Proceedings of ICFTBA 2024 Workshop: Finance in the Age of Environmental Risks and Sustainability
DOI: 10.54254/2754-1169/94/2024OX0182

Identification and Analysis of Credit Card Fraud Based on Machine Learning Methods

Meng Zhang[1,a,*]

[1]*School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, United Kingdom*
a. Zhangmeng8128@gmail.com
**corresponding author*

Abstract: Credit card has become an indispensable payment tool in People's Daily life. However, credit card fraud has also increased, resulting in huge financial losses for banks and individuals. Based on the real world credit card transaction data, this paper uses a variety of machine learning algorithms to identify and analyze credit card fraud. Firstly, the problem of missing and unbalanced data is solved through data preprocessing and feature selection, and important features are screened out, Such as transaction amount, product code, payment card type, etc. Then, integrated logistic regression, random forest, LightGBM, XGBoost, and Stacking models were constructed, and the models were trained and evaluated through cross-validation and parameter optimization. The experimental results show that the integrated Stacking model performs best in credit card fraud detection, with an AUC(Area Under the Curve) value of 0.960, which can accurately identify fraudulent transactions and provide an effective fraud warning mechanism for financial institutions. This mechanism significantly benefits both financial institutions and credit card users by reducing fraud losses and enhancing transaction security. However, it is acknowledged that the study has limitations, including the potential impact of data imbalance on model performance, the need for further testing of model generalization ability to new fraud patterns, and the importance of optimizing real-time performance to ensure practical applicability.

Keywords: Credit card fraud detection, Machine learning, Feature selection, Stacking integrated model.

1. Introduction

As a convenient payment tool, credit card has been widely used in modern society. However, credit card fraud is also common, causing serious financial losses to banks and individuals. According to statistics released by the Federal Trade Commission (FTC) in 2023, tens of billions of dollars are lost each year due to credit card fraud. Therefore, establishing an effective credit card fraud detection system to detect and prevent fraud in time is of great significance for maintaining financial stability and protecting consumers' rights and interests.

In the field of credit card fraud detection, relevant scholars have conducted extensive research. Early research often used traditional statistical methods, such as decision trees [1], Bayesian networks [2], etc. With the rapid development of machine learning technology, more and more scholars are beginning to apply machine learning algorithms to credit card fraud detection. For example, Zhou

DOI: 10.54254/2754-1169/94/2024OX0182

proposed a fraud detection method based on decision trees and Boolean functions [3]; Lu et al. used neural network models for fraud identification [4]; Xu compared the performance of decision trees and support vector machines in fraud detection [5]; Based on cost sensitive machine learning methods, Chen constructed new features through feature engineering and applied them to decision tree models [6].

In recent years, significant progress has been made in the research of credit card fraud detection. Li proposed a credit card risk management scheme based on logistic regression and decision tree [7]; Ding constructed a fraud detection prototype system on a deep belief network [8]; Mao proposed a hybrid credit card fraud detection model AWFD, which improves the recognition rate of fraud detection by integrating anomaly detection and integrated models [9]. However, although significant progress has been made in improving the accuracy and efficiency of fraud detection through existing research, existing methods still face challenges in the face of increasingly complex and ever-changing fraud methods, such as data imbalance, the recognition ability of new fraud patterns, and real-time performance optimization. Based on this, this article summarizes previous research and further explores the application of various machine learning algorithms in credit card fraud detection, with a special focus on solving data imbalance problems, feature selection and optimization, and model integration strategies.

Based on real world credit card transaction data, this paper employs machine learning algorithms to identify and analyze credit card fraud. The structure of the full text is as follows: Chapter 2 introduces related theories and methods, including logistic regression, random forest, Light Gradient Boosting Machine (LightGBM for short), eXtreme Gradient Boosting (XGBoost for short), and Stacking integrated models. Chapter 3: Data preprocessing and feature selection; Chapter 4 builds a credit card fraud prediction model and analyzes the results. Chapter 5 summarizes the full text and looks forward to the future research direction.

