Proceedings of the 10th International Conference, ADMA 2014, Guilin, China, 19–21 December 2014*; Springer International Publishing: Berlin/Heidelberg, Germany,, 2014; pp. 488–498.
12. Meng, X.; Nie, L.; Song, J. Big data-based prediction of terrorist attacks. *Comput. Electr. Eng.* **2019**, *77*, 120–127. [CrossRef]
13. Gundabathula, V.T.; Vaidhehi, V. An Efficient Modelling of Terrorist Groups in India Using Machine Learning Algorithms. *Indian J. Sci. Technol.* **2018**, *11*, 1–10. [CrossRef]
14. Khan, F.A.; Li, G.; Khan, A.N.; Khan, Q.W.; Hadjouni, M.; Elmannai, H. AI-Driven Counter-Terrorism: Enhancing Global Security Through Advanced Predictive Analytics. *IEEE Access* **2023**, *11*, 135864–135879. [CrossRef]
15. Zhang, L.; Qiao, F.; Wang, J.; Zhai, X. Equipment Health Assessment Based on Improved Incremental Support Vector Data Description. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 3205–3216. [CrossRef]
16. Rodriguez-Galiano, V.F.; Luque-Espinar, J.A.; Chica-Olmo, M.; Mendes, M.P. Feature selection approaches for predictive modelling of groundwater nitrate pollution: An evaluation of filters, embedded and wrapper methods. *Sci. Total Environ.* **2018**, *624*, 661–672. [CrossRef] [PubMed]
17. Zhang, X.; Jin, M.; Fu, J.; Hao, M.; Yu, C.; Xie, X. On the Risk Assessment of Terrorist Attacks Coupled with Multi-Source Factors. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 9. [CrossRef]
18. Jiang, L.; Kong, G.; Li, C. Wrapper Framework for Test-Cost-Sensitive Feature Selection. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 1747–1756. [CrossRef]
19. Michalewicz, Z.; Schoenauer, M. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evol. Comput.* **1996**, *4*, 1–32. [CrossRef]