Fraud Feature Boosting Mechanism and Spiral Oversampling Balancing Technique for Credit Card Fraud Detection

Lina Ni, Jufeng Li, Huixin Xu, Xiangbo Wang, and Jinquan Zhang

***Abstract*— With the flourishing of the credit card business and Internet technology, the risk of fraudulent credit card transactions is ever-increasing due to the complex information involved in the credit card business. Since the high redundancy of feature information and imbalance of class distribution in transaction data, the performance of the existing machine learning-based models for detecting credit card fraudulent transactions still needs to be improved. Therefore, it is crucial to build fraud detection models for effective feature engineering and sampling techniques. This article proposes a credit card fraud detection model incorporating a fraud feature-boosting mechanism with a spiral oversampling balancing technique (SOBT). Specifically, we present a compound grouping elimination strategy to exclude highly redundant and correlated features from the credit card transaction dataset and improve the data quality. Furthermore, we design a multifactor synchronous embedding mechanism, which combines the performance evaluation metrics of the embedding model for each feature and improves the decision-making ability of each feature for the target domain. Moreover, we propose an SOBT to balance the ratio of legitimate to fraudulent transactions, which improves the ability of the fraud detection model to distinguish legitimate from fraudulent transactions. Extensive experimental results based on two real-world datasets demonstrate that our methods can facilitate efficient credit card fraud detection and achieve better performance than state-of-the-art methods.**

***Index Terms*— Credit card fraud detection, feature boosting, multifactor embedding, oversampling technique.**

I. Introduction

CREDIT card fraud has become a significant financial security issue with considerable costs for banks and card issuers. Financial institutions try to adopt different security solutions to prevent account abuse. Since fraudsters change their strategies over time, the more sophisticated the security solution, the more sophisticated the fraudster's methods; therefore, it is essential to improve fraud detection approaches and the security modules that attempt to prevent transaction fraud [1]. Fraud detection has become a critical activity in reducing the impact of fraudulent transactions on service delivery, cost-effectiveness, and company reputation.

As is known to all, detecting events in credit card fraud detection is a prediction problem, also known as a data classification problem [2], [3]. The fundamental task of credit card fraud detection is to obtain better feature separability and feature discrimination to ensure the performance and stability of the fraud detection model [4]. A credit card fraud detection method based on machine learning has become an inevitable trend. There are several essential factors in the model training phase of the entire workflow for completing fraud detection: category imbalance, feature redundancy complexity, validation delay, data feature preprocessing, and concept drift [5], [6], [7]. However, credit card fraud detection still faces severe challenges in enhancing the quality of transaction data.

On the one hand, low correlation between features and low redundancy are essential properties of high-quality transaction datasets. Unfortunately, existing methods [8], [9] fail to consider the redundancy of transaction records in credit card transaction scenarios. A large number of redundant features therein will increase the discriminative computational load of fraud detection models, making it difficult for these models to identify fraudulent transactions. Therefore, it is necessary to design an effective feature processing mechanism to improve the quality of credit card transaction data and fraud detection efficiency while reducing the fraud detection model's feature correlation, redundancy, and spatiotemporal complexity.

On the other hand, the subset of features, after removing redundant features, may change the behavior representation of the original transaction data, which cannot maximize the decision capability of each transaction feature for the target domain in credit card fraud detection tasks [10]. For better performance of the fraud detection task, it is essential to adaptively adjust the joint decision capability of each feature on the target domain.

Moreover, due to the unbalancedness of credit card transaction data, it is vulnerable to predicting wrong results by fraud detection models. For traditional machine learning models,

Manuscript received 30 October 2022; revised 11 December 2022 and 23 January 2023; accepted 28 January 2023. This work was supported in part by the National Science Foundation of Shandong Province under Grant ZR2022MF338 and Grant ZR2022LZH014, in part by the Humanity and Social Science Fund of the Ministry of Education under Grant 20YJAZH078 and Grant 20YJAZH127, and in part by the Open Project of Tongji University Embedded System and Service Computing of Ministry of Education of China under Grant ESSCKF2022-02. *(Corresponding author: Jinquan Zhang.)*

Lina Ni is with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China, and also with the Key Laboratory of the Ministry of Education for Embedded System and Service Computing, Tongji University, Shanghai 201804, China (e-mail: nln2004@163.com).

Jufeng Li, Huixin Xu, Xiangbo Wang, and Jinquan Zhang are with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China (e-mail: lijufeng8532@163.com; x15554956758@163.com; w159357xb@163.com; tjzhangjinquan@126.com).

Digital Object Identifier 10.1109/TCSS.2023.3242149

such as logistic regression [11], support vector machines [12], and decision trees [13], they are challenging to identify the difference between legitimate and fraudulent transactions when dealing with imbalanced data. Thus, it is of vital importance to filter out the features subset among all feature data that can represent the original data distribution and effectively distinguish legitimate transactions from fraudulent ones.

Accordingly, to provide services such as intelligent fraud detection to financial institutions, they urgently require a robust credit card fraud detection system that can achieve satisfactory performance in blocking fraudulent transactions. Hence, we introduce a four-layer system model, i.e., input-data processing-predictive classification output, to support transaction identification in financial institutions' credit card fraud detection tasks.

Note that previous work ignored the high redundancy and high similarity among features in credit card fraud detection tasks. To ensure high-quality training data, we apply feature correlation as an indicator to eliminate redundant data in improving data quality, which can judiciously filter out the optimal feature subset to support the predictive classification layer, efficiently improving data quality and reducing the computational load of credit card fraud detection models.

To facilitate the representation capabilities of the optimal feature subset on the fraudulent transaction samples, inspired by genetic algorithms [14], we investigate an embedding and redistribution mechanism to fit the original data distribution [15], readjust each feature's weight, and synchronously combine the multiple guiding metrics of the embedded model on the actual transaction dataset.

Moreover, ensuring high homogeneity and low overlap between the artificial fraud samples and the original fraud transactions is indispensable. Therefore, motivated by the synthetic minority oversampling technique (SMOTE), we formulate an artificial sample generation technique, which not only reduces the overlap rate of artificial samples in the geometric probability model but also dynamically adjusts the distribution ratio of transaction data to improve the ability of the fraud detection model to distinguish legitimate and fraudulent transactions.

The contributions of this article are given as follows.

1) We design a four-layer credit card fraud detection system model incorporating a fraud feature-boosting mechanism with a spiral oversampling balancing technique (SOBT) to facilitate efficient credit card fraud detection.
2) We propose a feature compound grouping elimination algorithm (CGEA) for highly complex and redundant features, which can effectively eliminate redundant features based on missing values and interfeature correlation in the credit card transaction dataset while enhancing the quality of fraud detection data.
3) We present a multifactor synchronous embedding feature-boosting (MSEFBoost) algorithm to address the problem that an individual feature cannot judge the fraud-labeled data. Incorporating an embedded decision tree and a light gradient boost machine model (LightGBM), MSEFBoost evaluates each feature in credit card data for target domain prediction and adaptively fuses multiple evaluation metrics to improve the performance of fraudulent transactions in credit cards.
4) We propose an SOBT combined with a geometric model to balance the ratio of legitimate and fraudulent transactions.

The rest of this article is organized as follows. Section II presents the related work. The preliminary is given in Section III. Section IV presents the proposed methods. The experiment setup and results are introduced in Section V. Section VI concludes this article.

II. Background and Related Work

A. Credit Fraud Detection

Credit card fraud detection based on machine learning has been an important research direction. Supervised [16] and unsupervised [17] learning methods have been widely used. Most current studies consider not only historical data of the user transactions but also necessary data processing in the pre-training phase. Taha and Malebary [18] proposed a LightGBM method for credit card fraud detection. The hyperparameters of LightGBM are tuned using a Bayesian-based optimization algorithm. In addition, Randhawa et al. [19] investigated the performance of some standard machine learning algorithms and hybrid classifiers, including an ensemble classifier based on majority voting. Zhu et al. [20] proposed a dandelion algorithm with probability-based mutation to achieve better classification performance in credit card fraud detection tasks.

Bahnsen et al. [10] analyzed the transaction time based on the von Mises distribution for cyclical behavior and created a new set of features. Although the feature-derived approach improves the model accuracy, it is slow in efficiency. In application fraud detection scenarios with fast information transfer, fraud detection models pose a severe challenge to the detection of spatiotemporal complexity.

Elyan et al. [21] proposed a new hybrid approach to balance the dataset using class decomposition to reduce the advantage of majority class instances. Bunkhumpornpat and Sinapiromsaran [22] proposed a density-based majority undersampling technique (DBMUTE) to eliminate negative models in overlapping regions. Devi et al. [23] proposed an updating method based on T-Link, which aims to remove noise and redundant instances from overlapping areas to balance the class distribution in the data. However, the most significant disadvantage of undersampling is that a large amount of feature information is lost during the undersampling process, which may increase the difficulty of fraud detection models distinguishing legitimate transactions from fraudulent ones.

The problem of imbalanced distribution of data classes in credit card fraud detection datasets exists throughout the fraud detection field. Although imbalanced data are not the only problem hindering fraud detection, it is an unavoidable issue in the fraud detection domain. The performance of most machine learning algorithms may be biased toward the legitimate transaction class, so more fraudulent samples are misclassified than legitimate transaction samples [24], resulting in enormous financial loss.

B. Feature Engineering

Feature selection is an effective method in risk management to improving the performance of fraud detection models. However, in this field, finding an optimal subset of features in high-dimensional data is difficult.

Zheng et al. [9] found that some old transaction data may be outdated and cannot predict the future transaction behavior of users using credit card transaction datasets. The distribution of transaction data changes over time, resulting in the distribution of transaction data during this time not following the distribution of data in the latter period [25]. However, the predictive effect on the target domain is not entirely lost. They considered how to increase the impact of source domain instances on predicting the target domain data and proposed ITrAdaBoost [25], which can effectively solve the concept drift problem by adjusting the data distribution of the source and target domains through filtering.

Therefore, one of the critical tasks of feature engineering is how to retain valuable data or explore the hidden information at different data levels [26] before training the model to bring into play the potential value of historical data existing for model training.

C. Resampling and Ensemble Learning Model

The predictive performance of machine learning models depends on the quality of the data used to train the models. Islam et al. [27] proposed the K-nearest neighbor oversampling (KNNOR) technique to identify critical and safe regions and generate synthetic sample points for minority instances. Chen et al. [28] proposed a random search-synthetic minority oversampling technique-gradient boosting tree (RS-SMOTE-GBT) algorithm to outperform other machine learning algorithm models in linear and nonlinear classification spaces. Jo and Kim [29] used generative adversarial networks with minority-class oversampling in the boundary region to effectively capture the features of the minority-class data. Yi et al. [30] proposed an adaptive synthesizer neighbor SMOTE (ASN-SMOTE) to filter the noise in the minority class by determining whether the nearest neighbor of each minority instance belongs to the minority or majority class. ASN-SMOTE adaptively selects qualified minority instances for each minority instance by the adaptive neighbor selection scheme to synthesize new minority instances.

