Interpretable Electronic Transfer Fraud Detection with Expert Feature Constructions

Yu-Yen Hsin[1], Tian-Shyr Dai[2], Yen-Wu Ti[3] and Ming-Chuan Huang[4]

[1]*Institute of Computer Science and Engineering, National Yang Ming Chiao Tung University,Hsinchu 300, Taiwan*

[2]*Department of Information Management and Finance and Institute of Finance, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan*

[3]*College of Artificial Intelligence, Yango University, Fujian 350015, China*

[4]*Institute of Finance, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan*

Abstract

Since the magnitude of financial frauds grow rapidly with low clearance rates, detecting and avoiding frauds has been a tremendous challenge for financial institutions. Both the detection performance and interpretability are critical for fraud detection to profile the fraudsters' modus operandi and to spot vulnerabilities of financial systems/processes. Traditional rule-based approaches yield poor detection performances. Recent machine learning methods basically generate recency, frequency, and temporal features to extract patterns from raw transaction data. On the other hand, this paper generates behavioral and (financial organization's) segmentation features based on financial expertise and characteristics solely belonging to (non)-fraudulent accounts. While inputting aforementioned features into different models and using accumulated features from past literature generate unstable prediction results, our features generate the best and stable results for the decision-tree-base approach like Extreme Gradient Boosting and Light Gradient Boosting Machine. By using Kolmogorov–Smirnov test, we discover the instable predictive results are caused by vastly different distributions of features that reflects the fast-changing modus operandi in the training/testing sets. Thus, generating training/testing sets by random sampling (compared to chronological separation) is improper for modeling time varying data. Combining XGBoost with our expertise-based features provides clear causal-effect between features and fraudulent labels for further interpretations. The high precision and recall rates allow banks to save screening labor costs and identify frauds without interfering with normal transactions. The quality of our features can be examined by showing that they occupy three out of the five most important features under the ranking procedure in a premium finance publication by Butaru et al. [*Journal of Banking and Finance (72) 218–239* (2016)].

Keywords

Electronic Transfer Fraud Detection, Feature Engineering, Boosted Decision Tree, Interpretability

1. Introduction

As financial technologies and services evolve, the magnitude and variations of financial frauds have spawned rapidly. Common financial frauds include (electronic) transfer frauds, credit-card frauds, money laundering, insurance frauds, and so on. These frauds not only cause substantial financial losses but also induce a significant management cost for law enforcement units and financial institutions. Specifically, electronic transfer frauds denote that malicious scammers guide victims by phones or social media to transfer their savings to accounts controlled by scammers. Communication fraud control association showed that the worldwide fraudulent loss in 2019 is 28.3 billion with extremely low clearance rates[1]. Various electronic funds transfer EFT scams, like buyer overpays, romance scams, $\cdots$[2] make them hard to be prevented and detected. Fraud prevention acts have also been enacted in many countries [1], and developing effective and efficient automatic EFT fraud precaution mechanisms is important in practice and in academic researches. For example, fraud prevention acts such as the Money Laundering Control Act, the Money Laundering Prevention Act, and the Proceeds of Crime Act, (see [1]) have been enacted in Taiwan.

Many commercial banks have adopted the rule-based method for fraud detection which takes the guidelines in the fraud prevention acts and established a set of static rules to spot suspicious accounts. However, this method fails to capture complex features of fraudulent behaviors and the fast-changing modus operandi [2]. Our co-working bank (denoted as Bank L) reported that the "Rule-Based" method produces lousy precision rate (40%) and recall rate(5.56%). As a consequence, substantial screening labor costs and frequent disturbance of normal clients are incurred without effective crime prevention. Therefore, constructing a fraud detection system with high

Woodstock'21: Symposium on the irreproducible science, June 07–11, 2021, Woodstock, NY

dismentopia.dif04@nctu.edu.tw (Y. Hsin); cameldai@mail.nctu.edu.tw (T. Dai); ywdi@ygu.edu.cn (Y. Ti); a870523tw.cs09@nycu.edu.tw (M. Huang)

This research was funded by the Ministry of Science and Technology (MOST) of Taiwan under grants MOST 109-2218-E-009 -015

CEUR Workshop Proceedings (CEUR-WS.org)

[1] See https://cfca.org/sites/default/files/Fraud Loss Survey_2019_Press_Release.pdf.

[2] See https://www.worldremit.com/en/stories/story/2020/01/20/money-transfer-scams

precision and recall rates is thus critical.

Recent researches broadly apply machine learning to detect EFT frauds. To address the high-dimensional-nature of the raw transaction data, [3] and [4] suggest to construct features to retrieve patterns from raw transaction data. [5, 6, 7, 8] collect features proposed in past literature and categorize them into recency, frequency, monetary, (unsupervised) anomaly detection, and other feature engineering techniques. Most of the aforementioned features are constructed based on math or statistical properties without involving much financial expertise. [5] argues that clever feature constructions could yield good detection results, we create features based on financial expertise[3] and observations. By following [9] idea that creating features to capture the patterns for either positive or negative observations, we created features specifically for fraudulent accounts and (non)-fraudulent ones. For example, fraudsters usually try to empty an account by withdrawing at the largest amount available in the ATM to maximize the fund-transfer speed. And the account's last withdrawal amount is usually larger than its afterwards account balance. In addition to detecting fraudulent behaviors, we also create features that describe non-fraudulent behaviors to reduce harassment on non-fraudulent accounts. For example, over-the-counter services require account holders to be present in the business office and are unlikely to be utilized by fraudsters. On top of the aforementioned five feature categories, we constructed two new categories: behavioral and segmentation for our new features.

To compare the performances of our proposed behaviors, we first collected and implemented features from past literature [10, 11, 8, 3, 4, 12, 7, 13, 1, 14, 5, 15, 16, 17]. Then we compare the performance of Extreme Gradient Boosting (abbreviated as XGBoost), Bayes-Point machine, Random Forest, support vector machine (abbreviated as SVM), Neural Network, Logistic regression and light gradient boosting machine (abbreviate as LGBM) by inputting the features generated in past works and the features generated in this paper. Our experimental results suggest that XGBoost produces the best detection results unless certain "noisy" features are inputted. These noisy features are examined to have different distributions by Kolmogorov–Smirnov test; in addition, the recall (precision) rates deteriorate with the level of difference in features distributions of fraudulent (non-fraudulent) observations. Thus, instead of separating the data into training and testing sets by chronological orders, generating training/testing set by randomly sampling the data could faultily generate good detection results since the time-varying properties of noisy features are alleviated. Besides, XGBoost also produces comparable results with our features even though the number of our features is much lower than features in past works; this entails that generating features by expertise does work.