2. Related Theories and Methods

2.1. Logistic Regression

Although the name logistic regression includes "regression", it is actually a widely used classification algorithm, especially for binary classification problems. The core idea is to map the continuous output values of the linear model to the (0,1) interval through the sigmoid function on the basis of linear regression, thereby transforming them into a probability value. This probability value represents the likelihood that the sample belongs to a positive class. Logistic regression models are simple, fast to train, and easy to understand and interpret. Model parameters can directly reflect the contribution of each feature to the prediction results. The sigmoid function is shown in (1):

$$g(z) = \frac{1}{1+e^{-z}}, \tag{1}$$

Among them, z is the output of the linear regression model, and g (z) is the probability value transformed by an S-shaped function. The function curve is shown in Figure 1:

DOI: 10.54254/2754-1169/94/2024OX0182

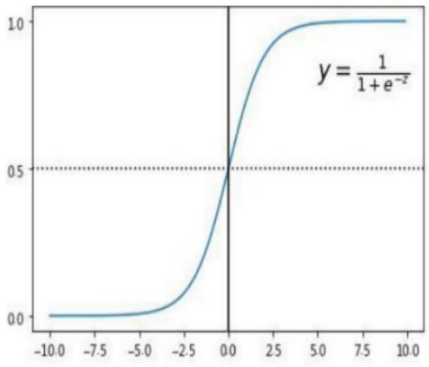

Figure 1: Sigmoid function curve

The loss function of logistic regression usually uses log loss, which takes the form of:

$$P(Y = 1 \mid x) = \frac{\exp(\beta^T x)}{1+\exp(\beta^T x)}, \tag{2}$$

As shown in (2), where (x) is the total number of samples, (y) is the true label of the first sample, and (P) is the probability value predicted by the model.

2.2. Random Forest

Random Forest is an ensemble learning method that improves the generalization ability of a model by constructing multiple decision trees and integrating their prediction results. When constructing each decision tree, random forests not only randomly extract samples from the original dataset (known as bootstrap sampling), but also introduce randomness when selecting splitting attributes, that is, randomly selecting a subset of attributes from all attributes and then selecting the optimal attribute for splitting. This dual randomness enables random forests to reduce model variance and improve model stability.

The advantages of random forests encompass their capability to handle high-dimensional data without requiring explicit feature selection, their impressive classification performance even with imbalanced datasets, their innate ability to evaluate the significance of individual features, and their robust resistance to noise.

2.3. LightGBM

LightGBM (Light Gradient Boosting Machine) is an efficient decision tree algorithm based on a gradient boosting framework. It significantly reduces the computational complexity and memory consumption of the model through optimization techniques such as histogram based algorithm and exclusive feature bundling (EFB), enabling LightGBM to efficiently process large-scale datasets.

The histogram algorithm employed by LightGBM involves discretizing continuous floating-point eigenvalues into k integers, subsequently constructing a histogram with a defined width of k. During the traversal of data, these discretized values serve as indices to accumulate statistical information within the histogram. The optimal segmentation point is then identified based on the distribution of these discretized values within the histogram, significantly reducing the overall computational complexity. Furthermore, the Mutual Exclusion Feature Binding (EFB) algorithm leverages the mutual exclusivity among features, where specific features do not concurrently assume non-zero values. By binding such mutually exclusive features into a single feature, the EFB algorithm effectively diminishes the total number of features processed, thereby contributing to a further reduction in computational complexity.

DOI: 10.54254/2754-1169/94/2024OX0182

LightGBM supports multiple loss functions and evaluation metrics, allowing for flexible handling of different classification and regression tasks.

2.4. XGBoost

XGBoost (eXtreme Gradient Boosting) is an optimized gradient enhanced decision tree algorithm proposed by Chen et al [10]. XGBoost makes a number of improvements on the basis of the traditional algorithm GBDT (Gradient Boosting Decision Tree), including introducing second-order Taylor expansions to approximate the loss function more accurately, and adding regularization terms to control the complexity of the model and prevent overfitting.

The main advantages of XGBoost include: supporting multiple types of loss functions, including regression, classification, etc; Introducing regularization terms improves the model's generalization ability; Support column sampling to further reduce computational complexity; Support custom evaluation indicators with strong flexibility.

2.5. Stacking Integration Model

Stacking (Stacked Generalization) is an advanced ensemble learning method that trains a secondary learner by taking the predicted results of multiple base learners as input, thereby integrating the advantages of multiple base learners and improving the overall predictive performance of the model.

The process of stacking is usually divided into two stages:

1. Basic learner training stage: Firstly, use the original training set to train multiple different basic learners. Then, use these base learners to predict the training set and obtain a new training set (i.e. the prediction results of the base learners).

