The Behavioral Sign of Account Theft: Realizing Online Payment Fraud Alert

Cheng Wang[1,2,3*]
[1]Department of Computer Science, Tongji University, Shanghai, China
[2]Key Laboratory of Embedded System and Service Computing, Ministry of Education, Shanghai, China
[3]Shanghai Institute of Intelligent Science and Technology, Tongji University
cwang@tongji.edu.cn

Abstract

As a matter of fact, it is usually taken for granted that the occurrence of unauthorized behaviors is necessary for the fraud detection in online payment services. However, we seek to break this stereotype in this work. We strive to design an ex-ante anti-fraud method that can work before unauthorized behaviors occur. The feasibility of our solution is supported by the cooperation of a characteristic and a finding in online payment fraud scenarios: The well-recognized characteristic is that online payment frauds are mostly caused by account compromise. Our finding is that account theft is indeed predictable based on users' high-risk behaviors, without relying on the behaviors of thieves. Accordingly, we propose an account risk prediction scheme to realize the ex-ante fraud detection. It takes in an account's historical transaction sequence, and outputs its risk score. The risk score is then used as an early evidence of whether a new transaction is fraudulent or not, before the occurrence of the new transaction. We examine our method on a real-world B2C transaction dataset from a commercial bank. Experimental results show that the ex-ante detection method can prevent more than 80% of the fraudulent transactions before they actually occur. When the proposed method is combined with an interim detection to form a real-time anti-fraud system, it can detect more than 94% of fraudulent transactions while maintaining a very low false alarm rate (less than 0.1%).

1 Introduction

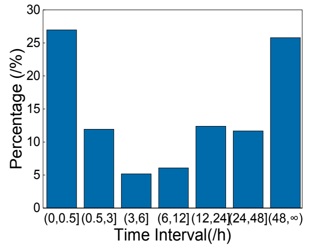
Online payment services bring convenience to people's daily lives. Meanwhile, they also face many challenges, e.g., transaction fraud, where a fraudster might have stolen an account and intends to transfer its funds quickly by purchasing some merchandises at online shopping platforms. Transaction fraud has caused huge economic loss to financial platforms every year [Yang *et al.*, 2014; Mittal and Tyagi, 2020]. It is really imperative to build effective online fraud detection systems for financial platforms to lock fraudsters out.

Most existing methods for transaction fraud detection usually depend on real-time online payment behaviors. These methods actively examine every transaction when a user starts a payment, and try to detect and terminate any fraudulent transaction before fraudsters finish the transfer of funds. Among these methods, the rule-based system was once commonly used [Jarovsky *et al.*, 2018]; it is generally built on business experiences and character statistics of historical risk events. Nowadays, machine learning techniques are increasingly applied to active fraud detection and have shown great effectiveness [Nami and Shajari, 2018]. Thereinto, supervised learning models have been widely used in real-time transaction fraud detection [Jing *et al.*, 2019]. They are often combined with the usage of transaction aggregation [Kim *et al.*, 2019]. In the meantime, unsupervised and semi-supervised learning methods also have been taken into consideration [Porwal and Mukund, 2019; Elshaar and Sadaoui, 2020]. In addition, deep learning techniques, such as Recurrent Neural Networks (RNN) [Wang *et al.*, 2017] and Convolutional Neural Networks (CNN) [Pozzolo *et al.*, 2018], have shown their great advantages on transaction fraud detection.

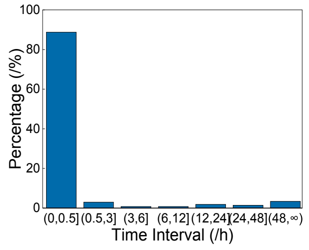
All of these methods can be regarded as *interim* detection methods, which work passively and take effect only when a transaction occurs. We take a different point of view, however, to ask whether transaction fraud can be detected in an *ex-ante* manner. That is to say, *can we predict a fraudulent transaction before its occurrence?*

We start our investigation for solutions on the basis of a fact that most transaction frauds in the online payment services are caused by account compromise. Thus, we resort to the prediction of account theft based on user behaviors. By examining the collected real-world transaction data, we find that account compromise is highly associated with risky payment behaviors of users. In Figure 1(a), we show the percentages of time intervals between adjacent legal and fraudulent transactions in a real dataset of a commercial bank. For more than a quarter of the compromised accounts, each of them has time intervals of less than half an hour between its last legal transaction and the first following fraudulent transaction. This indicates that there is possibly a relation between a user's historical payment behavior and his/her account compromise. Figure 1(b) shows the percentages of time intervals between adjacent fraudulent transactions of the same account. It reveals the behavior patterns of fraudsters to some extent. To

[*]Corresponding author: cwang@tongji.edu.cn



(a)



(b)

Figure 1: **(a)** The percentages of different intervals between adjacent legal and fraudulent transactions of the same account. **(b)** The percentages of different intervals between adjacent fraudulent transactions of the same account.

be concrete, once a fraudster gets hold of an account, he/she will transfer the funds as quickly as possible.

From the above observation, our objective is to check out to what extent an account is in risk after its holder has generated a series of transactions. Accordingly, we propose a fraud risk prediction method based on transaction sequences. The fraud risk of an account can be determined by only several recent transactions performed by its holder. Obviously, in contrast to typical fraud detection techniques, which are mainly interim or ex-post, ours is an ex-ante one.

