Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Fraud detection within bankcard enrollment on mobile device based payment using machine learning*

Hao ZHOU†‡1,2, Hong-feng CHAI1,3,4, Mao-lin QIU2
1*School of Cyber Security, Shanghai Jiao Tong University, Shanghai 200240, China*
2*Department of Risk Control, China UnionPay, Shanghai 200135, China*
3*Office of Board of Directors, China UnionPay, Shanghai 200135, China*
4*Chinese Academy of Engineering, Beijing 100088, China*
†E-mail: zhouhaounionpay@sjtu.edu.cn

Received Sept. 17, 2018; Revision accepted Dec. 16, 2018; Crosschecked Dec. 24, 2018

Abstract: The rapid growth of mobile Internet technologies has induced a dramatic increase in mobile payments as well as concomitant mobile transaction fraud. As the first step of mobile transactions, bankcard enrollment on mobile devices has become the primary target of fraud attempts. Although no immediate financial loss is incurred after a fraud attempt, subsequent fraudulent transactions can be quickly executed and could easily deceive the fraud detection systems if the fraud attempt succeeds at the bankcard enrollment step. In recent years, financial institutions and service providers have implemented rule-based expert systems and adopted short message service (SMS) user authentication to address this problem. However, the above solution is inadequate to face the challenges of data loss and social engineering. In this study, we introduce several traditional machine learning algorithms and finally choose the improved gradient boosting decision tree (GBDT) algorithm software library for use in a real system, namely, XGBoost. We further expand multiple features based on analysis of the enrollment behavior and plan to add historical transactions in future studies. Subsequently, we use a real card enrollment dataset covering the year 2017, provided by a worldwide payment processor. The results and framework are adopted and absorbed into a new design for a mobile payment fraud detection system within the Chinese payment processor.

Key words: Fraud detection; Mobile payment; Bankcard enrollment; Mobile device based; GBDT; XGBoost
https://doi.org/10.1631/FITEE.1800580 **CLC number:** TP309.2; TP181

1 Introduction

With the rapid growth in credit and debit card use, fraud has increased significantly, especially in mobile payments. According to the risk report delivered by UnionPay, during 2017, the total fraud cost in mobile device based payments reached 14.2 million RMB in a single UnionPay network, while the total transaction amount was 90.8 billion RMB (China UnionPay, 2017). Payment fraud has rapidly shifted to mobile and Internet channels. Furthermore, fraudsters dynamically change their methods to avoid being detected (Correa Bahnsen et al., 2016). To face this situation, the following solutions are inadequate: (1) traditional fraud detection tools, i.e., expert rules, and (2) machine learning methods without adjustment to new fraud attempts, i.e., static models that are updated infrequently (Dal Pozzolo et al., 2014).

The two widespread methods of mobile payment are mobile device based and mobile app based.

1. The core idea of the mobile device based solution is physical bankcard simulation in the embedded secure element (eSE) within a device. Following this idea, the focus of the solution is on the device hardware. The process of adding bankcards to a

‡ Corresponding author
* Project supported by China UnionPay
ORCID: Hao ZHOU, http://orcid.org/0000-0002-0869-6570

mobile wallet through a mobile device is called "bankcard enrollment" (Fig. 1). Bankcard enrollment usually includes user authentication, device primary account number (DPAN) provisioning, and lifecycle management. Multi-factor user authentication is adopted to validate the operation of each bankcard enrollment process, including short message service (SMS) validation, personal identification number (PIN) verification, and real name matching. Furthermore, to adapt to the mobile scenario, information such as the mobile device name, subscriber identification module number, international mobile equipment identity number, media access control address, Internet protocol address, and location is collected and introduced into the detection system (Zhou and Chai, 2017). However, due to phishing and data loss, card data, SMSs, and even PIN codes could be stolen or leaked on a large scale. Thus, multi-factor user authentication is not enough for security. Currently, between steps 5 and 6 in Fig. 1, the Chinese payment processor, UnionPay, is running rule-based expert risk analysis in real time.