2. Secondary learner training stage: Use the new training set obtained in the first stage to train the secondary learner. Finally, a secondary learner is used to predict the test set and obtain the final prediction result.

As shown in Figure 2,The advantage of Stacking is that it can fully utilize the advantages of different base learners, integrate these advantages through the learning of secondary learners, and thereby improve the prediction accuracy and stability of the overall model. However, Stacking has a high computational complexity and requires careful selection of base learners and secondary learners to avoid overfitting.

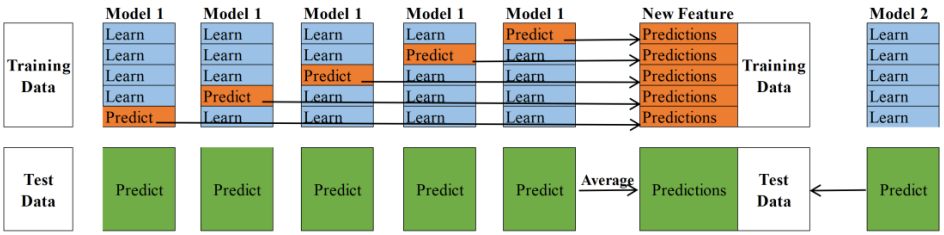

Figure 2: Example Stacking Algorithm

3. Data Preprocessing and Feature Selection

3.1. Data Source and Explanation

The dataset used in this article is sourced from the IEEE-CIS fraud detection competition, which was provided by Vesta and contains real e-commerce transaction records. The dataset consists of two parts: transaction table and identity table. By merging these two tables, This paper obtained a summary table containing 144233 transaction records and 434 attributes. Each transaction record details various

DOI: 10.54254/2754-1169/94/2024OX0182

attributes of the transaction, such as transaction amount, transaction time, transaction product code, payment card information, etc. Among them, the target variable isFraud is a binary variable used to identify whether there is fraudulent behavior in the transaction, with a value of 0 indicating non fraudulent transactions and a value of 1 indicating fraudulent transactions.

3.2. Data Description

Descriptive statistical analysis of the dataset is crucial before data preprocessing, as it helps us better understand the characteristics and distribution of the data. Due to the unclear meanings of some features in the dataset, we will select several attributes with more comprehensive descriptions of their meanings for detailed analysis.

3.2.1. Transaction Amount Analysis

The TransactionAMT property in the dataset represents the payment amount during the transaction. As shown in Figure 3, although the proportion of fraudulent transactions in total transactions is very low (0.17%), the proportion of fraudulent transactions in total transaction amount is relatively high, reaching 8.34%, indicating that the average amount of fraudulent transactions may be higher than that of non-fraudulent transactions.

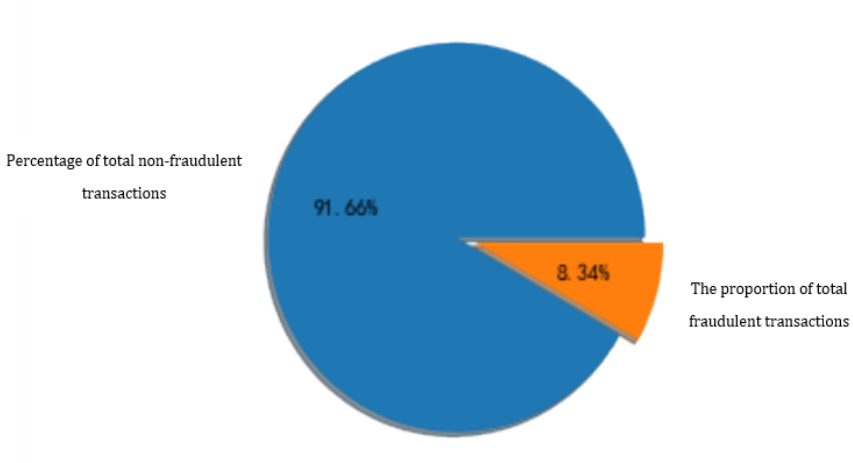

Figure 3: The proportion of transaction amount

Table 1 shows the quartile statistical information of payment amounts for fraudulent and non fraudulent transactions:

Table 1: Quartile table of fraudulent and non-fraudulent transaction amounts

isFraud	count	mean	std	min	25%	50%	75%	max
0	121,804	82.096	98.003	0.141	24.467	50.000	100.000	1,800.00
1	10,207	87.798	106.881	0.181	24.000	50.000	100.047	1,800.00

From the table 1, it can be seen that the average payment amount for fraudulent transactions (87.798$) is slightly higher than the average payment amount for non fraudulent transactions (82.096$), although the difference between the two is not significant. In addition, the maximum and median values for fraudulent and non fraudulent transactions are the same, indicating that fraudulent transactions may be hidden in transactions with normal transaction amounts, increasing the difficulty of detection.

