

Credit Card Anti-Fraud Prediction Based on Ensemble learning

Chenrui Liang

Beijing Changping Xinxuedao Linchuan International School, Beijing, China

Abstract. In recent years, due to the development of Internet technology in the financial industry, cardless and cashless payments have become increasingly popular. At present, people only need to bind their cards to their cell phones to scan the code for payment. In the meanwhile, as the use of credit cards is vigorously promoted nationwide, younger generations relied more on the credit card debt for consumption. However, although these developments have brought a lot of convenience to people's travel and their life, the virtual nature of the Internet and the "Enjoy First, Pay Later" feature of credit cards have led to a significant increase in potential credit risk in the entire transaction, triggering more and more new credit card fraud incidents. While the overall financial environment in China is relatively healthy, the fact that fraudulent behaviors such as credit card overdrafts, counterfeit cards, and credit card frauds are all personal conduct, making it difficult to detect and prevent these individuals from acting illegally in a timely manner. Although fraud represents a small percentage of the overall transaction size, its bad debt losses to merchants and banks can be significant. In order to make a timely prediction of fraud, there are already many single-model machine learning methods such as decision trees and logistic regression. However, the generalization ability of these models is not good enough in the face of complex user behavior features. In addition, since the probability of fraudulent behavior is very small, resulting in few samples that can be trained, even if a model has a high accuracy rate, there is no guarantee that it can accurately predict fraudulent behavior. Therefore, this paper proposes the Adaboost method based on ensemble learning and uses SMOTE oversampling based on a few classes of fraudulent samples of user features to solve the above existing problems.

Keywords: Credit Card, Anti-Fraud, Adaboost method.

1. Background and Significance

In the information science era of the 21st century, traditional businesses are getting more closely connected to the Internet, which has disrupted the business models of many industries and facilitated their transformation. However, no industry can grow without the act of financial transactions in the end. According to statistics, billions of credit card transactions are made every day worldwide in 2018, and that scale is still growing all the time [1]. A survey by the Federal Reserve Bank of San Francisco in the United States found that the use of credit cards increased by 9% from 2016 to 2018, and credit cards have overtaken cash as one of the preferred payment tools for most people [2]. In order to respond to the national call, accelerate the transformation process of China's economic market from "large-scale" to "high-quality" development, and provide more convenient payment services for society, the credit card business has become the main force in China's financial consumption business. By the end of 2021, a total of 800 million credit cards and deposit cards were activated in China [3].

However, although digital transformation has brought great convenience to the credit card business, after all, online transactions are virtual, and since the credit card itself has the feature of "Enjoy Now, Pay Later", many problems may arise during the payment period. This has led to frequent credit card fraud, which poses a huge challenge to credit card security control. According to the survey, the direct losses caused by credit card fraud in 2018 amounted to more than \$27.8 billion worldwide. Common credit card frauds include fraud at the time of application, fraud at the time of transaction, and fraud for illegal purposes. Fraud at the time of application is the use of false information, or impersonating someone else to apply for a credit card, which is now less likely to occur. Fraud at the time of transaction is quite common, and can be subdivided into the following four situations: (1) frequent credit card overdraft: overspending that leads to the inability to repay on time; (2) card overdraft and revolving overdraft: multiple credit cards repaying each other; (3) cash flow cashing out: a broken capital chain that leads to the inability to repay debts; and (4) fraudulent transaction cashing out.

As of the end of 2019, the card fraud rate in China was 0.87 %, which shows that this data category is quite unbalanced. Generally, we call the class that is higher than the number of other samples as the majority class (in this paper no fraud is the majority class), or else, is called as the minority class (i.e. fraud in this paper). Although the fraud rate is extremely low, it is tremendously harmful. The problem of class imbalance is common in real scenarios, such as bank customer credit rating assessment [4], fault detection [5], anomaly processing [6], medical diagnosis [7], and fraud risk control [8], etc. Due to the complexity and missing nature of their data, traditional prediction methods do not have a good grasp of the characteristics of the data, ultimately leading to poor prediction results.