2. The mobile app based solution lacks an eSE within the mobile device. Thus, it is a logical rather than a physical bankcard simulation, similar to the quick response code and host card emulation modes. For security, the added bankcard's DPAN within the mobile app based solution is dynamic; in contrast, the DPAN in mobile device based payment is static.

Bankcard fraud detection is by definition a cost-sensitive problem, meaning that the cost of a false positive is different from the cost of a false negative. When incorrectly classifying a transaction as fraudulent, there is an administrative cost incurred by the financial institution. In contrast, failing to detect a fraud means the transaction amount is lost (Hand and Henley, 1997).

When training a bankcard fraud detection model, it is important to use features that enable good classification. Typical models use only raw transactional features, such as the amount, time, transaction type, location of the transaction, and merchant category. However, when detecting fraud during bankcard enrollment, these approaches do not take into account the cardholder's card enrollment data, such as device information, historical card enrollment records, and historical card transactions before the enrollment.

In this study, we introduce machine learning algorithms into a real dataset of mobile device based bankcard enrollment. We then compare three algorithms: random forest (RF), logistic regression (LR), and gradient boosting decision tree (GBDT). Finally, GBDT was determined as the best algorithm to be applied to our real fraud detection project. Additionally, as it supports distributed processing frameworks, we use XGBoost as our software library for implementation (Li XR et al., 2018; Wang et al., 2018; Wu et al., 2018). We attempt to combine different fraud detection tools and strategies in the enrollment stage and in the following transaction stage, which will help to combat fraud more effectively, while also realizing our aim, "to recognize suspicious enrollment in time and to be prepared for blocking fraudulent transactions later on." The dataset is provided by a worldwide payment processor headquartered in China, namely, China UnionPay. The dataset includes bankcard enrollment data in 2017, where five major mobile device based payment platforms exist, namely, Apple Pay, Samsung Pay, Huawei Pay, Xiaomi Pay, and Meizu Pay. Similarly, the approach introduced in the paper could be applied to mobile app based solutions.

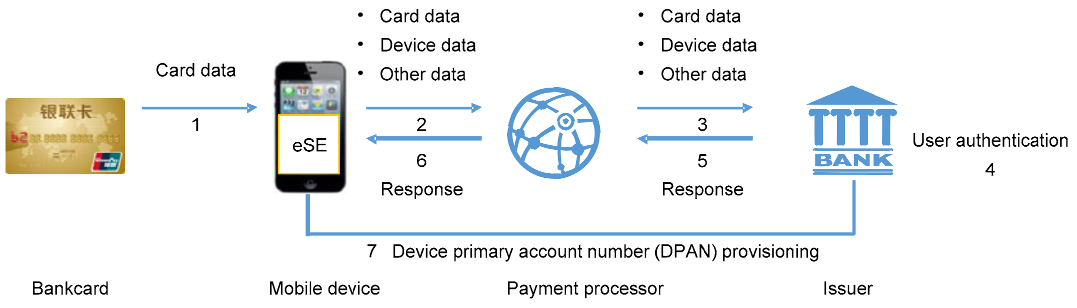

Fig. 1 Flowchart of bankcard enrollment for mobile device based payment

2 Related works

A bankcard fraud detection algorithm usually involves identifying transactions with a high possibility of being fraudulent based on historical fraud modes. The use of machine learning (Tian et al., 2017) in fraud detection has been an area of interest. Different detection tools based on machine learning have been used for this problem, particularly neural networks (Maes et al., 2002; Fu et al., 2016; Cheng et al., 2018), Bayesian learning (Maes et al., 2002), association rules (Sánchez et al., 2009), artificial immune systems, hybrid models (Krivko, 2010), RF (Correa Bahnsen et al., 2013; Dal Pozzolo et al., 2014), and support vector machines (Bhattacharyya et al., 2011).

