IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 25, 2018, accepted March 3, 2018, date of publication March 26, 2018, date of current version April 23, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2816564

CoDetect: Financial Fraud Detection With Anomaly Feature Detection

DONGXU HUANG, DEJUN MU, LIBIN YANG, AND XIAOYAN CAI
School of Automation, Northwestern Polytechnical University, Xi'an, 710072, China

Corresponding author: Dongxu Huang (huangdongxu21@mail.nwpu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672433 and in part by the Basic Research of the Shenzhen Science and Technology Innovation Committee under Grant 201703063000511.

ABSTRACT Financial fraud, such as money laundering, is known to be a serious process of crime that makes illegitimately obtained funds go to terrorism or other criminal activity. This kind of illegal activities involve complex networks of trade and financial transactions, which makes it difficult to detect the fraud entities and discover the features of fraud. Fortunately, trading/transaction network and features of entities in the network can be constructed from the complex networks of the trade and financial transactions. The trading/transaction network reveals the interaction between entities, and thus anomaly detection on trading networks can reveal the entities involved in the fraud activity; while features of entities are the description of entities, and anomaly detection on features can reflect details of the fraud activities. Thus, network and features provide complementary information for fraud detection, which has potential to improve fraud detection performance. However, the majority of existing methods focus on networks or features information separately, which does not utilize both information. In this paper, we propose a novel fraud detection framework, CoDetect, which can leverage both network information and feature information for financial fraud detection. In addition, the CoDetect can simultaneously detecting financial fraud activities and the feature patterns associated with the fraud activities. Extensive experiments on both synthetic data and real-world data demonstrate the efficiency and the effectiveness of the proposed framework in combating financial fraud, especially for money laundering.

INDEX TERMS Anomaly feature detection, CoDetect, financial fraud.