We firstly adopt a feature aggregation method to deal with transaction sequences, and then feed the aggregated features into some state-of-the-art machine learning models to predict account risks. Surprisingly, upon doing so, our ex-ante method can achieve nearly the same detection performance as those interim ones on a real B2C transaction dataset from a commercial bank. In addition, to exploit the complementary effects, we design an anti-fraud scheme by combining the ex-ante and interim paradigms in order to further improve the effectiveness.

The rest of the paper is organized as follows. Section 2 provides the related works on transaction fraud detection. In Section 3, we propose the method. In Section 4, we present the experimental evaluation, and devise an enhanced anti-fraud scheme in Section 5. We conclude the work in Section 6.

2 Related Work

With the advent of large-scale e-commerce platforms and online payment platforms, transaction fraud detection has become a widely studied research area, and attracts a lot of attention from researchers. Existing studies on real-time transaction fraud detection mainly work in an interim manner.

Interim fraud detection tries to discover and identify fraudulent activities as they enter the detecting systems and report them to a system administrator [Wang *et al.*, 2019]. Rule-based expert system used to be the most widely used technique, which incorporates various areas of knowledge like economics, finance and business practices [Jarovsky *et al.*, 2018; Milo *et al.*, 2018]. However, the capability of this approach is limited because it heavily depends on predefined rules. These rules can only be maintained by domain experts, which is quite labor-intensive. Moreover, statistical properties of transactions may change over time, making the rules obsolete.

Fraud detection methods based on machine learning and artificial intelligence techniques are becoming increasingly popular. Supervised learning techniques make use of users' behavior data to create a classification model to determine whether a user's ongoing behavior deviates from the global behavior [Nami and Shajari, 2018]. From the point of view the fraudster, Jing et al. [Jing *et al.*, 2019] have found some behavior patterns of fraudsters from raw data. They combined these behavior patterns with machine learning models. As it is not always possible to label all data, unsupervised learning approaches are utilized to overcome this defect [Porwal and Mukund, 2019]. Later, deep learning techniques were increasingly applied to financial transaction fraud detection scenarios [Wang *et al.*, 2017; Pozzolo *et al.*, 2018]. Recently, Zhang et al. [Zhang *et al.*, 2019] applied the deep forest to the task of fraud detection. Zheng et al. [Zheng *et al.*, 2019] proposed one-class adversarial nets for fraud detection with only benign users as training data to detect the malicious users. Fraud detection methods based on network and knowledge graphs have also attracted more and more attentions [Cao *et al.*, 2017; Ying *et al.*, 2018; Liu *et al.*, 2019; Sangers *et al.*, 2019].

The most similar work to ours was done by [McGlohon *et al.*, 2009]. It proposed an account risk evaluation system using link analysis, and achieved great success in pinpointing misstated accounts from their dataset. Their method is offline in nature, and needs to be updated periodically for predictive effectiveness. Ours is a real-time method of risk prediction, so the risk of an account could be evaluated every time it generates a transaction. The evaluated risk can be further utilized to predict fraudulent transactions.

3 Fraud Prediction

We present a real-time account risk prediction method to prevent the occurrence of future fraudulent transactions.

3.1 Transaction Sequence Window

Given an account of a user, we can use a sequence in chronological order, i.e., $T = \{(x_1, y_1), (x_2, y_2), ..., (x_t, y_t)\}$, to represent its historical transaction records, where x_i is the i-th transaction of the account, y_i is the label of transaction x_i, x_t is the latest completed transaction of the account, and the risk value of the account is r_t. The risk of the account is closely related to the user's recent transaction behaviors. We devise a transaction sequence window W for each account. The size of the window is w. The so-called transaction sequence window of an account, denoted by $W = \{x_{t-(w-1)}, ..., x_{t-1}, x_t\}$, contains a sequence of w transaction records generated by this account recently. We formulate here risk prediction problem to evaluate the probability of fraud in the next transaction, where $r_t = \Pr\left(y_{t+1} = 1 | W = \{x_{t-(w-1)}, ..., x_{t-1}, x_t\}\right)$.

When an account finishes a transaction, its transaction sequence window is updated by maintaining a first-in-first-out transaction queue. In order to make real-time fraud risk predictions on the account, every time when the transaction sequence window of an account is updated, the predicted risk of this account is immediately re-estimated. Based on the updated risk value, we can judge the risk of the account continuing

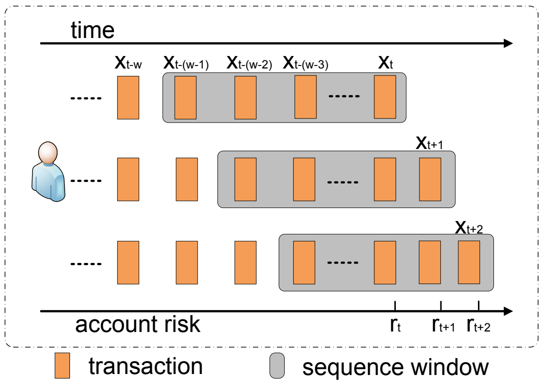


Figure 2: The process of updating transaction sequence window and account risk.

to trade and prevent the occurrence of fraudulent transactions in advance. If the risk is beyond a given threshold RT, we immediately lock the account and prevent any of its subsequently requested transactions.