2. Related Work

In terms of the development time of the industry, the credit card business in foreign countries emerged earlier than in China, so the related technologies also developed earlier and better. Traditional single-model machine learning models such as SVM, DT, and NB were widely used for credit card anti-fraud prediction, and then the arrival of multi-model classification algorithms such as random forests and neural networks led to an improvement in the accuracy of anti-fraud prediction. Kokkinaki [9] et al. combined decision trees and Boolean logic functions to distinguish whether each user's transaction behavior is illegal or not by clustering analysis. D. Ignatov [10] et al. proposed a new decision tree-based prediction model that can solve the two problems of high model complexity and overfitting of data. This approach constructs a new architecture-decision flow that can overcome recursive node partitioning, which leads to a geometric reduction in the amount of leaf node data. Maes et al [11] verified through a comparative analysis that Bayesian networks are faster and more accurate than neural networks, but have slightly lower learning speeds. E. A. Morales [12] et al. proposed a risk analysis model applying a newer technique by using the Bayesian algorithm and used it for prediction and analysis, other than allowing the client to evaluate the data to verify the sensitivity and specificity of the results. In addition to this, there is also a relatively well-developed work abroad related to credit card anti-fraud detection based on clustering techniques. For example, M. Zamini [13] et al. proposed an unsupervised fraud detection method using autoencoder-based clustering and used a European credit card transaction dataset to confirm that the accuracy of the method outperformed other methods.

In recent years, the economic development in China is getting better and better, and credit cards are being more widely used. Research related to the anti-fraud prediction of credit cards has also gradually increased, and there are many works in this direction with very accurate results. H. Tingfei et al [14] proposed an over-sampling method based on the automatic encoding of variances, and after submitting the extended dataset to a training baseline, the model was tested with good performance in terms of precision, F-parameters, accuracy and specificity. Z. Zhang et al [15] integrated gradient boosting decision trees and neural networks to significantly improve performance and to further correct for model deterioration.

All the above research methods have their own advantages, which can reduce the occurrence of credit card fraud cases to a certain extent; however, they all have certain problems, such as the handling of imbalanced data sets, tuning of model training overfitting or underfitting, avoiding falling into local optimal solutions, etc. Moreover, there are not many research contents for the integration of imbalance classification problem and integration learning, which makes the accuracy of credit card transaction fraud prediction unsatisfactory.

3. Construction of Classification Prediction Model

3.1 Prediction Framework Construction Process

The full flow of our model architecture in this paper is shown in Figure 1. In Chapter 3 we will first describe how to process the unbalanced data set and treat the two types of data as 1:1. Then we will explain the principles of decision tree and Adaboost model.

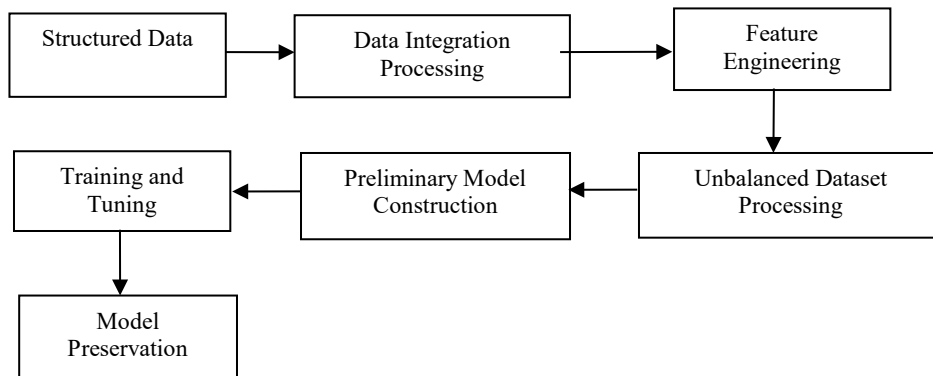


Figure 1. Flow Chart of Model Construction

3.2 Data Pre-processing

A reasonable and effective pre-processing of the dataset can improve the subsequent modeling prediction ability. First, we need to adopt feature engineering to decompose the feature information in the dataset in order to guide the subsequent operation to improve the accuracy of modeling. Most of the credit card data are positive and negative sample imbalance datasets, and special treatment of sample imbalance is required when processing such data. Section 3.1 of this chapter introduces the dataset used in this thesis and the feature engineering commonly used in data analysis and processing. Section 3.2 introduces a data processing method and results of oversampling negative samples used in this experiment. Section 3.4 and Section 3.5 introduce the two specific algorithms used in this experiment, respectively.