Wang and Sun [31] used ensemble learning to solve the problem of getting classification anomalies due to imbalanced data. Wang et al. [32] developed a LightGBM model to predict the financial risk profile of 186 firms. The ensemble learning model exhibited better prediction results than the decision tree and K-nearest neighbor algorithms. Yang et al. [33] proposed a clustering algorithm to dynamically determine the noise points during the ensemble learning iterations while flexibly adjusting the feature weights according to the misclassification rate of the noise points.

In conclusion, more balanced feature processing strategies are needed to apply the potential value in historical data efficiently. Our proposed compound group elimination strategy and multifactor synchronous embedding mechanism can effectively filter out the optimal feature subset while adjusting the decision power of features on the target domain. In addition, due to the high imbalance of credit card transaction data, in this article, we use an SOBT to solve the problem of a high imbalance of legitimate and fraudulent transaction data. Furthermore, we adopt the ensemble learning classifier LightGBM as the final fraud detection model to effectively predict the transaction attributes of a large amount of transaction data.

III. Preliminary

In this section, we conduct an exploratory analysis of the actual transaction data and describe our motivation for the idea of feature engineering and imbalanced data to present our four-layer credit card fraud detection system model.

A. Feature Analysis of Transaction Data

It is worth noting that each feature in the real-world credit card transaction dataset contains the necessary information. However, due to the extensiveness and uniformity of information in the transaction data, which has not only the personal information of cardholders but also the information on various transaction goods, the presence of similar users and similar interests may lead to the prevalence of a large amount of redundant data [34]. There is an equivalent missing distribution and high similarity between the derived and original features. It means that the derived features do not apply to all credit card fraud detection tasks.

It is a normal phenomenon in financial transactions that the number of legitimate transactions is much larger than that of fraudulent transactions. To further distinguish the difference in the distribution of legitimate and fraudulent transaction data and facilitate fraud detection models to more easily distinguish legitimate from fraudulent transactions [35], which means that it is not enough for us to enhance the data in terms of feature engineering, we also need to fit the fraudulent transaction data distribution and generate artificial fraud samples. Thus, we can draw the following conclusions from the credit card fraud transaction distribution.

1) There are different degrees of heterogeneity between legitimate and fraudulent transactions in user transaction data.
2) Eliminating redundant features can effectively enhance data quality.
3) Artificial samples maintain substantial homogeneity between fraudulent transaction data and heterogeneity with legitimate transaction data.
4) It is essential to avoid creating data overlap and boundary blurring.

Therefore, our research necessity focuses on dealing with redundant feature data and reassigning the feature weights on the new dataset without the redundant data. We adopt an oversampling technique to generate a balanced dataset of fraudulent samples for imbalanced datasets to maximize the distinction between legitimate and fraudulent transactions and to learn fraudulent transaction features more accurately.

B. Feature Engineering

Feature engineering converts the original data into optimal features suitable for prediction models [5]. The compound grouping elimination strategy improves the prediction accuracy of anonymous transaction data by eliminating features with a high percentage of missing and high correlation.

We propose a multifactor synchronous embedding mechanism to fit the original data distribution. This mechanism embeds the decision tree and LightGBM models [32] into the fraud detection model and trains the predictive ability of single features on the target domain. It finally introduces the three indicators of area under the ROC curve (AUC), MSE, and the feature importance of the embedding model test data, adaptively fuses them into metamorphic factors, and embeds them into the final fraud detection model to improve the fraud detection accuracy.

C. Resampling

Resampling techniques are used to rebalance the sample space of an imbalanced dataset to mitigate the effect of class distribution during the learning process. They are more general than other techniques because they are independent of the selected fraud detection model and can construct more complimentary samples for fraud detection models.

Oversampling techniques eliminate the hazards of skewed distributions by creating new minority-class samples. Two widely used methods for creating synthetic minority samples are random replication of minority-class samples and SMOTE [36]. The basic idea of SMOTE is to analyze the minority-class samples and add to the dataset based on a small number of synthetic samples, which can be defined as

$$X_{\text{new}} = X + \text{rand}(0, 1) * (X_{\text{old}} - X) \tag{1}$$

where $X_{\text{old}} - X$ is the line that identifies the fraudulent transaction X_{old} in the original data with one of its neighbors X, and then, the artificial sample points X_{new} are selected by linear random.

Many variants of SMOTE, such as adaptive synthetic sampling (ADASYN) [37] and borderline SMOTE (BorderlineSMOTE) [38], work to create better artificial samples [39] that fit the original data distribution.

Considering the existing sampling techniques, most researchers used a synthesis of fraud samples in linear space to balance the transaction dataset. To reduce the overlap rate [40] and solve the problem of homogeneity and heterogeneity of the sample, we propose an SOBT to bridge the imbalance of transaction data ratio. The original data are placed in a 3-D space and combined with clustering unsupervised classification methods to synthesize 3-D artificial samples on the spiral line, which are finally mapped to the original sample space to balance the dataset.

D. Predictive Classification Model

Before LightGBM [41] was proposed, XGBoost [42] was the most commonly used gradient boosting decision tree (GBDT) algorithm model. Tian and Liu [43] proposed a black-box explanation method to capture the subtle and implicit cross features hidden in fraud behaviors through deep learning models and GBDT models to uncover why transactions are considered fraudulent. However, due to the considerable time and space overhead in finding data splitting points using a sorting algorithm, each iteration of computation is not friendly to cache optimization. In contrast, LightGBM can optimize the GBDT algorithm by relying on its unique histogram algorithm, one-sided gradient sampling algorithm, mutually exclusive feature bundling, and depth-limited leaf growth strategy [44], which can significantly reduce the time overhead and decrease the memory footprint while supporting category features with efficient parallelism.

The massive size of the credit card fraud detection dataset poses a more significant challenge to all fraud detection models. In the face of the massive amount of data in credit card transactions, ordinary GBDT algorithms cannot meet the demand. LightGBM can effectively solve the problem of GBDT training efficiency under high data volume in credit card transactions in order to support large data volume and ensure training efficiency during real-time transaction data transmission.

E. System Model

We design a four-layer credit card fraud detection system model shown in Fig. 1. It can be seen that our system model consists of an input layer, a data progressing layer, a predictive classification layer, and an output layer. The basic principle of the model is given as follows.

1) Input Layer: After exploratory analysis of the original input dataset, processing perspectives to enhance the data quality of this dataset are determined. We start the study from two perspectives: handling redundant data and balancing legitimate and fraudulent transactions.

2) Data Processing Layer: The following conditions hold.

1) Redundant information is processed to improve data quality by screening redundant features through a compound grouping elimination strategy and a multifactor synchronous embedding mechanism in the data processing layer.
2) The number of legitimate and fraudulent transactions is balanced by an SOBT, which enhances the training learning effect of the predictive classification layer on fraudulent samples.

3) Predictive Classification Layer: The predictive classification layer takes the output results of the data processing layer as input and divides the training and test sets on balanced transaction data. The LightGBM classifier can efficiently learn the difference between legitimate and fraudulent transactions and thus can effectively identify fraudulent transactions.

4) Output Layer: The output results represent the predicted labels of the test data, and we compare their actual labels to evaluate our proposed fraud detection model by combining six machine learning metrics [45], and this model achieves a high fraud detection rate.

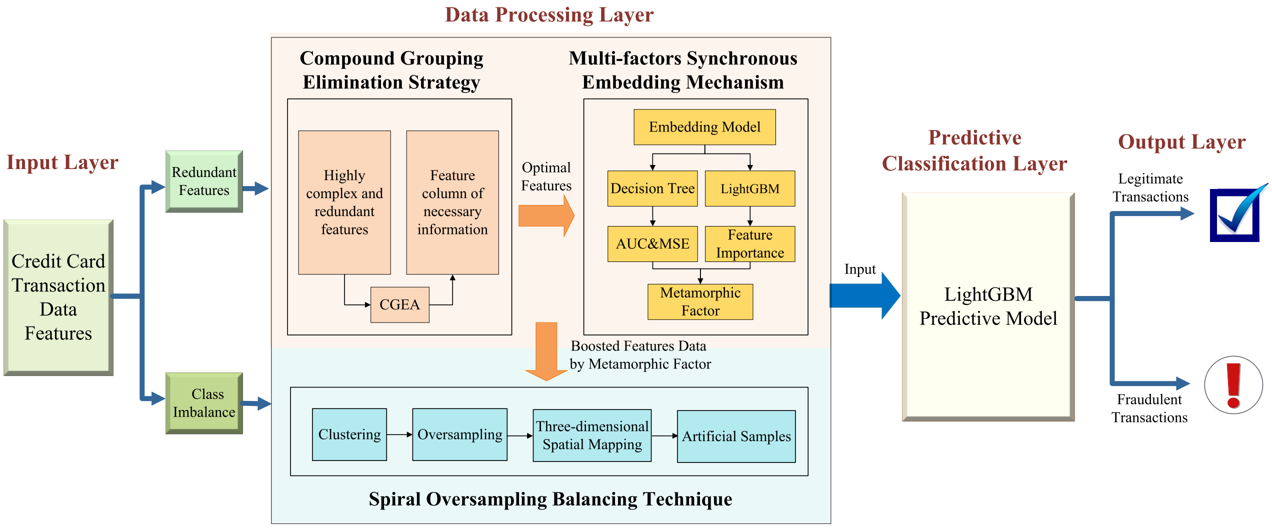

Data Processing Layer
Input Layer
Credit Card Transaction Data Features
Redundant Features
Class Imbalance
Compound Grouping Elimination Strategy
Highly complex and redundant features
Feature column of necessary information
CGEA
Optimal Features
Multi-factors Synchronous Embedding Mechanism
Embedding Model
Decision Tree
LightGBM
AUC&MSE
Feature Importance
Metamorphic Factor
Boosted Features Data by Metamorphic Factor
Clustering
Oversampling
Three-dimensional Spatial Mapping
Artificial Samples
Spiral Oversampling Balancing Technique
Input
Predictive Classification Layer
LightGBM Predictive Model
Output Layer
Legitimate Transactions
Fraudulent Transactions

Fig. 1. System model.