Figure 2 provides an illustration for the updating process of the window. The current status of the window is $W_t = \{x_{t-(w-1)}, ..., x_{t-1}, x_t\}$, and the risk of the account is r_t, when a transaction x_{t+1} is completed, the window is updated as $W_{t+1} = \{x_{t-(w-2)}, ..., x_t, x_{t+1}\}$, we need to re-estimate the fraud risk of the account

$$r_{t+1} = P\left(y_{t+2} = 1 | W_{t+1} = \{x_{t-(w-2)}, ..., x_t, x_{t+1}\}\right).$$

If r_{t+1} is bigger than the threshold RT, the account is immediately locked, and the transaction x_{t+2} will not happen; otherwise, the transaction x_{t+2} is allowed to occur.

We vary the amount of transaction records in a transaction sequence window by choosing the different window sizes. We examine the performance of diverse window sizes ranging from 1 to 5, respectively. Note that the transaction sequence window is a rolling window and is updated with every transaction. This scheme ensures the feasibility of real-time fraud prediction. That is, we no longer have to wait for some periods of time until the next transaction is actually generated.

3.2 Feature Engineering

Feature Selection

A transaction record $x = (x^1, ..., x^M)$ is a feature vector $x \in R^M$, which contains various types of features, where x^i is the i-th feature of the transaction x. However, the data sparsity is often a big challenge in fraud detection task. People are becoming more and more aware of the importance of privacy and information security. As a result, platforms are not allowed to collect or utilized some sensitive information about users.

To assess the importance of different features, we introduce the mean decrease impurity (MDI, [Breiman *et al.*, 1984]) as a standard metric. The metric is defined as

$$MDI\left(x^m\right) = \frac{1}{N_T} \sum\nolimits_T \sum\nolimits_{i \in T: s_i = x^m} p(i) \bigtriangleup f(s_i, i),$$

where x^m is a feature, N_T is the number of trees within a Random Forest, s_i is the selected feature of one of splitting

Time	Amt	Aut	Merchant	Addr
7:50	100.00	Face	7	25
8:27	20.50		12	25
11:15	500.00	Password	7	25
11:17	125.35	Fingerprint		25
20:37	135.00	Password		25

Table 1: The Example of Transaction Records.

node i in tree T, $p(i)$ is the proportion of samples reaching node i, and $f(s_i, i)$ is an impurity decreasing measure. In our work, Gini index [Busa-Fekete *et al.*, 2017] is adopted as the impurity decreasing measure. The larger of the MDI, the more important the feature is. We retain features of high importance and remove features of low importance from all transaction records.

Feature Aggregation

Within a transaction sequence window for each account, there are several independent transaction records there. We need to perform feature transformations on the contained transaction records. A user's payment behavior changes across different transactions. The values of the same feature may be different for different records in the transaction sequence window. Therefore, we need aggregate several records into one record in the window to feed the machine learning model.

We adopt different feature extraction methods for different categories of features.

Discrete features. A discrete feature can be denoted by $x^i = \left\{f_i^1, f_i^2, ..., f_i^j, ...\right\}$, where x^i is the i-th feature of a transaction record, and f_i^j is the j-th value of the feature. We aggregate each feature by replacing its values with its occurrence frequencies in the window. For features with too many values, we need to group the values of features, and count the number of occurrences of different groups separately. We reassign the j-th value of feature x^i to be $count\left(f_i^j\right)/w$, where w is the size of transaction sequence window.

Continuous features. We do some mathematic operations on transaction amount and time which are continuous features, including calculating the sum, average, variance, maximum and minimum.

Other features. We simply digitize users' demographic or registering information whose values usually do not change across different transactions of the same account.

Table 1 presents an example of transaction sequence in the window ($w = 5$) of an account. There are 5 features for each transaction, including transaction time (Time), transaction amount (Amt), authentication method (Aut), online merchant (Mer), and account registration address (Addr). These five features can be divided into three different types: Time and Amt are the continuous features, Aut and Merchant are the discrete features, and Addr is the registering information. The values of Merchant and Addr are represented by numbers, which means that the values of these features have been grouped, and the number denotes the group id of the value. Table 2 provides the results of feature aggregation.

Min(Interval)	...	Sum(Amt)	Avg(Amt)	...	Face	Password	Fingerprint	Mer7	Mer12	Addr
2	...	880.85	176.17	...	0.2	0.4	0.2	0.4	0.2	25

Table 2: The Example of Feature Aggregation.

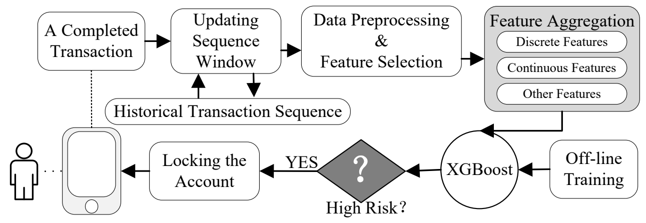


Figure 3: A real-time ex-ante risk prediction system.

3.3 Sampling

Highly imbalanced data is a practical challenge that we are faced in fraud detection [Wu *et al.*, 2017; Mohammed *et al.*, 2018]. Usually, there are much more non-fraudulent transactions than fraudulent ones. This problem seriously declines the performance of classifiers, as they are inclined to be overwhelmed by the majority class and thus ignore the minority class. Besides, the fraudulent transactions usually occur in a small number of accounts. As a matter of fact, we need to balance class labels in training data before feeding it to machine learning models.