Most of these studies compare their recommended algorithms with a benchmark algorithm and make the contrast with a standard binary classification measure (Hand and Henley, 1997; Jurgovsky et al., 2018), such as misclassification errors, receiver operating characteristics, Kolmogorov–Smirnov (KS) tests, F_1 scores (Bolton et al., 2002; Hand et al., 2008), or area under curve (AUC) statistics. Most of these measures are extracted from the confusion matrix shown in Table 1, in which the prediction of the algorithm C is a function of the k features of transaction i, $\boldsymbol{x}_i=[x_i^1, x_i^2, ..., x_i^k]$, and y_i is the real class of transaction i.

Table 1 Classification confusion matrix

	Actual positive $y=1$	Actual negative $y=0$
Predicted positive $C=1$	True positive (TP)	False positive (FP)
Predicted negative $C=0$	False negative (FN)	True negative (TN)

From Table 1, multiple metrics are extracted:

$$\text{Accuracy}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}, \tag{1}$$

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{2}$$

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}}, \tag{3}$$

$$F_1\ \text{score}=2\frac{\text{Precision}\times\text{Recall}}{\text{Precison}+\text{Recall}}. \tag{4}$$

However, these measures may not be the most appropriate evaluation criteria for evaluating fraud detection methods, because it is assumed that the cost added by misclassification is similar to that added by correct classification. This premise is not reasonable in practice, as failing to identify a fraudulent transaction will lead to a totally different financial cost. Moreover, the accurate method assumes that the classification among transactions is constant and balanced, and the distributions of a fraud detection dataset are typically skewed, with a percentage ranging from 0.0005% to 0.05% for mobile device based payments.

To consider the different costs of fraud detection during the assessment of an algorithm, we can use the adjusted cost matrix. In Table 2, the cost matrix is presented, in which the costs for accurate classification, namely, true positives C_{TP_i} and true negatives C_{TN_i}, and the two types of misclassification errors, namely, false positives C_{FP_i} and false negatives C_{FN_i}, are presented. This is an extension of Table 1, but in this situation the costs are event dependent (in other words, specific to every transaction i).

Table 2 Cost matrix

	Actual positive $y_i=1$	Actual negative $y_i=0$
Predicted positive $C_i=1$	C_{TP_i}	C_{FP_i}
Predicted negative $C_i=0$	C_{FN_i}	C_{TN_i}

In the situation of a false positive, the cost is the administrative cost $C_{\text{FP}_i}=C_\text{a}$ related to analyzing the individual transaction and notifying the card holder. This cost is also assigned to a true positive $C_{\text{TP}_i}=C_\text{a}$ because in this situation, the card holder will be contacted to confirm the transaction. However, in the situation of a false negative, where a fraud is missed, the cost is defined as the transaction amount itself. The costs are listed in Table 3.

Furthermore, this framework is flexible to include extra costs such as the expected cost caused by an irritated card holder due to a false positive, or the profit due to a satisfied card holder that feels safe by being called by the bank.

Table 3 Bankcard cost matrix

	Actual positive y_i=1	Actual negative y_i=0
Predicted positive C_i=1	$C_{\mathrm{TP}_i}=C_\mathrm{a}$	$C_{\mathrm{FP}_i}=C_\mathrm{a}$
Predicted negative C_i=0	$C_{\mathrm{FN}_i}=\mathrm{Amt}_i$	$C_{\mathrm{TN}_i}=0$

Let S be a group of N transactions under the class label $y_i\in\{0, 1\}$. A function f generates label C_i for each transaction i. We observe the results when using the algorithm and those when not using any algorithm or rules. The savings are then calculated as

$$\mathrm{savings}(f(s))=\frac{\sum_{i=1}^{N}\left(y_iC_i\mathrm{Amt}_i-C_iC_\mathrm{a}\right)}{\sum_{i=1}^{N}y_i\mathrm{Amt}_i}, \quad (5)$$

which is the sum of the amounts of the precisely predicted fraudulent transactions minus the cost caused while detecting them, divided by the sum of the amounts of the fraudulent transactions. Finally, it can be summarized as follows:

1. This event-dependent strategy focuses only on large amount transactions, and smaller amount frauds would not matter. Nevertheless, this framework is flexible to allow adjustment of the cost matrix to refer to the available amount of transactions as the cost of a false negative. Furthermore, small amount frauds but with frequent transactions would have a higher importance because a significant amount of money is available within the bankcard.