Interpretability is important for an EFT fraud detection system to profile fraudulent behaviors and to discover EFT system vulnerabilities. This gives us clear guidelines to avoid EFT systems from being utilized by fraudulent transfers (see [18]) and to fulfill the "risk-oriented"[4] property (of a money laundry system) required by the Financial Action Task Force[5](see [19] and [20]). Combining XGBoost with our features not only produces very good detection results but ranks the importance of features that can be directly explained by expertise. The good qualities of our features and categories can also be examined by showing that they occupy three out of the five most important features under the ranking procedure published in a premium financial journal [21].

The rest of this paper is organized as follows. Section 2 reviews related works on financial fraud detection and feature construction. Section 3 describes the formats of our raw data and the data-preprocessing procedures. Section 4 first collects the features of past fraud detection researches. Then we describe how our features and new feature categories are generated based on financial expertise and observations. Section 5 compare fraud detection performances with different training models and input features. We explore the properties of bad features that deteriorate detection performances and show that clever feature constructions with decision tree models could yield good detection rates and interpretability. Section 6 concludes this paper.

2. Related Works

There are many types of financial frauds, such as credit card frauds [22, 16, 17, 14], phone fraud [23], online transaction fraud [24], instant payment fraud [25], and so on. The rule-based method is commonly used to detect frauds by conventional banks for high interpretability. However, it is difficult for the rule-based method to capture complex and time-varying fraud patterns and the detection performance is hence related low. In addition, fixed rules may be easily cracked once fraudsters became aware of them.

Training a machine learning model with raw transaction data is impractical since fraudulent transactions are too rare to meet very high dimensionality of raw data. [4] and [3] suggest to construct features to extract information from raw transaction data to train a machine learning model for fraud detection. The aforementioned feature engineering process is called feature construction

[3] Like the anti-money laundry (AML) guidelines from Taiwan's financial supervisory commission.

[4] Efforts should be allocated where the risk of money laundering is higher.

[5] https://www.fatf-gafi.org/

(generation) (see [26]), and it can develop a more profound insight into characteristics of fraudulent and non-fraudulent accounts. Obviously, the qualities of feature constructions significantly influence detection results. [7] argue that many features are generated based on the frequency of transactions. But only exploring temporal features without considering financial/fraud detection expertise could significantly prevent a machine learning model from recognizing complete fraudulent behaviors. [6] studies suggest that most recent fraudulent detection works construct their input features based on RFM (recency, frequency. and monetary) categories. In addition to RFM, [5] show that some past studied features can be categorized into two new categories: (unsupervised) anomaly detection, and other feature engineering techniques. By implementing past studied features, they empirically show that clever feature engineering can yield very good detection results even with simple learning models like classification trees. Similarly, in addition to frequency features, [7] explained that frequency features are unable to capture chronological relationships in transactions. For example, fraudsters tend to first make some small transactions and then make a big one. They also show that incorporating interpretable monetary features derived from fraudulent behaviors can greatly enhance detection performance. In light of their observations, this paper creates new features through financial expertise and meticulous observations of (non-)fraudulent behaviors. we also augment two more categories of features according to typical (non-)fraudulent behaviors and segmentational properties.

Since a decision tree is inherently interpretable (see [18]) and our proposed features (categories) are constructed based on financial expertise and observations, combining our features and the XGBoost provides good detection results and proper causal explanations as discussed later. Thus our model is practical and can avoid flawed or unfair AI usages. The interpretability of AI models has attracted widespread attention in both academics and especially practical applications in finance and law. According to the evaluation indicators based on human subject-based evaluation metrics proposed by Moraffah [27], our model is causal interpretable that can explain and predict the classification results. Our proposed features can be explained by financial expertise and the fraudulent detection results also conform to human intuition. Good interpretability makes our model more suitable to meet financial institutions' requirements.

3. Data Descriptions and Preprocessing

Our data set contains transaction data during Apr. 2018 to Sep. 2018 from Bank L and the fraudulent accounts from the National Police Agency. Sophisticated features are designed in this paper based on real raw transaction data that contains featureful details as in Table 1. We do not test our method with public fraudulent detection data sets on the Internet [6] since the limited disclosure information provided by public datasets due to strict protection of privacy regulations prevent the construction of expertise features. The survey paper [28] also show that some anti-fraud works, like [29] and [30], only raise their methods without conducting experiments due to the aforementioned problem. Inputting the entire transaction record to a fraud detection system is impractical (see [4]) due to very high dimensionality of raw data and heterogeneity of transactions. Thus they aggregate $\boldsymbol{n}$ transactions for each user within a fixed time interval and extract features from these transactions. To our knowledge, this technique is widely used in recent researches. However, deciding the value of the hyper-parameter $\boldsymbol{n}$ results in a trade-off. Specifically, as $\boldsymbol{n}$ decreases (increases), less (more) transactions are aggregated to describe the characteristics of an account but more (less) accounts are eligible to be included as training/test data. This is because many accounts do not have frequent transactions and will be removed if $\boldsymbol{n}$ becomes large. But removing samples can be detrimental to fraud detection since fraudulent samples are scarce and some of them are seldomly transacted. To strike a balance between the number of aggregated transactions and the removed accounts, we choose $\boldsymbol{n}$ to be 9 as illustrated in Table 2. It can be observed that the percentage of fraudulent accounts being included in the trading data drop rapidly when $\boldsymbol{n}$ exceeds 9. The ratio of aggregated transactions of fraudulent accounts to all

Table 1
Structure of Transaction Data

Item	Description
Account ID	Unique identification number for each account
Transaction Date	The date when the transaction took place
Transaction Type	The transaction type (in code) such as ATM intra-bank withdrawal and at the counter deposit.
Withdrawal Amount	The withdrawal amount. It is empty if the transaction is not withdrawal.
Deposit Amount	The deposit amount. It is empty if the transaction is not deposit.
Account Savings	The balance of the account after performing the transaction.
Note	Textual information of the transaction, like "transferred to company X", or the ATM ID for performing the transaction.
Internet	Whether the transaction is performed through E-bank services
Voice	Whether this transaction is performed through telephony services
Warned	Whether the account is fraudulent or not.