We *under-sample* the legitimate transaction samples by random-skipping, and keep all the fraudulent samples. We define a class ratio C_R. It represents the fraction of the number of legal transactions and fraudulent transactions. For different classifiers, the ratio C_R is different.

3.4 Ex-ante Risk Prediction System

We devise an architecture of ex-ante risk prediction system, as illustrated in Figure 3, that is composed of an offline training procedure and an online prediction procedure.

Offline Training

Given all transactions of each account stored in a chronological order, we design a sliding window of fixed size for each account. Every time the window slides on the historical transaction sequence of an account, a training sample can be generated. When a training set is generated, we use the under-sampling technique to balance the ratio of legitimate and fraudulent transaction records. Afterwards, the data preprocessing, feature selection and feature aggregation modules are sequentially performed on all training samples. Finally, we use the aggregated features to train a classifier, e.g., XGboost [Chen and Guestrin, 2016], as the risk prediction model.

Online Prediction

When a user completes a transaction with his/her account, the system would update its transaction sequence in the window immediately, and re-generate desired features for the available classifier. The classifier will output a risk value that represents the risk of the account caused by its recent transaction operations. When the risk value is higher than a predefined threshold, the system will immediately lock the account and disable its following-up transactions.

4 Experimental Evaluation

We evaluate the performance of our method at different benchmark classifiers, and evaluate the influence of different setting parameters in our method. Furthermore, We verify the performance of our ex-ante prediction method by comparing it with the interim detection methods using the state-of-the-art classifiers.

4.1 Data Description and Analysis

We collect 3.5 million real-world B2C transaction records from a commercial bank. The transaction records have a time interval from April 1, 2017 to June 30, 2017 generated by 0.1 million accounts. All these records have been labelled as legitimate/fraudulent manually, and the whole data is encrypted and desensitized for security and privacy issues. Each transaction consists of 54 features, which include 17 user features (users' demographic information and accounts' registering information) and 37 online payment behavior features.

Although the adopted dataset has 54 features, many of them have invalid values for most transaction records, due to the limited permission offered by users and payment carries. From Figure 4, we can observe that more than half of the data records have a feature missing rate over 60%. We apply the random forest feature importance to rank the other 52 features except account ID and merchant ID. The features (with the corresponding types) selected in this way include Account ID (Int), Transaction Amount (Float), Transaction Time (Datetime), Card Type (Int), Authentication (Int), Frequent-Used IP (Bool), Registered Addr (String), and Merchant ID (Int).

Data imbalance is another big challenge. In our data, there are only 65291 fraudulent transactions, which means that only 1.8% of transactions are fraudulent transactions. Furthermore, only 8% of the accounts have been compromised in our all accounts. Additionally, the numbers of transactions and fraudulent transactions approximately follow pow-law distributions. As shown in the left side of Figure 5, most of the accounts completed very few transactions, while only few accounts completed most of transactions. The right side of Figure 5 presents the distribution of fraudulent transactions in the 8% fraud accounts, and shows that the occurrence fraudulent transactions concentrate on a few accounts.

4.2 Experiment Setting

We partition the dataset into two parts, with transaction records from April 1, 2017 to May 31, 2017 as the training data, and transaction records from June 1, 2017 to June 30, 2017 as the testing data.

The objective is to make a real-time prediction of account risk based on historical transaction sequences. Clearly, the

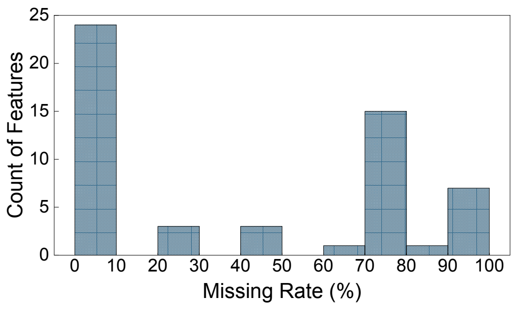


Figure 4: The missing rates of raw features.

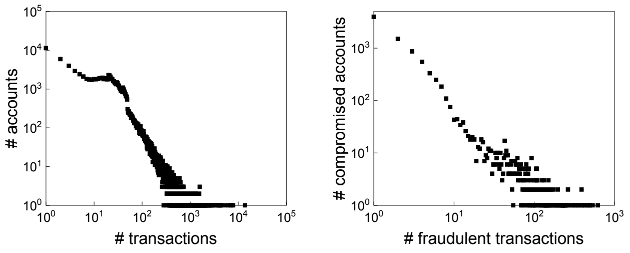


Figure 5: The imbalanced distributions of transactions and fraudulent transactions.

size of transaction sequence windows, denoted as w, is a critical parameter. It not only decides how much historical information of an account is used, but also determines the start point of the risk prediction task.

For each account, we set a fixed sequence window size w. Then, for each user, we only start predicting the risk of account after the user completes its $w-th$ transaction. For example, if we set $w=3$, we can predict the risk of an account only after it generates three transactions.