2. For bankcard enrollment, the amount is zero. However, for false negatives, the fraud would occur later. Thus, it is unreasonable to evaluate the enrollment fraud detection just by savings(f(s)). We use recall and precision during the bankcard enrollment stage. We would then detect the following transaction stage using savings(f(s)). Thus, for recording statistics of card enrollment fraud, true positives stand for the cardholder denying the enrollment, true negatives stand for no fraud occurring afterwards, false negatives stand for fraud occurring afterwards even if they are detected later, and false positives stand for card enrollments that are wrongly detected as positive.

3 Feature engineering for fraud detection based on machine learning

When training a bankcard fraud detection algorithm during the enrollment, the initial group of features (raw features) are classified into the following categories: bankcard data, user data, and device data. The typical bankcard enrollment fraud detection raw features are summarized in Tables 4–6.

Table 4 Summary of typical raw features of bankcard enrollment fraud detection: bankcard data

Attribute name	Description
Transaction ID	Transaction identification number
Time	Date and time of the enrollment
Primary account number (PAN)	Identification of the bankcard
Bank	Issuer bank of the card
Bankcard type	Debit, credit, or quasi-credit
Type of product	Apple Pay, or other mobile device based Payment, such as Huawei Pay

Table 5 Summary of typical raw bankcard enrollment fraud detection features: user data

Attribute name	Description
Method of PAN capture	Method of bankcard input, such as by camera, by manual input
Registered mobile number	The mobile number which is recorded as a contact number by banks
Cardholder name	Name of the cardholder, optional
HASH of merchant user ID	HASH value of user ID in mobile device manufacturer's account, like Apple ID

Bankcard data is almost identical between the bankcard enrollment stage and the transaction stage. Different types of payment tools may provide different device data. Thus, we classify the mobile device based payments into two categories: Apple Pay and non-Apple Pay.

User data is useful for judging whether bankcard enrollment is conducted by the cardholder. Some mobile device manufacturers transfer the hash value of the registered user ID in their account list to facilitate fraud detection. Furthermore, device data is helpful for recognizing a unique device and analyzing the mapping relationship between the device and bankcard.

Table 6 Conclusion of typical raw features of bankcard enrollment fraud detection: device data

Attribute name	Description
IP address	IP during bankcard enrollment
Number of SIMs	For iPhone, just 1; for Android based phone, 1 or 2
SIM card 1 number	SIM card numbers during bankcard enrollment
SIM card 2 number	If two SIMs, the second SIM number during bankcard enrollment
Type of mobile device	iPhone 7, Huawei P10, etc.
Language of device	Language for use in a device
Name of device	Name set up by a device user
SEID	Security entity ID number which is unique for each mobile device
Risk score by a mobile device manufacturer	The score of risk level provided from the perspective of the manufacturers
Geolocation	Latitude and longitude data of a mobile device
Other data	Optional, such as screen resolution, OS type and version, and network mode

Thus, it is a two-category machine learning issue. According to Bayesian theory, for an input $X=x$, the value of output is $y=\{C_1, C_2, ..., C_k\}$.

$$P\left(y=C_k \mid X=x\right)=\frac{p(X=x \mid y=C_k)p(y=C_k)}{\sum_{l=1}^{K} p(X=x \mid y=C_l)p(y=C_l)}. \tag{6}$$

For two-category machine learning, $Y=\{0, 1\}$. We have taken the LR algorithm (Correa Bahnsen et al., 2014) given in Eq. (7), the GBDT algorithm, and the RF algorithm as candidate algorithms.

$$y=\text{sigmoid}(x)=\frac{1}{1+\mathrm{e}^{-x}}. \tag{7}$$

RF is a learning method for classification, regression, and other tasks. It operates by training a set of decision trees and outputting the class that is the mode of the classes or the mean prediction of the individual trees. Overall, the random decision forests method helps limit the decision trees' tendency to overfit to their training dataset.