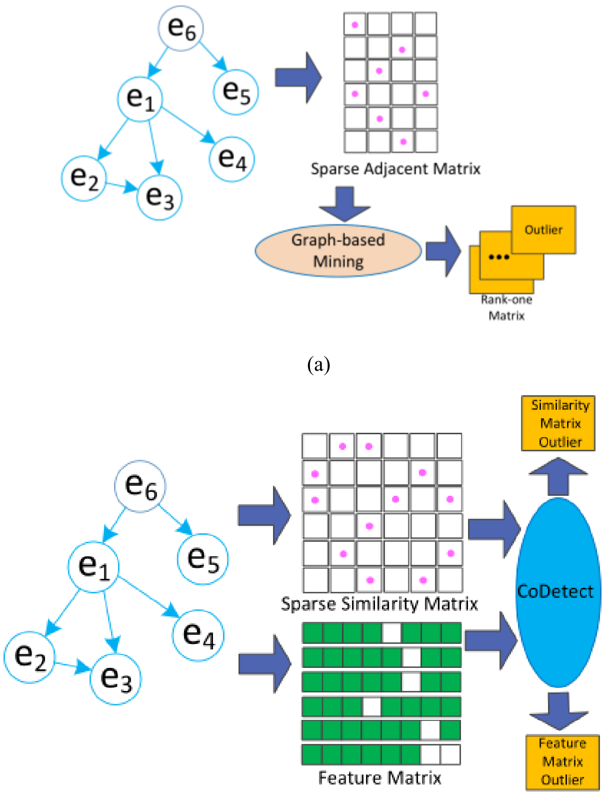
I. INTRODUCTION

In recent years, financial fraud activities such as credit card fraud, money laundering, increase gradually. These activities cause the loss of personal and/or enterprises' properties. Even worse, they endanger the security of nation because the profit from fraud may go to terrorism [1], [25]. Thus, accurately detecting financial fraud and tracing fraud are necessary and urgent. However, financial fraud detection is not an easy task due to the complex trading networks and transactions involved. Taking money laundering as an example, money laundering is defined as the process of using trades to move money/goods with the intent of obscuring the true origin of funds. Usually, the prices, quantity or quality of goods on an invoice of money laundering are fake purposely. The misrepresentation of prices, quantity or quality of goods on an invoice merely exposes slight difference from regular basis if we use these numbers as features to generate detection policy. Under certain circumstances, this kind of detector may work well with relatively stable trading entities. Unfortunately, the real world situation is more complicated, especially within Free Trade Zones (FTZs) where international trade involves complex procedures and exchange of information between trading entities. The fraud activities, especial money laundering, are deeper stealth. Money laundering activities may take different forms [1] such as the concealing transportation of cash using trading operations; the acquisition and sale of intangibles; and related party transactions. Not only the trading of goods shows on much more diversity, but also different type of companies, shell and front companies involve in to facilitate money laundering. In contrast with other fraud activities, money laundering demonstrates special characteristic which presents high risk to financial system with obscuring the money trail, collectivization behavior and wild trading regions in FTZs.

Many fraud detection models work with attribute-value data points that are generated from transactions data.

Some aggregation methods are also used to enrich the information of data [28]. After generating feature points from transactions, supervised and unsupervised methods can be used to perform detection [26], [27], [34]. Usually, these data points are assumed to be independent and identically distributed (i.i.d.). However, the characteristic of money laundering is different from attribute-value data. The collectivization behavior means the data is inherently linked or partly linked. Obviously, trading activity involves at least two business entities. Linked data is patently not independent and identically distributed, which contradicts the assumptions of traditional supervised and unsupervised methods. On the other side, some linked data is auto correlated. For example, trading between business entity A and B implies that feature points A and B are correlated. Some features used to describe the properties of trading goods can be identical between A and B. This characteristic of auto correlation reduce the effective size of data for learning. Furthermore, feature points don't fuse the interaction information in data. The relations between any business entities indicate the potential causality that means, if businesses on going, fraud entity can be located by other identified fraud entity. This means the entity, which have conection with fraud entity, are suspicious. Consequently, feature based detection models with supervised or unsupervised methods have inherent limitation of incapacity of identifying what the fraud relations are.

Graph-based mining methods are one of the most important theories that attempt to identify relations between data points [3], [7], [13], as Fig. 1(a) shows. Financial activities can be modeled as a directed graph, then a sparse adjacent matrix can represent this graph. With graph-mining method, the sparse matrix can be approximated as summation of low-rank matrix and outlier matrix. The outlier matrix is a sign of suspicious fraud activities. Exploiting the graph-based mining provides a new perspective for fraud detection and enables us to do advanced research on fraud detection. With the fraud activities detected by graph-based detection technique we are able to draw the conclusion that several business entities involved in fraud, however, we still don't know how these fraud activities are operated and why these activities labeled as fraud, i.e., the detailed features of the fraud activities. The majority of this how-and-why information is fused in features points, which have essential meaning for financial fraud detection because of the tracing necessity. For example, doing business with misrepresentation of the price may transfer additional value to exporter. The value in this example reveals how did the fraud happen. This simple example requires the detection system to mark value as fraud property. Another example, fraud activities might go deeper stealth with multi-entities involved. If the same good or service invoices a number of different business entities to make the payments, then there are several properties should be consider as suspicious: business location, name, direction, good or service etc. With the knowledge of these suspicious properties, tracing fraud can be much easier for executives.



(a)

(b)

FIGURE 1. Fraud detection using graph mining techniques. (a) Existing fraud detection framework. (b) The proposed framework.

Thus, graph-based methods can detection suspicious interactions between entities while attribute-feature based methods can reveal the features of the fraud. Graph and attributes provides two complementary information for financial fraud activity detection and fraud property tracing. However, the majority of the existing algorithms exploits these two information separately and thus can not provide a system that can detect the fraud entities and reveal suspicious properties for easy tracing simultaneously.

