

Article

Denying Evolution Resampling: An Improved Method for Feature Selection on Imbalanced Data

Li Quan , Tao Gong * and Kaida Jiang

College of Information Science and Technology, Donghua University, Shanghai 201620, China; xcunited@hotmail.com (L.Q.); kaida.jiang@hotmail.com (K.J.)

* Correspondence: taogong@dhu.edu.cn; Tel./Fax: +86-021-6779-2312

Abstract: Imbalanced data classification is an important problem in the field of computer science. Traditional classification algorithms often experience a decrease in accuracy when the data distribution is uneven. Therefore, measures need to be taken to improve the balance of the dataset and enhance the classification accuracy of the model. We have designed a data resampling method to improve the accuracy of classification detection. This method relies on the negative selection process to constrain the data evolution process. By combining the CRITIC method with regression coefficients, we establish crossover selection probabilities for elite genes to achieve an evolutionary resampling process. Based on independent weights, the feature analysis improves by 3%. We evaluated the resampled results on publicly available datasets using traditional logistic regression with cross-validation. Compared to the other resampling models, the F1 score performance of the logistic regression five-fold cross-validation is more stable than the other methods using the two sampling results of the proposed method. The effectiveness of the proposed method is verified based on F1 score evaluation results.

Keywords: classification algorithms; imbalanced data; similarity measure; evolutionary process



Citation: Quan, L.; Gong, T.; Jiang, K. Denying Evolution Resampling: An Improved Method for Feature Selection on Imbalanced Data. *Electronics* **2023**, *12*, 3212. <https://doi.org/10.3390/electronics12153212>

Academic Editors: Min Xia, Xiangcheng Chen, Haoxiang Lang, Haidong Shao and Darren Williams

Received: 1 July 2023

Revised: 17 July 2023

Accepted: 20 July 2023

Published: 25 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Imbalanced data analysis is a common problem in machine learning and data mining, referring to the situation where there is a significant imbalance in the number of samples among different classes or labels in a training dataset. For example, in a binary classification problem, one class has a much larger number of samples compared to the other. This situation often occurs in the medical field, such as the diagnosis and prediction of rare diseases; in text classification tasks, such as sentiment analysis and spam email detection; in image classification, such as anomaly detection in medical imaging and vehicle recognition; and in credit scoring, where imbalanced data are a common issue due to the majority of individuals having good credit records. These mentioned fields and datasets are just a small part of the research on handling imbalanced data, and researchers choose appropriate methods based on specific application domains and data characteristics. Currently, traditional machine learning methods for imbalanced data classification mainly include the following approaches:

- (1) Traditional machine learning methods, including: decision trees, random forests, support vector machines, etc. [1,2].
- (2) Deep learning methods, including: convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long- and short-term memory networks (LSTM) [3–5].
- (3) Ensemble learning methods, including: Bagging, Boosting, etc. [6,7].
- (4) Feature engineering methods, including: feature selection, feature extraction, etc. [8,9].
- (5) Model fusion methods, including: voting method, weighted average method, etc. [10,11].

In the past few years, traditional machine learning methods have demonstrated certain advantages in handling imbalanced data classification. Algorithms such as decision trees

and random forests can effectively handle high-dimensional data and nonlinear relationships. However, these methods have limited scalability and generalization capabilities when dealing with large-scale data. On the other hand, deep learning methods have shown better accuracy in imbalanced data classification. Convolutional neural networks, for instance, can automatically extract features and handle high-dimensional data. Additionally, ensemble learning methods and feature engineering techniques can enhance the performance of classifiers. Model fusion methods can combine the strengths of multiple classifiers, thereby improving classifier accuracy.

However, the problem of imbalanced data still brings challenges to traditional classification algorithms, which reduces the classification accuracy and leads to poor performance of the model in practical applications. To solve this problem, it is necessary to take measures to improve the balance of the data set. Here are a few ways:

- (1) **Resampling methods:** This includes over-sampling, under-sampling, synthetic sample generation, and data augmentation techniques to improve the robustness and accuracy of classification algorithms. Taking the synthetic sample generation algorithm as an example, SMOTE (Synthetic Minority Over-sampling Technique) generates minority class samples to increase their quantity.
- (2) **Data collection and preprocessing:** During the data collection stage, the distribution of samples should be considered, and efforts should be made to ensure the balance of the dataset [12,13]. Methods such as feature selection and feature extraction can be used to remove irrelevant or redundant features and improve model performance.
- (3) **Class weighting:** During the model training process, assign greater weight to minority class samples to emphasize their importance. Class weighting is often implemented in the loss function.

To solve this problem, we try to build an integrated resampling model to alleviate the impact of imbalanced data on the detection model. The main points are as follows:

- (1) Utilizing the combination of negative selection process and evolutionary theory, constraints are placed on the selection of resampled data, elite gene crossovers, and mutation processes.
- (2) Based on independent weights to enhance feature selection efficiency, supervised learning methods are used to improve the recognition process of data feature boundaries, such as special treatment for minority classes (e.g., sampling them or assigning them special weights).

The remaining parts of this chapter are arranged as follows: In Section 2, the current research related to imbalanced data is discussed, and possible directions for optimization are presented. In Section 3, a Negatively Evolved Resampling (NER) model is constructed based on the identified optimization and improvement directions. In Section 4, the model is tested, and the results are compared with other resampling detection models. Section 5 analyzes the test results.

2. Related Work

The limitations of resampling methods include high computational complexity, the need to store all training data, and vulnerability to noise and outliers. Therefore, optimizing these processes is crucial for evolutionary methods. Currently, the relevant research work is summarized in Figure 1.

2.1. Data Sampling

In fields such as medical diagnosis, earth sciences, and engineering machinery, these domains often exhibit noticeable characteristics of imbalanced data.

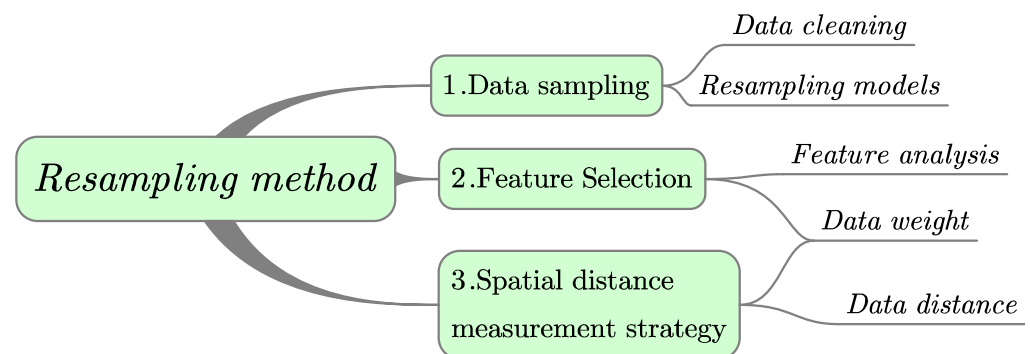


Figure 1. Research focus of related work.

In the traditional k-nearest neighbors (kNN) model, it is crucial to have known labeled samples for predicting the labels of unknown samples [14]. The majority voting rule may overlook outliers or anomalies in uncertain data, resulting in significant bias in the prediction outcomes. A commonly employed approach for handling imbalanced data is resampling based on the local mean or median of kNN. Jaafor et al. [15] proposed a model to deal with data imbalance. By learning the training set in advance and then distance-weighting the test data, the model's learning noise was reduced, which increased the sensitivity of the model to the characteristics of data samples. Bao [16] and Ying [17] proposed a method of data resampling and filtering to improve the anomaly detection model recognized by the kNN algorithm through data mining of high-dimensional unbalanced data. Han [18], Li [19], and Xu [20] all adopted the support vector machine (SVM) in their respective studies to realize a kNN optimization model for data sample class allocation in the data statistics process. Mavrogiorgos et al. [21] proposed a rule-based data cleaning mechanism that uses Natural Language Processing (NLP) to address imbalanced data in the healthcare domain. By using specific grammar similarity algorithms, the "NLP engine" in the proposed method helps to validate the reusability of rules and data patterns, saving time and minimizing computational costs.

Resampling methods can be categorized into two categories: over-sampling and under-sampling. Over-sampling involves increasing the number of samples in the minority class to balance the dataset. Examples of over-sampling methods include random over-sampling, SMOTE, and ADASYN. On the other hand, under-sampling reduces the number of samples in the majority class to achieve balance. Examples of under-sampling methods include Random under-sampling, Tomek links, and Edited Nearest Neighbors. Zhang et al. [22] proposed the instance-weighted SMOTE (IW-SMOTE) algorithm, which indirectly utilizes distribution data to enhance the robustness and universality of SMOTE, achieving good results.

Additionally, there are hybrid methods that combine over-sampling and under-sampling, such as SMOTE-ENN and SMOTE-Tomek. These methods simultaneously increase the number of minority class samples and reduce the number of majority class samples to create a more balanced dataset. The choice of resampling method depends on the specific problem, dataset, and algorithm. Sometimes, using a combination of different resampling techniques can yield better results. However, it is important to note that excessive over-sampling may lead to overfitting problems, and under-sampling may result in the loss of useful information. Therefore, when using resampling methods, it is necessary to adjust and evaluate them reasonably based on the specific situation.

2.2. Feature Selection

Choosing the most relevant features reduces the impact of noise and improves the generalization ability of the model. Since high-dimensional features increase the computational cost of a classification detection model, it is necessary to obtain a set of important features

through feature selection methods to improve the efficiency of the model. Adding sample weights when calculating data distances improves the accuracy of model classification.