GBDT is a machine learning method for classification and regression problems, contributing to a prediction model in a similar manner to an ensemble of weak prediction models, especially decision trees. It constructs the model with stages like those made by other boosting methods and sums them up by adopting the optimization of an arbitrary differentiable loss function.

The generic gradient boosting algorithm at the m^{th} step would suit a decision tree $h_m(x)$ to pseudo residuals. Let J_m be the number of its leaves. The tree divides the input space into J_m uncrossed regions R_{1m}, R_{2m}, …, $R_{J_m m}$ and predicts a constant value in every region. Using the indicator representation, the output of $h_m(x)$ for input x would be written as

$$h_m(x)=\sum_{j=1}^{J_m} b_{jm}R_{jm}(x), \tag{8}$$

where b_{jm} is the value calculated in region R_{jm}.

From an implementation perspective, XGBoost is a widely used open-source software library that provides a gradient boosting design framework suitable for C++, Java, Python, R, and Julia. XGBoost works in Linux, Windows, and macOS environments. From the program description, its purpose is to provide a scalable (Wang et al., 2016), portable, and distributed gradient boosting (GBM, GBRT, GBDT) library. Among the 30 challenge-winning solutions published on Kaggle's blog during the year 2015, 17 solutions use XGBoost (Chen and Guestrin, 2016). The results above demonstrate that the XGBoost system provides state-of-the-art performance on a wide range of problems. The extension of XGBoost is due to certain vital system- and algorithm-level optimizations. These innovations include a novel tree learning algorithm for handling sparse data and a theoretically reasonable weighted quantile sketch procedure for dealing with instance weights in similar tree learning. Parallel computing and distributed computing make tree learning faster, which leads to quicker model exploration. Furthermore, XGBoost uses out-of-core computation and enables data scientists to process hundreds of millions of examples on a desktop.

Our goal is to design a fraud scoring framework to reflect the probability of fraud of bankcard enrollment with precision. We put three categories of data as the model's input:

1. the data of current bankcard enrollment;
2. historical records of the same bankcard's enrollment;
3. historical records of bankcard enrollment on the same mobile device.

4 Experimental setup and results

In this section, the dataset used in the experiments is described first. The partitioning of the dataset is then explained and presented. Finally, the algorithms used to effectively detect fraud are shown.

4.1 Database

In this study we adopted a dataset provided by a worldwide payment processor. The dataset includes fraudulent and legitimate Chinese bankcard enrollment during the year 2017. The dataset contains 52 541 638 individual bankcard enrollment records, each with almost 20 attributes on average, including a fraud label. This label was created and confirmed internally in the payment processor and, if necessary, through issuers to reconfirm, and thus could be regarded as accurate. Within the dataset, only 5753 bankcard enrollment records were labeled as fraud; thus, the dataset has a fraud ratio of 0.0109%.

Moreover, the use of the methods for feature extraction is described, with approximately 614 features obtained. For the experiments, a smaller subset includes 0.1 million bankcard enrollment records and a fraud ratio of 5.7%. Within this dataset, the total financial losses are 14.2 million RMB. This dataset was selected because of its relatively high fraud ratio.

4.2 Database partitioning

From the dataset, four different sub-datasets are extracted as follows: training, validation 1, validation 2, and testing, containing 71.5%, 8%, 8.5%, and 12% of the bankcard enrollment records, respectively. Table 7 lists the different datasets.

4.3 Algorithms

In the experiments, we used three cost-insensitive classification algorithms, namely, LR (Correa Bahnsen et al., 2016), RF, and GBDT. Additionally, we converted the probability into a score ranging from 1 to 999 by linear conversion.