DOI: 10.54254/2754-1169/94/2024OX0182

3.2.2. Analysis of Trading Products

The ProductCD attribute in the dataset represents the product code for each transaction, giving four possible values: C, H, R, and S. By analyzing the distribution of ProductCD, we can get the proportion of each product in the transaction and the fraud situation of each product (see Figure 4).

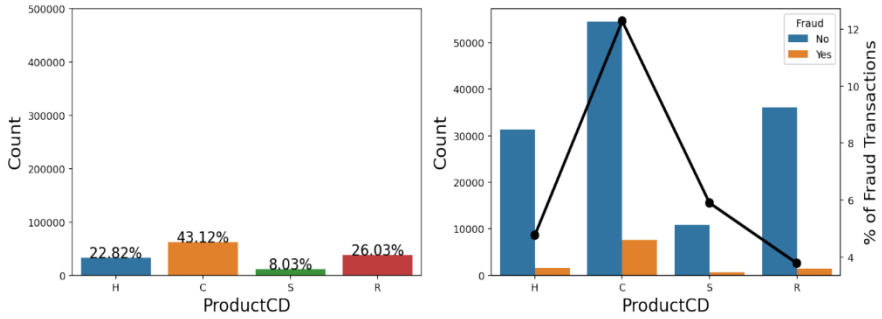

Figure 4: Distribution of ProductCD

Figure 4 shows the quantity and fraud rate of various products in the transaction. It can be seen that category C products have the highest transaction volume, but the fraud rate is also relatively high; Although the transaction volume of R category products is not low, the fraud rate is the lowest. Therefore, in credit card transactions, it is important to focus on the target audience for C-category products to reduce the probability of fraudulent transactions.

3.2.3. Transaction Card Analysis

There are multiple attributes related to payment card information in the dataset, among which card4 is an important categorical variable that represents the brand of the payment card. Card4 has four possible values: mastercard visa、discover、American express, Some data is still missing. By analyzing the distribution of card4 and its corresponding fraud situation (as shown in Figure 5), This text can obtain the proportion and fraud rate of different payment card brands in transactions.

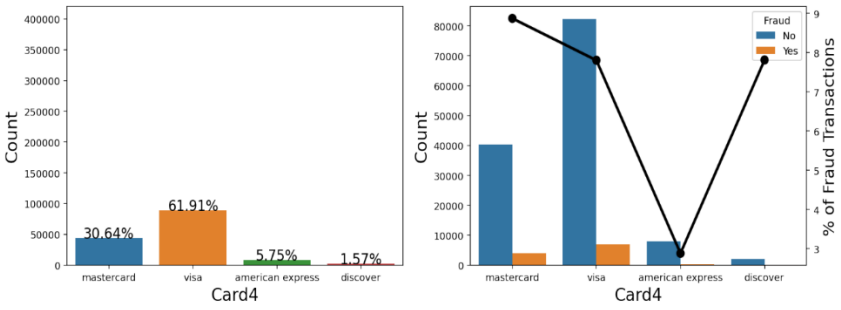

Figure 5: Card4 distribution map

Figure 5 shows that mastercard cards have the highest rate of fraud. As a result, financial institutions can pay special attention to transactions using mastercard when conducting credit card fraud detection.

DOI: 10.54254/2754-1169/94/2024OX0182

3.3. Data Preprocessing

In order to build an effective fraud detection model, we need to perform a series of preprocessing steps on the original dataset.

3.3.1. Missing Value Handling

The dataset may contain some missing values, which may be caused by omissions or errors during the data collection process. In order to not affect the training effect of the model, we need to handle missing values. Firstly, we removed features with a missing rate greater than 40% and meaningless features (such as transaction IDs, timestamps, etc.), as these features may contain a large amount of irrelevant information and may not contribute significantly to model predictions. Then, for the remaining missing values, we filled them in differently based on the feature type: for numerical features, we filled them in using the mean; For categorical features, we use mode to fill in. After filling, there are no missing values in the dataset.