Overall, we have improved data quality in two directions: feature engineering and balanced samples, which improves the fraud detection model's ability to distinguish legitimate from fraudulent transactions and enhances the fraud detection accuracy.

IV. Proposed Method

After analyzing the user transaction data and confirming the perspective of designing our fraud detection model, we propose a fraud feature-boosting mechanism and SOBT to enhance the data quality and improve the fraud detection performance.

A. Data Processing Sequence

There are many problems when building a credit card fraud detection model using existing transaction data. For example, avoiding the impact of redundant features in all credit card features and extracting the optimal subset of features in the feature space while ensuring that the original data distribution [3] can be mapped are the problems we need to solve. Conversely, effectively removing irrelevant, redundant features and reducing the risk of overfitting are both major tasks of feature engineering [33]. The data processing sequence of this article is shown in Fig. 2. As can be seen from Fig. 2, the original fraud transaction dataset is filtered by CGEA, and features with high interfeature similarity and low data variety are eliminated to reduce the redundancy of the original data.

The MSEFBoost algorithm readjusts the training weights of features on the prediction of the target domain and decreases the proportion of decisions of features with high feature importance while increasing the proportion of features with low feature importance [5]. Each feature predicts the target domain based on its metamorphic factor, which reduces the major decision of some features on the target domain prediction.

The SOBT generates artificial fraud samples within each cluster in the 3-D space to balance the dataset and reduce the probability that the fraud detection model results are biased toward legitimate transactions due to data imbalance.

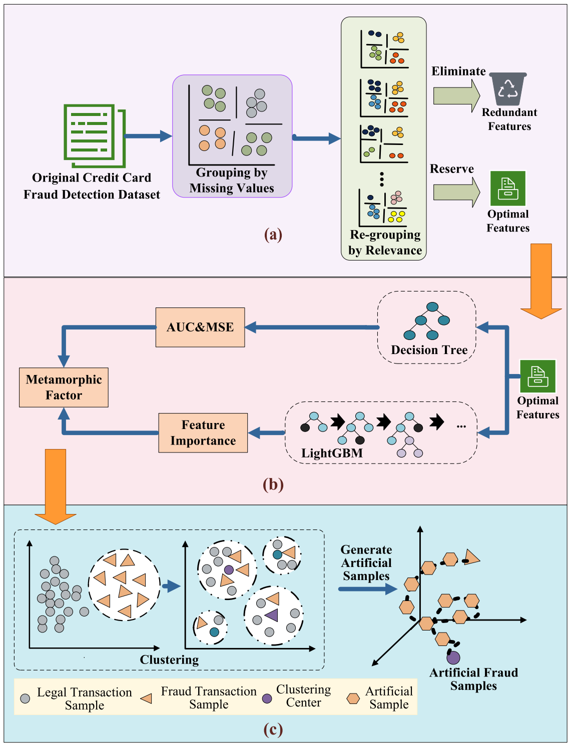


Fig. 2. Illustration of data processing sequence. (a) Compound grouping elimination algorithm based on Pearson correlation coefficient. (b) Multifactor synchronous embedding strategy. (c) Spiral oversampling balancing technique.

In subsequent experiments, CGEA, MSEFBoost, and SOBT successfully solve the feature redundancy and distribution imbalance of the dataset in the credit card fraud transaction domain and improve the classification performance of fraud prediction.

B. Compound Grouping Elimination Strategy

Usually, we do not remove the data columns in the dataset, and most of the features are correlated with some extent with the labeled features. Xie et al. [5] showed that the derived columns usually add essential information as well, which brings another problem that too many derived features tend

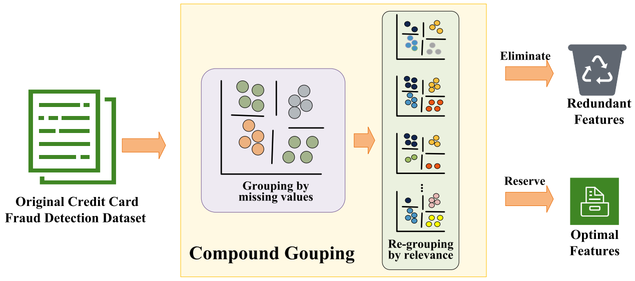


Fig. 3. CGEA framework.

to cause feature redundancy and increase the burden of the fraud detection model.

For highly redundant feature columns, we propose a CGEA based on Pearson correlation coefficients. The algorithm framework is shown in Fig. 3. As can be seen from Fig. 3, CGEA divides the first grouping according to the default value case, followed by a correlation group for feature columns whose relevance exceeds a specific threshold.

1) Grouping According to Feature Missing Values: Some features in the credit card transaction dataset are derived from the original features and have the same data distribution as the original features. The missing value distribution is also the same, so we group them according to the missing values for the first time.

2) Regrouping According to Feature Correlation: We adopt the Pearson correlation coefficient to reflect the degree of linear correlation between features in the fraud transaction dataset. Its calculation involves two statistics: covariance and standard deviation, which are defined as

$$\mathrm{Cov}(x, y) = \frac{\sum_{i=1}^{n} (x_i - \mu_x)(y_i - \mu_y)}{n-1} \tag{2}$$

where u_x and u_y denote the mean values of the random variables x and y, respectively. The covariance reflects the degree of correlation [43] between the two variables. Since there are often outliers in the fraud transaction data or when the dispersion of the transaction records changes, it may affect the value of the covariance. In order to portray the dispersion of the data better, as well as to standardize the data, both covariance and standard deviation are introduced to define the correlation [46], i.e., the Pearson correlation coefficient, which is defined as

$$\begin{aligned} \rho_{xy} &= \mathrm{corr}(x, y) = \frac{\mathrm{Cov}(x, y)}{\delta_x \delta_y} \\ \delta_x &= \sqrt{\frac{\sum_{i=1}^{n} (x_i - \mu_x)^2}{n-1}} \\ \delta_y &= \sqrt{\frac{\sum_{i=1}^{n} (y_i - \mu_y)^2}{n-1}}. \end{aligned} \tag{3}$$

The simplified form is described as

$$\rho_{xy} = \frac{\sum_{i=1}^{n} (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n} (x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n} (y_i - \mu_y)^2}}. \tag{4}$$

For feature columns with low correlation with each other feature, this situation indicates that there is no feature column with its redundancy, so we keep the features in independent groups. The diversity of data types of fraudulent transactions tends to correlate positively with the information they contain. For correlation groups consisting of more than one feature, we keep the feature columns with more variety of feature column data in the group.

3) Algorithm Framework: Fig. 3 shows the specific process of two groupings. Correlation grouping is a secondary grouping that continues on the result of missing value grouping, and the grouped data have equal missing values and higher similarity. It is possible to set a specific relevance threshold in the relevance grouping result to eliminate redundant features and retain features with low correlation and a large variety of data.

4) Algorithm Implementation: CGEA will record the distribution of missing values when traversing the feature data in the first grouping and record the data type of the feature in the second grouping. The pseudocode of CGEA is elaborated in Algorithm 1.

Algorithm 1 CGEA

```
Input: Imbalanced data with the original features $D_s$;
Output: Subset of features after eliminating redundant
    features $F_{res}$;
$F_m$ ← statsMV($D_s$);
$F_c$ ← corrAna($D_s$);
$G_m$, $G_c$ ← cpdGroup($F_m$, $F_c$);
$N^f_{amt}$, $N^f_{\max}$ ← Count the amount and maximum of data
    contained in a single feature;
for i ← 1 to len($G_m$) do
    for j ← 1 to len($G_c$) do
        $N^f_{amt}$ ← Update the number of data in each feature
            in $D_s$;
        if $N^f_{amt} \geq N^f_{\max}$ then
            $N^f_{\max}$ ← Update the current data amount $N^f_{amt}$
                to the maximum data amount $N^f_{\max}$;
            $F_{res}$ ← Preserve features;
        end if
    end for
end for
return $F_{res}$.
```

In CGEA, functions *statsMV* and *corrAna* represent missing value statistics and correlation analysis of fraudulent transaction datasets, respectively. The function *cpdGroup* performs sequential grouping based on missing value cases and correlation analysis results. Finally, when traversing the results of the compound grouping, features with high redundancy, high relevance, and low data variety are eliminated.

C. Fraud Feature-Boosting Mechanism

For all the credit card fraud detection classification methods, most of them can identify fraudulent transactions hidden in similar user transactions [47]. However, there are some drawbacks among them, which cannot show optimal classification when faced with a class-imbalanced dataset. Thus, it is essential to execute feature engineering before fraud detection.

Some studies directly use original transaction data. However, there is usually a very high similarity in the actual transaction data since data collected during credit card transactions must comply with the international financial reporting standards [48].

To solve this problem, some feature engineering techniques have been studied in this regard. In this article, we propose the MSEFBoost algorithm. Univariate feature selection can rely on machine learning algorithms and models to train and test each feature to obtain the weight coefficients of each feature, indicating the contribution of that feature to the prediction of the target variable. Since single attribute values cannot derive a subset of features with maximum discriminative power, identifying a subgroup with full discriminatory ability requires multifactor feature engineering [49].

1) Multifactor Indexes: AUC indicates the performance of the fraud detection model, the higher the AUC score, the greater the prediction of the target domain. MSE reflects the degree of difference between the predicted and actual values. Feature importance refers to the percentage of decisions played by the feature for the prediction of the target domain, i.e., the contribution to the target prediction.

2) Synchronous Embedding: Before all training data are input to the final fraud detection model for predictive classification, the embedded decision tree model and the LightGBM model based on the MSEFBoost algorithm train the CGEA output synchronously. AUC scores and MSE of each feature are obtained in the decision tree model; the feature importance of each feature is received on the LightGBM model.

3) Metamorphic Factor: Inspired by genetic algorithms, we present the concept of a metamorphic factor in fraud feature-boosting mechanism to describe the contribution of each feature to the target domain, adjusting features' weight in the adaptive fusion process.

The metamorphic factor of each feature is calculated according to the AUC, MSE, and feature importance, which is defined as follows:

$$\mathrm{meFacs}_i = \frac{\mathrm{IMP}_i}{\sum_{i\in X\cdot\mathrm{columns}} \mathrm{IMP}_i} + (\mathrm{AUC}_i - \mathrm{MSE}_i)^{-1} \tag{5}$$

where IMP_i denotes the feature importance of each feature in the embedded LightGBM learner, AUC_i is the ROC_AUC score of a single feature in the decision tree model for the target domain prediction, MSE_i is the mean squared error of a single feature in the decision tree model for the target domain, and $X \cdot$ columns refers to all features in the credit card data.