Table 7 Summary of the datasets

Set	Number of bankcard enrollment activities	Percentage of frauds (%)
Total	100 000	5.75
Training	71 434	5.54
Validation 1	8048	5.78
Validation 2	8505	5.73
Testing	12 013	7.02

4.3.1 LR

Consider a model with two predictors, x_1 and x_2. There may be continuous variables (taking a real number as the value) or indicator functions for binary variables (taking value 0 or 1). The general form of the log-odds is

$$l = \beta_0 + \beta_1 x_1 + \beta_2 x_2, \tag{9}$$

where coefficients β_i (i=0, 1, 2) are the parameters of the model. Note that l is a linear combination of predictors x_1 and x_2, including a constant term β_0. The corresponding number of odds is

$$\text{odds} = b^l. \tag{10}$$

b is usually set as e. Here we set b=10 for easy understanding of the odds.

We realize the LR in a Spark environment.

4.3.2 RF

The main idea of the RF model is to establish multiple decision trees as weak classifiers and combine multiple weak classifiers to form strong classifiers. The RF algorithm is a method of averaging multiple deep decision trees, which are trained on different parts of a single training set mainly to reduce the variance. This comes at the cost of a small increase in bias and loss of interpretability; however, they generally enhance the performance of the final model. The RF algorithm applies the technique of bootstrap aggregating, or bagging, to the tree learners. Given a training set X=x_1, x_2, ..., x_n with responses Y=y_1, y_2, ..., y_n, bagging repeatedly (B times) selects a random sample of the training set with replacement and is suitable for trees which are applicable to these samples.

For b=1, 2, ..., B:

1. Select, with replacement, n training samples from X, Y, and denote them as X_b, Y_b;

2. Construct a classification or regression tree f_b on X_b, Y_b.

After training, predictions for unknown samples x' could be generated by averaging the predictions from the whole set of individual regression trees on x':

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x'), \tag{11}$$

or by means of taking the majority vote in the situation of classification trees.

4.3.3 GBDT

As described in Section 3, we realize GBDT using the XGBoost library (Chen and Guestrin, 2016). For a dataset containing n examples and m features, $D=\{(\boldsymbol{x}_i;\ y_i)\}$ ($|D|=n$, $\boldsymbol{x}_i\in\mathbb{R}^m$, $y_i\in\mathbb{R}$), a tree model (Fig. 2) uses K additive functions to foresee the result as output.

$$\hat{y}_i = \phi(\boldsymbol{x}_i) = \sum_{k=1}^{K} f_k(\boldsymbol{x}_i),\quad f_k\in F, \tag{12}$$

where $F=\{f(\boldsymbol{x})=\boldsymbol{w}_{q(\boldsymbol{x})}\}$ (q: $\mathbb{R}^m\rightarrow T$, $\boldsymbol{w}\in\mathbb{R}^T$) is the space of regression trees (known as CART).

To learn the group of functions in the model, we minimize the regularized objective as follows:

$$L(\phi) = \sum_i l(\hat{y}_i y_i) + \sum_k \Omega(f_k), \tag{13}$$

where $\Omega(f)=\gamma T+0.5\lambda\|\boldsymbol{w}\|^2$, and l is a differentiable convex loss function which measures the difference between prediction value $\hat{y}_i$ and target value y_i. Ω penalizes the complexity of the model (i.e., the regression tree functions).

4.4 Results

We compared the performances of current rule-based expert, LR-based, RF-based, and GBDT-based algorithms by the number of bankcard enrollment activities. Table 8 shows the precision and recall performance on Apple Pay and non-Apple Pay bankcard enrollment.

