Stacking Model for Financial Fraud Detection with Synthetic Data

Zichuan Fu(✉)

Department of Computer Science, The University of Sheffield, Sheffield S10 2TN, UK
fzkuji@gmail.com

Abstract. With the fast pace development of the Internet nowadays, financial frauds have also emerged continuously, which has seriously affected the development of the financial sector. Due to the lack of data in the financial field and the loose structure of transaction information, financial fraud detection remains a significant challenge. Based on the traditional machine learning model, this paper combines the three basic logistic regression models, support vector machine and random forest, and designs a two-layer stacking prediction model to detect financial transaction fraud. For unbalanced samples, this article uses up-sampling, under-sampling, and fusion methods to test and help search for optimal parameters through GridSearchcv. The final experiment shows that the Stacking model has a 97% recall rate and 87% accuracy for fraud samples on synthetic financial datasets. It can quickly detect most fraud samples while keeping false positives within a reasonable range. The model designed in this paper enriches the research of model fusion in financial fraud detection.

Keywords: Stacking Model · Financial Fraud Detection · Synthetic Financial Datasets · Logistic Regression · Support Vector Machines · Random Forest

1 Introduction

The development and popularization of Internet technology have promoted the transformation of mass consumption patterns. Various mobile online payments have appeared, significantly increasing the volume of online financial transactions. Worldwide e-commerce transaction volume has increased from 1336 billion in 2014 to 4280 billion in 2020 [3]. However, with the increase in transaction data, online financial transaction fraud has also grown and gradually turned online, causing tens of billions of losses every year. Especially in the past two years, there have been many pandemic-related scams, using fear of the new coronavirus to trick them into providing funds, thereby avoiding bank security measures [4].

Financial fraud has brought considerable losses to this industry. The research on financial anti-fraud provides a viable path for future prevention and control. Anti-online financial fraud can minimize the possibility of individuals defrauded in financial investments and online transactions and the losses they bear. However, traditional manual detection methods can no longer adapt to the ever-changing market. Financial fraud

D. Qiu et al. (Eds.): ICBBEM 2022, AHIS 5, pp. 60–67, 2023.
https://doi.org/10.2991/978-94-6463-030-5_8

shows new features such as specialization, industrialization, concealment, and cross-regional. New machine learning and techniques can identify the current anomalies and continuously optimize with the large-scale data, which is very suitable for online finance with huge transaction volume today, and the cost is much lower than traditional methods. The sound development of data-related industries first requires the truthfulness and accuracy of the data itself. Real-time features and data quality are crucial for establishing a good Internet financial environment. Therefore, the application of machine learning to online financial anti-fraud has excellent potential.

2 Related Work

Considering the outstanding performance on classification problems, applying machine learning to financial fraud detection is natural. Typically, they can be classified into unsupervised learning and supervised learning.

Some recent methods tried to combine supervised learning and unsupervised techniques to improve the proposed model's performance further. Carcillo et al. demonstrated that using unsupervised outlier scores to extend the feature set of a supervised fraud detection classifier achieved better accuracy than the random forest benchmark on the company's transaction dataset [1]. The integrated method considered the various level of aggregation through clustering. However, the results were not convincing on global and local granularity situations, probably because of misinterpretations of unsupervised information. Wang et al. proposed a semi-supervised attentive graph neural network (SemiGNN) to address the problem that few users in the dataset are labeled [5]. A hierarchical attention mechanism exploits the relation between different users completely. The result, tested on a large dataset from a third-party cashless payment, showed that SemiGNN outperformed previous methods such as Xgboost and graph convolutional network (GCN) and had better interpretability.

3 Data

3.1 Introduction of Datasets

Due to the lack of real data sets, this paper uses a synthetic data set, Synthetic Financial Datasets [2], for experiments. The dataset uses PaySim to generate data, which simulates mobile currency transactions based on real samples extracted from one-month financial logs of mobile currency services implemented in African countries. The entire dataset includes more than 6 million transaction data and 1 million user accounts, as the structure demonstrated in Table 1.

The original dataset is large and contains many missings accounts of both sides of the transaction. Therefore, this paper considers two cases in the experiment: the original and after deleting all the missing entries.

Table 1. Overview of the headers in Synthetic Financial Datasets.