4) Algorithm Theoretical Basis: The metamorphic factor in the MSEFBoost algorithm is inspired by the fitness function in the genetic algorithm [14]. The metamorphic factor is defined in which we fit the feature importance of a single feature to the fitness function. At this time, the range of $(\mathrm{IMP}_i/(\sum_{i\in X\cdot\mathrm{columns}} \mathrm{IMP}_i))$ values is mapped between 0 and 1.

AUC score reflects the degree of importance of the feature to the target domain prediction from the perspective of judging correctness. MSE demonstrates that the deviation range from the judging discrepancy perspective reflects the content of bias generated when the feature is predicted for the target domain.

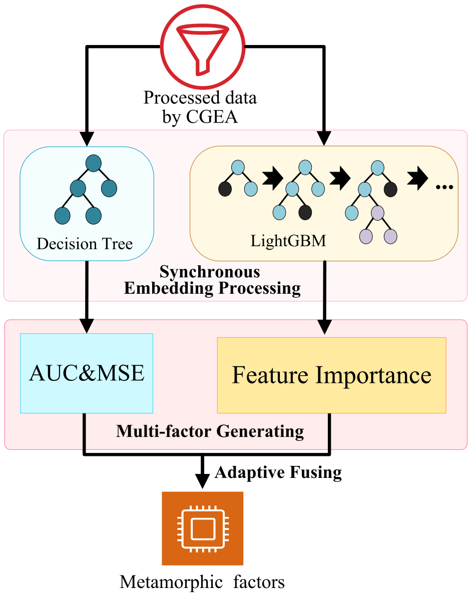


Fig. 4. MSEFBoost algorithm framework.

To adjust the contribution of each feature to the prediction of the target domain, it is experimentally confirmed that when the difference between AUC score and MSE is taken as the inverse of the fit feature importance, the prediction weight of features with solid importance to the target domain can be weakened. The prediction weight of features with low importance to the target domain can be enhanced. The weights of the features are rebalanced, resulting in better fraud detection results.

5) Algorithm Framework: The algorithm framework is shown in Fig. 4. The MSEFBoost algorithm takes the feature data with low redundancy retained by CGEA output as input. After the embedding model is trained on the input feature data, each feature's AUC, MSE, and feature importance are obtained, and the metamorphic factor is fused adaptively and embedded into the XGBoost classifier training process. The feature weights readjusted by MSEFBoost can be effectively applied to the fraud detection model, differentially increasing the weight percentage of each feature in predicting the target domain. MSEFBoost still relies on the complete feature set, and the final prediction effect will be more accurate to the fraud detection model.

6) Algorithm Implementation: MSEFBoost algorithm introduces AUC and MSE, combining feature importance to obtain a reweighting factor, the metamorphic factor, to improve the weights of sample features to different degrees. We have conducted extensive experiments on the XGBClassifier [50] and compared the ability of the classifier to distinguish between transaction types before and after setting the metamorphic factor. The pseudocode of MSEFBoost is elaborated in Algorithm 2.

In MSEFBoost, functions Roc_Auc_Score, Mean_Squared _Error, and PredictedByDTR are to analyze the AUC and MSE of each feature on the decision tree regressor embedding model based on the feature data output from Algorithm 1. Then, the predictions are trained on LightGBM to obtain the

Algorithm 2 MSEFBoost

```
Input: Data after removing redundant features $F_{res}$;
Output: Adjusted feature metamorphosis factor $meFacs$;
Train data on decision tree regression;
$y_{pred} \leftarrow$ PredictedByDTR($F_{res}$);
$V_{AUC} \leftarrow$ Roc_Auc_Score($y_{test}, y_{pred}$);
$V_{MSE} \leftarrow$ Mean_Squared_Error($y_{test}, y_{pred}$);
$V_{AMsub} \leftarrow V_{AUC} - V_{MSE}$;
Train data on LightGBM classifier;
for i, (t_index, v_index) in ($X_{train}, y_{train}$) do
    $F_{imp} \leftarrow$ Output feature importance of training dataset on LightGBM classifier;
end for
for i $\leftarrow$ 1 to len($V_{AMsub}$) do
    $meFacs_i = \frac{IMP_i}{\sum_{i \in X.columns} IMP_i} + (AUC_i - MSE_i)^{-1}$;
end for
return $meFacs$.
```

individual feature importance F_{imp}. Finally, AUC, MSE, and feature importance F_{imp} are adaptively fused into metamorphic factors $meFacs$ to control the feature weights in the fraud detection model.

D. *Spiral Oversampling Balancing Technique*

Random oversampling-based technique, which duplicates some samples, increases the number of minority-class samples but cannot well extract the samples nearby the cross edge of majority and minority classes due to its randomness [51]. SMOTE-based oversampling techniques tend to expand the data within a few class samples based on some rules. These methods overcome the drawback of repeated sampling but still cannot reduce the overlap rate of artificial samples.

We want to generate artificial samples that maintain the original attributes and increase the proportion of fraudulent transaction samples in the space with minor overlap. SOBT combines the sample space with the spiral line model in 3-D space to minimize the linear overlap of fraudulent transaction data.

1) Clustering: In credit card fraud detection, it is difficult to classify some feature sets into specific behavioral patterns. Clustering methods can automatically organize the unsupervised learning of high-level abstract knowledge so that the original ambiguous fraudulent behavior patterns can become clear before this balanced sampling [52]. SOBT uses KMeans++ clustering [53], where the more distant points from the current cluster center have a higher probability of being selected as another cluster center. It makes the heterogeneity [54] between fraudulent transactions in different classes more robust and maximizes the homogeneity of fraudulent transactions between classes.

2) Constructing a 3-D Spiral Model: In each clustering result, we construct a spiral curve starting with the cluster center and ending with the fraudulent sample points within the cluster until we have traversed all fraudulent sample points. The number of artificial samples generated in the cluster is determined by the proportion of fraudulent transactions within the cluster to the total number of fraudulent transactions. The 3-D model illustration of SOBT artificial sample generation is shown in Fig. 5.

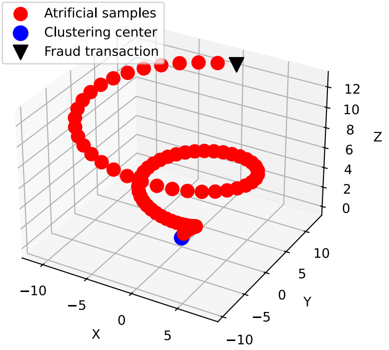


Fig. 5. Illustration of SOBT artificial sample generation.

3) Generating Artificial Samples: The SOBT takes the datasets feature labels and classification labels as input to the algorithm and calculates the difference between the number of legitimate and fraudulent transactions in the training set, i.e., the number of artificial samples to be generated, namely,

$$N_{\text{sample}} = X_{\text{maj}} - X_{\text{min}} \tag{6}$$

where X_{maj} is the number of legitimate transactions and X_{min} is the number of fraudulent transactions.

With the help of the cluster centers and the fraudulent transaction data in each cluster, we calculate the percentage of fraudulent transaction data and the number of artificial samples that need to be generated, i.e.,

$$\text{Min}_{\text{prop}} = \frac{C_{\text{min}}}{X_{\text{min}}} \tag{7}$$

$$\text{NC}_{\text{count}} = \text{Min}_{\text{prop}} * N_{\text{sample}} \tag{8}$$

where C_{min} is the number of fraudulent transactions in a single cluster.

To maintain the homogeneity of fraudulent transactions in the original sample, we transform the spatial 3-D coordinate data into the same dimensionality as the original data points by 3-D random weights. The transformation method is defined as

$$\text{artiSam}_i = \frac{\sum_{i \in \text{artiSam}} \text{rand}(0, 1) * (X_i + Y_i + Z_i)}{3} \tag{9}$$

where X_i, Y_i, and Z_i are the coordinates of the artificial sample points in the 3-D space.

4) Theoretical Basis: SMOTE determines a minority-class sample and then randomly selects one of its K-nearest neighbors (generally, K is taken as 5) and then randomly chooses a location on the linear space of the two as an artificial sample point whose probability expression can be summarized as $((\text{rand}(0, 1))/K)$.

SOBT selects one of the minority classes in all clusters, constructs artificial samples in the 3-D space and finally maps to the linear space. The probability expression of SOBT to produce synthetic samples can be summarized as $\text{Min}_{\text{count}}^{-1} * \text{rand}^3(0, 1)$. Its probability distribution is much smaller than that in SMOTE, which can reduce the artificial sample overlap rate.

5) Algorithm Implementation: Algorithm 3 demonstrates the entire process of SOBT, where we employ KMeans++ clustering for initial unsupervised classification and obtain the clustering labels C_{tag} and cluster centers C_{cen}. The proportion of the number of internal fraudulent transactions to the total number of fraudulent transactions $\text{Min}^i_{\text{prop}}$ is counted within each cluster. While the number of artificial samples to be generated for a single cluster $\text{NC}^i_{\text{count}}$ is derived from the percentage of fraud data $\text{Min}^i_{\text{prop}}$ and the total number of samples to be generated N_{sample}.

Algorithm 3 SOBT

```
Input: Imbalanced training set data $X^{imb}_{train}$;
Output: Original data and artificial samples $Sample_{re}$;
$C_{cen}$, $C_{tag}$ ← KMeans++($X^{imb}_{train}$, KMeans args);
for i ← 1 to range($C_{tag}$) do
    $Min^i_{prop}$ ← Divide all minority classes into several clusters;
    $NC^i_{count}$ ← $Min_{prop} * N_{sample}$;
    if $Min^i_{prop} == 0$ then
        Delete $C^i_{cen}$, $C^i_{tag}$;
        Continue;
    end if
end for
for j ← 1 to $C_{tag}$ do
    for k ← 1 to $Min_{count}$ do
        $N^j_{gen}$ ← $NC^i_{count}$ / $Min^j_{count}$;
        $Z_k$ ← Construct spiral curves starting with $C^j_{cen}$ and ending with the fraudulent sample point;
        $X_k$ ← $\sin(Z_k) * Z_k$;
        $Y_k$ ← $\cos(Z_k) * Z_k$;
    end for
end for
$Arti_{sam}$ ← Iterate X, Y, Z and return the list of artificial data samples;
$Sample_{re}$ ← Map 3D data to linear space using $rand(0, 1) * Arti_{sam}$;
return $Sample_{re}$.
```

Then, each fraud sample data within a cluster needs to be combined with the cluster center C^j_{cen} to generate N^j_{gen} artificial fraud data Arti_{sam}. Finally, by randomly mapping to the same dimension as the original data and combining with the original data, SOBT forms the balanced data $\text{Sample}_{\text{re}}$.