Currently, on Apple Pay, the real performance with the rule-based expert algorithm was fluctuating in precision each month in 2017, from a minimum of 1.26% in January to a maximum of 6.94% in

Table 8 Comparison of the performance between Apple Pay and non-Apple Pay

Recall (%)	Precision (%)							
	Rule based*		LR based		RF based		GBDT based	
	AP	nAP	AP	nAP	AP	nAP	AP	nAP
25	N/A	N/A	42.31	34.09	41.77	34.42	50.83	41.36
50	N/A	N/A	17.95	11.37	17.49	13.42	21.03	16.74
75	2.72	2.24	4.13	2.88	3.02	2.64	6.27	5.85

* Average value. AP: Apple Pay; nAP: non-Apple Pay

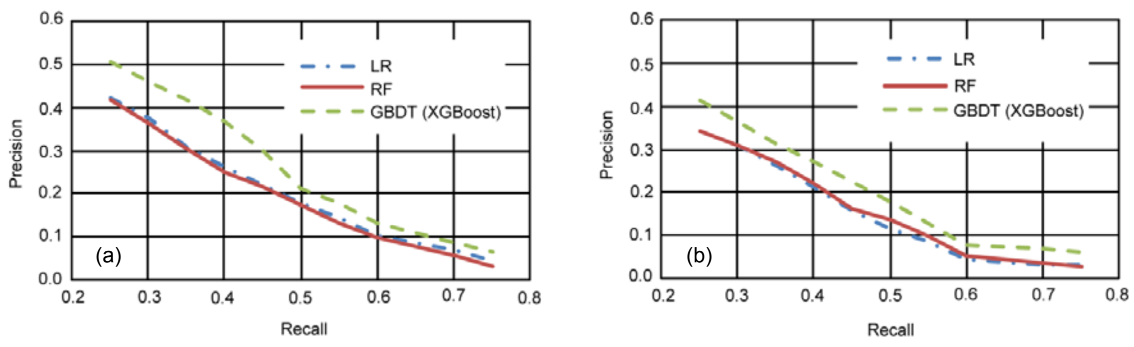

Fig. 2 Precision–recall curves in the dataset: (a) Apple Pay; (b) non-Apple pay

September. The average precision of year 2017 is 2.72%, while the average recall is close to 75%, at 73.48%. For non-Apple Pay applications, the average precision with the rule-based expert algorithm is 2.24%, which is a minor decrease compared to Apple Pay, while recall remains close to 75%. Because the real system aims to cover almost 75% of fraud, we do not have more performance data.

The precision–recall curves are shown in Fig. 2. We can see that our final GBDT-based model implemented by XGBoost outperforms the other methods significantly, with an increase of approximately 22% in performance.

5 Conclusions and discussion

In this study, we have shown different means of implementing machine learning algorithms to perform fraud detection of bankcard enrollment. Machine learning performs better than expert rules, especially in differentiating high and low risk. However, LR, RF, and GBDT have different performances. GBDT, implemented by XGBoost, is the most efficient.

In future work, we will implement the GBDT-based model and score framework in a real system. Furthermore, we will attempt to add historical bankcard transactions before enrollment into the training for observation. Additionally, we will test the unsupervised machine learning (Li S et al., 2018) algorithm for training to recognize teams of fraudsters more accurately.

Acknowledgements

We want to thank Drs. Jun WU, Jin-tao ZHAO, and Jian-hua LI for their advice on machine learning based bankcard fraud scoring and its applications design.

References

Bhattacharyya S, Jha S, Tharakunnel K, et al., 2011. Data mining for credit card fraud: a comparative study. *Dec Support Syst*, 50(3):602-613. https://doi.org/10.1016/j.dss.2010.08.008

Bolton RJ, Hand DJ, 2002. Statistical fraud detection: a review. *Stat Sci*, 17(3):235-255. https://doi.org/10.1214/ss/1042727940

Chen TQ, Guestrin C, 2016. XGBoost: a scalable tree boosting system. Proc 22nd ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.785-794. https://doi.org/10.1145/2939672.2939785

Cheng J, Wang PS, Li G, et al., 2018. Recent advances in efficient computation of deep convolutional neural networks. *Front Inform Technol Electron Eng*, 19(1):64-77. https://doi.org/10.1631/FITEE.1700789

China UnionPay, 2017. China Bank Card Annual Fraud Report 2017.