In this paper, we would like to develop a novel framework for fraud detection by considering the special detecting and tracing demanding of fraud entities and behaviors. Specifically, we investigate: (1) how to utilize both graph matrix and feature matrix for fraud detection and fraud tracing; (2) how to mathematically model both graph matrix and feature matrix so as to simultaneously achieve the tasks of fraud detection and tracing. In an attempt to solve these challenges, we proposed a novel detection framework CoDetect, as Fig. 1(b) shown, for financial data, especially for money laundering data. We incorporate fraud entities detection and anomaly feature detection in the same framework to find fraud patterns and corresponding features simultaneously. Combining entities detection and feature detection enables us to build a novel fraud detection framework for noisy and sparse financial data: relevant fraud patterns help the identification of fraud identities, and relevant features in turn help revealing of the nature of fraud activities.

Our empirical study on synthetic and real world data sets demonstrates the effectiveness of CoDetect, which does discover the fraud pattern and decide the fraud related properties in an unsupervised manner by seeking the low-rank approximation representations and residual for complex network matrix and feature matrix simultaneously. The major contributions of the paper can be summarized as follows:

1) Provide an approach to establish weighted graph from financial network, incorporating properties of nodes and links;
2) Demonstrate different scenarios of financial fraud and formulate the patterns of fraud in term of graph and sparse matrix;
3) Propose a novel unsupervised framework, CoDetect, for the problem of complex patterns discovery and anomaly features identification, employing two matrices residual analysis on graph-based financial network;
4) Evaluate framework using synthetic and real world data to demonstrate both effectiveness and efficiency of the proposed framework.

The rest of the paper is organized as follows: the characteristic of financial data and typical fraud scenario is demonstrated in Section 2; our detection framework for financial data, CoDetect, with an optimization methods and its detailed convergence analysis, is introduced in Section 3; and the algorithm is introduced in Section4, empirical evaluation on data set from synthetic and real-world data is presented in Section 5; the related work is shown in Section 6; and conclude in Section 7.

II. CHARACTERISTIC OF FINANCIAL DATA

Usually, financial transactions involve complex information exchanges between business entities and third party (supervision). Financial fraud activities (money laundering) range from simple technique, such as misrepresentation of the price, quantity or quality of goods on an invoice, to complex networks of financial transactions. For better interpretation of financial activities, we introduce an example case from APG2008 [1], [25] to extract elements which we are interested in for analyzing fraud:

Case Study: Directors of a company were involved in purchasing large quantities of duty free cigarettes and alcohol to sell on the domestic market contrary to their export duty free status, thus avoiding tax obligations. The company generated fake receipts with an export company detailing their alleged cigarette exports. Investigations confirmed that no such exports had ever been made. Payment was made for the cigarettes on a cash-on-delivery basis. A large number of the company's sales occurred over the internet from customers paying via credit card. A majority of the sales on the internet were illegitimate and came from three different email addresses. Payments for these orders were made from one of two credit cards linked to Belize bank accounts. One card was held in the company name. The money in the Belize bank account was sent there by one of the directors using several fake names from not only Australia but also Belize, Hong Kong and Vietnam. The director conducted structured wire transfers under fake names and front company accounts. The funds were purchased at well-known banks with multiple transactions occurring on the same day at different bank locations and all of the cash transfers conducted in amounts of just under AUD 10,000 to avoid the reporting threshold.

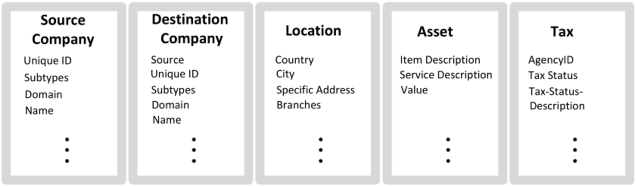


FIGURE 2. SDLAT data in financial network.

The words in bold type are some properties which can be used for representation of financial activities. Typical these words can be grouped into transactional data such as names, tax ids, addresses and value. In FTZs, service can be traded with less standard value or price which apparently more difficult to substantiate. The services trade presents a much more significant challenge to fraud detection. Consequently, the type and quality of information we summarized influences detection performance of fraud. As the case illustrate, detecting complex fraud schemes requires better integration and summarization of data from disparate financial entities often interconnected in Source company, Destination company, Location, Asset and Tax status (SDLAT) networks, show in Fig. 2.