Feature recognition is crucial for nearest neighbor classification models as the boundary of data samples is determined by the feature values. Currently, research on feature recognition algorithms mainly falls into two aspects: traditional feature extraction methods and deep learning feature extraction methods. Traditional feature extraction methods are mainly used to extract useful information from raw data. These methods are commonly used for structured data such as images, audio, text, and time series data, converting them into numerical features for further processing by machine learning models. Some classical data feature extraction algorithms are used for dimensionality reduction, noise removal, and increasing discriminability, including Principal Component Analysis (PCA) [23], Linear Discriminant Analysis (LDA) [24], Singular Value Decomposition (SVD) [25,26], and Wavelet Transform (WT) [27,28], among others.

Deep learning feature extraction methods can automatically learn complex feature representations from data, helping models recognize and understand patterns and information in the data. Some popular deep learning data feature extraction methods include convolutional neural networks (CNNs) [29], recurrent neural networks (RNNs) [30], autoencoders (AEs) [31], and other pre-training models.

The research status of feature recognition algorithms is still evolving and currently faces several challenges. Firstly, the processing of large-scale data requires more efficient and accurate algorithms, especially in deep learning algorithms, where better training sets and reliable network structure designs are needed. Secondly, the robustness of features needs to be further improved to combat the influence of noise, fuzziness, and other issues. Finally, the interpretability of features and their applicability in different scenarios are also issues that need to be addressed.

Researchers have proposed various approaches to address these challenges. Keller [32] proposed a fuzzy boundary kNN model based on the membership relationship between data, which has higher recognition efficiency for specific data samples. Xia et al. [33] achieved certain results in detecting and recognizing the boundaries of data points using the reverse k-nearest neighbor technique. Kumar et al. [34] proposed a variant of a neural-network-based kNN model that restricts the boundaries of unknown data within the maximum distance of known data types, optimizing the overall computational efficiency of the classification model. Czarnowski et al. [35] combined a weighted ensemble classification method with a technique for handling class imbalance problems. They proposed and validated a new hybrid method for data stream learning, which uses over-sampling and instance selection techniques to transform class distribution into a more balanced dataset. Experimental results showed that this method effectively improves the performance of online learning algorithms.

Overall, feature recognition algorithms are continuously evolving, aiming to improve efficiency, robustness, interpretability, and applicability in various scenarios.

2.3. Spatial Distance Measurement Strategy

Distance metric optimization strategies include standardizing features to have zero mean and unit variance, reducing dimensionality using techniques like PCA or t-SNE, assigning appropriate weights to features based on their importance, learning a discriminative metric using methods like Siamese Networks or Mahalanobis distance learning, and combining multiple distance measures through ensemble learning or distance measure combination. These strategies aim to improve the performance and generalization ability of the model in machine learning and pattern recognition tasks by optimizing the choice and calculation of distance metrics.

- (1) Kernel Method: Optimization of distance measurement is realized by mapping samples to high-dimensional feature spaces [36,37]. Larson, N. B. et al. [38] evaluated the nuclear method for the genetic association analysis process and concluded that the nuclear method has good robustness and flexibility. Cakmakhusi S. D. et al. [39] pro-

- posed a DDoS detection scheme based on the Mahalanobis distance nuclear method, and the experimental results proved the effectiveness of the scheme.
- (2) Local distance weighting: According to the non-uniformity of data distribution, the sample vectors in different regions are weighted by different distances, which can effectively deal with the problem of uneven data distribution. Tang, C. et al. [40] proposed an unsupervised feature selection method based on multi-graph fusion and feature weight learning to carry out weighted reconstruction of local data of samples. Experimental data verified the effectiveness of local distance weighting.
 - (3) Learning distance metrics: By utilizing training data to establish an optimal distance metric matrix or metric function, the performance of such models can be enhanced by optimizing for specific tasks and datasets. Sung, F. et al. [41] used the distance measurement method based on the neural network model to carry out classification tasks in small sample scenarios and achieved good results.
 - (4) Distance measurement based on information theory: Based on the principle of information theory, optimization is achieved by maximizing the information entropy of distance measurement. This method can effectively improve the robustness and generalization ability of the model. Campo D. S. et al. [42] proposed a new distance metric mutual information and entropy to identify phenotypic correlation of genetic distance measurement. Shannon entropy was used to measure the heterogeneity of individual locations, and coordinated replacement of mutual information metric locations was realized.

The bottleneck of distance measurement lies in the strong support needed from mathematical theory, and existing technology may not be able to achieve higher accuracy and a wider range of measurement. Additionally, the dimensionality of data plays a crucial role in data measurement. Factors such as weather conditions, light intensity, and terrain changes can affect the accuracy of temperature measurement, necessitating additional data processing. Processing and analyzing large-scale measurement data require efficient algorithms and computing power, which also brings cost pressure. Furthermore, the development and improvement of new technology require substantial investment. When the high cost directly relates to the use of the technology, it can limit the promotion and application of new technology.

3. The Negation Evolutionary Resampling Method

In this section, based on the research related to the negative selection algorithm model [43], combined with evolutionary algorithms, a construction of the resampling method is aimed at improving the resampling process through supervised learning of the spatial attribute feature relationships between data. The overall framework of the negative evolutionary resampling method is shown in Figure 2, with the goal of improving the prediction efficiency of the model.

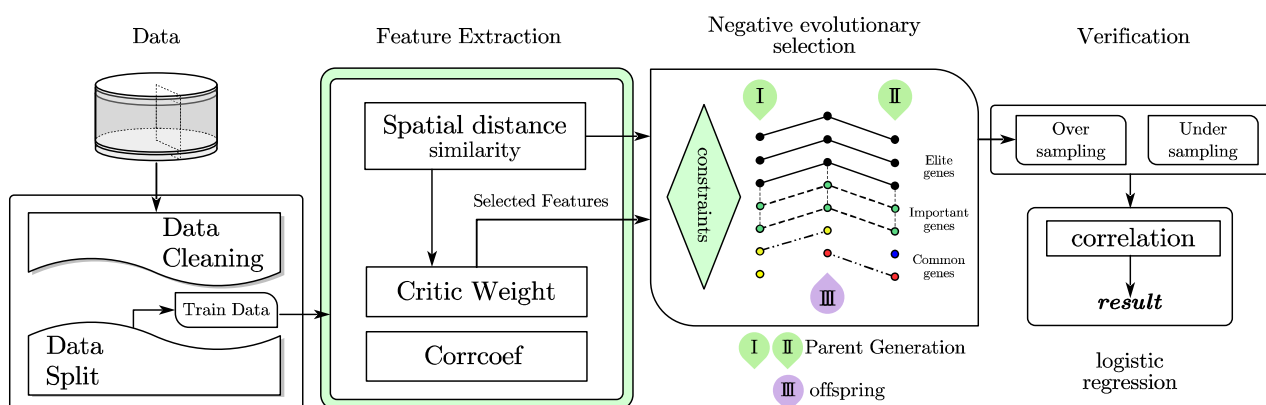


Figure 2. Overall architecture of the detection model.

3.1. Spatial Distance Similarity

In the negation evolutionary model, the choice of distance function is very important for the accuracy of classification results. Common distance functions include Euclidean distance and cosine similarity. In order to ensure the reliability of evolutionary data, it is necessary to prioritize which distance measurement method should be adopted. We compared three common spatial distance measurement methods by considering the performance of real-valued data in space.

For given vectors X and Y , the Euclidean distance formula is:

$$d_{euclidean} = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (1)$$

The cosine similarity can be represented as:

$$d_{cosine} = \sum_{i=1}^n (X_i \times Y_i) / \left(\sqrt{\sum_{i=1}^n X_i^2} \times \sqrt{\sum_{i=1}^n Y_i^2} \right) \quad (2)$$

The Bray–Curtis distance can be represented as:

$$d_{BrayCurtis} = \sum_i |X_i - Y_i| / \sum_i (X_i + Y_i) \quad (3)$$

Under the condition of high-dimensional data, the detection model uses the Bray–Curtis distance measure for the similarity measure of data, as shown in Formula (3). Figure 3 is obtained by using randomly generated multi-class data sets for verification. Because Bray–Curtis distances take into account the ratio between features rather than the absolute number, the classification results may be more accurate for some datasets. Specifically, the Bray–Curtis distance is more robust for features with vastly different numbers, so it can better handle a large number of outliers and is suitable for very sparse data or data with a large number of zeros, in which case the Euclidean distance loses its effect, and the cosine distance may not be applicable.

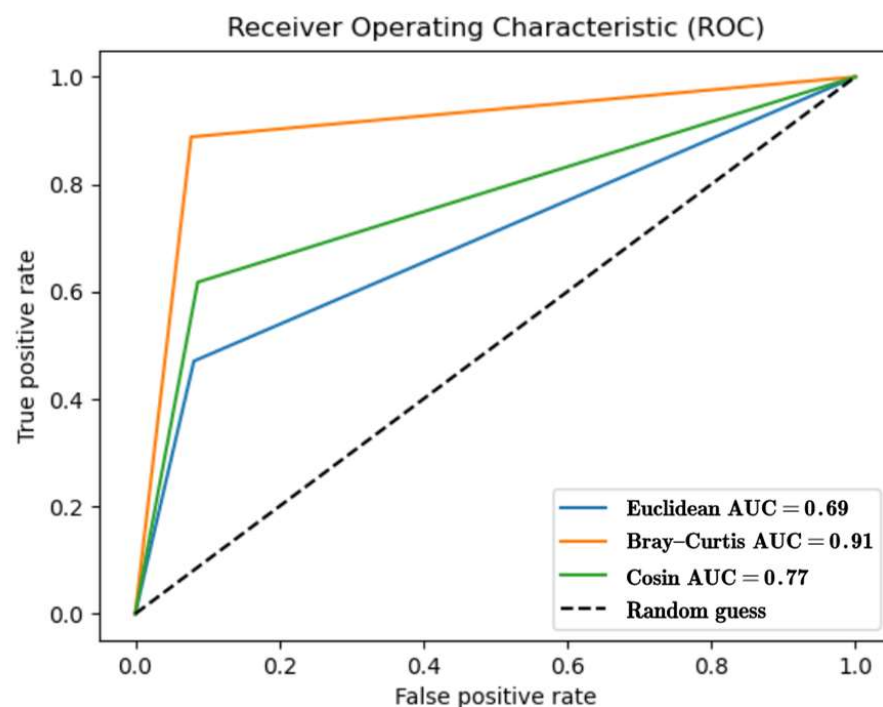


Figure 3. ROC comparison of three distance measures.