[6] Like https://www.kaggle.com/ealaxi/paysim1?fbclid=IwAR1wwa2npiZsoLHf1yNUTODJU z_x JoCQ5eKOLLpDMBkmyGDnNz2OIsmxcac

Table 2
Percent of Used Accounts and Transactions Given n

Per1: % of fraudulent accounts that contains more than n transactions to all fraudulent accounts
Per2: % of aggregated transactions of fraudulent accounts to all fraudulent account transactions

n	Per1	Per2
4	100.00%	19.40%
5	95.36%	23.13%
6	93.30%	27.15%
7	90.72%	30.80%
8	87.63%	34.00%
9	85.57%	37.35%
10	78.87%	38.25%
11	69.59%	37.13%
12	64.43%	37.50%
13	58.76%	37.05%
14	51.55%	35.00%
15	47.42%	34.50%
16	43.81%	34.00%
17	38.66%	31.88%

transactions of these accounts during the time period is also the highest for $n \leq 9$ scenarios.

4. Feature Construction

[4, 3] create features to extract information from aggregated transaction data and then use these features to train a machine learning model.

This approach is generally adopted in machine-learning-based fraud detection papers. To comprehensively analyze the characteristics of past features and analyze their effectiveness, we collect features that can be applied to our raw data in Table 4 from past literature, including [10, 11, 8, 3, 4, 12, 7, 13, 1, 14, 5, 15, 16, 17], and these features[7] are denoted as ***Others*** in the following experiments. On the other hand, we create two new sets of features: behaviors and segmentation, that are generated based on financial expertise or observations. Features that are first proposed in this paper will be denoted as ***Ours***. The categories of ***Ours*** and ***Others*** features are illustrated in Table 5. The definitions of ***Others*** and ***Ours*** features are listed in Table 4 and later in this section, respectively. Our experiments suggest that our proposed features can significantly improve the performance of the fraudulent detection model.

In addition to the aforementioned RFM features, [7] and [5] create features only based on anomaly properties of fraudulent accounts; this is, they focus solely on identifying typical fraudulent behaviors. This is because fraudsters act very similarly to a normal user for most of the time, and fraudulent behaviors usually take place in a short period of time. Thus it is intuitive to pinpoint "what fraudsters would do" to identify fraudsters. On the other hand, profiling certain normal behaviors could also be beneficial since fraudsters avoid these behaviors due to the risk of getting caught or due to potential disturbances to their criminal schemes. Taking account of "what fraudsters would not do" in addition to "what fraudsters would do" is beneficial to identifying normal users whose transaction characteristics are closer to fraudsters; intuitively, this improvement could alleviate the harassment to normal clients and hence reduce screening labor costs for fraud detection as well as reducing the heavy cost of dealing false alarms for banks [8]. But we find that our way of constructing features with respect to "what fraudsters would (not) do" could not be easily classified into categories proposed by [5] and therefore we augmented the categories by inserting "Behavioral" and "Segmentation".

Behavioral features denote specific transaction attributes that do not relate to RFM and anomaly detection techniques but are considered to be important in fraud detection according to financial expertise. These features include:

ATM_Transaction: It is defined as the number of ATM transactions of all 9 aggregated transactions (defined in Table 2) of an account. A Bank-L's expert suggests that most EFT frauds include transferring and dispatching money through ATM services for it is faster and involves less risks of being caught. In fact, there are 405 types of transactions and creating features for every type results in unnecessary dimensions of inputs, deteriorating fraud detection performances.

Immediate_Withdraw:

It is defined as the number of times a withdrawal happening right after a deposit within the same trading day. This is because fraudsters strive to withdraw the illicit money prior to polices' investigations and freezing the suspicious accounts.

Internet:

It is defined as the number of times an account uses the E-bank service. Very few fraudsters ever used the E-bank service since additional personal information should be provided at the bank counter in advance to enable certain E-bank services like wire transfer.

Voice: Similar to the descriptions for **Internet**.

LT_Count:

It is defined as the number of times an account conducted "likely-legal" transactions, which are defined as transaction types that have never been used by fraudulent accounts. Such transaction types may increase the risk of being caught or identified, like withdrawal/deposit at bank counters, Or they are unrelated to criminal schemes, such as the purchase/redemption of funds provided by Bank-L.

Last_Withdrawal_Larger_Than_Savings: It indicates a

[7] Some features like transaction locations that cannot be retrieved from the raw data in Table 1 are ignored.

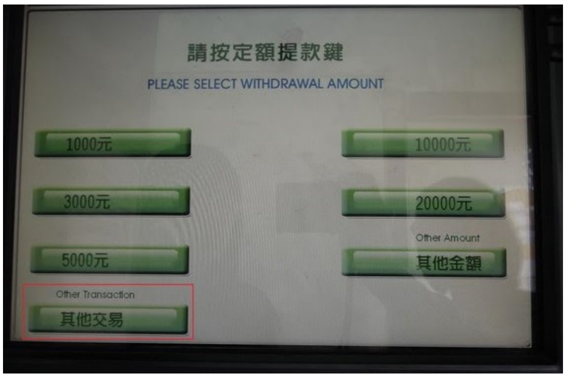
specific scenario that the last withdrawal amount is larger than the account balance. This is because fraudsters would transfer as much illicit money as possible from a fraudulent account under the limitation of the ATM: the minimum withdraw amount is a 1000 or 100 Taiwan dollar note. Note that no withdrawals can be made after the fraudulent account is frozen.

Suspicious_Score: It computes the suspicious likelihood of frauds by the products of several suspicious features and the equation is defined as follows:

$$\begin{aligned}
&(Last\_Withdrawal\_Larger\_Than\_Savings) \\
&\times (Suspicious\_Amount\_Count)/n \\
&\times (Immediate\_Withdraw)/n \\
&\times (ATM\_Transaction)/n,
\end{aligned} \tag{1}$$

where "Suspicious_Amount_Count" is a monetary feature proposed by [7] that denotes the maximum or the quick-transfer amount in an ATM. In Taiwan, the maximum inter-bank withdraw amount is $60,000 (Taiwanese Dollar). The maximum cross-bank withdraw amount is $20,000 plus the service fee $5. The maximum quick-transfer service is $10,000 plus the fee $5. The ATM withdraw menu is illustrated in Fig. 1 . Besides, n denote the number of aggregated transactions defined in Table 2. This feature allows us to capture simultaneous occurrences of suspicious features to precisely identify fraudulent transactions.