SDLAT integrates much more properties from financial network, which enable executives to detect fraud through pattern detection over evolving SDLAT. As we know, the five key elements in SDLAT have large number of physical things. So the SDLAT data is in high dimensional and sparse which present extremely challenge for fraud detection.

A. FINANCIAL FRAUD SCENARIOS

There are mainly three scenarios in final fraud. In this subsection, we analyze these three scenarios, which can help us to develop algorithms for fraud detection.

Scenario 1 (Outlier Point): Over and under-invoicing of goods and services. The primary activities of this kind of fraud are misrepresentations of price of the good or service for the purpose of illegally transferring additional value between the importer and exporter. Fig. 3 gives an example of this scenario. In Fig. 3, nodes represent entities and links between nodes means trade between them. The thickness of the link can represent the price of the good or service. As we can see, the price of good traded among these four entities are relatively small except that the line between Entity C and Entity D is very thick, which is suspicious and it is likely that there's money laundering between Entity C and Entity D.

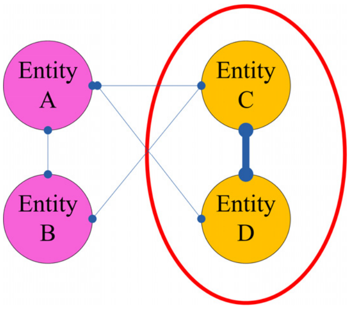


FIGURE 3. Outlier point.

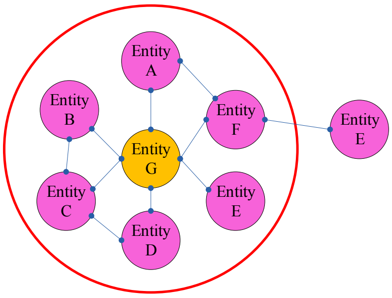


FIGURE 4. Merge.

Scenario 2 (Merge): Multiple invoicing of goods and services. This kind of fraud makes no misrepresentation of price of the good or services on commercial invoice. It involves more complicated web of transactions whereby the same good or service is invoiced more than once, often using a number of different financial institutions to make the payments, as Fig. 4 shown. This scenario explains a financial fraud called Ring fraud. Entities A to F all had business trades with Entity G. Because there is no misrepresentation of price of good or services in this kind of fraud, the lines between each pair of entities have no obvious difference in thickness. Even the fraud group in ring can be detected, further information about each entities properties (features) is need for tracking and forensic. So we need to detect the suspicious features. In our framework, we employ additional residual term on feature matrix for this purpose.

Scenario 3 (Ring): Related party transactions. Transaction based money laundry requires collusion between commercial entities at both ends of import/export chain, but they don't need to be linked directly. The good can be traded from one entity to another, and then from another to third party, as Fig. 5 shown. From Entity A to E, each entity has connection with its neighbor, and Entity A and Entity E also has connection which forms a Ring. This is a steal fraud activity which involve tight cooperation in this group. The thickness of line can not work as a sign of detection of fraud. Finally, the trading value may go to the desired entity without triggering the alarm. There is more information needed for tracking and forensic each entity in Ring group. The executive need to know where and how the money go. Under this condition, detecting the suspicious properties of each entity is necessary. In our framework, residual term on feature matrix can perform this task in good manner.

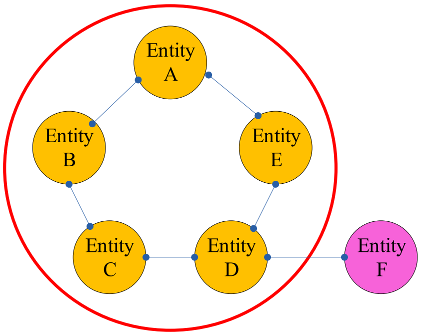


FIGURE 5. Ring.