However, the Bray–Curtis distance measurement method needs to standardize the values between samples, and different standardization methods may lead to differences in distance calculation results, which makes it difficult to compare models between different data sets. Especially when there is a small abundance in the sample, there may be a miscalculation and a transitive error, which may cause the distance structure between certain environmental features to become complicated, and when there are a large number of species, calculating the distance structure requires more computing time and memory.

3.2. Objective Weighting Strategy

The model uses the CRITIC objective weight [44] method to calculate the independent weight value of data samples. CRITIC is based on the comparison strength of evaluation indicators and the conflict between indicators to comprehensively measure the objective weight of indicators [45]:

- (1) Comparison intensity uses the standard deviation to represent the difference between the values of the same index in each scheme.
- (2) The correlation coefficient represents the conflict between indicators. If there is a strong positive correlation between two indicators, the smaller the conflict, the lower the weight will be.

In the Ss-kNN model, the relevant definition of the CRITIC objective weight process is as follows:

- Step 1 If $D = (x_1, x_2, \dots, x_n)$ exists, for any $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$ composed of m vectors, the original index matrix of the data sample to be detected is:

$$D = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix} \quad (4)$$

- Step 2 Conduct dimensionless processing in a positive way for any $x_{ij} | i \in n, j \in m$ according to Formula (4), to eliminate the influence caused by dimensional differences in multiple cases.

$$x_{ij} = \frac{x_j - x_{\min}}{x_{\max} - x_{\min}} \quad (5)$$

- Step 3 Calculate the form of standard deviation through Formula (5), conduct comparative processing on data sample indicators, and increase the difference of data sample values for specific indicators.

$$\begin{cases} \bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij} \\ S_j = \sqrt{\frac{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}{n-1}} \end{cases} \quad (6)$$

where S_j represents the standard deviation of the j index of any sample x_i .

- Step 4 In the objective weight method of CRITIC, indicator conflict determines the correlation between indicators. The smaller the conflict is, the greater the similarity between indicators; therefore, it is necessary to reduce the allocated weight of the indicator. This index is calculated using the correlation coefficient as in Formula (6).

$$R_j = \sum_{i=1}^p (1 - r_{ij}) \quad (7)$$

r_{ij} represents the correlation coefficient between i and j . The information carrying capacity j of specific index C_j can be obtained as follows.

$$C_j = S_j \cdot R_j = S_j \sum_{i=1}^p (1 - r_{ij}) \quad (8)$$

Step 5 Calculate the objective weight w_j of the j index.

$$w_j = \frac{C_j}{\sum_{j=1}^p C_j} \quad (9)$$

3.3. Negative Evolutionary Resampling

To solve the data imbalance problem, we need to resample the data. In the negation evolution process, based on Formula (9), the evolution process is constrained, and the relevant definitions are as follows:

- Def 1** If parental data exists, the determination of a corresponding similarity space radius for each parental data can be achieved by locating the centroid of the parental data.
- Def 2** According to the random selection process, randomly select two parents from the parent data for crossover and mutation, while satisfying the following two constraints:
- (1) If the similarity between parents is lower than the parent similarity threshold, it is considered that the correlation between the selected parents is poor, and it needs to be selected again.
 - (2) If the parent–offspring similarity threshold between the offspring produced by the crossover and mutation process of the two selected parents exceeds the spatial radius of any parent generation, the correlation between the offspring produced by the crossover and mutation is poor, and it needs to be selected again.
- Def 3** Divide the gene crossover process of the offspring into three parts according to the feature distribution obtained by the objective weighted feature selection process; namely, high elite genes part, median elite cols part, and low elite cols part. The corresponding crossover rate is in the following Table 1.
- Def 4** The time complexity of the random selection process in the process of negative selection evolution is usually $O(n)$, where n represents the number of individuals in the candidate set. Specifically, the selection process can be carried out in the following steps:
- (1) Calculate the initial selection probabilities *rate*: The probability of selecting each individual can be calculated based on their fitness values. By establishing a decreasing selection count for each individual, the selection probability for unselected individuals can be increased. Assuming that, in each iteration, the chances of each individual in the parent sample array being selected are dependent on the array total n and the number of iterations *iteration*, the sampling chance *op* for the two cases is as follows:

$$\begin{cases} \text{Over } op = n \times (r^{-iteration}) \\ \text{Under } op = n \times (1 - (1 - r)^{iteration}) \end{cases} \quad (10)$$

- (2) The initial selection probability *rate* is calculated based on the following conditions: r is a positive value, and when over-sampling is performed, it takes the value $r = [1, 2)$. The larger r is, the faster the data n will reduce. On the other hand, when under-sampling is performed, it takes the value $r = [0, 1)$. The smaller r is, the faster the data n will grow:

$$rate = op_x / \text{sum}(op_n) \quad (11)$$

Selection process: Based on the calculated selection probabilities, individuals are randomly selected from the candidate pool as parents for the evolutionary process. If any selected parent individual meets the constraints, their selection chance is reduced, and their selection probabilities are recalculated using Formula (11). As the recalculated

selection process usually involves traversing the entire candidate pool, the time complexity remains $O(n)$.

Table 1. Evolutionary crossover rate.

Elite Cols	Cross Rate
High	100%
Median	50%
Low	25%

4. Model Testing

4.1. Data Analysis

The problem of imbalanced data affects various aspects of people's lives, such as the healthcare sector. Imbalanced medical data pose a significant challenge as the collection and sharing of data are often limited by legal, privacy, and security concerns. This leads to an imbalance in datasets, which can have a negative impact on medical research and diagnostics.

In the field of education, there exists an imbalance in learning resources and opportunities among students from different socio-economic backgrounds and geographical locations in certain areas. This can result in a dataset where data from specific groups are more abundant while data from other groups are relatively scarce, thereby affecting the accurate analysis of overall educational development trends.

In the socio-economic domain, imbalanced socio-economic data are also a common issue. Economic data from some countries or regions may receive complete and accurate recording, while data from other countries or regions may be incomplete or inaccurate. This can impact the overall understanding of global or regional economic development and decision making.

The Credit Card Fraud Detection dataset [46,47] is primarily used as the testing dataset in the model evaluation process. It consists of credit card transaction records from a subset of European regions.

Based on the observation of the time distribution in Figure 1, there is no clear time boundary between fraudulent and non-fraudulent data. According to the description of the dataset, it is highly imbalanced, with only a small portion of transactions labeled as fraudulent. The dataset contains a total of 284,807 transactions, of which only 492 are fraudulent, accounting for 0.172% of the total transactions.

In addition, this dataset contains 30 variables, 28 of which are the variables obtained by PCA, which are used to protect the privacy of traders.

Therefore, the exact meaning of these variables is unknown; only the time and amount variables represent the time and the numerical value of the specified feature. Figure 4 is based on these two numerical values. At the same time, the 28 variables obtained by the PCA method should have a linear relationship, and the preliminary analysis results of data characteristics were verified by using the Lasso regression method, as shown in Figure 5 below. From the correlation analysis in Figure 5, it can be seen that there is a significant correlation between the relevant indicators in V1-V18 for fraudulent data, while the correlation coefficients for V19-V29 and Amount are relatively low. Using the Lasso feature correlation analysis results for ranking, we obtained Figure 6, which shows that in fraudulent data, the columns after the Time attribute make almost no contribution to feature analysis.

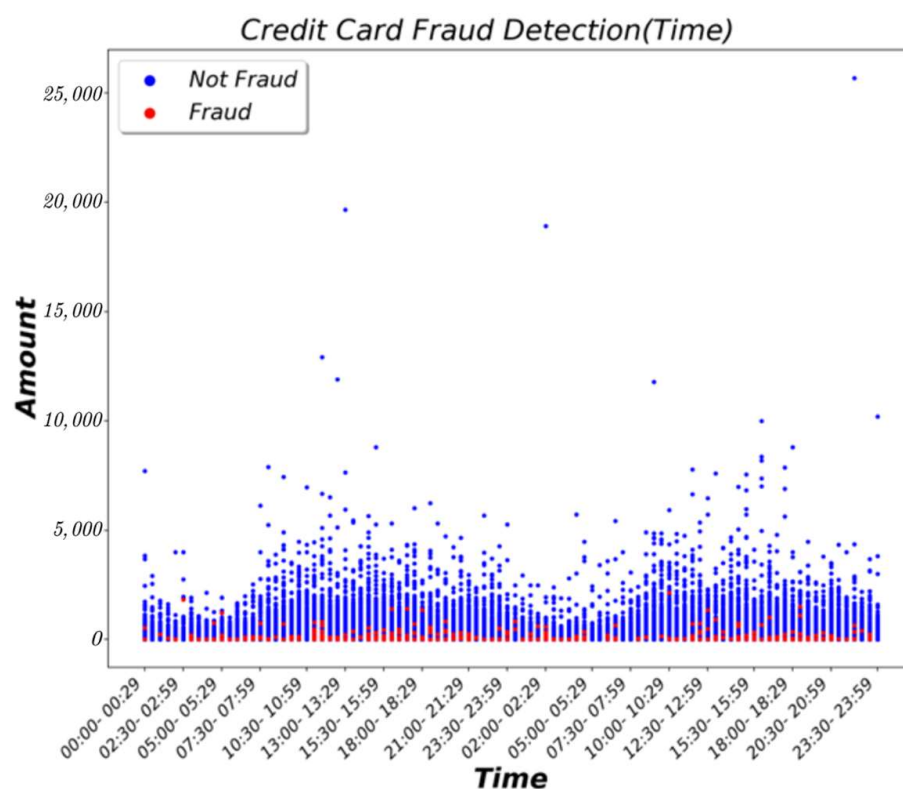


Figure 4. Temporal periodic distribution of credit card transaction fraud data.