V. Experiment

In this section, we conduct experiments on the more classical datasets Creditcard [55] and IEEE-CIS Fraud Detection [56] in the credit card transaction domain to evaluate the performance of our proposed CGEA, MSEFBoost, and SOBT. The experimental flow design is shown in Fig. 6.

We compare the fraud detection effect of the same fraud detection model LightGBM [41] before and after the MSEFBoost algorithm. SOBT compares SMOTE [36], ADASYN [37], and BorderlineSmote [38] sampling techniques and does a comparative analysis with other methods in the same field of study. The experimental results show that our proposed methods significantly improve the credit card fraud detection rate.

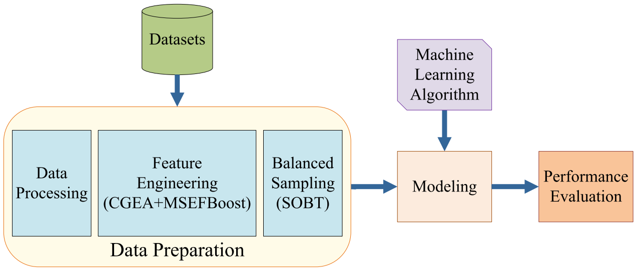
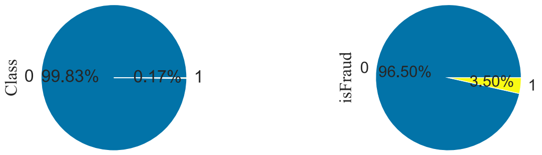


Fig. 6. Experimental flowchart.



(a) (b)

Fig. 7. Imbalance ratio of the dataset. (a) Creditcard dataset label ratio. (b) IEEE-CIS Fraud Detection dataset label ratio.

A. Datasets

1) CreditCard Dataset: The data source is the dataset CreditCard from the Kaggle website [55]. This dataset is the transactions made by European cardholders via credit cards in two days in September 2013. A total of 284 807 transactions are recorded in this dataset, with 492 fraudulent transactions, which represents approximately 0.17% of the total data. The data label ratio is shown in Fig. 7(a). The extremely imbalanced ratio significantly impacts the classification effect of the fraud detection model, which is highly susceptible to bias toward the legitimate transaction class data and thus cannot achieve the effect of accurately predicting fraudulent transactions. There are no missing values for feature labels and classification labels in this dataset, so there is no need to do feature filtering and feature removal.

2) IEEE-CIS Fraud Detection Dataset: IEEE-CIS Fraud Detection dataset is the actual e-commerce transactions data from September to December 2017 from Vesta [56]. It contains data on a variety of features from product codes, payment card information, address information, device types, product functionalities, and other various features. The dataset contains two tables: the transaction table and the identity table, and the combined dataset contains 590 540 transaction records and 435 data features. The 435 data items contain both category-based and numeric data. The data label ratio is shown in Fig. 7(b). The features with 99% missing values in this dataset are removed to facilitate our subsequent processing.

B. Performance Evaluation Metrics

The field of fraud detection research often employs several evaluation metrics to address the learning problem of the imbalanced datasets. In this work, we use recall and F1-score, as well as AUC, as the primary performance evaluation metrics.

AUC score is the area under the ROC curve. It is created by plotting the relationship between the true positive rate (TPR) and the false positive rate (FPR) for different threshold settings. The formulas for TPR and FPR are defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{10}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \tag{11}$$

where TP is the accurate positive prediction, FN is the false negative prediction, FP is the false positive prediction, and TN is the true negative prediction.

Accuracy refers to the proportion of successfully classified results to the total samples, which can be calculated as

$$\text{accuracy} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}. \tag{12}$$

Precision refers to the percentage of samples that are actually fraudulent transactions to total fraudulent transaction samples. The formula for this is denoted as

$$\text{precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \tag{13}$$

Recall is the ratio of the amount of data correctly predicted to be fraudulent transactions to all actual fraudulent transactions. In general, the cost loss of not detecting a fraudulent transaction is greater than that of judging a legitimate transaction as fraudulent. Thus, we require a higher recall rate than anything else between the accuracy rate and the recall rate. The formula for this is denoted as

$$\text{recall} = \text{sensitivity} = \frac{\mathrm{TP}}{P}. \tag{14}$$

F1-Score is the weighted summed average of precision and recall. The formula for this is denoted as

$$F1 = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{15}$$

F1-score considers both the precision and recall factors and achieves a reconciliation of them. It is noting that using F1-score as an evaluation metric can avoid extreme cases [57].

G-mean is a combination of positive and negative case accuracy. According to previous studies, in order to obtain as much information as possible about the magnitude of each category's contribution to the final performance and to consider the imbalance rate of the data, many researchers have tried to investigate new evaluation metrics in the field of imbalance fraud detection, such as the tuned G-mean. The formula for this is denoted as

$$\text{G-mean} = \sqrt{\text{recall} * \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}}. \tag{16}$$

C. Experimental Setup

1) Preprocessing: The data features of the Creditcard dataset are disclosed as $V1, V2, \ldots, V28$, which are the principal components obtained after principal component analysis (PCA) [58]. Three features, time, amount, and class, are not transformed by PCA. Xie et al. [59] adopted timestamps to explore the differences in cardholders' transaction behavior between legitimate and fraudulent transactions. The long- and short-term transaction behavior habits of fraudsters are summarized. The values of the time and amount columns need to be feature scaled because of their significant differences in magnitude from other features after PCA.

The IEEE-CIS Fraud Detection dataset contains a wide range of features from device types to product functionalities. The features also contain category and numerical variable features. We encode the category features using weight of evidence (WOE) coding [60], which is defined as

$$\mathrm{WOE}_i = \ln\frac{py_i}{pn_i} = \ln\frac{y_i}{y_T} - \ln\frac{n_i}{n_T} \tag{17}$$

where py_i is the ratio of fraudulent transactions in this group to all fraudulent transactions in the sample, pn_i is the ratio of legitimate transactions in this group to all legitimate transactions in the sample, y_i is the number of fraudulent transactions in this group, n_i is the number of legitimate transactions in this group, y_T is the number of all fraudulent transactions in the sample, and n_T is the number of all legitimate transactions in the sample.

Before performing WOE coding, these variables need to be grouped first. The higher the WOE, the higher the lousy rate, i.e., with the WOE transformation, the eigenvalue represents not only a classification but also the weight of this classification.

As shown in Fig. 8, D11 denotes the timestamp, which indicates the number of days from the previous transaction; most of the V-series features are new features derived from existing features. First, CGEA groups the features according to the missing value case. The V-series features have the same missing value.

Second, we group the features according to their relevance. As shown in Fig. 9, among *V1–V11*, *V10* has the highest correlation with *V11*, while there are high correlations between *V4* and *V5*, and *V6* and *V7*. We count the different values in the feature variables for all features in the credit card data. For the variables in the group of features with high correlations, we keep the variables with a wide variety of variable values. We remove the features with a sparse variety of variable values. In the end, we filter out 128 features with low correlation from the 339 features in the V series. Other series features are also selected accordingly using this method, significantly reducing credit card data redundancy and improving data quality.

In general, feature selection also optimizes all feature columns, and subsequent experimental results show that CGEA efficiently and quickly filters out the feature columns containing the necessary information.

2) Balancing Dataset: We employ SOBT to generate artificial samples. Based on the clustering results in the first part of SOBT, we calculate the number of artificial samples to be generated for each cluster and traverse the fraud sample points in each cluster. Starting from the cluster's center and ending with each fraud sample point in that cluster, we construct

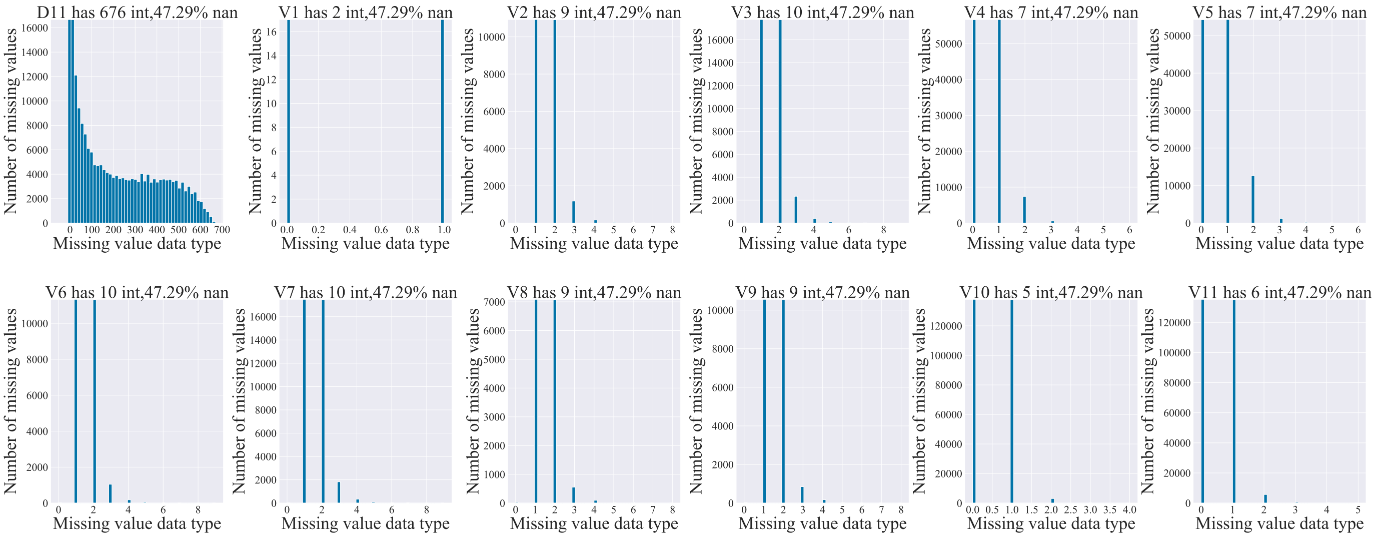


Fig. 8. Distribution of missing values of different transaction features for the IEEE-CIS Fraud Detection dataset.

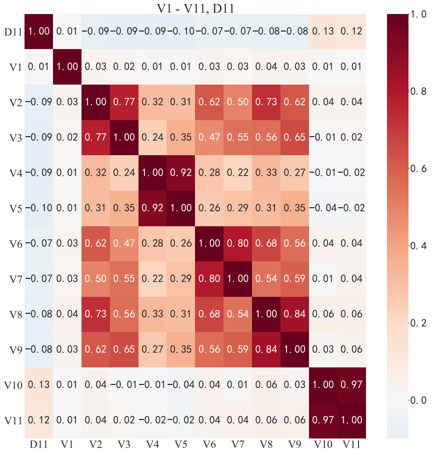


Fig. 9. Illustration of $D11$ and $V1$–$V11$ correlation heat map for the IEEE-CIS Fraud Detection dataset.