4.3 Evaluation Metrics

In general, there are many metrics to evaluate the performance of binary classifiers, such as AUC (area Under the ROC curve) score, F-measure, and KS (Kolmogorov-Smirnov) score. However, these metrics *cannot* directly reflect the economic influence of models, especially in the case of imbalanced data. With the consideration of practical usage, we use *precision*, *recall*, *false positive rate* and *F1-score* as evaluation metrics. As a classification model often outputs the numerical probability of a transaction being fraudulent, we need to set a threshold to determine whether fraud occurs. The threshold actually provides a tradeoff between precision and recall, and may be different for different classifiers. We adopt false positive rate (FPR) as a principle for choosing thresholds.

With a fixed value of FPR, we can get different thresholds for different classifiers. We can also calculate their recall, precision and F1-score under the fixed FPR. This can be used for model selection. For models with the same FPR value, large values of the chosen metrics suggest good model performance. In our online payment scenario, FPR must be smaller than 0.001, otherwise the model makes no sense.

FPR	Classification	Precision	Recall	F1-score
0.001	XGB	0.9515	0.9421	0.9468
	RF	0.9511	0.9370	0.9440
	LR	0.8885	0.3839	0.5362
	DNN	0.9499	0.7637	0.8467
0.0005	XGB	0.9746	0.9273	0.9504
	RF	0.9745	0.9208	0.9469
	LR	0.9306	0.3201	0.4762
	DNN	0.9582	0.4634	0.6247
0.0001	XGB	0.9879	0.4091	0.5786
	RF	0.9638	0.3259	0.4871
	LR	0.2293	0.0014	0.0028
	DNN	0.7529	0.0469	0.0883

Table 3: Comparison of Classifiers at Different FPR.

4.4 Comparison of Benchmark Classifiers

We compare the performance of four popular classification models, including XGBoost (XGB), Random Forest (RF), Logistic Regression (LR), and Deep Neural Network (DNN). In the DNN model, the sigmoid function is used as activation function and there are 3 hidden layers in addition to the input and output layers; the neurons at the three hidden layers are 20, 30 and 20, respectively. We set the window size $w=2$ and repeat this process 3 times. Table 3 shows the average precision, recall and F1-score on testing data. Note that we set $FPR=0.001$, $FPR=0.0005$ and $FPR=0.0001$, respectively. We learn that XGBoost not only performs better, but shows superior stability and robustness. Then, we choose XGBoost as the classifier of benchmark interim method.

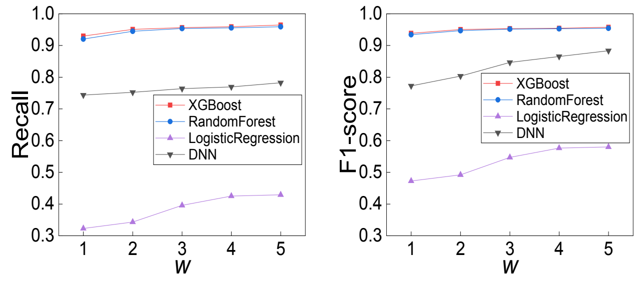
4.5 Impact of Window Size on Prediction

We need to analyze the impact of the transaction sequence window size on the perdition performance. We increase the window size w from 1 to 5, and compare the performance of corresponding models. The size of transaction sequence window decides when to start the risk evaluation. For example, when $w=1$, we can predict the risk of an account after it generates one transaction; but when $w=5$, we cannot make a prediction until the account finishes five transactions. In order to analyze the impact of window size reasonably, our prediction starts from the sixth transaction for each account in the testing data.

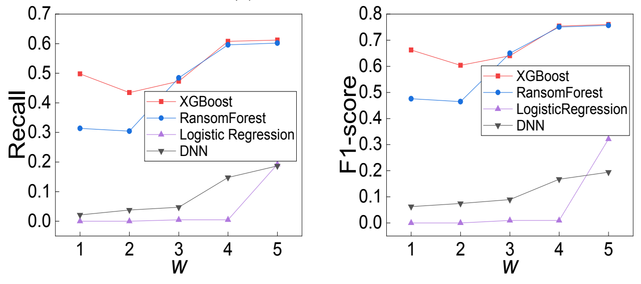
Figure 6 shows that the window size has a significant impact on performance. The recall and F1-score of all models increase with the increase of window size given fixed FPR. The reason lies in that larger window size ensures more behavioral information from account holders. In practice, we can choose a properly large window size, but for our testing data, when the window size increases by 1, almost 20 thousand transactions are made unpredictable. For example, when $w=5$, there are about 0.1 million transactions that cannot be predicted in our testing data. As an explanation, most accounts have only a few transactions. Therefore, in our experiment, we do not consider window size larger than 5.

4.6 Impact of Class Ratio for Prediction

We compare the effect of class ratio C_R on the performance of classifiers. In our experiment, we increase the class ra-

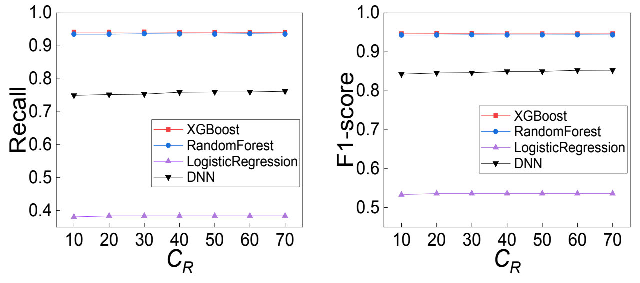


(a) $FPR = 0.001$

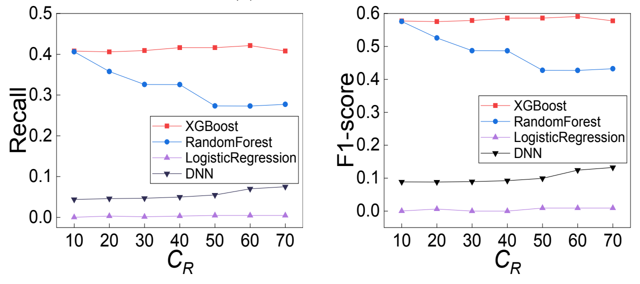


(b) $FPR = 0.0001$

Figure 6: Recall, F1-score at different FPR and w ($C_R = 30$).