Headers	Meanings
Step	The time when the transaction occurred, with a total of 720 h
Amount	Transaction amount
NameOrig, NameDest	Name of the transfer-in and out accounts
OldbalanceOrg, NewbalanceOrig	Account balance of the transfer-out account before and after transaction
OldbalanceDest, NewbalanceDest	Account balance of the transfer-in account before and after transaction
IsFraud	Whether it is a fraud transaction

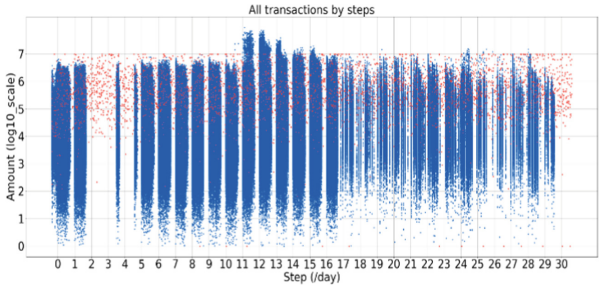

Fig. 1. Transaction statistics with steps. The step lasted 31 days, and log10 processed the transaction amount in advance.

3.2 Features of Datasets

The original data includes 6.3 million transaction data with 743 h, including five transaction types, with transaction amounts ranging from 1 to 50 million. The number of positive and negative samples is extremely unbalanced with a ratio of approximately 800:1.

The statistics of transactions over time are shown in Fig. 1, in which the data step represents time. Most transaction amounts are between 10^2 and 10^7, which also affects the number of transactions.

Firstly, a certain connection between transaction types and amounts is obvious through correlation analysis. The average amount of TRANSFER transactions is the largest, reaching 900,000; the amount of CASH_IN and CASH_OUT is in medium level with about 170,000, and PAYMENT and DEBIT are only around 10,000. Secondly, there is also a certain proportional relationship between the amounts before and after the transaction. However, because of the inevitable relation and missing data, for other correlations, this paper uses the data after deleting all missing entries to perform analysis.

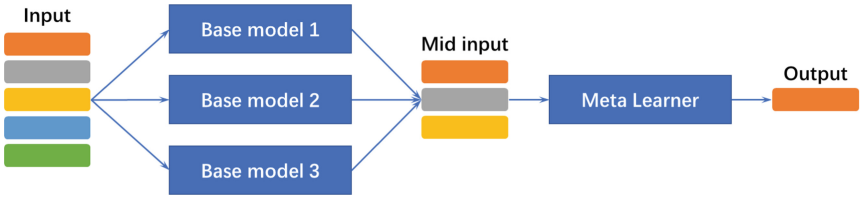

Fig. 2. The demonstration of a staking model with three base models.

4 Model

Logistic Regression, support vector machine (SVM) and Random Forest are used in this paper.

Logistic regression is a generalized linear regression model widely used in various data mining fields and is mainly used to solve two classification problems. SVM is also a binary classification model, and its decision boundary is the hyperplane of the maximum margin for solving the learning sample. Random Forest a classifier that contains multiple decision trees, belongs to the Bagging method in the ensemble learning algorithm.

Stacking refers to the technique of training one model to combine other models. First, train several different models and then use the output of the previous training as input to train a new model to obtain a final model. This paper uses the above-mentioned logistic regression, SVM, and random forest to form a fusion model.

The integrated knowledge can be put on a simple classifier using the stacking method. The Stacking method has passed K-fold cross-validation and has superior performance, which is much better than traditional machine learning. The regular term in the second layer prevents overfitting so that there is no need to adjust parameters and features too much. Using various models for fusion can effectively reduce variance and enhance the robustness of the network.

Stacking has a layered structure. Each layer includes different models, as shown in Fig. 2. There are several operations for all models of each layer:

Divide the training set into n parts. For the m base models of this layer, each one is trained by n-fold cross-validation.

Then, the m model gets the prediction result P_m of the entire training set, and finally, pass $n \times m$ dimensional predictions of all models to next layer.

For test set, n trained different models of each base model are used to make predictions. Therefore, each base model obtains n predictions for each sample. The whole results are averaged to finally get the test set input of the next layer model.