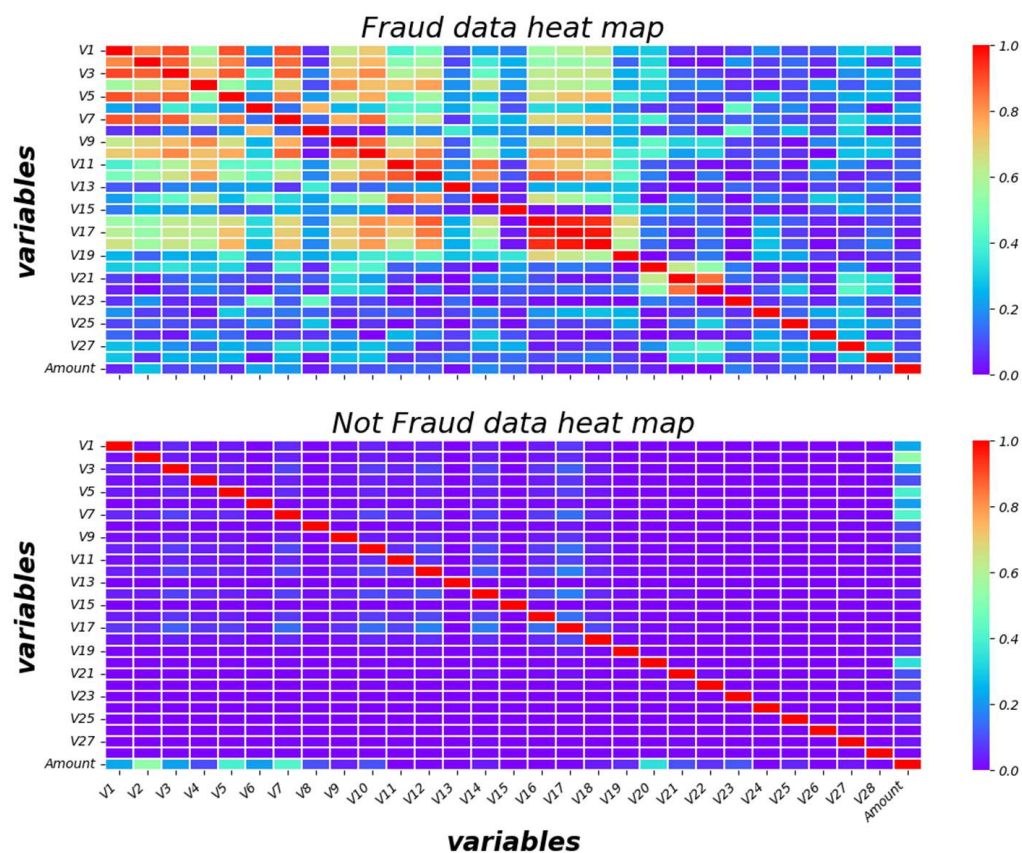


Figure 5. Attribute correlation of credit card transaction fraud data.

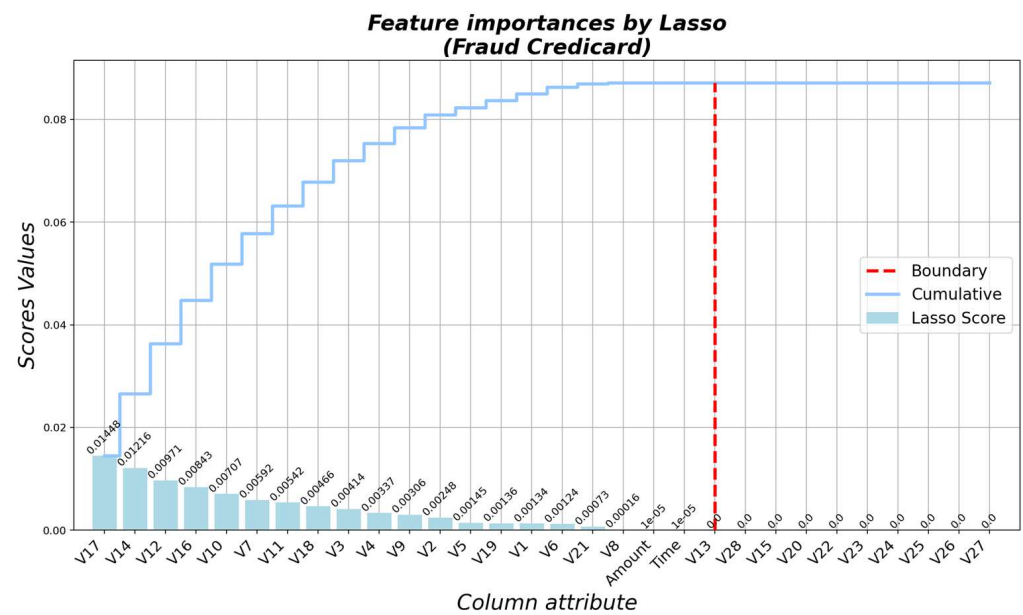


Figure 6. Feature selection results based on traditional method.

4.2. Test Environment and Parameters

The algorithm test environment was deployed based on the Ali Cloud ecs.c5.large computing server (8 vCPU 16 GiB (I/O optimization)). The system environment was Linux Ubuntu 21.10 5.4.1-117-Generic Desktop version, and the test Python version was 3.9.

During the test, the sample of the data set was set as D , based on the train_test_split function (split parameters were 0.2), and the training set was randomly split into $Train$ and $Test$, where $Train \cup Test \neq \emptyset$.

4.3. Resample the Comparison Models

The resampling comparison model used was the imblearn toolkit, which includes over-sampled and under-sampled models, as shown in Table 2.

Table 2. Resampling comparison models and their parameters.

Over_Sampling	Parameters	Under_Sampling	Parameters
RandomOverSampler	sampling_strategy = 0.15	RandomUnderSampler	sampling_strategy = 0.1
BorderlineSMOTE		NearMiss	
ADASYN		ClusterCentroids	
SMOTE		EditedNearestNeighbours	
SVMSMOTE		TomekLinks	
		OneSidedSelection	-

5. Results

5.1. Feature Weighting Analysis

Through the objective weighted feature selection strategy, the results of retained feature values are obtained in Figure 7. The weighted feature selection score is 3% higher than the unweighted feature selection score on average, which means that, under the weighted condition, the data features are effectively expanded, and the column attribute order has also changed to some extent.

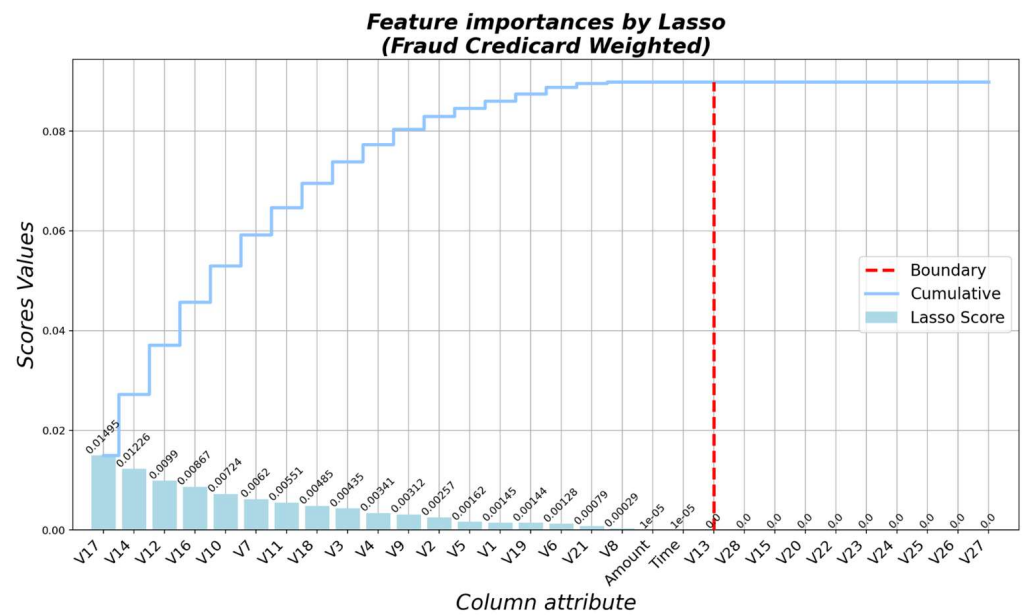


Figure 7. Feature selection results of feature selection strategy based on objective weighting.