3.2.1 Dataset

The dataset used in this experiment includes data on transactions made by European cardholders via credit cards over a two-day period in September 2013, which contains a total of 492 fraudulent records out of 284,807 transactions with a total of 31 features.

The data in the dataset are processed by the PCA algorithm with dimensionality reduction, and the private data of users are hidden. The features in the dataset are classified into four categories, and the specific attribute descriptions are shown in Table 1.

Table 1. Dataset Attributes

Feature Name	Data Type	Feature Description
time	float	Elapsed time (unit: second) between this transaction and the first transaction in the dataset
V1-V28	float	Principal component feature data obtained with PCA
amount	float	Transaction amount per sample
class	float	1 means the transaction is fraudulent; 0 means the transaction is normal

3.2.2 Dealing with Missing Values and Outliers

The treatment of missing values is very important, and how it is handled will affect the subsequent model building and the accuracy of the results. Commonly used methods to handle missing values include: (1) No treatment. When the model to be used is a tree model such as decision tree or random forest, we can put the missing values aside because some models have a very reasonable way to accept and handle missing values. (2) Remove features. If there is too much missing data for a particular feature in the dataset, then this feature in the modeling can be omitted. (3) Interpolation fitting. If the number of missing values is small, interpolation fitting can be used to fill in the missing values. After analyzing and observing the data, the mean, plural or maximum value is usually selected for estimation, and a model can also be built to predict the missing values. (4) Feature binning. The data are boxed, and the missing values are set as a separate box.

Outliers in the dataset may also have a significant impact on the subsequent modeling analysis processing. Similar to the common approach of dealing with missing values in a dataset, we can refer

to the approach of dealing with missing values when dealing with outliers. For example, we can remove outliers that are clearly visible when observing the data, and we can also perform reasonable transformation or grouping of outliers, or estimation or other statistical methods for the data.

In the following, common methods typically used to dispose of outliers in a data set are listed: (1) Direct deletion. If obvious outliers are found during the initial preview of the data and the number is small, you can choose to dispose of the outliers by directly deleting them, or by using the method of trimming at both ends. (2) Estimation. Similar to the estimation method used when dealing with missing values, the abnormal values in the dataset can be estimated using the mean, median, etc. (3) Separate processing. If a large number of outliers are found in the process of analyzing the data, such outliers can be treated separately. One solution is to separate the abnormal and non-abnormal data, and then divide them into two groups for subsequent processing. (4) Feature transformation. Mathematical transformation of the characteristic data in the data set can also achieve the purpose of eliminating outliers, such as logarithmic transformation of the anomalous characteristics caused by extreme values; operations such as addition, subtraction, multiplication and division between data with different characteristics can also effectively deal with outliers; data boxing is also a very common processing method.

3.2.3 Feature Construction and Selection

The reconstruction of features in a dataset usually requires the use of some specialized background knowledge. There are also many factors to consider, such as different numerical, categorical and temporal features, etc. For numerical features, we usually consider simple addition or subtraction transformations between different features in a certain way; for categorical features, we can try to crossover between features or use embedding; for temporal features, we need some specialized ways and techniques, such as calculating time intervals, etc.

Feature selection can effectively reduce the number and dimensionality of features, and the number of features within a reasonable range can better ensure the generalization ability of the model and reduce the degree of overfitting.

In this experiment, the following principles were used for feature selection.

(1) The features with very small variation between different samples of the same feature are removed first. The specific formula for calculating the variance is as follows.

$$M = \frac{x_1 + x_2 + \dots + x_n}{n}, n = \text{number of sample} \quad (1)$$

$$s^2 = \frac{(M - x_1)^2 + (M - x_2)^2 + \dots + (M - x_n)^2}{n} \quad (2)$$

(2) The individual feature variables in the dataset are tested and scored for each dimensional feature, and the score indicates the importance of this column of features, and the features with low scores can be thrown away before the model is built.