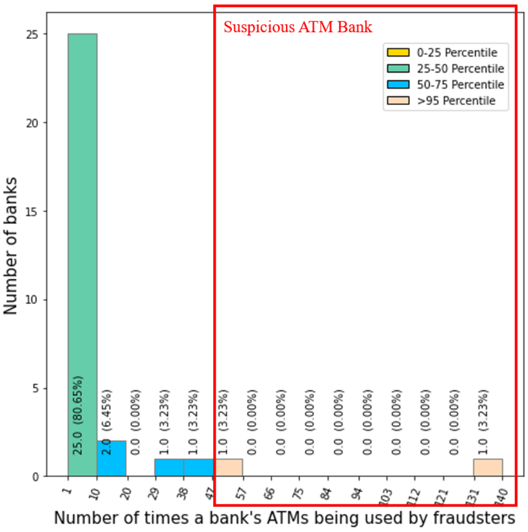
Figure 1: An Example Interface of Taiwanese ATM for quick-transfer options



Segmentation features label each account/ATM machine with the bank, branches, or other meaningful classifications that they belong to. These features include:

Suspicious_ATM_Bank: We first calculate the number of times each ATM has been accessed by fraudulent accounts and recognize its owner bank. Then we label the banks with top 5% lump sums (see Fig 2) of fraudulent ATM accesses as "Suspicious ATM Banks". 5% is a common statistical threshold for significance tests. High fraudulent accessed numbers of a specific Bank's ATMs are probably due to its management and location selection policies[8]. For instance, ATMs located in the vicinity of police station are less likely to be accessed by fraudsters.

Figure 2: Histogram and Percentile of the Number of Fraudulent Accesses of a Bank's ATMs. Each bar shows the the number of banks and the ratio (in parenthesis) to the total number of banks.

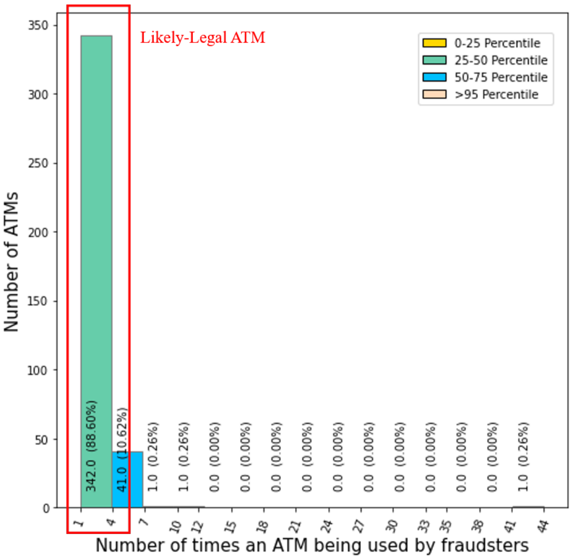


LATM_Count: It is defined as the number of times for an account to access "likely-legal ATMs", which are defined as ATMs that have been accessed by fraudsters less than 6 times. 6 is decided due to 95%(a widely adopted threshold in statistics) of ATMs in our training set data have been accessed more than 6 times (see Fig 3) by fraudulent accounts.

Suspicious_Branch: Label Whether an account's transactions are performed in areas with dense fraudulent activities. Although actual ATM locations are not accessible from our raw data, we can identify the branch to which each ATM belongs by comparing a bank's branches serial numbers and its ATM's serial numbers. Since ATMs belonging to a bank's branch office are normally located in its proximity, we can then profile each branch office with its own ATM data. We found that some ATMs have only been accessed by fraudsters with Bank-L's accounts. We then label the ATM's owner branches as suspicious branches (see Fig. 4). Note that this label does not imply a suspicious branch's ATMs are only accessed by fraudsters since we can only access Bank-L accounts' transaction data during a limited time span.

[8] For example, a bank may corporate with a chain store to deploy ATMs to its branch stores.

Figure 3: Histogram and Percentile of Fraudulent Accesses of ATMs. Each bar shows the the number of ATMs and the ratio (in parenthesis) to the total number of ATMs.



In addition to the aforementioned features, we also add one frequency feature, "Most_Frequent_Object_Count" that has never been studied before. It denotes the number of times an account transfers money to its most frequently transferred account.

Table 3
Top 5 important features in **XGBOOST with Others+Ours**

Xgboost Feature Importance		
Features	Importance	rank
LT_Count	2.08410446	1
Average_Transaction_Interval	1.696345674	2
untrusted_frequent_trade_count	1.401884328	3
Big_onetime_deal_count	1.320201917	4
BranchID_CPP	1.098959776	5

Table 3 ranks the importance of all features in Table 5, including the features constructed exclusively by us, based on the method proposed in a top financial expertise publication [21]. It can be observed that 3 out of 5 top important features belong to our proposed behavioral or segmentation categories. It confirms [5] claims that clever feature engineering can yield good detection results even with simple machine learning models.

5. Experiments

After constructing a feature vector defined in Table 5 for each account, we compare the fraud detection performance with different machine learning models and input features to show that clever feature constructions could yield good performance. The qualities of our proposed features (and corresponding categories) are examined by feature importance defined in [21] and detection results. We examine the following machine learning models: XGBoost, BayesPoint machine, Random Forest, SVM, Neural Network, Logistic regression and Light GBM.

In addition to detection accuracy, the performance of precision and recall are also demanded by commercial banks for practical reasons as follows. Precision reflects the ratio of actual fraudulent accounts detected to all accounts detected as fraudulent. A lower precision rate denotes that our fraudulent detection system would significantly affect the false positive observations since bank staffs will make phone calls and even freeze these accounts if necessary; these precaution procedures inevitably annoy normal users. On the other hand, recall measures how many real fraudsters can be identified in advance to block their further actions from causing potential financial losses. Financial regulatory authorities provide anti-money laundering documents[9] and require banks to provide

qualified detection mechanisms. To strike a balance between precision and recall, this paper use F1-Score to measure model performances, but how to find a proper β for F-score to reasonably measure the loss for annoying a normal user or for missing a fraudster from a bank point of view is still an open problem.