III. A FRAMEWORK OF CoDetect

Before introducing the details of the proposed framework, we first introduce the notations used in this paper. Throughout the paper, matrices are written as boldface capital letters such as $\mathbf{A}$ and vectors are denoted as bold face lowercase letters such as $\mathbf{a}$. $\mathbf{I}$ is the identity matrix. For an arbitrary matrix $\mathbf{A}$, $\mathbf{A}_{ij}$ is the i-th row and j-th column of $\mathbf{A}$. $||\mathbf{A}||\ \mathbf{1}$ is the l_1 norm of $\mathbf{A}$, which is defined as the summation of absolute elements of $\|\mathbf{A}\|_1 = \sum i \sum j |\mathbf{A}ij|$ Next, we will first introduce how we construct graph and feature matrix from SDLAT. We will then introduce how to perform financial detection on the constructed graph matrix and feature matrix.

A. GRAPH MATRIX AND FEATURE MATRIX FROM SDLAT

In this subsection, we introduce how we construct graph matrix and feature matrix from SDLAT data. Since SDLAT contains the source company and destination company, we can construct a network $G = \{v, \varepsilon\}$ where $\mathbf{V} = \{v_1, \ldots, v_N\}$ is a set of N nodes with each node being a company and $\varepsilon \subset v \times v$ is a set of edges. If v_i is the source company and v_j is the destination company, we add and edge $e_{ij} = 1$ between v_i and v_j. In addition to the network, each node is also associated with a set of attributes such as location, asset, tax status. We use $\mathbf{F} \in R^{N \times d}$ to represent the feature matrix, where d is the dimension of the features. The network G contains the interactions among companies. From the network structure of G, we might able to detect scenario 1 and 3 financial fraud. $e_{ij} = 1$ only means that there's an interaction from v_i to v_j. To reflect the similarity between the source and destination company, which doesn't reflect the price of the goods or other properties and thus cannot be used to detect scenario 2. To incorporate information for detecting scenario 2, we also use $\mathbf{S}_{ij}$ to represent the weight between v_i and v_j. The weight $\mathbf{S}_{ij}$ is defined as:

$$S_{ij} = e^{\frac{||f_i - f_j||^2}{\sigma^2}} \tag{1}$$

where f_i means the i-th row of $\mathbf{F}$ and σ is a scalar to control the scale of the weight. Thus, $\mathbf{S}$ is the weighted graph information and **F** is the feature matrix. The problem is formally defined as:

Given graph matrix **S** and feature matrix **F**, find a function f which can simultaneously detect fraud activities and trace the properties of the fraud.

B. ANOMALY DETECTION ON GRAPH MATRIX

In real-world, the trade are usually among companies of similar type, i.e., companies that deal with similar business are more likely to have interaction. For example, for an IT company v_i it is more likely to see v_j to have trade/business with IT companies than fruit companies. This fact makes the graph matrix to have block structures. Companies within the same block are of similar business type and there are more interactions of companies within each block than that of between blocks. In other words, the graph matrix is low-rank [4], [44]. Thus, we can represent as **S**:

$$\mathbf{S} = \mathbf{U}\mathbf{V}_s^{\mathbf{T}}+\mathbf{R}_s \tag{2}$$

where $\mathbf{U} \in R^{N\times r}$ and $\mathbf{V}_s \in R^{N\times r}$ are two low-rank latent feature matrix with $r << \mathrm{N}$. The interaction between v_i and v_j is recovered by the interaction between the latent features of u_i and v_j as $\mathbf{U}(i,:)\,\mathbf{V}_s(j,:)^T$, $\mathbf{U}\mathbf{V}_s^T$will give a low rank matrix, which mainly recovers the within blocks interactions. $\mathbf{R}_s$ is the residual matrix, which mainly includes the interaction between the blocks. As we know, the fraud transaction is rare, each two businesses in trading is interdependent. In graph mining, low-rank matrix is used to represent the transactions data [4], [45]. Since the interaction between blocks, i.e., the trade between companies of different types, are very rare and very suspicious, $\mathbf{R}_s$ can be used to capture the suspicious interaction and can thus be used to detect fraud [7]. Given the fact that the majority of interactions are normal and are not financial fraud, we would expect the captured financial fraud to be very sparse. Based on this, we add the l_1 norm on $\mathbf{R}_s$, so as to make $\mathbf{R}_s$ sparse and can capture real financial fraud. Then the objective function becomes:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V_s},\mathbf{R}_s} \quad \|\mathbf{R}_s\|_1 \\ &s.t.\ \mathbf{S} = \mathbf{U}\mathbf{V}_S^T + \mathbf{R}_s \end{aligned} \tag{3}$$