The commonly used scoring criterion is the Pearson correlation coefficient, which is a more concise and intuitive way to represent the correlation between the characteristic variables. The calculation formula is as follows.

$$\rho_{x,y} = \frac{cov(X,Y)}{\sigma_x \sigma_y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)}\sqrt{E(Y^2) - E^2(Y)}} \quad (3)$$

The features are processed according to the results of the Pearson correlation coefficient formula above. If the Pearson correlation coefficient of a feature is very close to 0, it means that there is no too strong positive or negative correlation between the feature and the classification label, and the feature can be removed.

3.3 Unbalanced Dataset Processing

Unbalanced data refers that one or some classes in the dataset have much higher sample sizes than others, while some classes have smaller sample sizes, and the classes with larger sample sizes are usually referred to as majority classes and those with smaller sample sizes as minority classes [16]. Samples with imbalanced data classes are usually encountered by two methods of handling, namely,

undersampling and oversampling. The undersampling method can improve the classification accuracy of the model for minority class samples, but this method loses a large amount of real data and does not make full use of the information available. For the data of minority samples, the oversampling method can generate some synthetic data with higher similarity around the existing data sample points, so as to extend the data diversity and avoid overfitting in the future model training. By comparison, this paper will use the method of oversampling a few classes of samples to solve the problem of unbalanced dataset and improve the accuracy of subsequent model classification. Specifically, we use the method of SMOTE, which is an improved oversampling method that restricts the rules and range of sample point generation to make the generation of samples more scientific.

The flow of the SMOTE algorithm is as follows.

1). Let the set of few classes be S_{little} , take any one of the points of this $x \in S_{little}$, compute x to S_{little} , all the distances (e.g., Euclidean distance) of all points in the set to get its k nearest neighbors.

2). Determine the sampling rate N , which is usually based on the proportion of category imbalance artificially set based on experience. For each of the $x \in S_{little}$, from its k nearest neighbors, a number of samples are randomly selected, and it may be useful to set the selected sample of nearest neighbors as x_n .

3). By the nearest neighbor sample x_n and the original sample x , we construct a new sample x_{new} that

$$x_{new} = x + rand(0,1) \times |x - x_n|.$$

Table 2 compares the datasets before and after SMOTE processing.

Table 2. Comparison of Datasets Before and After SMOTE Processing

	Number of Samples	Percentage of Positive Samples	Number of Negative Samples
Before Processing	284,807	99.83%	0.17%
After Processing	568,630	50.00%	50.00%

3.4 Decision Tree Algorithm

Due to its natural branching structure, decision trees are often used to build classifiers based on feature scores, and each node split is actually the process of making a choice. The process of building a decision tree is very much in line with how humans think when they actually classify, so the decision tree algorithm has a high degree of interpretability. Starting from the root node, the specified data samples are divided according to a certain feature at the next node, and the next branches represent the possible outcomes of this feature attribute. Until finally, all samples in the dataset are assigned to specific categories and become leaf nodes.

In general, different algorithms choose different methods of dividing data samples by internal nodes. For example, ID3 selects split nodes based on information gain; C4.5 is an improved version of ID3, which controls the fraction of nodes based on information gain rate; and CART uses the Gini index. Using different algorithms and choosing different feature orders will result in different tree structures.

The specific steps of the decision tree algorithm are as follows.

(1): Assume that T is the training sample set.

(2): Select an attribute $c_i \in Characters$ from the feature set $Characters$ that best distinguishes the samples in T according to the feature scoring rules.

(3): Based on the features selected from (2) $c_i \in Characters$, create a tree node G_1 and its child nodes G_2, G_3, \dots, G_m (m is the length of the $Characters$ set).

(4): Repeat step (2) and step (3), stopping the cycle when one of the following two conditions is encountered:

(a). All features in the set $Characters$ have formed nodes in the tree.

(b): Certain node G_i has only one classification result (i.e., entropy of 0).

The three choices of features for decision tree classification are briefly described below.

(1) Information gain: with or without feature c_i of the dataset, the difference in information entropy obtained from the dataset D , i.e.

$$g(D, c_i) = H(D) - H(D|c_i)$$

where

$H(D) = -\sum_{i=1}^k \frac{|D^i|}{|D|} \log \frac{|D^i|}{|D|}$ is the information entropy of the dataset D , k is the number of categories, and

$H(D|A) = \sum_{i=1}^n \frac{|D^i|}{|D|} H(D^i)$ is the gap between the conditional entropy of the condition c_i , and n is the number of values of features c_i .