(a) $FPR = 0.001$



(b) $FPR = 0.0001$

Figure 7: Recall, F1-score at different FPR and C_R ($w = 2$).

tio C_R from 10 to 70, and set the window size $w = 2$. We repeat this process 3 times, Figure 7 presents the average recall, F1-score of testing data when $FPR = 0.001$ and $FPR = 0.0001$, respectively, where the x-axis is class ratio.

From Figure 7, we obtain that when $FPR = 0.001$, the change of C_R has no significant impact on all models. However, if we decrease the FPR to 0.0001, the performance of XGBoost has hardly changed with the increase of C_R, but the performance of Random Forest is decreased greatly. Although the change of C_R has little effect on Logistic Regression and DNN, their performance is very poor when $FPR = 0.0001$.

4.7 Comparison with Interim Detection

In this part, we compare our ex-ante fraud detection method with the traditional interim fraud detection method. As stated before, we adopt XGBoost as the classifier in both methods.

Activation detection on the fly is widely-used fraud detection method. Different from ours, the interim fraud detection method examines every currently ongoing transaction of an account by comparing it with the account's historical transactions. Once the currently ongoing transaction is determined to be fraudulent, the system will immediately terminate the transaction. In order to train an effective interim fraud detection model, it is important to extract useful features with data and business experience. The feature extraction process for interim fraud detection is described as follows:

User historical behaviour features. We use the most recent month and day as the time interval to calculate the payment behaviour of users in these intervals. We extract a set of user-behaviour-related features, including the statistics of transaction volume, the statistics of transaction amount, the trading period and the statistics of other historical trading characteristics.

Merchant historical behaviour features. Similar to the users' historical behaviour feature extraction, we extract 9 statistical features related to merchants.

User-merchant features. We extract 2 features related to user-merchant, including the number of transactions per user in different merchants and the average amount per transaction.

Ongoing payment features. To extract features of an ongoing transaction, we digitize its non-numeric fields, and calculate the difference between the ongoing transaction and its previous transaction, such as the time intervals and amount differences. We extract 12 features related to ongoing transaction in all.

Figure 8 illustrates the comparison of the receiver operating characteristic (ROC) between two methods. Although the interim fraud detection method outperforms ours, the advantage is very limited. The strong point of our method is that it can prevent fraudulent transactions from occurring, while still keeping very good performance. This enables financial platforms to inform users of account risks in advance, and remind them to protect their accounts timely.

5 Enhanced Anti-Fraud Scheme

As the fraud prevention and fraud detection are the two mostly used schemes to combat fraud in practice, we design a hybrid anti-fraud system that combines both methods.

5.1 Hybrid System Architecture

As depicted in Figure 9, the enhanced scheme consists of two modules as follows:

Ex-ante risk prediction module. When an account finishes a transaction x_t, the system immediately uses the ex-ante risk prediction method to predict the risk of account. If the risk is too high, the system will immediately lock the account

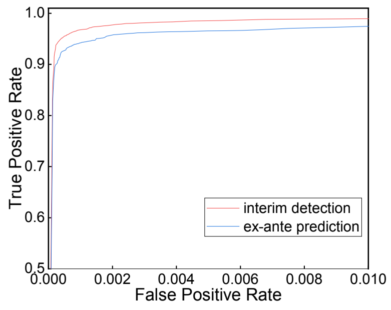


Figure 8: ROC Curves.

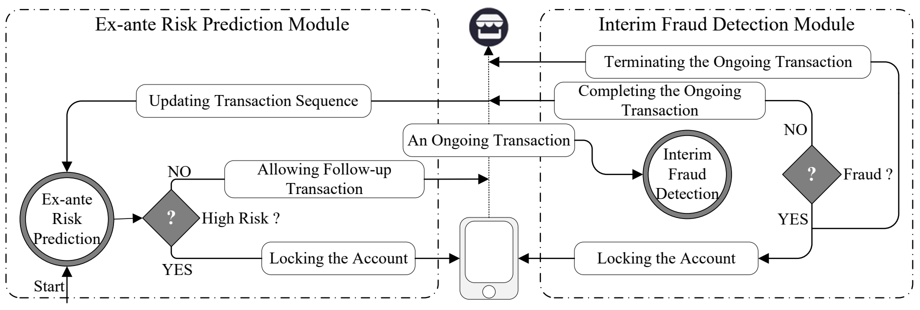

Ex-ante Risk Prediction Module
Interim Fraud Detection Module
Terminating the Ongoing Transaction
Updating Transaction Sequence
Completing the Ongoing Transaction
An Ongoing Transaction
NO
Allowing Follow-up Transaction
Interim Fraud Detection
Ex-ante Risk Prediction
?
High Risk ?
Fraud ?
YES
Locking the Account
Locking the Account
Start

Figure 9: A real-time fraud prevention and detection system.

and prohibit all subsequent transactions. Otherwise, the system will allow the occurring of next transaction, say x_{t+1}.