Since **U** is the latent features of companies and companies form groups, i.e., some companies do similar business, we would expect the latent features of companies within the same group have similar latent features. Based on this, we add the orthogonal constraint on **U**, which is widely used for di fferentiating groups of features [8]. After adding the orthogonal constraint, Equation (3) becomes:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V_s},\mathbf{R}_s} \quad \|\mathbf{R}_s\|_1 \\ &s.t.\ \mathbf{S} = \mathbf{U}\mathbf{V}_S^T + \mathbf{R}_s \\ &\mathbf{U}^T\mathbf{U} = \mathbf{I} \end{aligned} \tag{4}$$

Where norm 1 is used to ensure the detected fraud is rare. **U** is pseudo class label.

C. ANOMALY DETECTION ON FEATURE MATRIX

With residual matrix $\mathbf{R}_s$ we can easily clarify how many business entities involve in fraud and what is the pattern of the fraud, e.g. merge or ring. There are still vast of information we don't know about the fraud, such as location, value, tax etc. which can be represented by SDLAT feature. Those fraud information is valuable to financial executive for fraud tracing. Therefor, anomaly detection on matrix **F** is necessary. As for normal financial business, we would expect similar feature patterns to have within companies of the same type, such as the price, the location. Therefore, the feature matrix **F** is naturally low-rank as companies of the same type has similar feature patterns [25]. Based on this observation, we first decompose the feature matrix as **F** as:

$$\mathbf{F} = \mathbf{U}\mathbf{V}_f^T+\mathbf{R}_f \tag{5}$$

where $\mathbf{U} \in R^{N\times r}$ is the latent representations of the companies and $\mathbf{V}_f$ are the latent representations of the SDLAT features. $\mathbf{R}_f$ is the residual matrix. For features that cannot be well reconstructed, the corresponding residual will be large, which reflects the anomaly features. Thus, with the residual matrix $\mathbf{R}_f$, we can trace the fraud patterns. Since the majority companies don't involve in financial fraud, we can expect that the residual matrix $\mathbf{R}_f$ is sparse. Thus, we also add l_1 norm on $\mathbf{R}_f$ to make it sparse, which gives us the objective function as:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}_f,\mathbf{R}_f} \quad \left\|\mathbf{R}_f\right\|_1 \\ &s.t.\ \mathbf{F} = \mathbf{U}\mathbf{V}_f^T + \mathbf{R}_f \end{aligned} \tag{6}$$

Similarly, we add the orthogonal constraint on U to make it discriminative as:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}_f,\mathbf{R}_f} \quad \left\|\mathbf{R}_f\right\|_1 \\ &s.t.\ \mathbf{F} = \mathbf{U}\mathbf{V}_f^T + \mathbf{R}_f \\ &\mathbf{U}^T\mathbf{U} = 1 \end{aligned} \tag{7}$$

D. THE CoDetect FRAMEWORK

Equation (4) models the graph matrix **S** to detect fraud activities while (7) handles **F** to trace the fraud patterns. To fully leverage these two matrices for simultaneously detecting financial fraud and tracing fraud patterns, we can combine (4) and (7) together, which results in the objective function of CoDetect:

$$\begin{aligned} &\arg\min_{\mathbf{U}} \quad \|\mathbf{R}_s\|_1 + \alpha \left\|\mathbf{R}_f\right\|_1 \\ &s.t.\ \mathbf{S} = \mathbf{U} * \mathbf{V}_S^T + \mathbf{R}_S \\ &\mathbf{F} = \mathbf{U} * \mathbf{V}_f^T + \mathbf{R}_f \end{aligned} \tag{8}$$

Where α is a scalar to leverage the contribution of the graph matrix **S** and feature matrix **F**. The latent company feature matrix **U** is learned from both **S** and **F** as by the constraint $\mathbf{S} = \mathbf{U} * \mathbf{V}_s^T + \mathbf{R}_S$ and $\mathbf{F} = \mathbf{U} * \mathbf{V}_f^T + \mathbf{R}_f$. Thus information of **S** and **F** can flow through **U** and thus proposed CoDetect is a unified framework that leverages both **U** and **V** simultaneously.