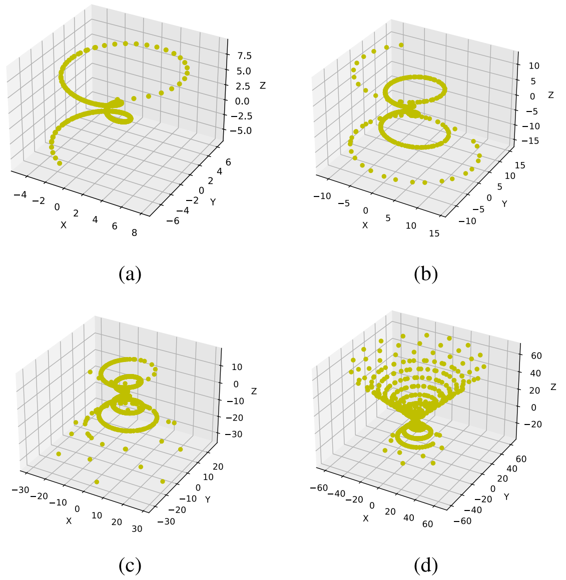


Fig. 10. Illustration of the artificial samples distribution by SOBT in different clusters of the Creditcard dataset. (a) First cluster. (b) Second cluster. (c) Third cluster. (d) Fourth cluster.

a spiral curve, and each fraud sample point needs to generate artificial samples. Fig. 10 shows the artificial sample's distribution by SOBT in different clusters of Creditcard dataset.

The points at both ends of the spiral in Fig. 10 are the sample point of the fraud data in the original dataset and the cluster center point of the cluster in which the sample is located. The 3-D spiral curve is constructed between this sample point and the cluster center. The spiral is constructed differently because of the different distribution of fraud data within different clusters.

3) Fraud Detection: Most common machine learning algorithms are trained in a mini-batch fashion to reduce memory limitation due to the size of the training data. In contrast, the gradient-boosting decision tree (GBDT) model must traverse the entire data during each training iteration. However, the standard GBDT model cannot meet the engineering requirements in facing with a massive industrial-grade dataset.

LightGBM [23] smoothly solves the problem of GBDT when dealing with massive data and can be better used for experiments. We employ LightGBM as the final predictive classifier in our experiments. The LightGBM algorithm model computes split nodes more efficiently on the features processed by the MSEFBoost algorithm. LightGBM also solves the same problems of poor performance at the beginning of the random forest model, ignoring the correlation between attributes and easy overfitting when encountering noise.

D. Experimental Results

1) Creditcard Dataset Experimental Results: We test the optimization of MSEFBoost for fraud detection model performance on the Creditcard dataset. There is always a tradeoff between the choice of high accuracy and high recall, which depends on the ultimate goal of building the model. For most cases, the choice of high accuracy may be better than

TABLE I
MSEFBoost Algorithm Effect Statistics on Creditcard

Metrics	Feature weights		Boost Rate
	Result Exclusive of MSEFBoost	Result Inclusive of MSEFBoost	
Accuracy	0.9993	0.9994	+0.01%
Precision	0.876	0.878	+0.2%
Recall	0.7211	0.7347	+1.36%
F1-score	0.7910	0.8	+0.9%
Auc	0.8605	0.8673	+0.68%
G-mean	0.8491	0.8581	+0.8%

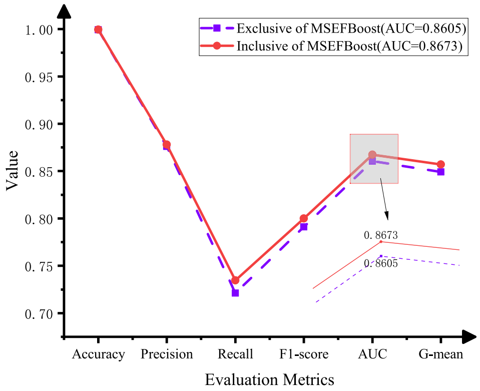


Fig. 11. Comparison of inclusive and exclusive MSEFBoost algorithm for Creditcard dataset feature boosting.

TABLE II
Performance of SOBT and Other Sampling Techniques

Algorithm Metircs	SOBT	SMOTE	ADASYN	BorderlineSMOTE
Accuracy	**0.9995**	0.9989	0.9974	0.9992
Precision	**0.9153**	0.6429	0.3782	0.7389
Recall	**0.8095**	0.7959	0.8027	0.7891
F1-score	**0.8592**	0.7112	0.5142	0.7631
Auc	**0.9081**	0.8976	0.9002	0.8943
G-mean	**0.8996**	0.8918	0.8949	0.8881

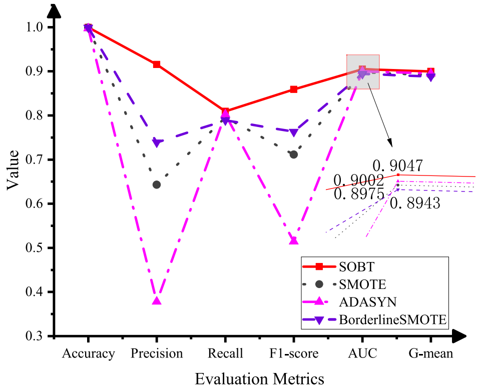


Fig. 12. Comparison of four methods for balancing Creditcard dataset.

high recall. However, for fraud prediction models, high recall is usually favored, even if some accuracy is sacrificed [61].

Table I shows the experimental results before and after applying the MSEFBoost algorithm on the Creditcard dataset. The data visualization analysis in Fig. 11 illustrates that the MSEFBoost algorithm successfully improves the recall, that is, not only the accuracy does not decrease, but also the recall increases by 1.36% and the F1-score increases by 0.9%. The fraud detection model learns better with MSEFBoost-treated features training than without, which is an essential advancement for the fraud detection model.

MSEFBoost redistributes each feature metamorphic factor to optimize the decision tree single feature importance, AUC, and MSE on the decisive role of target features. MSEFBoost improves the detection accuracy of fraudulent transaction data and promotes the learning classification performance of the fraud detection model. More fraud detection transactions will be detected, and a more significant reduction in economic may lose.

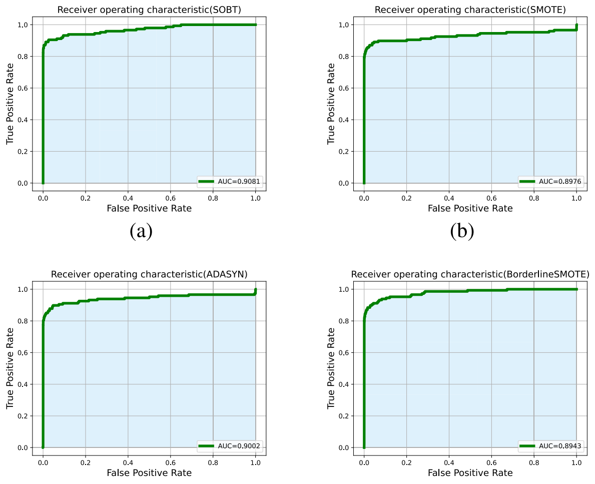
After the above processing of the Creditcard dataset, we put the preprocessed data through SOBT, SMOTE, ADASYN, and BorderlineSMOTE to compare the performance of the resampling techniques. The test results of the above algorithms on the Creditcard dataset are shown in Table II.

As shown in Fig. 12, SOBT is excellent in theoretical studies and practical applications of fraud detection, with the best performance in all evaluation metrics among other similar algorithms. After balancing the Creditcard dataset, SOBT achieves a recall of 81.63% on LightGBM fraud detection model, compared to the test of SMOTE recall results [62], an improvement of 114%, substantially improving the detection rate of fraudulent transactions. Compared to the F1-score of 85% for SMOTETomek [63], the F1-score of SOBT test is improved to 85.92%, which can detect more fraudulent transactions and block financial losses. We performed SMOTE test on LightGBM, and the test recall result is 79.5%, which is lower than the fraud detection recall of our proposed SOBT.

Karthika and Senthilselvi [64] tested the recall of the Creditcard dataset under different machine learning techniques processing in their study and measured a recall of 11% on the PAC regression algorithm proposed by Crammer, 86.95% on LDA algorithm; 21.06% on RNC algorithm, 65.47% on BNB algorithm, 15.11% on GNB algorithm, and 40.59% on ETC algorithm. From the above data compared with SOBT, it is clear that SOBT in the field of fraud detection is superior in detecting fraudulent transactions.

SOBT balances the fraud class data. Moreover, it maximizes the homogeneity of elements between classes and the heterogeneity between classes, ensuring that the generated artificial sample points fit the original data to a high degree and avoiding sample overlap, significantly improving the sample quality.

As shown in Fig. 13, our comparative study finds that SOBT has a greater improvement in the classification performance of the fraud detection model for artificial samples generated on a balanced dataset. Compared with SMOTE, ADASYN, BorderlineSMOTE, and F1-score of SOBT improved by 14.9%, 34.6%, and 9.71%, respectively, and AUC improved by 1.06%, 0.79%, and 1.38%, respectively. These data show that SOBT can better distinguish legitimate transactions from fraudulent



(a) (b) (c) (d)

Fig. 13. RUC curve and AUC area output on LGBMClassifier after balancing techniques for Creditcard dataset. (a) SOBT. (b) SMOTE. (c) ADASYN. (d) BorderlineSMOTE.

TABLE III
MSEFBOOST ALGORITHM EFFECT STATISTICS ON IEEE-CIS FRAUD DETECTION

Metrics	Feature weights		Boost Rate
	Result Exclusive of MSEFBoost	Result Inclusive of MSEFBoost	
Accuracy	0.9731	0.9734	+0.03%
Precision	0.8722	0.8781	+0.59%
Recall	0.2611	0.2671	+0.6%
F1-score	0.4018	0.4096	+0.78%
Auc	0.6299	0.6329	+0.3%
G-mean	0.5106	0.5164	+0.58%

ones, detect more fraudulent transactions hidden in legitimate ones, and minimize financial and economic losses.

Experimental results show that our algorithm can achieve better detection performance on the class-imbalanced dataset. Our balanced data show higher recall on LGBMClassifier [41], indicating that our model can detect more fraudulent transactions learned in the balanced artificial data samples and more fraudulent transactions recorded.