5.2. Data Negation Evolutionary Resampling Results

According to the analysis results of the above data characteristics, the first two columns of data characteristics are selected based on the ranking order. The data distribution diagram using the over-sampling results of the traditional nearest neighbor model is obtained in Figure 8, while Figure 9 shows the data distribution using the over-sampling results of the negative evolutionary resampling method. It can be seen that, after using the negative evolutionary resampling method to over-sample the data, the distribution of the data is consistent with the distribution shape of the minority class data. Compared with other comparison models, this method can better preserve the characteristics of the minority class data and enhance the data, improving the recognition rate of classification detection models for minority types.

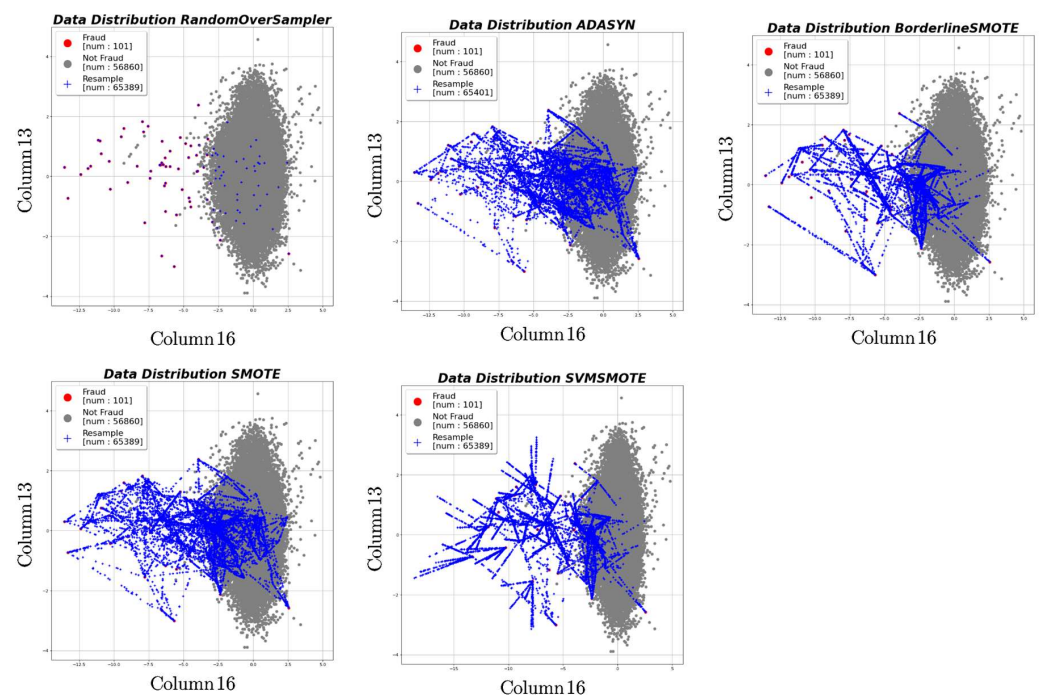


Figure 8. Compare the over-sampling result data distribution of model resampling.

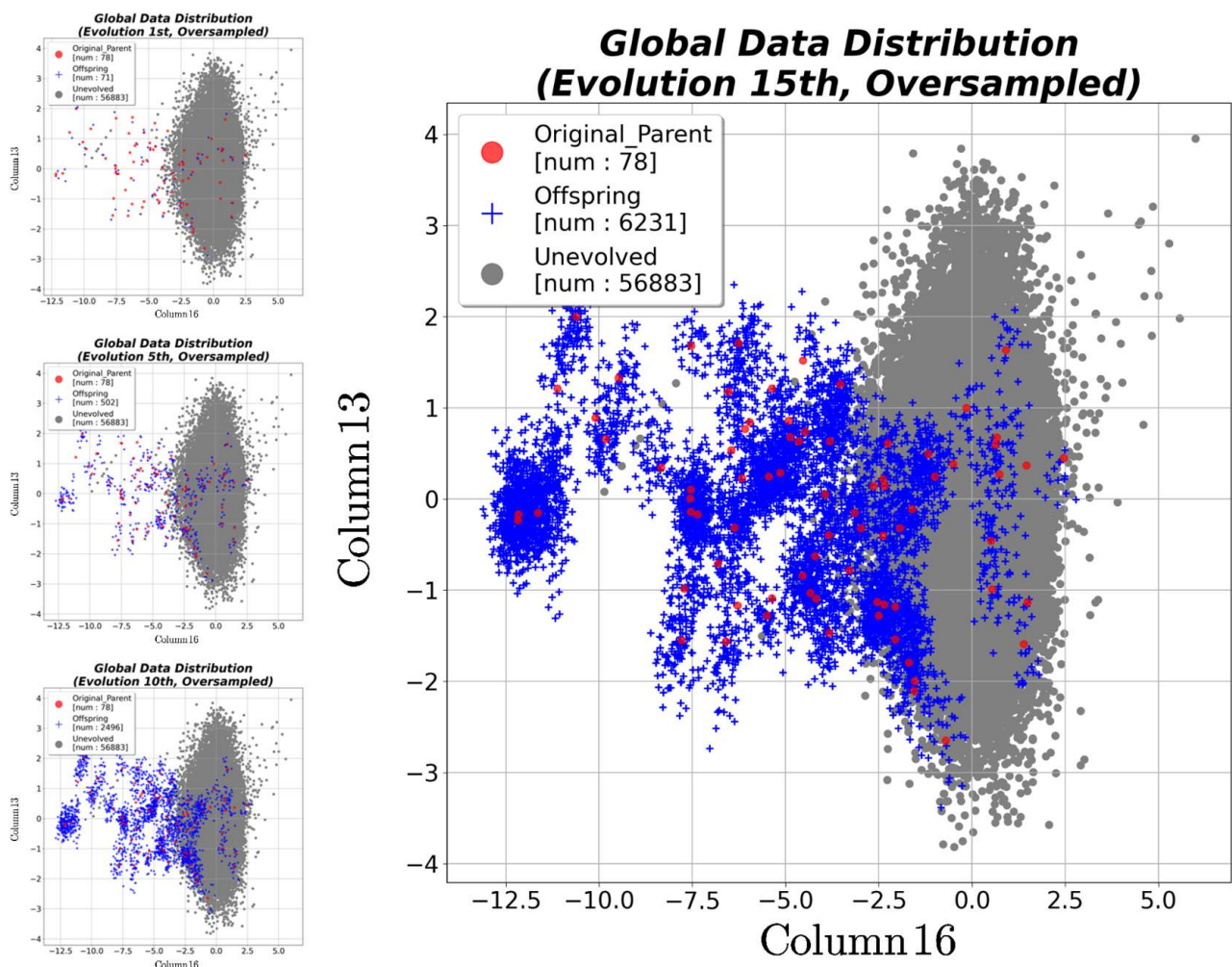


Figure 9. Data distribution of over-sampling results based on the negation evolutionary resampling process.

Similarly, based on the analysis results of the data characteristics, the first two columns of data characteristics are selected according to the ranking order. The data distribution diagram using the under-sampling results of the traditional nearest neighbor model is obtained in Figure 10, while Figure 11 shows the data distribution using the under-sampling results of the negative evolutionary resampling method. It can be seen that, after using the negative evolutionary resampling method to under-sample the data, the distribution of the data is consistent with the distribution shape of the majority class data. Compared with other comparison models, this method can greatly retain the characteristics of the majority class data.

The F1 score is informative for evaluating the performance of resampled data. In the cases of imbalanced data, the F1 score combines precision and recall to accurately assess the model's predictive ability on minority classes. By using the F1 score, we can gain a comprehensive understanding of the model's performance on minority classes and subsequently optimize the model to improve the accuracy of minority class detection. Logistic regression cross-validation F1 scores for different sampling methods are shown in Figure 12 below.

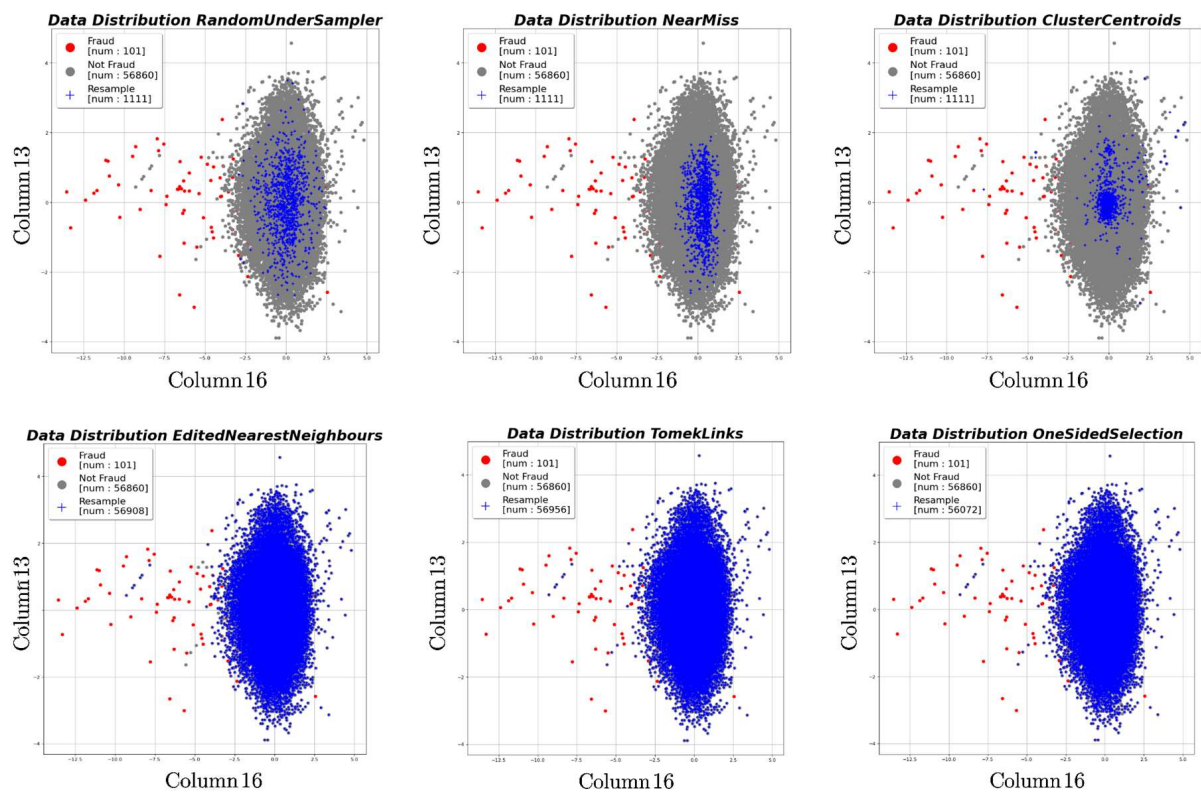


Figure 10. Compare the data distribution of under-sampling results of model resampling.