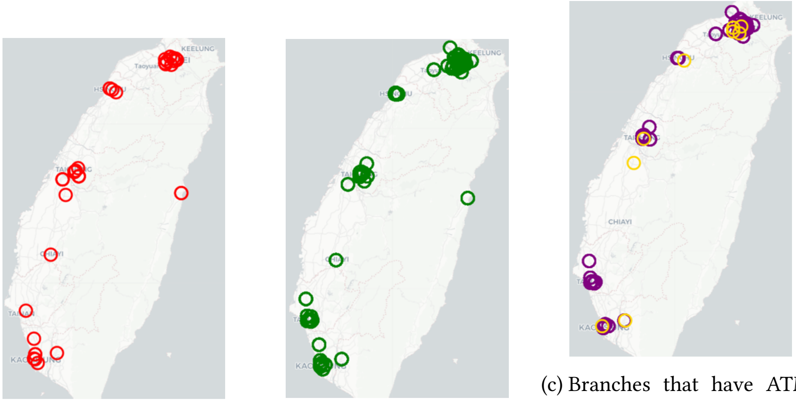
5.1. Analyzing Different Learning Models and Inputs

Here we first sort the observations by chronological order and divide the first 60% of the data as the training set and the last 40% for testing as illustrated in Table 6. Then we compare the fraud detection performance with different machine learning models and input features as illustrated in the "Model+Data" column. It can be observed that decision-tree-based models like XGBoost and LGBM can often yield very good detection results but on rare occasions perform badly due to different input features.[10] Specifically, using only ***Ours*** features produce a very good F1-Score (73.95%) while incredibly low Recalls and F1-Scores are produced if all features in Table 5 are included. It confirms [5] arguments that clever feature constructions do influence machine learning results since we could use less features (marked in red colors in Table 5) to achieve similar prediction results. To check for the causes for the severe drop in performance, we adopted

[9] See https://www.banking.gov.tw/en/home.jsp?id=17&parentpath=0,3

[10] https://medium.com/@aravanshad/gradient-boosting-versus-random-forest-cfa3fa8f0d80 also suggests that boosted decision trees such as XGBoost and LGBM are more sensitive to overfitting if the data is noisy.

Figure 4: Fraudulent and Normal Accesses of ATMs and Corresponding Owner Branches' Locations



(a) Branches that have ATMs being accessed by fraudsters in Bank-L

(b) Branches that have ATM being accessed by normal users in Bank-L

(c) Branches that have ATMs being accessed by only fraudsters (denoted by the yellow color) and normal users (in purple color) in Bank-L

leave-one-out feature selection to monitor the change in detection performance by repeatedly singling out one feature each time for all features. We discover that the significant drops in recalls and F1-scores are caused by two features from the (unsupervised) anomaly detection category: LOF and KNN_ distance. Removing these two features as in the right panel of Table 6 could restore the XGBoost performance which the F1-score is 74.34% (with Others+Ours without LOF&KNN_distance) and 68.42% (with Others without LOF&KNN_distance). Besides, although the performance of the non-linear-kernel SVM is worse than XGBoost when input parameters are properly designed, SVM performance tends to increase with the number of features without suffering from the problem of the aforementioned improper features, the non-linear-kernel SVM cannot provide feature importance for a bank to profile fraudulent behaviors or to identify the weaknesses of transaction process.

To find the reason of this anomaly, we repeat the aforementioned experiments with different proportions of the training/testing data in Table 7. If both LOF and KNN_ distance are removed, the F1-scores remain high and stable (from 71% to 80%) regardless of the changes of the proportion of the training data. On the other hand, incorporating LOF and KNN_ distance varies fraud detection results significantly with the changes of the training data proportion. Note that precision rates are stable and high(87.5%∼100%) but the recall rates vary significantly (1.49% to 70.59%). Because KNN_distance and LOF profiles an account's overall behavior (or the modus operandi for a fraudster), we can examine whether the patterns of modus operandi change in the training and the testing period by checking whether KNN_distance (or LOF) from the training and the testing period are drawn from the same distribution. We use Kolmogorov–Smirnov test (K-S test) to check for distribution similarity by calculating the likelihood of two realized distributions of samples being drawn from the same distribution.

The null hypothesis is "the two distributions are drawn from the same distribution" and we reject the null hypothesis to adopt the alternative one – the two distributions are different– if the p-value is small[11]. It can be observed that while the distributions of KNN_distance of legal accounts stays relatively coherent in training and testing periods (i.e., the p-values are high), the distributions of both KNN_distance and LOF of fraudulent accounts change drastically represented by very small p-values. This varying distribution phenomenon implies that the modus operandi changes over time. It can also be observed that the increment of the p-values of KNN_distance and LOF of fraudulent accounts also increase the recall rate– the likelihood to detect fraudulent accounts. Besides, the p-value of LOF of legal accounts also vary mildly, implying that the behaviors of legal accounts also change with time but not as severe as fraudulent accounts. The precision value even drops to 87.5% at the extreme case (p-value=0.003% when the training set accounts for 70% of data).

To confirm the aforementioned argument, we construct the training and the testing set by randomly sampling 60% and 40% of the aggregated transaction data, respectively, instead of separating the training/testing data chronologically. Unlike the results in Table 6, the ex-

[11] The null hypothesis is rejected with 99% confidence interval if $p < 0.01$.

Table 4
Features Proposed in Past Literature(***Others***)