5 Results and Discussion

This section will specifically analyze the experimental part of this article, Sect. 5.1 will introduce the operating environment and configuration of the experiment, Sect. 5.2 will analyze the settings of a single model parameter, Sect. 5.3 will analyze the preprocessed data, and finally, Sect. 5.4 will analyze the optimal solution of the fusion model.

Table 2. Comparison of model performance under different solver and balanced of logistic regression.

	Label	Precision	Recall	Specificity	F1-score	Geo	Iba
lbfgs	0	**1**	**1**	**0.49**	**1**	**0.7**	**0.51**
	1	**0.89**	**0.49**	**1**	**0.63**	**0.7**	**0.46**
linear	0	1	1	0.43	1	0.66	0.46
	1	0.93	0.43	1	0.59	0.66	0.41
linear+ balanced	0	1	0.95	0.96	0.97	0.95	0.91
	1	0.02	0.96	0.95	0.05	0.95	0.91

5.1 Experimental Configuration

The experimental hardware is Intel Core i7-11800H @ 2.30 GHz, and the running memory is 16 GB. The program runs in the Windows 11 environment, using python3.7, scikit-learn, and the GridSearchCV to optimize parameters.

5.2 Single Model Analysis

Apart from the optimal solution, when solver is 'liblinear', and class_weight is 'balanced,' the recall rate reaches 96%, but the accuracy rate is less than 1%. The specific comparison is shown in Table 2.

After manual fine-tuning, the maximum number of iterations was increased to 500 to ensure complete convergence of the model. However, the highest performance of AUC is 0.74, which is still far from the application standard.

Table 3 shows the influence of the coefficient and the performance comparison of different kernel functions.

The penalty coefficients and dimensional search results of GridSearchCV on the polynomial kernel function SVM are shown in Fig. 3. Higher-dimensional kernel functions and larger penalty coefficients improve the performance of the model. Since the final model evaluation results are close to 1, the values are actual values − 0.998 in Fig. 3, and the model performance is consistent with Table 3. Different parameters mainly affect the performance of the recall rate.

5.3 Preprocessed Data

After deleting all missing values, the distribution characteristics of the visualized sample are more obvious. Only a single logistic regression model can obtain 100% accuracy in experiments. After analysis, it turns out that all fraudulent samples have the same feature in the preprocessed dataset. All balance is transferred from one account and stored into the target account. However, the transfer-out account of the non-fraud sample will still retain a certain amount of funds after the transfer, which is easy to identify. Therefore, more challenges exist in the missing data.

Table 3. SVM performance under different parameters. For example, poly-3-1000 represents the final performance of the model with a penalty coefficient of 1000 and a third-order polynomial kernel function.

	Label	Precision	Recall	Specificity	F1-score	Geo	Iba
poly-5-1000	0	1	1	0.7	1	0.84	0.73
	1	0.94	0.7	1	0.81	0.84	0.68
poly-5-1	0	1	1	0.49	1	0.7	0.52
	1	0.99	0.49	1	0.66	0.7	0.47
poly-3-1000	0	**1**	**1**	**0.71**	**1**	**0.84**	**0.73**
	1	**0.98**	**0.71**	**1**	**0.82**	**0.84**	**0.69**
poly-3-1	0	1	1	0.44	1	0.66	0.46
	1	1	0.44	1	0.61	0.66	0.41
linear-\-300	0	1	1	0.46	1	0.67	0.48
	1	0.99	0.46	1	0.62	0.67	0.43

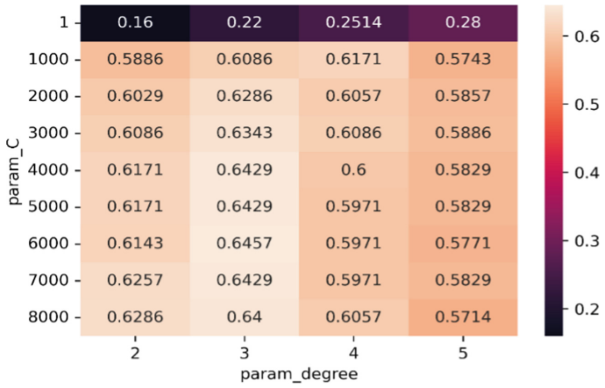

Fig. 3. Two-dimensional heat map of SVM performance under different penalty coefficients and dimensions. The lighter the color, the higher the performance.