3.3.2. One Hot Encoding

The dataset contains some categorical features, such as transaction product codes, payment card types, etc. These features are represented in textual form in the original dataset and cannot be directly used for model training. Therefore, we need to perform One Hot Encoding on these categorical features. Unique hot encoding is a method of converting categorical variables into numerical forms that are easy to process by machine learning algorithms. After undergoing unique hot encoding processing, each categorical variable is transformed into a vector composed of 0 and 1, with a length equal to the number of categories of the categorical variable.

3.3.3. Data Standardization

Data standardization is an important step in data preprocessing aimed at eliminating the impact of dimensional differences between different features on model training. Due to the fact that numerical features in the dataset may have different dimensions and distribution ranges, if directly used for model training, it may lead to excessive dependence or neglect of certain features by the model. Therefore, we need to standardize numerical features by scaling their values to the same range. The standardized formula used in this article is:

$$X_{normal} = \frac{X - X_{min}}{X_{max} - X_{min}}. \tag{3}$$

As shown in (3), where (X) is the original eigenvalue, all numerical eigenvalues are scaled to the range [0,1] after normalization.

3.3.4. Partition Dataset

In order to evaluate the predictive performance of the model, we need to divide the data set into a training set and a test set. The training set is used to train and optimize the model, and the test set is used to evaluate the generalization ability of the model. As shown in Table 2, this paper divides the data set into a training set and a test set in an 8:2 ratio, where the training set contains 80% of the data samples and the test set contains the remaining 20% of the data samples.

DOI: 10.54254/2754-1169/94/2024OX0182

Table 2: Dataset partitioning table

Data types	Non-fraudulent data	Fraudulent data	Total
Training set	106,221	9,043	115,275
Test set	26,694	2,275	28,958
Total	132,915	11,318	144,233

3.3.5. Unbalanced Data Processing

We adopted a sampling method combining SMOTE (Synthetic Minority Over sampling Technique) and Tomek Links to address the issue of class imbalance in the dataset. SMOTE is an oversampling technique that generates new minority class samples through linear interpolation to increase the number of minority classes and balance class distribution. However, simple SMOTE oversampling may introduce some noise samples or overgeneralization issues. Therefore, we combined Tomek Links undersampling technology to further improve data quality. Tomek Links refer to a pair of sample points that are nearest neighbors to each other and belong to different categories. Deleting these sample points can remove noisy and overlapping samples on the boundary, improving the purity of the dataset. After processing with SMOTE and Tomek Links, the proportion of positive and negative samples in the training set became 1:1, effectively alleviating the problem of class imbalance.

3.4. Feature Selection

Feature selection is one of the key steps in building efficient machine learning models. By selecting features that are highly correlated with the target variable and have strong predictive ability, we can reduce the complexity of the model and improve prediction accuracy. This article adopts a combination of univariate selection method and recursive elimination method for feature selection.

3.4.1. Single Variable Selection Method

The univariate selection method is a simple and intuitive feature selection method that selects features by calculating the correlation or statistics between each feature and the target variable. This article uses chi square test (for categorical features) and Pearson correlation coefficient (for numerical features) to evaluate the correlation between features and target variables. Call the f_classif function in Python to calculate the score and p-value of the target variable. Then, select features with high correlation as candidate feature sets.

3.4.2. Recursive Elimination Method

Recursive elimination is a model-based feature selection method that gradually removes unimportant features by repeatedly constructing the model and evaluating the importance of the features. This article uses a random forest model as the base model to evaluate the importance of features. Firstly, we train a random forest model using all candidate features and calculate the importance score of each feature based on the model's prediction results. Then, we delete the feature with the lowest importance score and retrain the model to update the importance score of the feature. This process is repeated until the predetermined number of features is reached or the performance of the model no longer significantly improves. In the end, we obtained a subset containing 124 important features for

DOI: 10.54254/2754-1169/94/2024OX0182

subsequent modeling analysis. These features have high predictive ability and stability in fraud detection tasks.