IV. OPTIMIZATION ALGORITHM

The optimization problem in (8) in not convex in $\mathbf{U}$, $\mathbf{V}_s$ and $\mathbf{V}_f$. If we fixed two variables and update only one variable once a time, the problem is convex and can be optimized by Alternating Direction Method of Multiplier (ADMM) [Boyd *et al.* 2011]. The (8) can be converted into following equivalent ADMM problem:

$$\begin{aligned}\underset{\mathbf{U},\mathbf{V}_s,\mathbf{V}_f,\mathbf{Y}_1,\mathbf{Y}_2}{min}\quad & \|\mathbf{R}_s\|_1 + \alpha\left\|\mathbf{R}_f\right\|_1\\ & + tr\left[\mathbf{Y}_1^T\left(\mathbf{S}-\mathbf{U}\mathbf{V}_s^T-\mathbf{R}_s\right)\right]\\ & + tr\left[\mathbf{Y}_2^T\left(\mathbf{F}-\mathbf{U}\mathbf{V}_f^T-\mathbf{R}_f\right)\right]\\ & + \frac{\mu}{2}\left(\left\|\left(\mathbf{S}-\mathbf{U}\mathbf{V}_s^T-\mathbf{R}_s\right)\right\|_{\mathbf{F}}^2\right.\\ & \quad\left.+\left\|\left(\mathbf{F}-\mathbf{U}\mathbf{V}_f^T-\mathbf{R}_f\right)\right\|_{\mathbf{F}}^2\right)\\ s.t.\ \mathbf{U}^T\mathbf{U}=\mathbf{I} & \end{aligned} \tag{9}$$

Where $\mathbf{Y}_1$, $\mathbf{Y}_2$ are two lagrangian multipliers and μ is a scalar to control the penalty for the violation of equality constraints $\mathbf{S}-\mathbf{U}\mathbf{V}_s^T-\mathbf{R}_s$ and $\mathbf{F}-\mathbf{U}\mathbf{V}_f^T-\mathbf{R}_f$.

A. UPDATE U

To update $\mathbf{U}$, we fix the other variables except $\mathbf{U}$ and remove terms that are irrelevant to $\mathbf{U}$. Then (9) becomes:

$$\begin{aligned}\min_{\mathbf{U}^T\mathbf{U}=I}\ & tr\left[\mathbf{Y}_1^T\left(\mathbf{S}-\mathbf{U}\mathbf{V}_s^T-\mathbf{R}_s\right)\right]\\ & + tr\left[\mathbf{Y}_2^T\left(\mathbf{F}-\mathbf{U}\mathbf{V}_f^T-\mathbf{R}_f\right)\right]\\ & + \frac{\mu}{2}\left(\left\|\left(\mathbf{S}-\mathbf{U}\mathbf{V}_s^T-\mathbf{R}_s\right)\right\|_F^2+\left\|\left(\mathbf{F}-\mathbf{U}\mathbf{V}_f^T-\mathbf{R}_f\right)\right\|_F^2\right)\end{aligned} \tag{10}$$

The (10) can be simplified as:

$$\begin{aligned}\min_{U^TU=I}\ & -tr\left(\mathbf{U}^T\mathbf{Y}_1\mathbf{V}_s\right)+tr\left(\mathbf{U}^T\mathbf{Y}_2\mathbf{V}_f\right)\\ & + \frac{\mu}{2}\left(\left\|\mathbf{U}\mathbf{V}_S^T-\mathbf{A}\right\|_F^2+\left\|\left(\mathbf{U}\mathbf{V}_f^T-\mathbf{B}\right)\right\|_F^2\right)\end{aligned} \tag{11}$$