2) IEEE-CIS Fraud Detection Dataset Experimental Results: In addition, we test the MSEFBoost algorithm for fraud detection model performance on the IEEE-CIS Fraud Detection. The results of our experiments before and after applying MSEFBoost algorithm on the IEEE-CIS Fraud Detection dataset are shown in Table III.

As the experimental results shown in Fig. 14, MSEFBoost also performs better on the IEEE-CIS Fraud Detection dataset. It is because MSEFBoost assigns a different metamorphic factor to each feature, which can effectively avoid the computational complexity of the fraud detection model caused by the fusion of small features into new features. Moreover, it can avoid the "Hughes effect" of degradation of fraud detection due to too many features.

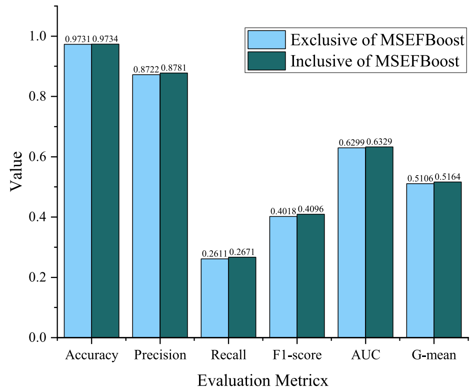


Fig. 14. Comparison of inclusive and exclusive MSEFBoost algorithm for IEEE-CIS Fraud Detection dataset feature boosting.

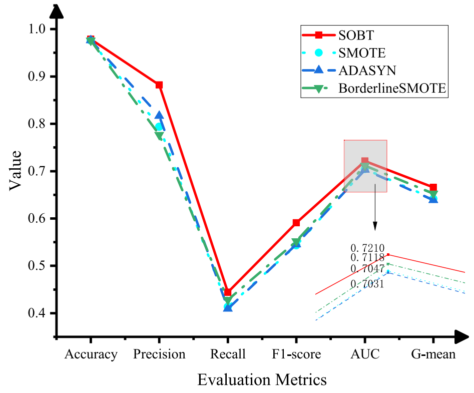


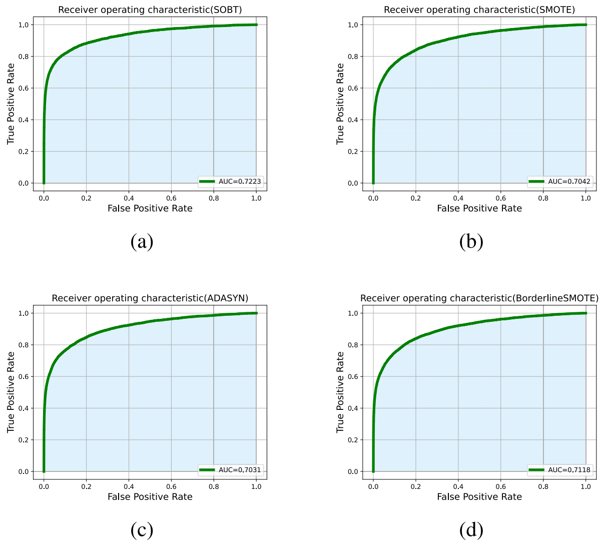
Fig. 15. Comparison of four methods for the IEEE-CIS Fraud Detection dataset balancing.

TABLE IV
PERFORMANCE OF SOBT AND OTHER SAMPLING TECHNIQUES

Algorithm Metircs	SOBT	SMOTE	ADASYN	BorderlineSMOTE
Accuracy	**0.9787**	0.9760	0.9764	0.976
Precision	**0.8819**	0.7935	0.8166	0.7765
Recall	**0.4442**	0.4132	0.4096	0.4280
F1-score	**0.5908**	0.5434	0.5455	0.5519
AUC	**0.7223**	0.7042	0.7031	0.7118
G-mean	**0.6659**	0.6415	0.6389	0.6528

Compared to previous studies that incorporated less critical features to derive new features, the embedding model in the MSEFBoost algorithm plays an important guiding role in controlling the classification results of feature decisions. As shown in the above experimental results, the feature-boosting mechanism in MSEFBoost significantly improves the fraud detection effect.

Fig. 15 shows that SOBT has the best performance among the other sampling techniques. The specific experimental data are shown in Table IV. The accuracy, recall, and F1-score are improved by 6.36%, 1.36%, and 3.62%, respectively. SOBT not only effectively balances the fraud detection dataset but



(a) (b)

(c) (d)

Fig. 16. RUC curve and AUC area output on LGBMClassifier after balancing techniques for IEEE-CIS Fraud Detection dataset. (a) SOBT. (b) SMOTE. (c) ADASYN. (d) BorderlineSMOTE.

also effectively improves the quality of fraud detection data and maximizes the fit to the original data distribution.

As shown in Fig. 16, our comparative study finds that SOBT generates artificial samples on the balanced dataset to improve the classification performance of the fraud detection model by 4.48%, 4.26%, and 3.62% compared to the F1-score of SMOTE, ADASYN, and BorderlineSMOTE, respectively, and the AUC improved by 1.56%, 1.66%, and 0.79%, respectively.

Regarding the placement of artificial fraud instances in the experiments, one of the critical elements of the oversampling techniques based on fraud instance generation is their placement of fraud instances in the feature space. Random positioning must be a nonoptimal way of a generation because we want to keep the original properties of the fraud-like data and enhance them in conflicted regions. These regions are mainly class boundaries, overlapping regions, and small discontinuities.

Although ADASYN automatically determines the number of samples to be synthesized for each minority sample based on the data distribution, it is susceptible to outliers. The more legitimate transactions in its immediate neighborhood, the more samples are generated around it. BorderlineSmote only oversamples the fraudulent samples near the boundary. SOBT ensures the homogeneity of the artificial sample with the original fraud sample and enhances the heterogeneity of the fraudulent transaction sample with the legitimate one.

Therefore, the optimal oversampling technique concentrates on the intelligent placement of instances, which balances the data distribution of legitimate and fraudulent transactions and reduces the learning difficulty. SOBT combines the geometry model with the idea of oversampling to reduce the credit card fraud detection error probability of fraudulent transaction data.

VI. Conclusion and Future Work

In this article, we investigate the credit card fraud detection problem. The compound grouping elimination strategy is designed to solve the redundant features in the dataset. Multifactor synchronous embedding mechanism can make full use of the evaluation of features by the embedding model to improve the fraud detection performance. The classifier adjusts the decision rate of the features to the target domain based on our adaptive fusion of metamorphic factors. SOBT is proposed to enhance the ability of classifiers to distinguish between legitimate and fraudulent transactions.

Experimental results on two real-world transaction datasets show the performance of our methods. Compared with other methods, we can identify more fraudulent transactions and recover more financial losses. Our proposed fraud detection model is dedicated to historical transaction data. The experimental results further demonstrate the significant effect of CGEA, MSEFBoost, and SOBT in improving the effectiveness of fraud detection models, which can be used to ensure effective recall rate improvement. Banks and other financial institutions can fully consider the detection results of the detection system when identifying potential default fraudulent customers. It can have a better early warning function for credit card fraud default risk, reducing fraud risk early and relieving the supervisory costs for managers.

In the future, we intend to consider in more detail the data distribution of artificial fraud samples and the behavioral performance associations between transaction samples. We plan to explore improvement measures in fraud detection performance. Moreover, we will conduct experiments with graph neural network knowledge [65], [66] to achieve credit card transaction anomaly detection tasks.

Declaration of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

References

[1] R. Bin Sulaiman, V. Schetinin, and P. Sant, “Review of machine learning approach on credit card fraud detection,” *Human-Centric Intelligent Systems*, vol. 2, pp. 1–14, Jun. 2022.

[2] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, “Learning from class-imbalanced data: Review of methods and applications,” *Exp. Syst. Appl.*, vol. 73, pp. 220–239, May 2017.

[3] G. Liu, *Petri Nets Theoretical Models and Analysis Methods for Concurrent Systems*. Singapore: Springer, 2022.

[4] Z. Li, G. Liu, and C. Jiang, “Deep representation learning with full center loss for credit card fraud detection,” *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 2, pp. 569–579, Apr. 2020.

[5] Y. Xie, G. Liu, R. Cao, Z. Li, C. Yan, and C. Jiang, “A feature extraction method for credit card fraud detection,” in *Proc. 2nd Int. Conf. Intell. Auto. Syst. (ICoIAS)*, Mar. 2019, pp. 70–75.

[6] A. Singh, R. K. Ranjan, and A. Tiwari, “Credit card fraud detection under extreme imbalanced data: A comparative study of data-level algorithms,” *J. Exp. Theor. Artif. Intell.*, vol. 34, no. 4, pp. 571–598, Jul. 2022.

[7] L. Zheng, G. Liu, C. Yan, and C. Jiang, “Transaction fraud detection based on total order relation and behavior diversity,” *IEEE Trans. Computat. Social Syst.*, vol. 5, no. 3, pp. 796–806, Sep. 2018.

[8] D. Gibert, J. Planes, C. Mateu, and Q. Le, “Fusing feature engineering and deep learning: A case study for malware classification,” *Exp. Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 117957.

[9] L. Zheng, G. Liu, C. Yan, C. Jiang, and M. Li, “Improved TrAdaBoost and its application to transaction fraud detection,” *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 5, pp. 1304–1316, Oct. 2020.

[10] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," *Exp. Syst. Appl.*, vol. 51, pp. 134–142, Jun. 2016.

[11] A. S. Hussein, R. S. Khairy, S. M. M. Najeeb, and H. T. S. Alrikabi, "Credit card fraud detection using fuzzy rough nearest neighbor and sequential minimal optimization with logistic regression," *Int. J. Interact. Mobile Technol. (iJIM)*, vol. 15, no. 5, p. 24, Mar. 2021.

[12] Y. Guo, Z. Zhang, and F. Tang, "Feature selection with kernelized multi-class support vector machine," *Pattern Recognit.*, vol. 117, Sep. 2021, Art. no. 107988.

[13] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *J. Appl. Sci. Technol. Trends*, vol. 2, no. 1, pp. 20–28, Mar. 2021.

[14] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, 2021.

[15] S. Han, K. Zhu, M. Zhou, and X. Cai, "Information-utilization-method-assisted multimodal multiobjective optimization and application to credit card fraud detection," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 4, pp. 856–869, Aug. 2021.

[16] X. Dong, U. Victor, and L. Qian, "Two-path deep semisupervised learning for timely fake news detection," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 6, pp. 1386–1398, Dec. 2020.