4. Construction of Credit Card Fraud Prediction Model

4.1. Model Construction and Parameter Optimization

4.1.1. Logistic Regression Model

As shown in Figure 6, a logistic regression model is established by calling the LogisticRegression function in Python, and the model parameters are optimized using grid search method. The final optimal parameters obtained are C=100 and Penalty=L2. The prediction accuracy on the test set is 82.7%, and the AUC value is 0.855.

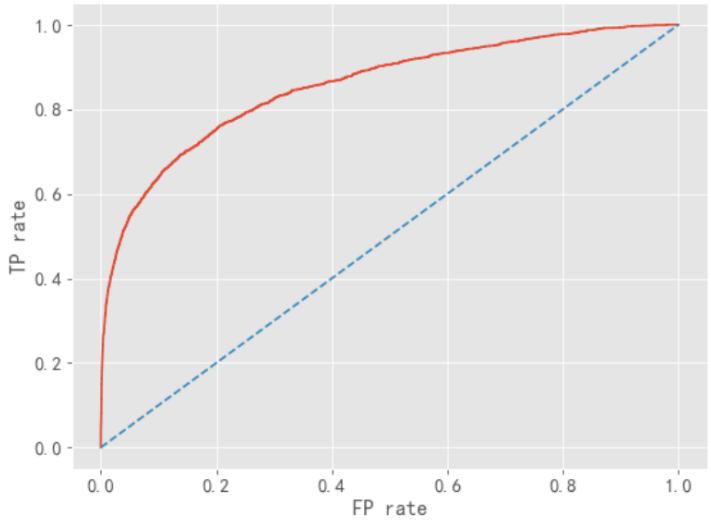

Figure 6: ROC curve of Logistic regression model

4.1.2. Random Forest Model

Establish a random forest model using the RandomForestClassifier function, and optimize the model parameters using Bayesian optimization methods. The final optimal parameter obtained is n_estimators=1410, max_depth=8,min_samples_leaf=82. The prediction accuracy on the test set is 93.3%, with an AUC value of 0.916.

DOI: 10.54254/2754-1169/94/2024OX0182

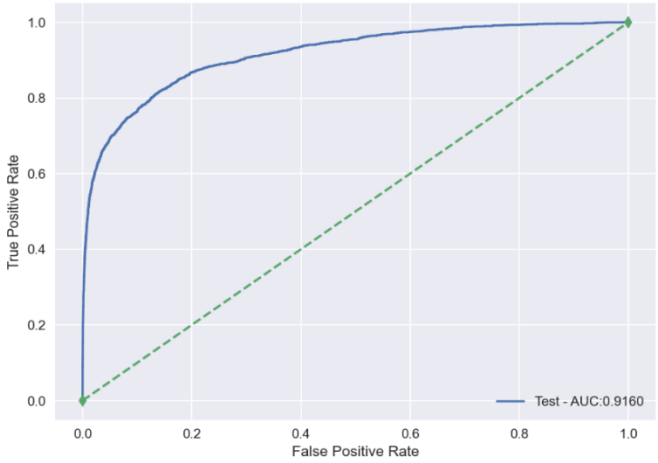

Figure 7: ROC curve of RF model

4.1.3. LightGBM Model

Establish a LightGBM model using the LGBMClassifier function and optimize the model parameters using Bayesian optimization methods. The optimal parameters obtained are shown in Table 3.

Table 3: Meaning of LightGBM model parameters

Parameter names	Optimal value
max_depth	6
num_leaves	203
feature_fraction	0.7523
bagging_fraction	0.6832
bagging_freq	6
min_data_in_leaf	53
min_split_gain	0.0728
reg_alpha	0.60
reg_lambda	2.44
learning_rate	0.06
n_estimators	2604

DOI: 10.54254/2754-1169/94/2024OX0182

As shown in Figure 8, the prediction accuracy of LightGBM on the test set is 97.0%, with an AUC value of 0.955.