Interim fraud detection module. If the ex-ante module does not lock the account and the account is making a new transaction x_{t+1}, the system will use the interim fraud detection module to check the ongoing transaction x_{t+1}. If x_{t+1} is determined to be fraudulent by this module, it will be terminated immediately and the account will be locked. Otherwise, it will wait for the ongoing transaction until it completes, update the transaction sequence of the account, and re-estimate its fraudulent risk using the ex-ante risk prediction module.

5.2 Anti-Fraud Performance Evaluation

We have applied the system to the bank's one-month B2C transaction data. There are 1028437 transaction records in this month, including 1003539 normal transactions and 24898 fraudulent transactions. In the ex-ante risk prediction module, we set $w = 2$. The core of our anti-fraud system is to select an appropriate risk threshold (RT) for the ex-ante risk prediction model. Figure 10 presents the recall and FPR with different RT, from which we can observe that the smaller the risk threshold, the larger the recall and FPR. It means that we can prevent and detect more fraudulent transactions at the cost of interrupting more legitimate ones. When $RT = 1$, it means that we only use the interim fraud detection method, without predicting the risk of accounts ahead of time. Although it interrupts very few legitimate transactions, many fraudulent transactions *cannot* be detected.

In practical applications, it is necessary to ensure that the false positive rate is less than 0.1%. Next, we present the results on the performance of ex-ante prediction module, interim detection module, and the integrated system, respectively, when $RT = 0.75$:

• Our real-time fraud prevention and detection integrated system can detect 94.46% of fraudulent transactions with less than 0.09% of legitimate transactions interrupted.

• It is remarkable that 80.42% of fraudulent transactions can be prevented by the ex-ante module with less than 0.04% of legitimate transactions interrupted, leaving only 14% of fraudulent transactions to be detected when they are ongoing.

• Figure 11 shows the distribution of fraudulent transactions over time intervals that the system can predict ahead of time. We observe that nearly 70% of fraudulent transactions can be predicted more than 5 seconds ahead of occurring.

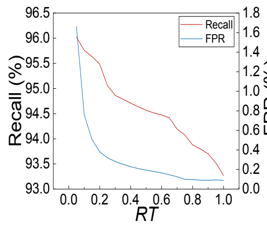


Figure 10: Recall and FPR with different thresholds.

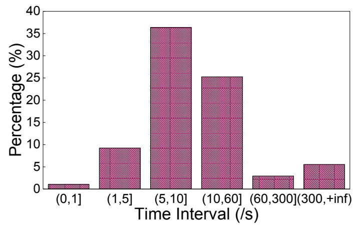


Figure 11: The cumulative distribution of predicted frauds.

6 Conclusion and Future Work

Different from most existing studies which usually aimed to design fraud detection methods of interim patterns, this work takes a different point of view to investigate whether transaction fraud can be detected in an ex-ante manner in an online payment service. We have obtained an insightful finding that there is really a relation between a user's historical payment behavior and his/her account compromise. Based on this finding, we can predict a fraudulent transaction before its occurrence without on-going transaction behaviors which are necessary for any interim fraud detection methods. Based on a real-world dataset, it is validated that this ex-ante fraud prediction can prevent more than 80% of fraudulent transactions before their actual occurrences. Moreover, utilizing their complementary effects, we have designed a real-time fraud prevention and detection system by combining the ex-ante and interim methods to further improve the effectiveness.

For future work, we will solve the *cold-start* problem of risk prediction for some accounts whose historical transaction count is smaller than the size of transaction sequence windows. In addition, it is also interesting to investigate how to overcome the *concept drift* problem [Lu *et al.*, 2019] in risk prediction.

Acknowledgments

The author sincerely thanks Pengfei Shu for his great efforts on this work, and thanks the anonymous referees for their helpful comments and suggestions. The research of author is partially supported by the National Natural Science Foundation of China (NSFC) under Grants 61972287, the Major Project of Ministry of Industry and Information Technology of China under Grant [2018] 282, and Municipal Human Resources Development Program for Outstanding Young Talents in Shanghai.

References

[Breiman *et al.*, 1984] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[Busa-Fekete *et al.*, 2017] Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, and Shie Mannor. Multi-objective bandits: Optimizing the generalized gini index. In *Proc. IEEE ICML 2017*, pages 625–634, 2017.

[Cao *et al.*, 2017] Bokai Cao, Mia Mao, Siim Viidu, and Philip S. Yu. Hitfraud: A broad learning approach for collective fraud detection in heterogeneous information networks. In *Proc. IEEE ICDM 2017*, pages 769–774, 2017.

[Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proc. ACM SIGKDD 2016*, pages 785–794, 2016.

[Elshaar and Sadaoui, 2020] Sulaf Elshaar and Samira Sadaoui. Semi-supervised classification of fraud data in commercial auctions. *Applied Artificial Intelligence*, 34(1):47–63, 2020.

[Jarovsky *et al.*, 2018] Ariel Jarovsky, Tova Milo, Slava Novgorodov, and Wang-Chiew Tan. Rule sharing for fraud detection via adaptation. In *Proc. IEEE ICDE 2018*, pages 125–136, 2018.