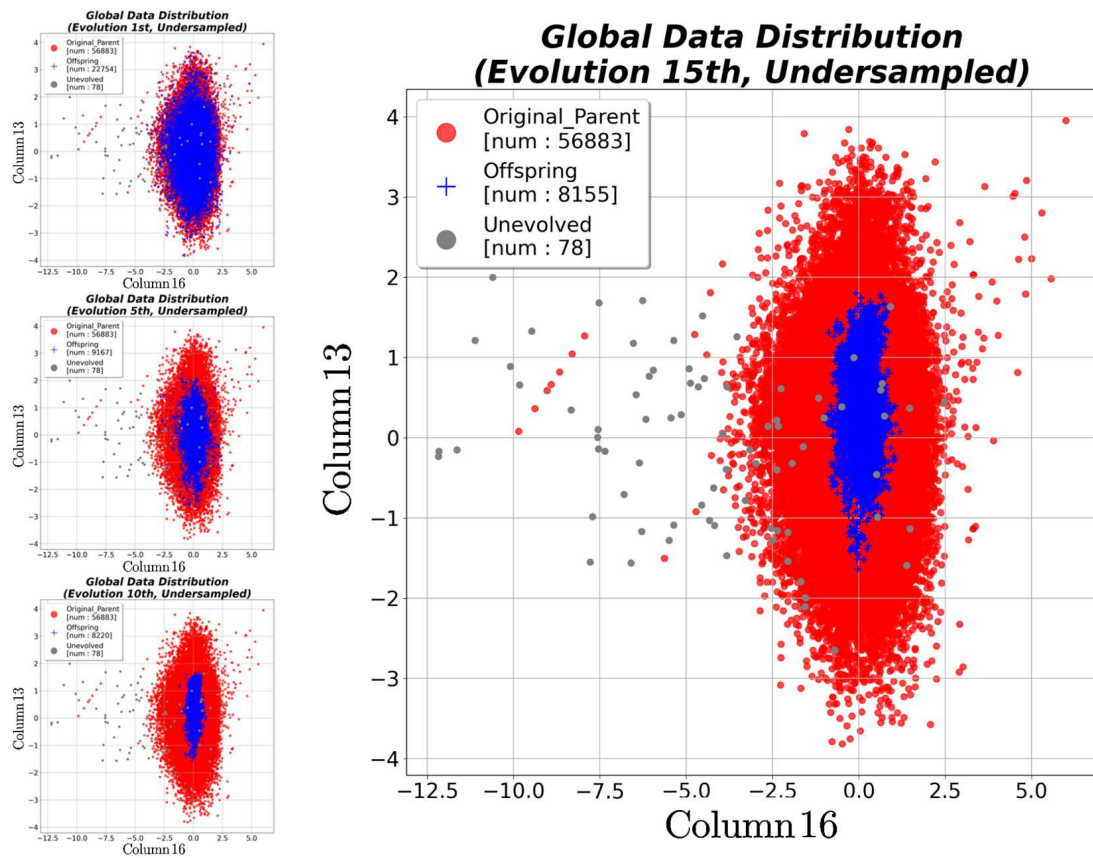


Figure 11. Data distribution of under-sampling results based on the negation evolution resampling process.

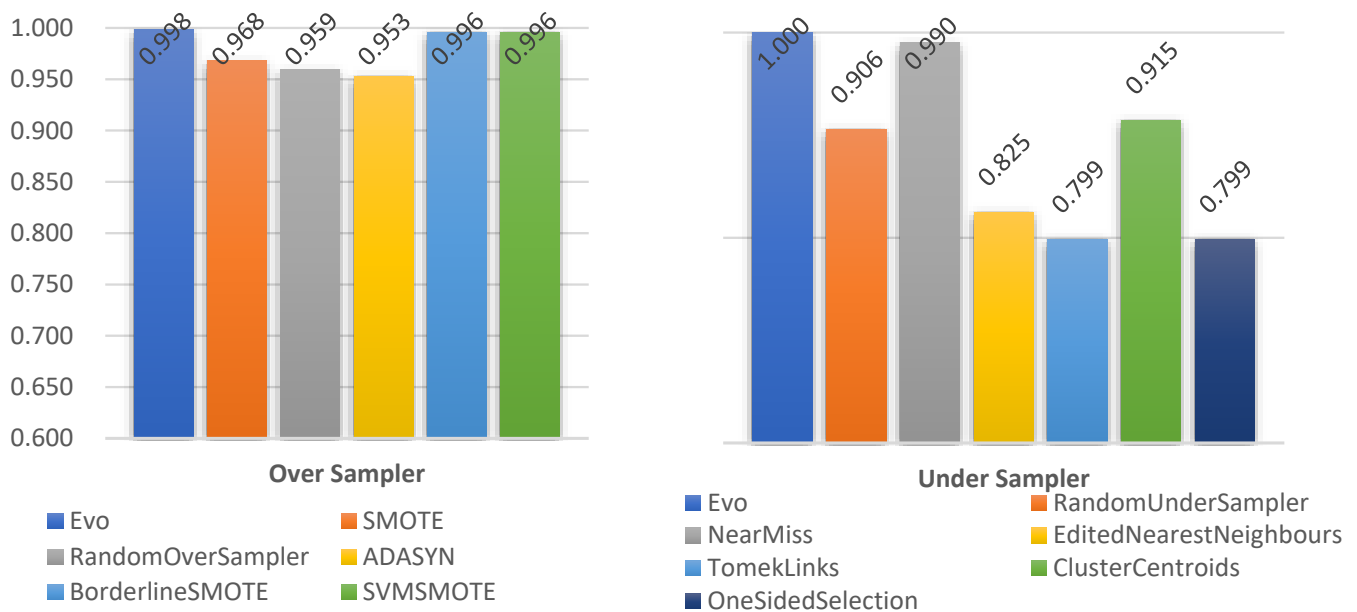


Figure 12. Logistic regression cross-validated F1 scores for different sampling methods.

To verify the effectiveness of the resampled data, we used the logistic regression method for 5-fold cross-validation and took the average to calculate the F1 score metric for the resampled data. Additionally, we conducted a correlation analysis on the data's attributes and obtained Figures 13 and 14. Figure 13 shows that the negative evolutionary resampling method outperforms other similar algorithms in terms of the F1 score metric for over-sampling data. Furthermore, Figure 14 clearly identifies the differences in the selection of minority class data through the attribute correlation analysis of the resampled data. The negative evolutionary resampling method makes the under-sampling data more intuitive, as shown in the adjacent heatmap, where the under-sampling results make the differences in attribute correlation between different types more prominent.

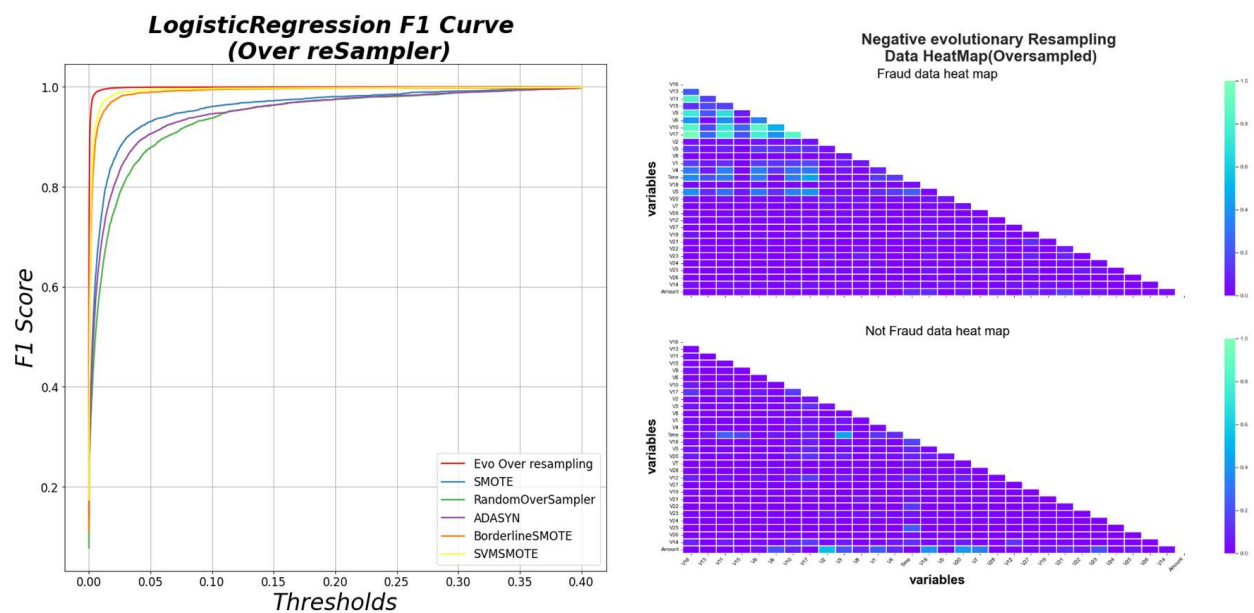


Figure 13. Compare the F1 score validation of over-sampled data based on logistic regression method and create a heatmap of attribute correlation.