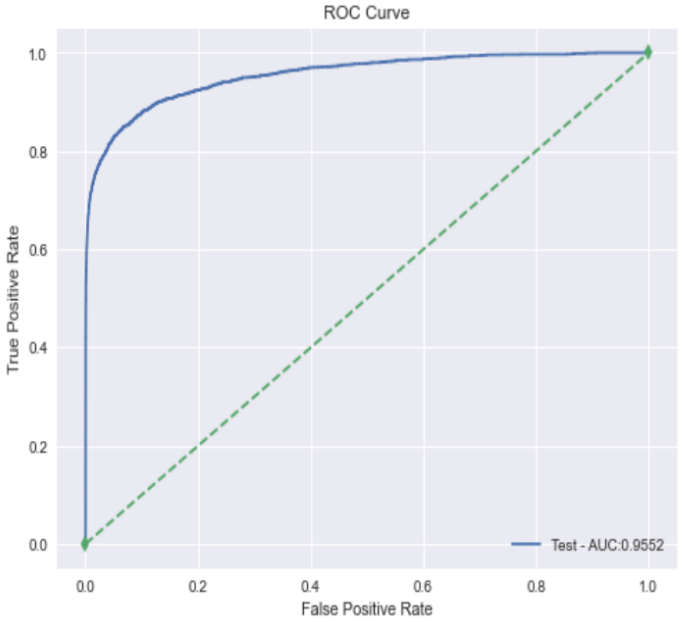

Figure 8: ROC curve of LightGBM model

4.1.4. XGBoost Model

Call the XGBClassifier function to establish the XGBoost model, and optimize the model parameters using Bayesian optimization methods. The prediction accuracy on the test set is 96.9%, indicating that the XGBoost model performs well in credit card fraud detection tasks and can accurately distinguish between fraudulent and non fraudulent transactions.

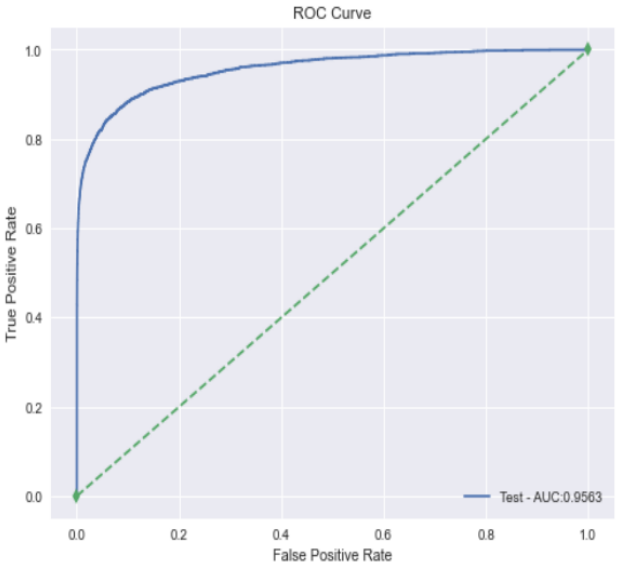

Figure 9: ROC curve of XGBoost model

DOI: 10.54254/2754-1169/94/2024OX0182

4.1.5. Stacking Integration Model

In order to further improve the predictive performance of the model, we constructed a Stacking ensemble model. In this model, we use logistic regression model, random forest model, and LightGBM model as the first layer learners, and XGBoost model as the second layer learners. Firstly, we train the first layer learner with the optimal parameters and use their predicted results as input for the second layer learner. Then, parameter optimization was performed on the second layer learner (XGBoost), and the optimal parameters obtained are shown in Table 4.

Table 4: Table of optimal parameters for XGBoost in the Stacking model

Parameter names	Optimal value
max_depth	2
subsample	0.7
colsample_bytree	0.7
gamma	0.0818
reg_alpha	8.7663
reg_lambda	4.1700
learning_rate	0.0445
n_estimators	1121

On the test set, the prediction accuracy of the Stacking integrated model is 96.9%, and the AUC value is 0.960, which has improved compared with the single model.

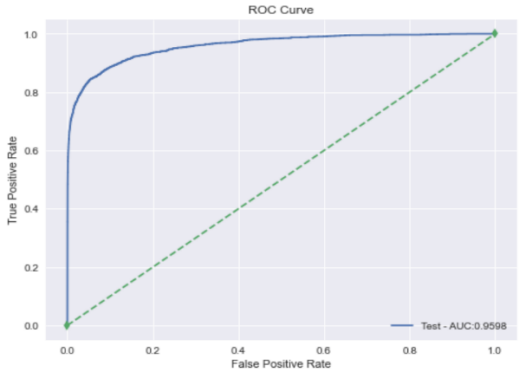

Figure 10: Stacking Model ROC Curve

DOI: 10.54254/2754-1169/94/2024OX0182

4.2. Comparison and Analysis of Model Results

In order to comprehensively evaluate the performance of each model, we compared the prediction results of logistic regression model, random forest model, LightGBM model, XGBoost model, and Stacking ensemble model on the test set. The main evaluation indicators include accuracy, precision, recall, F1 score, and AUC value. The comparison results are shown in Table 5.