(2) Information gain rate: This is the optimization of the information gain calculation method. The difference will be changed to ratio, reducing the interference of the outliers, that is

$$g_R(D, A) = \frac{g(D, c_i)}{H_A(D)}$$

where $H_A(D) = -\sum_{i=1}^n \frac{|D^i|}{|D|} \log \frac{|D^i|}{|D|}$ is the entropy of dataset D with respect to the values taken for the feature c_i .

$g(D, c_i)$ is the information gain of the dataset D , and n is the number of values of feature c_i .

The C4.5 algorithm uses this approach to select the optimal attributes, similar to the ID3 algorithm, which is improved by the C4.5 algorithm.

(3) Gini index: Unlike the previous two, this indicator is based on the definition of probability:

$$Gini(p) = \sum_{i=1}^k p_i(1 - p_i) = 1 - \sum_{i=1}^k p_i^2$$

where k is the number of categories, and p_i is the probability that the sample point belongs to class i . For a given set of samples D , its Gini index is:

$$Gini(p) = 1 - \sum_{i=1}^k \left(\frac{|D_i|}{|D|} \right)^2$$

where D_i is the set of samples belonging to class i in D , and k is the number of classes.

The sample set D , according to the whether the characteristics c_i takes a certain possible value, can be split into D_1 and D_2 two parts, i.e.

$$D_1 = \{(x, y) \in D | c_i(x) = a\}$$

$$D_2 = D - D_1$$

Then, under the characteristic c_i condition, the Gini index of the set D is defined as

$$Gini(D, c_i) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

where $Gini(D)$ denotes the uncertainty of the set D , $Gini(D, c_i)$ denotes the uncertainty of the D set after partitioning by c_i .

Decision trees generated using unconstrained methods are prone to overfitting. To prevent poor generalization performance of decision tree models, pruning methods can be used in the process of building decision trees to reduce the number of layers of the tree and thus have better predictive power for unknown data.

3.5 Adaboost

Adaboost improves the classification accuracy by integrating multiple weak classifiers and combining these weak classifiers to form a strong classifier. In fact, Adaboost is a serial integrated classifier and its results are influenced by the previous classifiers. Samples that were correctly classified in the previous training round are given a lower weight in the next training round, and conversely, samples that were incorrectly trained are given a higher weight. It is iterated in this way until the termination condition.

The specific learning algorithm is as follows:

Input: training dataset $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, where $x_i \in \mathfrak{X} = R$, and $y_i \in \mathfrak{Y} = \{-1, +1\}, i = 1, 2, \dots, N$.

Output: Final classifier $G(x)$.

Initialize the weight distribution of the training data

$$D_1 = (w_{11}, \dots, w_{1i}, \dots, w_{1N}), w_{1i} = \frac{1}{N}, i = 1, 2, \dots, N$$

It is assumed that the training set initially has a uniform distribution of weights.

For $m = 1, 2, \dots, M$, iterative learning of the basic classifier is needed.

The basic classifier is learned using a training dataset with a weight distribution D_m

$$G_m(x): \mathfrak{X} \rightarrow \{-1, +1\}$$

Calculate classification error rate of $G_m(x)$ on the training dataset.

$$e_m = \sum_{i=1}^N P(G_m(x_i) \neq y_i) = \sum_{i=1}^N w_{mi} I(G_m(x_i) \neq y_i)$$

Calculate the basic classifier $G_m(x)$ of the coefficients α_m , and α_m indicates the importance of $G_m(x)$ in the final classifier.

$$\alpha_m = \frac{1}{2} \log \frac{1 - e_m}{e_m}$$

The logarithm here is the natural logarithm.

Update the weight distribution of the training dataset.

$$D_{m+1} = (w_{m+1,1}, \dots, w_{m+1,i}, \dots, w_{m+1,N})$$

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} \exp(-\alpha_m y_i G_m(x_i)), i = 1, 2, \dots, N$$

Here, the Z_m is the specification factor.

$$Z_m = \sum_{i=1}^N w_{mi} \exp(-\alpha_m y_i G_m(x_i))$$

It makes D_{m+1} to be a probability distribution.