5.4 Stacking Model

Aiming at the problem of data imbalance, a single model test shows that the weighted loss function is not good, and it is easy to predict a large number of false-positive results. Although false-positive results are acceptable in fraud detection, too low an index makes the model unusable. Therefore, the experiment tried other methods to deal with unbalanced samples.

The first layer of the Stacking model are fixed, and the second layer still tries to use one of the three models mentioned above. Experiments have compared the effects of the three models on the final prediction results, as shown in Table 4.

Table 4. Performance comparison of different models as the second layer of stacking. Except for the case where SVM considers the poly kernel function, all other models use default parameters.

	Label	Precision	Recall	Specificity	F1-score	Geo	Iba
RF-default	0	**1**	**1**	**0.82**	**1**	**0.91**	**0.84**
	1	**0.96**	**0.82**	**1**	**0.89**	**0.91**	**0.81**
LR-default	0	1	1	0.79	1	0.89	0.8
	1	0.96	0.79	1	0.87	0.89	0.77
SVM-poly-3	0	1	1	0.09	1	0.3	0.1
	1	1	0.09	1	0.17	0.3	0.08
SVM-default	0	1	1	0.25	1	0.5	0.27
	1	0.98	0.25	1	0.4	0.5	0.24

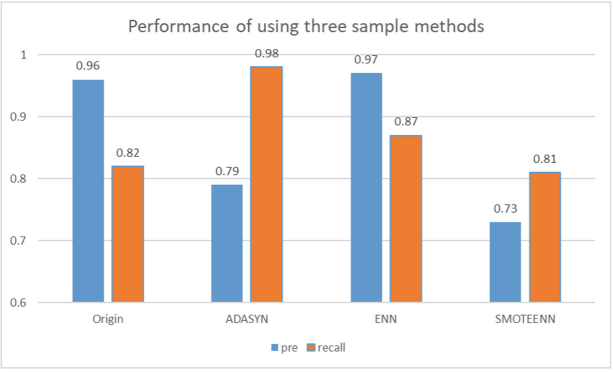

Fig. 4. The comparison of model performance between original data and processed data by three sample models including ADASYN, ENN, and SMOTEENN.

For unbalanced data, the experiment considers using up-sampling, under-sampling, and fusion methods for correction training, corresponding to ADASYN, ENN, and SMOTEENN, respectively. Compared with the model trained on the original data, the processed data archives better performance results, as shown in Fig. 4. Though the recall reached 98% after ADASYN, the training time consumption is unacceptable, making up-sampling unsuitable for big data. Finally, the model using ENN sampling method obtains the best performance of 97% accuracy and 85% recall rate in detecting fraud transactions.

6 Conclusions

In this paper, we design a stacking model to detect financial fraud. Specifically, we used an existing synthetic data set to conduct experiments and analyzed the characteristics under original and preprocessed data for the scarce data of financial fraud. At the same

time, we choose logistic regression, SVM, and random forest as the basic model and adjust the parameters for the optimal solutions. Finally, the stacking model is used for fusion, which increases the performance by 7%. Given the imbalanced characteristics of financial fraud data, the experiment considered up-sampling, under-sampling, and mixed-mode for correction training, and the AUC of the final model arrives 0.95. Since this article uses automatically synthesized data, which has differences from real-world data, future work can focus on using real data to verify the usability of the model.

References

1. Carcillo, Fabrizio, et al. "Combining unsupervised and supervised learning in credit card fraud detection." Information sciences 557 (2021): 317–331.
2. Lopez-Rojas, Edgar, Ahmad Elmir, and Stefan Axelsson. "PaySim: A financial mobile money simulator for fraud detection." 28th European Modeling and Simulation Symposium, EMSS, Larnaca. Dime University of Genoa, 2016.
3. Majchrzak-Lepczyk, Justyna. "Value for the Customer in E-Commerce." (2021).
4. UK, Financial Fraud Action. "The Definitive Overview of Payment Industry Fraud." (2021).
5. Wang, Daixin, et al. "A semi-supervised graph attentive network for financial fraud detection." 2019 IEEE International Conference on Data Mining (ICDM). IEEE, 2019.