Feature	Description
Sensitive single amount count	Counts the number of times an account's trans-action amount is abnormal, which is defined asan amount larger than the Maximum transac-tion amount minus the maximum quick-transferamoun.
Sensitive daily total amount count	Counts the number of times a client's single transaction amount over a single day is larger than the maximum daily transaction amount minus the maximum quick-transfer amount.
Sensitive test amount count	Counts the number of times the account has conducted exploratory tradings whose transaction amount is lower than the smallest quick-transfer amount.
Large amount count	Counts the number of times the account has made transactions of large amount. Large amount is defined as the amount larger 95% of all transaction amounts
Untrusted frequent trade count	Counts the number of times the account traded frequently (more than 4 times a day) to untrusted accounts (labeled by the bank).
Big one-time deal count	Counts the number of times large one-time transactions which transfers all the savings in the account.
Amount over month	Average transaction amount over the past 30 days.
Average daily over month	Average daily transaction amount over the past 30 days.
Average over 2 months	Average weekly transaction amount over the past 60 days.
Amount Transaction object over month	Average daily transaction amount with a specificcounter party over the 30 day period.
Number Transaction object over month	Total number of transactions with same counterparty over the 30 day period.
Amount Transaction object over 2 months	Average weekly transaction amount with a specific counter party over the 60 day period.
Max Amount same day	Maximum daily transaction amount of the account.
Max Number same day	Maximum number of transactions in the same day of the account.
Mahalanobis_Anomaly	Identify whether an account's attributes' Mahalanobis distance exceeds a given threshold, says the 97.5% quantile of the chi-squared distribution by setting the degrees of freedom as the number of features
KNN_distance	Average distance for an account to each of its k closest neighbors.
LOF	Average density around the k nearest neighbors divided by the density around the observation itself. It is considered anomaly if the ratio is above 1.
Isolation_Forest	Taking an ensemble of isolation trees that isolate each observation as quickly as possible. The final score is the average of the standardized path length (i.e. number of splits to isolate the observation) over all trees.
Fits_Benford	Judge whether the transaction amount's first leading digit fits Newcomb-Benford law's distribution.
Zscore_Outlier_count	Calculate Z-score separately for each account's transactions amount and count the number of times a transaction amount's Z-score is larger than 3.
Feature	Description
MAD	Median Absolute Deviation of each account's transaction amount.
Consecutive_transact_type_count	Count the number of times consecutive transaction type's being conducted.
Consecutive_transact_type_amount	Sum of the amount of consecutive transaction type being conducted.
Zero_Digit_Freq	Average number of the digit '0' in the transaction amounts.
Total Amount	Total transaction amount.
24hr_Transaction	Number of transactions over the last 24 hours.
Average_Transaction_Interval	Average length of period to perform a transaction.
Least_Frequent_Transaction_Type	The least frequently conducted transaction type (ie. Withdraw, Transfer etc.) of all transaction types that the account have conducted
Most_Frequent_Transaction_Type	The most frequently conducted transaction type (ie. Withdraw, Transfer etc.) of all transaction types that the account have conducted.
Second_Most_Frequent_Act	The second-most frequently conducted transaction type (ie. Withdraw, Transfer etc.) of all transaction types that the account have conducted.
Withdrawal_Stdev	The standard deviation of withdrawal amounts.
Deposit_Stdev	The standard deviation of deposit amounts.
Suspicious_Amount_Check	Count the number of times an account have conducted transactions with "suspicious amounts", like the maximum cross/inter-bank transaction amounts and the quick-transfer amounts defined on ATM machines.
Note_Not_Empty_Count	The number of times an account's transactions had attached addition text message.
BranchID	The Branch to which the account belongs.

perimental results illustrated in Table 9 suggest that the presence of LOF and KNN_ distance do not interfere the performance of XGBoost. Specifically, the F1-score is 86% for "XGBoost with Others+Ours" and 80% for "XGBoost with Others", even outperforming their counterpart models that remove LOF and KNN_ distance. This is because randomly sampling the training and testing set data from the same time span makes the KNN_distance (or LOF) distributions of the training and the testing set the same. This further implies that the model could foresee the change of future modus operandi beforehand, which is impractical in practice. Given the fast changing modus operandi, we conclude that it is inappropriate to examine a fraud detection model by random sampling.

6. Conclusion

This research constructs fraudulent detection features based on financial expertise and the characteristics of both fraudulent and non-fraudulent accounts. We identify the time-varying properties of the features that deteriorate detection performance. We show that combining XGBoost with our features provides very good detection performance and interpretability. The value of our feature constructions can be verified by showing that they occupy three out of the five most important features under the ranking procedure proposed in a premium financial journal [21].

References

[1] C. Tai, T. Kan, Identifying money laundering accounts, in: 2019 International Conference on System Science and Engineering (ICSSE), 2019, pp. 379–382.

[2] R. J. Bolton, D. J. Hand, Statistical fraud detection: A review, Statistical Science 17 (2002) 235–249.

[3] S. Bhattacharyya, S. Jha, K. Tharakunnel, J. C. Westland, Data mining for credit card fraud: A comparative study, Decision Support Systems 50 (2011) 602 – 613.

[4] C. Whitrow, D. Hand, P. Juszczak, D. Weston, N. Adams, Transaction aggregation as a strategy for credit card fraud detection, Data Mining and Knowledge Discovery 18 (2009) 30–55.

[5] B. Baesens, S. Höppner, T. Verdonck, Data engineering for fraud detection, Decision Support Systems (2021) 113492. doi:10.1016/j.dss.2021.113492.

[6] X. Zhang, Y. Han, W. Xu, Q. Wang, Hoba: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture, Information Sciences 557 (2021) 302–316. URL: https://www.sciencedirect.com/science/

Table 5
Feature Categories

Feature Categories Proposed in Bart Baesens (2021)

Recency	Frequency	Monetary	Features based on (unsupervised) [t] anomaly detection techniques [b]
24hr_Transaction	Average_Transaction_Interval	Withdrawal_Stdev	Mahalanobis_Anomaly
Amount_over_month	Least_Frequent_Transaction_Type	Deposit_Stdev	KNN_distance
Average_daily_over_month	Most_Frequent_Transaction_Type	Suspicious_Amount_Count	LOF
Amount_Transaction_object_over_month	Second_Most_Frequent_Act	Total Amount	Isolation_Tree
Average_over_2_months	Most_Frequent_Object_Count	Consecutive_transact_type_amount	Zscore_Outlier_count
Amount_Transaction_object_over_2_months		MAD	Fits_Benford
		Zero_Digit_Freq	
		Max_Amount_same_day	
		Max_Number_same_day	
		Sensitive_single_amount_count	
		Sensitive_daily_total_amount_count	
		Sensitive_test_amount_count	
		Large_amount_count	
		Big_onetime_deal_count	

Feature Categories Added in Our Research

Behavioral	Segmentation
Note_Not_Empty_Count	BranchID
Consecutive_transact_type_count	Suspicious_ATM_Bank
untrusted_frequent_trade_count	LATM_Count
ATM_Transaction	Suspicious_Branch
Immediate_Withdraw	
Internet	
LT_Count	
Voice	
Suspicious_Score	
Last_Withdrawal_Larger_Than_Savings	

Black: Others. Features used only by other researches
Red: Ours. Features used only by ours

Table 6
Fraudulent Detection Performances with Different Learning Models and Input Data with Chronologically separated Training (60%) and Testing (40%) Data