Build linear combinations of basic classifiers

$$f(x) = \sum_{m=1}^M \alpha_m G_m(x)$$

The final classifier is thus obtained:

$$G(x) = \text{sign}(f(x)) = \text{sign}\left(\sum_{m=1}^M \alpha_m G_m(x)\right)$$

4. Model Tuning and Result Analysis

In this section, we present the parameter settings of the decision tree model and the Adaboost model, and we perform the training set: test set as 7:3 division and ten-fold cross-validation on the dataset. We then give the result analysis based on the check accuracy, recall and F1-Score.

4.1 Decision Tree Model

The decision tree method constructs leaf nodes based on the tree structure, and the leaf nodes are used to make feature judgments for the final classification. Different decision tree algorithms use different methods on how to classify the choices.

In this thesis, a decision tree based on the ID3 algorithm is used for training and classification, and the splitting attributes are selected by the calculation of information gain.

The prediction results of the constructed decision tree model were recorded, where the parameters were selected as shown in Table 3.

Table 3. Decision Tree Model Parameter Settings

Parameters	Meaning	Set value
criterion	Feature selection criteria	'entropy'
splitter	Feature classification criteria	'best'
max_depth	Maximum depth of decision tree	7
min_samples_split	Minimum number of samples required for internal node repartitioning	30
min_samples_leaf	Minimum number of samples of leaf nodes (i.e. classification)	10
max_leaf_nodes	Maximum number of leaf nodes	'None'

The model prediction results are shown in Table 4.

Table 4. Decision Tree Model Prediction Results

	Check Accuracy Rate	Recall Rate	F1 Score	Number of Supported Samples
0	0.94	0.96	0.95	85048
1	0.90	0.92	0.91	85541
Macro Average	0.92	0.94	0.93	170589
Weighted Average	0.92	0.94	0.93	170589

4.2 Adaboost Model

The AdaBoost algorithm is an integrated iterative algorithm, and the core idea is to train different classifiers (weak classifiers) for the same training set, and then aggregate these weak classifiers to form a more powerful final classifier (strong classifier).

In this thesis, the weak classifier uses the decision tree classifier described above, and the decision tree classifier uses the information gain calculation for the selection of split attributes.

Based on the observation of the results of Adaboost prediction based on decision trees, the parameters are shown in Table 5.

Table 5. AdaBoost model parameter settings

Parameters	Meaning	Set value
base_estimator	Base classifier	Decision Trees
n_estimators	Number of base classifier boosts (cycles)	200
learning_rate	Learning rate, indicating the rate of gradient convergence	0.6
algorithm	Model Enhancement Guidelines	SAMME

The model prediction results are shown in Table 6.

Table 6. Prediction results of AdaBoost model

	Check Accuracy rate	Recall Rate	F1 Score	Number of Supported Samples
0	0.98	0.99	0.98	85198
1	0.96	0.97	0.96	85391
Macro Average	0.97	0.98	0.97	170589
Weighted average	0.97	0.98	0.97	170589

4.3 Comparative Analysis of Model Prediction Results

The performance evaluation of the prediction results modeled by the two machine learning algorithms were compared and the results are shown in Tables 7.

From the above table, it can be seen that the overall classification effect of the decision tree model is relatively good, but the construction process of the decision tree requires several calculations to sort the feature attributes in the data set, which affects the computational efficiency of the algorithm to some extent.

The Adaboost integrated algorithm has improved the result performance metrics compared to the separate decision tree algorithm, and it can be found that the prediction for both positive and negative samples has been improved to different degrees. Adaboost can adaptively adjust the assumed error rate based on the feedback from the weak classifiers, and the execution is efficient.