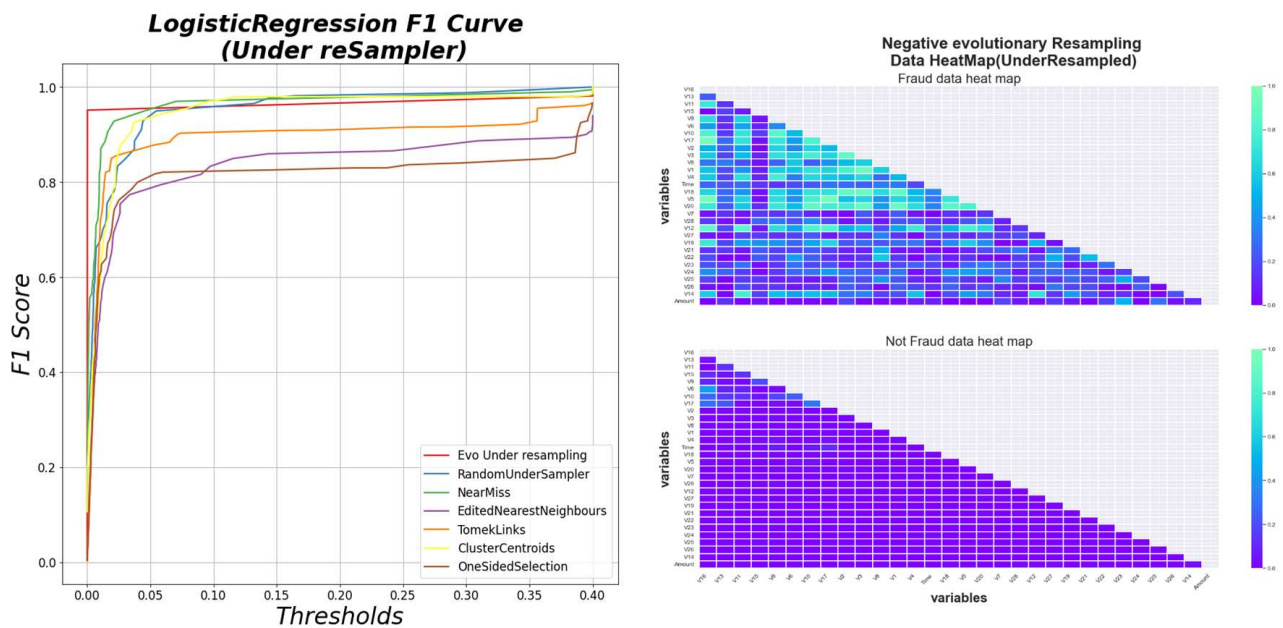


Figure 14. Compare the F1 score validation of under-sampled data based on logistic regression method and create a heatmap of attribute correlation.

6. Conclusions

A method combining the negative selection process with the evolutionary algorithm is proposed to address the severe bias and data imbalance issues in nearest neighbor classification models for high-dimensional and complex small-sample data. The effectiveness of this method has been verified through experiments. In the data evolution process, the negative selection method is used as an evolution filtering condition to ensure the accuracy of data evolution during over-sampling and under-sampling, while maintaining a certain correlation between parent and offspring generations. Through logistic regression verification, a higher F1 score indicates that the data obtained through the negative evolutionary resampling process not only avoids erroneous predictions as much as possible but also identifies all positive instances as much as possible. Therefore, the over-sampling and under-sampling results obtained through the negative evolutionary resampling process enhance the data. However, this method still has certain limitations:

- (1) The time complexity of the random selection process in the evolutionary method is usually $O(n)$, where n represents the number of individuals in the candidate set. In the selection process of the evolutionary algorithm, each individual has a probability of being selected, which is usually related to its fitness. This process has a complexity of $O(n)$ in data selection. When two parent generations are randomly selected, the time complexity of the random selection process is still proportional to the scale of the number of individuals in the candidate set, resulting in a sharp increase in computational time as the data increases, making the time efficiency of the model in the resampling process inferior to other resampling methods.
- (2) When using probability sampling or random sampling, there is a possibility that the sampled samples may not represent the population being studied, leading to problems related to "sample bias" or "selection bias" in statistics.
- (3) Mathematical models based on biological evolution theory are still a simplified and idealized representation. They are based on specific assumptions and premises and cannot fully cover the complexity of the real world. Therefore, the evolutionary data constraints and related parameters set by the model do not strictly follow the biological evolutionary process, resulting in limitations in the crossover and mutation process of data, which also affects the accuracy of data evolution.

To improve the efficiency of random value selection in the data evolution process, the following improvement directions are proposed for optimization in future experiments:

- (1) In addition to using parallel computing or utilizing GPUs for large-scale data calculations, explore using more feature analysis to obtain global elite genes, reducing the selection process of random values in the data evolution process and improving computational efficiency.
- (2) In determining the conditions for local elite genes, besides using objective weighting methods to determine the relationship between local features, utilize data cleaning methods in multi-label data classification to determine differentiated attributes of independent class labels as local elites for evolutionary operations.
- (3) Employing cloud computing [48] as a foundation for processing complex feature analysis eliminates redundant computations on locally similar data. This enables the establishment of a data analysis workflow, ultimately minimizing the execution time for each analysis.

By implementing these improvements, the computational efficiency of the method can be enhanced, making it more applicable to real-world applications with high-dimensional and imbalanced data. Additionally, we aim to promote the proposed mechanism to address the issue of elite gene selection without considering the domain of the dataset.

We also plan to add more ML algorithms to the process of negation selection so that the mechanism can choose the most suitable algorithm for each scenario. Lastly, one of our main goals is to implement the evaluation of basic attributes of unknown data, similar to the random selection of the parent generation in viruses and enable autonomous replication and growth processes. This process will enhance the applicability and efficiency of data resampling methods across different data types, especially in classification and identification models for medical diagnosis.

Author Contributions: Conceptualization and methodology, L.Q. and T.G.; software, L.Q.; validation, L.Q. and K.J.; formal analysis, L.Q.; investigation, K.J.; resources, T.G.; data curation, L.Q.; writing—original draft preparation, L.Q.; writing—review and editing, L.Q.; visualization, L.Q.; supervision, K.J.; project administration, T.G.; funding acquisition, T.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, No. 61673007.

Data Availability Statement: Data availability is not applicable to this article as no new data were created or analyzed in this study.

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

References

1. Tanouz, D.; Subramanian, R.R.; Eswar, D.; Reddy, G.V.P.; Kumar, A.R.; Praneeth, C.V.N.M. *Credit Card Fraud Detection Using Machine Learning*; IEEE: New York, NY, USA, 2021.
2. Mienye, I.D.; Sun, Y. A Machine Learning Method with Hybrid Feature Selection for Improved Credit Card Fraud Detection. *Appl. Sci.* **2023**, *13*, 7254. [\[CrossRef\]](#)
3. Zou, J.; Zhang, J.; Jiang, P. Credit card fraud detection using autoencoder neural network. *arXiv* **2019**, arXiv:1908.11553.
4. Esenogho, E.; Mienye, I.D.; Swart, T.G.; Aruleba, K.; Obaido, G. A neural network ensemble with feature engineering for improved credit card fraud detection. *IEEE Access* **2022**, *10*, 16400–16407. [\[CrossRef\]](#)
5. Al Balawi, S.; Aljohani, N. Credit-card Fraud Detection System using Neural Networks. *Int. Arab. J. Inf. Technol.* **2023**, *20*, 234–241. [\[CrossRef\]](#)
6. Deogade, K.R.; Thorat, D.B.; Kale, S.V.; Rajput, S.; Kaur, H. Credit Card Fraud Detection using Bagging and Boosting Algorithm. In Proceedings of the 2022 International Conference on Signal and Information Processing (IConSIP), Pune, India, 26–27 August 2022.
7. Han, Y.; Du, P.; Yang, K. FedGBF: An efficient vertical federated learning framework via gradient boosting and bagging. *arXiv* **2022**, arXiv:2204.00976.
8. Ni, L.; Li, J.; Xu, H.; Wang, X.; Zhang, J. Fraud feature boosting mechanism and spiral oversampling balancing technique for credit card fraud detection. *IEEE Trans. Comput. Soc. Syst.* **2023**, 1–16. [\[CrossRef\]](#)
9. Idrees, M.Q.; Naeem, H.; Imran, M.; Batool, A. Identifying Optimal Parameters and Their Impact For Predicting Credit Card Defaulters Using Machine-Learning Algorithms. *Lahore Garrison Univ. Res. J. Comput. Sci. Inf. Technol.* **2022**, *6*, 1–21.