[17] S. Zhu, H. Shaowu Yuchi, M. Zhang, and Y. Xie, "Sequential adversarial anomaly detection with deep Fourier kernel," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 3345–3349.

[18] A. A. Taha and S. J. Malebary, "An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine," *IEEE Access*, vol. 8, pp. 25579–25587, 2020.

[19] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit card fraud detection using AdaBoost and majority voting," *IEEE Access*, vol. 6, pp. 14277–14284, 2018.

[20] H. Zhu, G. Liu, M. Zhou, Y. Xie, A. Abusorrah, and Q. Kang, "Optimizing weighted extreme learning machines for imbalanced classification and application to credit card fraud detection," *Neurocomputing*, vol. 407, pp. 50–62, Sep. 2020.

[21] E. Elyan, C. F. Moreno-Garcia, and C. Jayne, "CDSMOTE: Class decomposition and synthetic minority class oversampling technique for imbalanced-data classification," *Neural Comput. Appl.*, vol. 33, no. 7, pp. 2839–2851, Apr. 2021.

[22] C. Bunkhumpornpat and K. Sinapiromsaran, "DBMUTE: Density-based majority under-sampling technique," *Knowl. Inf. Syst.*, vol. 50, no. 3, pp. 827–850, Mar. 2017.

[23] D. Devi, S. K. Biswas, and B. Purkayastha, "Redundancy-driven modified Tomek-link based undersampling: A solution to class imbalance," *Pattern Recognit. Lett.*, vol. 93, pp. 3–12, Jul. 2017.

[24] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[25] Y. Xie, G. Liu, C. Yan, C. Jiang, M. Zhou, and M. Li, "Learning transactional behavioral representations for credit card fraud detection," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 5, 2022, doi: 10.1109/TNNLS.2022.3208967.

[26] R. Cao, G. Liu, Y. Xie, and C. Jiang, "Two-level attention model of representation learning for fraud detection," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 6, pp. 1291–1301, Dec. 2021.

[27] A. Islam, S. B. Belhaouari, A. U. Rehman, and H. Bensmail, "KNNOR: An oversampling technique for imbalanced datasets," *Appl. Soft Comput.*, vol. 115, Jan. 2022, Art. no. 108288.

[28] J. Chen, H. Huang, A. G. Cohn, D. Zhang, and M. Zhou, "Machine learning-based classification of rock discontinuity trace: SMOTE oversampling integrated with GBT ensemble learning," *Int. J. Mining Sci. Technol.*, vol. 32, no. 2, pp. 309–322, Mar. 2022.

[29] W. Jo and D. Kim, "OBGAN: Minority oversampling near borderline with generative adversarial networks," *Exp. Syst. Appl.*, vol. 197, Jul. 2022, Art. no. 116694.

[30] X. Yi, Y. Xu, Q. Hu, S. Krishnamoorthy, W. Li, and Z. Tang, "ASN-SMOTE: A synthetic minority oversampling method with adaptive qualified synthesizer selection," *Complex Intell. Syst.*, vol. 8, pp. 1–26, Jun. 2022.

[31] W. Wang and D. Sun, "The improved AdaBoost algorithms for imbalanced data classification," *Inf. Sci.*, vol. 563, pp. 358–374, Jul. 2021.

[32] D.-N. Wang, L. Li, and D. Zhao, "Corporate finance risk prediction based on LightGBM," *Inf. Sci.*, vol. 602, pp. 259–268, Jul. 2022.

[33] C. Yang, G. Liu, C. Yan, and C. Jiang, "A clustering-based flexible weighting method in AdaBoost and its application to transaction fraud detection," *Sci. China Inf. Sci.*, vol. 64, no. 12, pp. 1–11, Dec. 2021.

[34] J. Singla et al., "Class balancing methods for fraud detection using deep learning," in *Proc. 2nd Int. Conf. Artif. Intell. Smart Energy (ICAIS)*, Feb. 2022, pp. 395–400.

[35] J. Cui, C. Yan, and C. Wang, "ReMEMBeR: Ranking metric embedding-based multicontextual behavior profiling for online banking fraud detection," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 3, pp. 643–654, Jun. 2021.

[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 28, pp. 321–357, Jun. 2002.

[37] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. IEEE World Congr. Comput. Intell.*, Jun. 2008, pp. 1322–1328.

[38] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.* Cham, Switzerland: Springer, 2005, pp. 878–887.

[39] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on K-means and smote," *Inf. Sci.*, vol. 465, pp. 1–20, Jun. 2018.

[40] Z. Li, M. Huang, G. Liu, and C. Jiang, "A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection," *Exp. Syst. Appl.*, vol. 175, Aug. 2021, Art. no. 114750.

[41] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–9.

[42] I. K. Nti and A. R. Somanathan, "A scalable RF-XGBoost framework for financial fraud mitigation," *IEEE Trans. Computat. Social Syst.*, early access, Oct. 10, 2022, doi: 10.1109/TCSS.2022.3209827.

[43] Y. Tian and G. Liu, "MANE: Model-agnostic non-linear explanations for deep learning model," in *Proc. IEEE World Congr. Services (SERVICES)*, Oct. 2020, pp. 33–36.

[44] Y. Zhang, W. Yu, Z. Li, S. Raza, and H. Cao, "Detecting ethereum Ponzi schemes based on improved LightGBM algorithm," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 2, pp. 624–637, Apr. 2022.

[45] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random forest for credit card fraud detection," in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC)*, Mar. 2018, pp. 1–6.

[46] D. Edelmann, T. F. Móri, and G. J. Székely, "On relationships between the Pearson and the distance correlation coefficients," *Statist. Probab. Lett.*, vol. 169, Feb. 2021, Art. no. 108960.

[47] D. Sehrawat and Y. Singh, "Comparative analysis on fraud detection in credit card transaction using different machine learning algorithms," in *Soft Computing: Theories and Applications*. Singapore: Springer, 2022, pp. 673–684.

[48] A. Rb and S. K. Kr, "Credit card fraud detection using artificial neural network," *Global Transitions Proc.*, vol. 2, no. 1, pp. 35–41, Jun. 2021.

[49] Y. Zhou, G. Chi, J. Liu, J. Xiong, and B. Wang, "Default discrimination of credit card: Feature combination selection based on improved FDAF-score," *Exp. Syst. Appl.*, vol. 206, Nov. 2022, Art. no. 117829.

[50] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[51] F. Zhang, G. Liu, Z. Li, C. Yan, and C. Jiang, "GMM-based undersampling and its application for credit card fraud detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[52] C. Jiang, J. Song, G. Liu, L. Zheng, and W. Luan, "Credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3637–3647, Oct. 2018.

[53] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, CA, USA, Tech. Rep., 2006.

[54] S. Carta, G. Fenu, D. R. Recupero, and R. Saia, "Fraud detection for E-commerce transactions by employing a prudential multiple consensus model," *J. Inf. Secur. Appl.*, vol. 46, pp. 13–22, Jun. 2019.

[55] *Credit Card Fraud Detection*. Accessed: 2022. [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[56] *IEEE-CIS-Fraud-Detection*. Accessed: 2022. [Online]. Available: https://www.kaggle.com/code/pavan1512/ieee-cis-fraud-detection/data

[57] R. Li, Z. Liu, Y. Ma, D. Yang, and S. Sun, "Internet financial fraud detection based on graph learning," *IEEE Trans. Computat. Social Syst.*, early access, Jul. 15, 2022, doi: 10.1109/TCSS.2022.3189368.

[58] R. Saia and S. Carta, "Evaluating the benefits of using proactive transformed-domain-based techniques in fraud detection tasks," *Future Gener. Comput. Syst.*, vol. 93, pp. 18–32, Apr. 2019.

[59] Y. Xie, G. Liu, C. Yan, C. Jiang, and M. Zhou, "Time-aware attention-based gated network for credit card fraud detection by extracting transactional behaviors," *IEEE Trans. Computat. Social Syst.*, early access, Mar. 30, 2022, doi: 10.1109/TCSS.2022.3158318.

[60] D. Gough, "Weight of evidence: A framework for the appraisal of the quality and relevance of evidence," *Res. Papers Educ.*, vol. 22, no. 2, pp. 213–228, Jun. 2007.

[61] A. D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, Sep. 2018.

[62] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Credit card fraud detection-machine learning methods," in *Proc. 18th Int. Symp. INFOTEH-JAHORINA (INFOTEH)*, Mar. 2019, pp. 1–5.

[63] A. K. Uttam and G. Sharma, "A comparison of data balancing techniques for credit card fraud detection using neural network," in *Proc. 5th Int. Conf. I-SMAC (IoT Social, Mobile, Analytics Cloud) (I-SMAC)*, Nov. 2021, pp. 1136–1140.

[64] J. Karthika and A. Senthilselvi, "Credit card fraud detection based on ensemble machine learning classifiers," in *Proc. 3rd Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, Aug. 2022, pp. 1604–1610.

[65] Y. Xu, J. Wang, M. Guang, C. Yan, and C. Jiang, "Multistructure graph classification method with attention-based pooling," *IEEE Trans. Computat. Social Syst.*, early access, May 3, 2022, doi: 10.1109/TCSS.2022.3169219.

[66] G. Zhang et al., "eFraudCom: An E-commerce fraud detection system via competitive graph neural networks," *ACM Trans. Inf. Syst.*, vol. 40, no. 3, pp. 1–29, Jul. 2022.

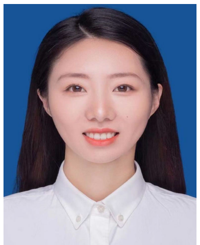
Lina Ni received the Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2009.

She is currently a Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong, China. Her main research interests include cloud/edge computing, Petri nets, distributed algorithms, information security, privacy preservation, and intelligent computing.

Dr. Ni is a Senior Member of the China Computer Federation (CCF). She is also a Committee Member of the Professional Committee of Network Information Service of the China Automation Federation.

Jufeng Li is currently pursuing the M.S. degree with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China.

His main research interests include information security, privacy preservation, natural language processing, and reinforcement learning.



Huixin Xu is currently pursuing the M.S. degree with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China.

Her main research interests include cloud computing, edge computing, information security, and privacy preservation.

Xiangbo Wang is currently pursuing the M.S. degree with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China.

His main research interests include distributed computing, and wireless and mobile security.

Jinquan Zhang received the Ph.D. degree in computer science and technology from Tongji University, Shanghai, China, in 2007.

He is currently an Associate Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong, China. His main research interests include cloud computing, Petri nets, information security, and parallel and distributed processing.

Dr. Zhang is also a member of the China Computer Federation (CCF).