Model+Data	Accuracy	Precision	Recall	F1-Score
XGBoost with Others+Ours	99.60%	100.00%	1.49%	2.94%
BayesPoint with Others+Ours	1.49%	0.39%	97.01%	0.79%
RandomForest with Others+Ours	99.20%	30.12%	74.63%	42.92%
SVM with Others+Ours	99.67%	57.32%	70.15%	63.09%
NN with Others+Ours	15.46%	0.43%	91.04%	0.86%
Logistic with Others+Ours	99.60%	0.00%	0.00%	0.00%
LGBM with Others+Ours	99.60%	100.00%	1.49%	2.94%
XGBoost with Ours	99.81%	84.62%	65.67%	73.95%
BayesPoint with Ours	8.33%	0.44%	100.00%	0.87%
RandomForest with Ours	94.76%	6.20%	85.07%	11.56%
SVM with Ours	98.44%	16.90%	73.13%	27.45%
NN with Ours	37.37%	0.54%	83.58%	1.06%
Logistic with Ours	38.57%	0.48%	73.13%	0.95%
LGBM with Ours	99.78%	81.63%	59.70%	68.97%
XGBoost with Others	99.60%	100.00%	1.49%	2.94%
BayesPoint with Others	1.49%	0.39%	97.01%	0.79%
RandomForest with Others	97.87%	12.53%	71.64%	21.33%
SVM with Others	98.05%	14.17%	76.12%	23.89%
NN with Others	92.55%	0.17%	2.99%	0.32%
Logistic with Others	99.60%	0.00%	0.00%	0.00%
LGBM with Others	99.60%	100.00%	1.49%	2.94%

Model+Data	Accuracy	Precision	Recall	F1-Score
XGBoost with Others+Ours without LOF&KNN_distance	99.83%	91.30%	62.69%	74.34%
BayesPoint with Others+Ours without LOF&KNN_distance	13.11%	0.46%	100.00%	0.92%
RandomForest with Others+Ours without LOF&KNN_distance	97.09%	10.71%	85.07%	19.03%
SVM with Others+Ours without LOF&KNN_distance	99.67%	57.32%	70.15%	63.09%
NN with Others+Ours without LOF&KNN_distance	0.73%	0.38%	94.03%	0.76%
Logistic with Others+Ours without LOF&KNN_distance	91.12%	1.45%	31.34%	2.76%
LGBM with Others+Ours without LOF&KNN_distance	99.77%	79.59%	58.21%	67.24%
XGBoost with Others without LOF&KNN_distance	99.78%	82.98%	58.21%	68.42%
BayesPoint with Others without LOF&KNN_distance	11.43%	0.45%	100.00%	0.90%
RandomForest with Others without LOF&KNN_distance	92.13%	4.41%	89.55%	8.40%
SVM with Others without LOF&KNN_distance	96.98%	9.65%	77.61%	17.16%
NN with Others without LOF&KNN_distance	29.04%	0.13%	22.39%	0.25%
Logistic with Others without LOF&KNN_distance	93.32%	2.29%	37.31%	4.31%
LGBM with Others without LOF&KNN_distance	99.75%	74.51%	56.72%	64.41%

article/pii/S002002551930427X. doi:https://doi.org/10.1016/j.ins.2019.05.023.

[7] Y. Xie, G. Liu, R. Cao, Z. Li, C. Yan, C. Jiang, A feature extraction method for credit card fraud detection, in: 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS), 2019, pp. 70–75. doi:10.1109/ICoIAS.2019.00019.

[8] A. Correa Bahnsen, D. Aouada, A. Stojanovic, B. Ottersten, Feature engineering strategies for credit card fraud detection, Expert Systems with Applications 51 (2016). doi:10.1016/j.eswa.2015.12.030.

[9] A. Abdallah, M. A. Maarof, A. Zainal, Fraud detection system: A survey, Journal of Network and Computer Applications 68 (2016) 90 – 113.

[10] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, G. Bontempi, Learned lessons in credit card fraud detection from a practitioner perspective, Expert Systems with Applications 41 (2014) 4915–4928. doi:10.1016/j.eswa.2014.02.026.

[11] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, G. Bontempi, Credit card fraud detection: A realistic modeling and a novel learning strategy, IEEE Transactions on Neural Networks and Learning Systems PP (2017) 1–14. doi:10.1109/TNNLS.2017.2736643.

[12] A. Correa Bahnsen, D. Aouada, A. Stojanovic, B. Ottersten, Detecting credit card fraud using periodic features, 2015, pp. 208–213. doi:10.1109/ICMLA.2015.28.

[13] V. Van Vlasselaer, C. Bravo, O. Caelen, T. Eliassi-Rad, L. Akoglu, M. Snoeck, B. Baesens, Apate: A novel approach for automated credit card transaction fraud detection using network-based exten-

Table 7
Training XGBoost with Different Proportion of Training Data

(a) Training and testing sets sampled chronologically. Feature set used: **Others** and **Ours**

Training set ratio to all data	Accuracy	Precision	Recall	F1-Score	Fraudulent Account KNN_distance KS-test p-value	Fraudulent Account LOF KS-test p-value	Legal Account KNN_distance KS-test p-value	Legal Account LOF KS-test p-value
50%	99.68%	100.00%	20.48%	34.00%	7.91E-46	1.88E-25	1	0.306120444
60%	99.60%	100.00%	1.49%	2.94%	9.61E-57	1.81E-31	1	0.992635613
70%	99.74%	87.50%	42.00%	56.76%	2.72E-33	1.90E-16	1	0.00359492
80%	99.81%	90.91%	58.82%	71.43%	1.07E-17	1.03E-09	1	0.490359292
90%	99.88%	100.00%	70.59%	82.76%	3.31E-07	0.000551005	1	0.027677963

(b) Training and testing sets sampled chronologically
Feature set used: **Others** and **Ours** without KNN_distance and LOF

Training set ratio to all data	Accuracy	Precision	Recall	F1-Score
50%	99.78%	74.03%	68.67%	71.25%
60%	99.81%	79.66%	70.15%	74.60%
70%	99.80%	82.05%	64.00%	71.91%
80%	99.80%	86.96%	58.82%	70.18%
90%	99.86%	92.31%	70.59%	80.00%

Table 9
Fraudulent Detection Performances with Different Learning Models and Input Data with Randomly Sampled 60% Training and 40% Testing Data

Model	Accuracy	Precision	Recall	F1-Score
XGBoost with Others+Ours	99.90%	98.00%	76.60%	86.00%
BayesPoint with Others+Ours	1.30%	0.40%	96.90%	0.80%
RandomForest with Others+Ours	97.40%	12.10%	89.10%	21.30%
SVM with Others+Ours	99.70%	58.80%	73.40%	65.30%
NN with Others+Ours	2.90%	0.40%	100.00%	0.80%
Logistic with Others+Ours	99.60%	0.00%	0.00%	0.00%
LGBM with Others+Ours	99.90%	92.30%	75.00%	82.80%
XGBoost with Ours	99.80%	84.80%	60.90%	70.90%
BayesPoint with Ours	64.00%	1.10%	98.40%	2.10%
RandomForest with Ours	91.00%	3.40%	81.30%	6.60%
SVM with Ours	96.90%	7.90%	64.10%	14.00%
NN with Ours	91.50%	3.70%	82.80%	7.10%
Logistic with Ours	91.40%	3.70%	84.40%	7.10%
LGBM with Ours	99.70%	75.00%	51.60%	61.10%
XGBoost with Others	99.90%	95.70%	68.80%	80.00%
BayesPoint with Others	1.30%	0.40%	96.90%	0.80%
RandomForest with Others	94.20%	5.50%	85.90%	10.40%
SVM with Others	97.20%	9.30%	71.90%	16.50%
NN with Others	8.10%	0.40%	84.40%	0.70%
Logistic with Others	99.60%	0.00%	0.00%	0.00%
LGBM with Others	99.80%	89.80%	68.80%	77.90%