10. Rakhshaninejad, M.; Fathian, M.; Amiri, B.; Yazdanjue, N. An Ensemble-Based Credit Card Fraud Detection Algorithm Using an Efficient Voting Strategy. *Comput. J.* **2022**, *65*, 1998–2015. [\[CrossRef\]](#)
11. Singh, D.; Samadder, J.; Nath, I.; Mitra, N.; Pal, S.R.; Bhattacharyya, S.; Sen, B.K. *Credit Card Fraud Detection Using Soft Computing*; Springer: Berlin/Heidelberg, Germany, 2022. [\[CrossRef\]](#)
12. Mavrogiorgou, A.; Kiourtis, A.; Manias, G.; Kyriazis, D. Adjustable Data Cleaning Towards Extracting Statistical Information. *Stud. Health Technol. Inform.* **2021**, *281*, 1013–1014.
13. Biran, O.; Feder, O.; Moatti, Y.; Kiourtis, A.; Kyriazis, D.; Manias, G.; Mavrogiorgou, A.; Sgouros, N.M.; Barata, M.T.; Oldani, I.; et al. PolicyCLOUD: A prototype of a Cloud Serverless Ecosystem for Policy Analytics. *arXiv* **2022**, arXiv:2201.06077. [\[CrossRef\]](#)
14. Rhee, F.C.H.; Hwang, C. An interval type-2 fuzzy K-nearest neighbor. In Proceedings of the IEEE International Conference on Fuzzy Systems, St. Louis, MO, USA, 25–28 May 2003. [\[CrossRef\]](#)
15. Jaafar, O.; Birregah, B. *KNN-LC: Classification in Unbalanced Datasets Using a KNN-Based Algorithm and Local Centralities*; Springer International Publishing: Cham, Switzerland, 2020.
16. Bao, F.; Wu, Y.; Li, Z.; Li, Y.; Liu, L.; Chen, G. Effect Improved for High-Dimensional and Unbalanced Data Anomaly Detection Model Based on KNN-SMOTE-LSTM. *Complexity* **2020**, *2020*, 9084704. [\[CrossRef\]](#)
17. Ying, S.; Wang, B.; Wang, L.; Li, Q.; Zhao, Y.; Shang, J.; Huang, H.; Cheng, G.; Yang, Z.; Geng, J. An Improved KNN-Based Efficient Log Anomaly Detection Method with Automatically Labeled Samples. *ACM Trans. Knowl. Discov. Data* **2021**, *15*, 1–22. [\[CrossRef\]](#)
18. Han, X.; Ruonan, R. *The Method of Medical Named Entity Recognition Based on Semantic Model and Improved SVM-KNN Algorithm*; IEEE: New York, NY, USA, 2011. [\[CrossRef\]](#)
19. Li, L.; Mao, T.; Huang, D. *Extracting Location Names from Chinese Texts Based on SVM and KNN*; IEEE: New York, NY, USA, 2005.
20. Xu, Q.; Liu, Z. *Automatic Chinese Text Classification Based on NSVMDT-KNN*; IEEE: New York, NY, USA, 2008.
21. Mavrogiorgos, K.; Mavrogiorgou, A.; Kiourtis, A.; Zafeiropoulos, N.; Kleftakis, S.; Kyriazis, D. Automated Rule-Based Data Cleaning Using NLP. In Proceedings of the 31th Conference of Open Innovations Association FRUCT, Helsinki, Finland, 27–29 April 2022; Volume 32, pp. 162–168. [\[CrossRef\]](#)
22. Zhang, A.; Yu, H.; Zhou, S.; Huan, Z.; Yang, X. Instance weighted SMOTE by indirectly exploring the data distribution. *Knowl. Based Syst.* **2022**, *249*, 108919. [\[CrossRef\]](#)
23. Jafarzadegan, M.; Safi-Esfahani, F.; Beheshti, Z. Combining hierarchical clustering approaches using the PCA method. *Expert Syst. Appl.* **2019**, *137*, 1–10. [\[CrossRef\]](#)
24. Zhu, F.; Gao, J.; Yang, J.; Ye, N. Neighborhood linear discriminant analysis. *Pattern Recognit.* **2022**, *123*, 108422. [\[CrossRef\]](#)
25. Hoecker, A.; Kartvelishvili, V. SVD approach to data unfolding. *Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrometers Detect. Assoc. Equip.* **1996**, *372*, 469–481. [\[CrossRef\]](#)
26. Abdi, H. Singular value decomposition (SVD) and generalized singular value decomposition. *Encycl. Meas. Stat.* **2007**, *907*, 912.
27. Arneodo, A.; Grasseau, G.; Holschneider, M. Wavelet transform of multifractals. *Phys. Rev. Lett.* **1988**, *61*, 2281. [\[CrossRef\]](#)
28. Zhang, D.; Zhang, D. Wavelet transform. In *Fundamentals of Image Data Mining: Analysis, Features, Classification and Retrieval*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 35–44.
29. Whig, P. More on Convolution Neural Network CNN. *Int. J. Sustain. Dev. Comput. Sci.* **2022**, *4*. Available online: <https://www.ijscds.com/index.php/ijscds/article/view/80> (accessed on 1 July 2023).
30. Wang, J.; Li, X.; Li, J.; Sun, Q. NGCU: A new RNN model for time-series data prediction. *Big Data Res.* **2022**, *27*, 100296. [\[CrossRef\]](#)
31. Yang, Z.; Xu, B.; Luo, W.; Chen, F. Autoencoder-based representation learning and its application in intelligent fault diagnosis: A review. *Measurement* **2022**, *189*, 110460. [\[CrossRef\]](#)
32. Keller, J.M.; Gray, M.R.; Givens, J.A. A fuzzy K-nearest neighbor algorithm. *IEEE Trans. Syst. Man Cybern.* **2012**, *SMC-15*, 580–585. [\[CrossRef\]](#)
33. Xia, C.; Hsu, W.; Lee, M.L.; Ooi, B.C. BORDER: Efficient Computation of Boundary Points. *IEEE Trans. Knowl. Data Eng.* **2006**, *18*, 289–303.
34. Kumar Sinha, P. Modifying one of the Machine Learning Algorithms kNN to Make it Independent of the Parameter k by Re-defining Neighbor. *Int. J. Math. Sci. Comput.* **2020**, *6*, 12–25. [\[CrossRef\]](#)
35. Czarnowski, I. Weighted Ensemble with one-class Classification and Over-sampling and Instance selection (WECOI): An approach for learning from imbalanced data streams. *J. Comput. Sci.* **2022**, *61*, 101614. [\[CrossRef\]](#)
36. Bach, F. Information theory with kernel methods. *IEEE Trans. Inf. Theory* **2022**, *69*, 752–775. [\[CrossRef\]](#)
37. Rezaei, M.; Montaseri, M.; Mostafaei, S.; Taheri, M. Application of Kernel-Based Learning Algorithms in Survival Analysis: A Systematic Review. **2023**, preprint.
38. Larson, N.B.; Chen, J.; Schaid, D.J. A review of kernel methods for genetic association studies. *Genet. Epidemiol.* **2019**, *43*, 122–136. [\[CrossRef\]](#)
39. Çakmakçı, S.D.; Kemmerich, T.; Ahmed, T.; Baykal, N. Online DDoS attack detection using Mahalanobis distance and Kernel-based learning algorithm. *J. Netw. Comput. Appl.* **2020**, *168*, 102756. [\[CrossRef\]](#)
40. Tang, C.; Zheng, X.; Zhang, W.; Liu, X.; Zhu, X.; Zhu, E. Unsupervised feature selection via multiple graph fusion and feature weight learning. *Inf. Sci.* **2023**, *66*, 152101. [\[CrossRef\]](#)

41. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to Compare: Relation Network for Few-Shot Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017. Available online: https://openaccess.thecvf.com/content_cvpr_2018/html/Sung_Learning_to_Compare_CVPR_2018_paper.html (accessed on 1 July 2023).
42. Campo, D.S.; Mosa, A.; Khudyakov, Y. A Novel Information-Theory-Based Genetic Distance That Approximates Phenotypic Differences. *J. Comput. Biol.* **2023**, *30*, 420–431. [[CrossRef](#)] [[PubMed](#)]
43. Jia, L.; Yang, C.; Song, L.; Cheng, Z.; Li, B. Improved Negative Selection Algorithm and Its Application in Intrusion Detection. *Comput. Sci.* **2021**, *48*, 324–331.
44. Diakoulaki, D.; Mavrotas, G.; Papayannakis, L. Determining objective weights in multiple criteria problems: The critic method. *Comput. Oper. Res.* **1995**, *22*, 763–770. [[CrossRef](#)]
45. Krishnan, A.R.; Kasim, M.M.; Hamid, R.; Ghazali, M.F. A Modified CRITIC Method to Estimate the Objective Weights of Decision Criteria. *Symmetry* **2021**, *13*, 973. [[CrossRef](#)]
46. Pozzolo, A.D.; Boracchi, G.; Caelen, O.; Alippi, C.; Bontempi, G. Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 3784–3797.
47. Pozzolo, A.D.; Caelen, O.; Johnson, R.A.; Bontempi, G. Calibrating Probability with Undersampling for Unbalanced Classification. In Proceedings of the 2015 IEEE Symposium Series on Computational Intelligence (SSCI), Cape Town, South Africa, 7–10 December 2015. [[CrossRef](#)]
48. Karabetian, A.; Kiourtis, A.; Voulgaris, K.; Karamolegkos, P.; Poulakis, Y.; Mavrogiorgou, A.; Kyriazis, D. *An Environmentally Sustainable Dimensioning Workbench towards Dynamic Resource Allocation in Cloud-Computing Environments*; IEEE: New York, NY, USA, 2022. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.