Correa Bahnsen A, Stojanovic A, Aouada D, et al., 2013. Cost sensitive credit card fraud detection using Bayes minimum risk. Proc 12th Int Conf on Machine Learning and Applications, p.333-338. https://doi.org/10.1109/ICMLA.2013.68

Correa Bahnsen A, Aouada D, Ottersten B, 2014. Example-dependent cost-sensitive logistic regression for credit scoring. Proc 13th Int Conf on Machine Learning and Applications, p.263-269. https://doi.org/10.1109/ICMLA.2014.48

Correa Bahnsen A, Aouada D, Stojanovic A, et al., 2016. Feature engineering strategies for credit card fraud detection. *Expert Syst Appl*, 51:134-142. https://doi.org/10.1016/j.eswa.2015.12.030

Dal Pozzolo A, Caelen O, Le Borgne YA, et al., 2014. Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Syst Appl*, 41(10):4915-4928. https://doi.org/10.1016/j.eswa.2014.02.026

Fu K, Cheng DW, Tu Y, et al., 2016. Credit card fraud detection using convolutional neural networks. Proc 23rd Int Conf on Neural Information Processing, p.483-490. https://doi.org/10.1007/978-3-319-46675-0_53

Hand DJ, Henley WE, 1997. Statistical classification methods in consumer credit scoring: a review. *J R Stat Soc Ser A*, 160(3):523-541. https://doi.org/10.1111/j.1467-985X.1997.00078.x

Hand DJ, Whitrow C, Adams NM, et al., 2008. Performance criteria for plastic card fraud detection tools. *J Oper Res Soc*, 59(7):956-962. https://doi.org/10.1057/palgrave.jors.2602418

Jurgovsky J, Granitzer M, Ziegler K, et al., 2018. Sequence classification for credit-card fraud detection. *Expert Syst Appl*, 100:234-245. https://doi.org/10.1016/j.eswa.2018.01.037

Krivko M, 2010. A hybrid model for plastic card fraud detection systems. *Expert Syst Appl*, 37(8):6070-6076. https://doi.org/10.1016/j.eswa.2010.02.119

Li S, Song SJ, Wu C, 2018. Layer-wise domain correction for unsupervised domain adaptation. *Front Inform Technol Electron Eng*, 19(1):91-103. https://doi.org/10.1631/FITEE.1700774

Li XR, Yu W, Luwang TY, et al., 2018. Transaction fraud detection using GRU-centered sandwich-structured model. https://arxiv.org/abs/1711.01434

Maes S, Tuyls K, Vanschoenwinkel B, et al., 2002. Credit card fraud detection using Bayesian and neural networks. Proc 1st Int NAISO Congress on Neuro Fuzzy Technologies, p.261-270.

Sánchez D, Vila M, Cerda L, et al., 2009. Association rules applied to credit card fraud detection. *Expert Syst Appl*, 36(2):3630-3640.
https://doi.org/10.1016/j.eswa.2008.02.001

Tian YH, Chen XL, Xiong HK, et al., 2017. Towards human-like and transhuman perception in AI 2.0: a review. *Front Inform Technol Electron Eng*, 18(1):58-67.
https://doi.org/10.1631/FITEE.1601804

Wang HZ, Zhang P, Xiong L, et al., 2016. A secure and high-performance multi-controller architecture for software-defined networking. *Front Inform Technol Electron Eng*, 17(7):634-646.
https://doi.org/10.1631/FITEE.1500321

Wang S, Wu J, Zhang YT, 2018. Consumer preference-enabled intelligent energy management for smart cities using game theoretic social tie. *Int J Distrib Sens Networks*, 14(4):1-11.
https://doi.org/10.1177/1550147718773235

Wu J, Dong MX, Ota K, et al., 2018. Big data analysis-based secure cluster management for optimized control plane in software-defined networks. *IEEE Trans Netw Serv Manag*, 15(1):27-38.
https://doi.org/10.1109/TNSM.2018.2799000

Zhou YK, Chai HF, 2017. Research and practice on system engineering management of a mobile payment project. *Front Eng Manag*, 2017, 4(2):127-137.
https://doi.org/10.15302/J-FEM-2017011