Model	Accuracy	Precision	Recall	F1-Score
XGBoost with Others and Ours without LOF&KNN_distance	99.90%	97.60%	64.10%	77.40%
BayesPoint with Others and Ours without LOF&KNN_distance	33.30%	0.60%	96.90%	1.10%
RandomForest with Others and Ours without LOF&KNN_distance	96.00%	7.80%	85.90%	14.40%
SVM with Others and Ours without LOF&KNN_distance	99.60%	47.30%	67.20%	55.50%
NN with Others and Ours without LOF&KNN_distance	99.60%	0.00%	0.00%	0.00%
Logistic with Others and Ours without LOF&KNN_distance	94.30%	5.80%	89.10%	10.80%
LGBM with Others and Ours without LOF&KNN_distance	99.70%	68.50%	57.80%	62.70%
XGBoost with Others without LOF&KNN_distance	99.80%	81.80%	56.30%	66.70%
BayesPoint with Others without LOF&KNN_distance	17.70%	0.50%	96.90%	0.90%
RandomForest with Others without LOF&KNN_distance	93.10%	4.70%	85.90%	8.90%
SVM with Others without LOF&KNN_distance	97.50%	10.20%	70.30%	17.80%
NN with Others without LOF&KNN_distance	94.50%	5.80%	85.90%	10.90%
Logistic with Others without LOF&KNN_distance	92.00%	4.30%	92.20%	8.20%
LGBM with Others without LOF&KNN_distance	99.70%	70.60%	56.30%	62.60%

sions, Decision Support Systems 75 (2015) 38–48. URL: https://www.sciencedirect.com/science/article/pii/S0167923615000846. doi:https://doi.org/10.1016/j.dss.2015.04.013.

[14] D. Cheng, S. Xiang, C. Shang, Y. Zhang, F. Yang, L. Zhang, Spatio-temporal attention-based neural network for credit card fraud detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 362–369.

[15] Y. Lucas, P.-E. Portier, L. Laporte, L. He-Guelton, O. Caelen, M. Granitzer, S. Calabretto, Towards automated feature engineering for credit card fraud detection using multi-perspective hmms, Future Generation Computer Systems 102 (2020) 393–402. URL: https://www.sciencedirect.com/science/article/pii/S0167739X19300664. doi:https://doi.org/10.1016/j.future.2019.08.029.

[16] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto,

P.-E. Portier, L. He-Guelton, O. Caelen, Sequence classification for credit-card fraud detection, Expert Systems with Applications 100 (2018) 234 – 245.

[17] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, P. Beling, Deep learning detecting fraud in credit card transactions, in: 2018 Systems and Information Engineering Design Symposium (SIEDS), 2018, pp. 129–134.

[18] R. Moraffah, B. Moraffah, M. Karami, A. Raglin, H. Liu, Causal Adversarial Network for Learning Conditional and Interventional Distributions, arXiv e-prints (2020) arXiv:2008.11376. arXiv:2008.11376.

[19] E. Savona, M. Riccardi, Assessing the risk of money laundering: research challenges and implications for practitioners, European Journal on Criminal Policy and Research 25 (2019) 1–4. doi:10.1007/s10610-019-09409-3.

[20] D. Vassallo, V. Vella, J. Ellul, Application of gradient boosting algorithms for anti-money laundering in cryptocurrencies, SN Computer Science 2 (2021). doi:10.1007/s42979-021-00558-z.

[21] F. Butaru, Q. Chen, B. Clark, S. Das, A. W. Lo, A. Siddique, Risk and risk management in the credit card industry, Journal of Banking & Finance 72 (2016) 218–239. URL: https://www.sciencedirect.com/science/article/pii/S0378426616301340. doi:https://doi.org/10.1016/j.jbankfin.2016.07.015.

[22] B. Wiese, C. Omlin, Credit Card Transactions, Fraud Detection, and Machine Learning: Modelling Time with LSTM Recurrent Neural Networks, Springer Berlin Heidelberg, 2009, pp. 231–268.

[23] Y.-J. Zheng, X.-H. Zhou, W.-G. Sheng, Y. Xue, S.-Y. Chen, Generative adversarial network based telecom fraud detection at the receiving bank, Neural Networks 102 (2018) 78–86.

[24] S. Cao, X. Yang, C. Chen, J. Zhou, X. Li, Y. Qi, Titant: Online real-time transaction fraud detection in ant financial, Proceedings of the VLDB Endowment 12 (2019).

[25] D. Wang, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, J. Zhou, S. Yang, Y. Qi, A semi-supervised graph attentive network for financial fraud detection, in: 2019 IEEE International Conference on Data Mining (ICDM), 2019, pp. 598–607.

[26] P.-N. Tan, M. Steinbach, A. Karpatne, V. Kumar, Introduction to Data Mining, 2nd ed., Pearson, 2018.

[27] R. Moraffah, M. Karami, R. Guo, A. Raglin, H. Liu, Causal interpretability for machine learning-problems, methods and evaluation, in: ACM SIGKDD Explorations Newsletter, 2020, pp. 18–33.

[28] Z. Chen, D. V.-K. Le, E. Teoh, A. Nazir, E. Karuppiah, K. Lam, Machine learning techniques for anti-money laundering (aml) solutions in suspicious transaction detection: a review, Knowledge and Information Systems 57 (2018). doi:10.1007/s10115-017-1144-z.

[29] K. Chitra, B. Subashini, Data mining techniques and its applications in banking sector, volume 3(8), 2013, pp. 219–226.

[30] R. Liu, X.-l. Qian, S. Mao, S.-z. Zhu, Research on anti-money laundering based on core decision tree algorithm, in: 2011 Chinese Control and Decision Conference (CCDC), 2011, pp. 4322–4325. doi:10.1109/CCDC.2011.5968986.