[Jing *et al.*, 2019] Chen Jing, Cheng Wang, and Chungang Yan. Thinking like A fraudster: Detecting fraudulent transactions via statistical sequential features. *Proc. International Conference on Financial Cryptography and Data Security 2019*, pages 140–155, 2019.

[Kim *et al.*, 2019] Eunji Kim, Jehyuk Lee, Hunsik Shin, Hoseong Yang, Sungzoon Cho, Seung-kwan Nam, Youngmi Song, Jeong-a Yoon, and Jong-il Kim. Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning. *Expert Syst. Appl.*, 128:214–224, 2019.

[Liu *et al.*, 2019] Shenghua Liu, Bryan Hooi, and Christos Faloutsos. A contrast metric for fraud detection in rich graphs. *IEEE Trans. Knowl. Data Eng.*, 31(12):2235–2248, 2019.

[Lu *et al.*, 2019] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Trans. Knowl. Data Eng.*, 31(12):2346–2363, 2019.

[McGlohon *et al.*, 2009] Mary McGlohon, Stephen Bay, Markus G. Anderle, David M. Steier, and Christos Faloutsos. SNARE: a link analytic system for graph labeling and risk detection. In *Proc. ACM SIGKDD 2009*, pages 1265–1274, 2009.

[Milo *et al.*, 2018] Tova Milo, Slava Novgorodov, and Wang-Chiew Tan. Interactive rule refinement for fraud detection. In *Proc. EDBT 2018*, pages 265–276, 2018.

[Mittal and Tyagi, 2020] Sangeeta Mittal and Shivani Tyagi. Computational techniques for real-time credit card fraud detection. In *Handbook of Computer Networks and Cyber Security, Principles and Paradigms*, pages 653–681. 2020.

[Mohammed *et al.*, 2018] Rafiq Ahmed Mohammed, Kok Wai Wong, Mohd Fairuz Shiratuddin, and Xuequn Wang. Scalable machine learning techniques for highly imbalanced credit card fraud detection: A comparative study. In *Proc. PRICAI 2018*, pages 237–246, 2018.

[Nami and Shajari, 2018] Sanaz Nami and Mehdi Shajari. Cost-sensitive payment card fraud detection based on dynamic random forest and *k*-nearest neighbors. *Expert Syst. Appl.*, 110:381–392, 2018.

[Porwal and Mukund, 2019] Utkarsh Porwal and Smruthi Mukund. Credit card fraud detection in e-commerce. In *Proc. IEEE TrustCom/BigDataSE 2019*, pages 280–287, 2019.

[Pozzolo *et al.*, 2018] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Trans. Neural Netw. Learning Syst.*, 29(8):3784–3797, 2018.

[Sangers *et al.*, 2019] Alex Sangers, Maran van Heesch, Thomas Attema, Thijs Veugen, Mark Wiggerman, Jan Veldsink, Oscar Bloemen, and Daniël Worm. Secure multiparty pagerank algorithm for collaborative fraud detection. In *Proc. International Conference on Financial Cryptography and Data Security 2019*, pages 605–623, 2019.

[Wang *et al.*, 2017] Shuhao Wang, Cancheng Liu, Xiang Gao, Hongtao Qu, and Wei Xu. Session-based fraud detection in online e-commerce transactions using recurrent neural networks. In *Proc. ECML PKDD 2017*, pages 241–252, 2017.

[Wang *et al.*, 2019] Cheng Wang, Bo Yang, Jipeng Cui, and Chaodong Wang. Fusing behavioral projection models for identity theft detection in online social networks. *IEEE Trans. Comput. Social Systems*, 6(4):637–648, 2019.

[Wu *et al.*, 2017] Fei Wu, Xiao-Yuan Jing, Shiguang Shan, Wangmeng Zuo, and Jing-Yu Yang. Multiset feature learning for highly imbalanced data classification. In *Proc. AAAI 2017*, pages 1583–1589, 2017.

[Yang *et al.*, 2014] Qinghong Yang, Xiangquan Hu, Zhichao Cheng, Kang Miao, and Xiaohong Zheng. Based big data analysis of fraud detection for online transaction orders. In *Proc. CloudComp 2014*, pages 98–106, 2014.

[Ying *et al.*, 2018] Josh Jia-Ching Ying, Ji Zhang, Che-Wei Huang, Kuan-Ta Chen, and Vincent S. Tseng. *FrauDetector*$^{+}$: An incremental graph-mining approach for efficient fraudulent phone call detection. *ACM Trans. on Knowledge Discovery from Data*, 12(6):68:1–68:35, 2018.

[Zhang *et al.*, 2019] Ya-Lin Zhang, Jun Zhou, Wenhao Zheng, Ji Feng, Longfei Li, Ziqi Liu, Ming Li, Zhiqiang Zhang, Chaochao Chen, Xiaolong Li, Yuan (Alan) Qi, and Zhi-Hua Zhou. Distributed deep forest and its application to automatic detection of cash-out fraud. *ACM Trans. Intelligent Systems and Technology*, 10(5):55:1–55:19, 2019.

[Zheng *et al.*, 2019] Panpan Zheng, Shuhan Yuan, Xintao Wu, Jun Li, and Aidong Lu. One-class adversarial nets for fraud detection. In *Proc. AAAI 2019*, pages 1286–1293, 2019.