Table 5: Model Comparison Table

Model	accuracy	Accuracy rate	Recall rate	F1 score	AUC value
Logistic regression	0.716	0.161	0.602	0.282	0.744
Random forest	0.822	0.452	0.567	0.504	0.805
LightGBM	0.860	0.775	0.601	0.672	0.844
XGBoost	0.858	0.770	0.587	0.668	0.845
Stacking	0.858	0.778	0.586	0.671	0.850

From the comparison results, it can be seen that the Stacking integrated model performs excellently in all indicators, especially reaching the highest AUC value, indicating its optimal overall performance. Although the logistic regression model is easy to implement, its accuracy is low due to data imbalance issues. The random forest model performs well in accuracy and recall, but its accuracy and F1 score are relatively low. LightGBM and XGBoost models are outstanding in a single model, but they are still slightly inferior when compared with the Stacking integrated model.

5. Conclusion

This article is based on real-world credit card transaction data and uses various machine learning algorithms to identify and analyze credit card fraud behavior. Through data preprocessing and feature selection, the problems of data loss and imbalance were solved, and integrated models such as logistic regression, random forest, LightGBM, XGBoost, and Stacking were constructed. The experimental results show that the Stacking ensemble model performs the best in credit card fraud detection, with an AUC value of 0.960, and can accurately identify fraudulent transactions. In addition, LightGBM and XGBoost models have also demonstrated good predictive performance, providing financial institutions with various effective fraud warning schemes.

Although this paper has made some achievements in the field of credit card fraud detection, there are still shortcomings, which need to be expanded and optimized in many aspects in the future research. Specifically, the potential features in the transaction data can be further mined, and the prediction accuracy of the model can be improved through strategies such as feature crossing and feature selection. Explore more diversified model fusion methods, such as weighted voting, dynamic integration, etc., to enhance the performance of integrated models; At the same time, the research results are transformed into practical applications, and efficient real-time credit card fraud detection

DOI: 10.54254/2754-1169/94/2024OX0182

system is developed to provide instant fraud early warning services for financial institutions. In addition, it is necessary to continue to study and apply more advanced unbalanced data processing techniques, such as cost-sensitive learning, adaptive sampling, etc., to effectively address the challenge of category imbalance. Through these continuous optimization and improvement, we are confident that we can build a more efficient and accurate credit card fraud detection system, and contribute more to maintaining financial stability and protecting consumer rights and interests.

References

[1] *Cao, S., & Min, J. (2017) Research on the Prediction Model of Credit Card High end Customer churn Based on Decision Tree. Beijing Financial Review, 03, 69-78.*

[2] *Lu, Z. (2018) Customer credit evaluation and research based on Bayesian networks. Wuhan: Huazhong University of Science and Technology.*

[3] *Zhou, M. (2011) Credit Card Reputation Detection Based on Decision Tree Method. Journal of Zhongyuan University of Technology, 22 (04), 75-78.*

[4] *Lu, H., Wei, Y, Jiao, L. (2023) A Credit Card Postloan Risk Rating Model and Empirical Study Based on GA-BP Neural Network. Operations Research and Management, 32 (06), 192-198.*

[5] *Xu, Y. (2011) Credit Card Fraud Detection Based on Support Vector Machine. Computer Simulation, 28 (08), 376-379+384.*

[6] *Chen, Q. (2018) Machine learning methods for identifying credit card fraud: a comparative study. China High tech, 24, 66-70.*

[7] *Li, J. (2023) Research on credit card overdue prediction method based on integrated algorithms. Chuangchun: Changchun University of Technology.*

[8] *Ding, W. (2015) Research on Credit Card Transaction Fraud Detection Based on Deep Learning Technology. Shanghai: Shanghai Jiao Tong University.*

[9] *Mao, M. (2021) A Hybrid Credit Card Fraud Detection Model. Computer Knowledge and Technology, 17(02), 194-196.*

[10] *Chen, T. (2016) XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.*