Table 7. Prediction Results of Two Algorithm Models

	Check accuracy rate	Recall Rate	F1 Score
Decision Trees	0.92	0.94	0.93
Adaboost	0.97	0.98	0.97

5. Conclusion

With the development of information technology and the progress of the times, more and more new credit card fraud methods are emerging. Although we cannot detect people with potential fraudulent behavior at the time of card issuance in time, we can accurately predict users who are about to be likely to commit fraud through their transaction behavior characteristics and their user profiles and take other coercive measures to stop them in time. In this paper, based on credit card fraud data of unbalanced categories, we adopt the Adaboost method of ensemble learning, which overcomes the weak generalization ability, low prediction accuracy of negative samples and slow training speed brought by single model. In addition, we have used SMOTE oversampling to process the dataset so that more potential features can be obtained during training, which better enhances the detection accuracy and recall rate of model prediction and provides a possible means for future credit card anti-fraud.

References

- [1] Xu, X.J. Credit Card Fraud Detection Based on Imbalanced Data Set [D]. Yangtze University, 2021. DOI:10.26981/d.cnki.gjhsc.2021.000326.
- [2] The Average Number of Credit Card Transactions Per Day & Year. <https://www.cardrates.com/advice/number-of-credit-card-transactions-per-day-year/>.
- [3] 2021 China's Credit Card Issuance Scale. <https://www.chyxx.com/industry/1106638.html>.
- [4] BROWN I, MUES C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets [J]. Expert Systems with Applications, 2012, 39(3): 3446 - 3453.
- [5] TAVALLAEE M, STAKHANVA N, GHORBANI A A. Toward credible evaluation of anomaly-based intrusion-detection methods [J]. IEEE Transactions on Systems, Man & Cybernetics, Part C: Applications and Reviews, 2010, 40(5): 516 - 524.
- [6] LIU Y-H, CHEN Y-T. Total margin based adaptive fuzzy support vector machines for multiview face recognition [C] // Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics. Washington, DC: IEEE Computer Society, 2005, 2: 1704 - 1711.
- [7] MAZUROWSKI M A, HABAS P A, ZURADE J M, et al. Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance [J]. Neural Networks, 2008, 21(2 /3) : 427 - 436.
- [8] Mo, Z. Gai, Y.R. Fan, G.L. Credit card fraud classification based on GAN-AdaBoost-DT imbalance classification algorithm[J]. Journal of Computer Applications, 2019, 39(02):618-622.
- [9] Kokkinaki A I. On atypical database transactions: identification of probable frauds using machine learning for user profiling [C] //Proc of IEEE Knowledge and Data Engineering Exchange Workshop.Piscataway, NJ: IEEE Press, 1997: 107-113.
- [10] D. Ignatov and A. Ignatov, "Decision Stream: Cultivating Deep Decision Trees," 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), Boston, MA, USA, 2017, pp. 905-912, doi: 10.1109/ICTAI.2017.00140.
- [11] Maes S, Tuyls K, Vanschoenwinkel B, et al. Credit card fraud detection using Bayesian and neural networks [C] //Proc of the 1st International NAISO Congress on Neuron Fuzzy Technologies. 2002: 261-270.
- [12] E. A. Morales, B. M. Ramos, J. A. Aguirre and D. M. Sanchez, "Credit Risk Analysis Model in Microfinance Institutions in Peru Through the use of Bayesian Networks," 2019 Congreso Internacional

de Innovación y Tendencias en Ingeniería (CONIITI), Bogota, Colombia, 2019, pp. 1-4, doi: 10.1109/CONIITI48476.2019.8960694.

- [13] M. Zamini and G. Montazer, "Credit Card Fraud Detection using autoencoder based clustering," 2018 9th International Symposium on Telecommunications (IST), Tehran, Iran, 2018, pp. 486-491, doi: 10.1109/ISTEL.2018.8661129.
- [14] H. Tingfei, C. Guangquan and H. Kuihua, "Using Variational Auto Encoding in Credit Card Fraud Detection," in IEEE Access, vol. 8, pp. 149841-149853, 2020 , doi: 10.1109/ACCESS.2020.3015600.
- [15] Z. Zhang, K. Niu and Y. Liu, "A Deep Learning Based Online Credit Scoring Model for P2P Lending," in IEEE Access, vol. 8, pp. 177307-177317, 2020, doi: 10.1109/ACCESS.2020.3027337.
- [16] SEARLE S R. Linear Models for Unbalanced Data[M]. New York: John Wiley& Sons, 1987:145 - 153.