Where $\mathbf{A} = \mathbf{S} - \mathbf{R}_s$ and $\mathbf{B} = \mathbf{F} - \mathbf{R}_f$. It can be further simplified as:

$$\min_{\mathbf{U}^T\mathbf{U}=I}\ -tr\left[\mathbf{U}^T\mathbf{N}\right] \tag{12}$$

Where $\mathbf{N} = \mathbf{Y}_1\mathbf{V}_s + \mathbf{Y}_2\mathbf{V}_f + \mu\mathbf{A}\mathbf{V}_s + \mu\mathbf{B}\mathbf{V}_f$. This problem is equivalent to:

$$\min_{\mathbf{U}^T\mathbf{U}=I}\ tr\left\|\mathbf{U}-\mathbf{N}\right\|_F^2 \tag{13}$$

which is the classical Orthogonal Procrustes problem. The following Lemma [Huang *et al.* 2014] gives the solution to this problem.

Lemma 1. Given the objective in (13), the optimal $\mathbf{U}$ is defined as:

$$\mathbf{U} = \mathbf{P}\mathbf{Q}^T \tag{14}$$

Where $\mathbf{P}$ and $\mathbf{Q}$ are the left and right singular vectors of the economic singular value decomposition of $\mathbf{N}$.

B. UPDATE $\mathbf{V}_S$

$$\min_{\mathbf{V}_S} tr\left[\mathbf{Y}_1^T\left(\mathbf{S}-\mathbf{U}\mathbf{V}_s^T-\mathbf{R}_s\right)\right]+\frac{\mu}{2}\left\|\mathbf{S}-\mathbf{U}\mathbf{V}_s^T-\mathbf{R}_s\right\|_F^2 \tag{15}$$

Which is equivalent to:

$$\min_{\mathbf{V}_s}\frac{\mu}{2}\left\|\mathbf{V}_s-\left(\mathbf{S}-\mathbf{R}_s\right)^T\mathbf{U}-\frac{1}{\mu}\mathbf{Y}_1^T\mathbf{U}\right\|_F^2 \tag{16}$$

Obviously, (16) has a closed form solution as:

$$\mathbf{V}_s = \left(\mathbf{S}-\mathbf{R}_s\right)^T\mathbf{U}+\frac{1}{\mu}\mathbf{Y}_1^T\mathbf{U} \tag{17}$$

Taking the advantage of $\mathbf{U}\mathbf{U}^T = \mathbf{I}$, the solution can be reformulated as:

$$\mathbf{Q}_S = \mathbf{V}_S\mathbf{U}^T = \left(\mathbf{S}-\mathbf{R}_S\right)^T+\frac{1}{\mu}\mathbf{Y}_1^T \tag{18}$$

C. UPDATE $\mathbf{V}_f$

$$\min_{\mathbf{V}_f} tr\left[\mathbf{Y}_2^T\left(\mathbf{F}-\mathbf{U}\mathbf{V}_f^T-\mathbf{R}_f\right)\right]+\frac{\mu}{2}\left\|\mathbf{F}-\mathbf{U}\mathbf{V}_f^T-\mathbf{R}_f\right\|_F^2 \tag{19}$$

Similarly, it is equivalent to:

$$\min_{\mathbf{V}_f}\frac{\mu}{2}\left\|\mathbf{V}_f-\left(\mathbf{F}-\mathbf{R}_f\right)^T\mathbf{U}-\frac{1}{\mu}\mathbf{Y}_2^T\mathbf{U}\right\|_F^2 \tag{20}$$

Then comes to a closed form solution:

$$\mathbf{V}_f = \left(\mathbf{F}-\mathbf{R}_f\right)^T\mathbf{U}+\frac{1}{\mu}\mathbf{Y}_2^T\mathbf{U} \tag{21}$$

Similar to update of $\mathbf{V}_s$, the solution can be formulated as:

$$\mathbf{Q}_f = \mathbf{V}_f\mathbf{U}^T = \left(\mathbf{F}-\mathbf{R}_f\right)^T+\frac{1}{\mu}\mathbf{Y}_2^T \tag{22}$$

D. UPDATE $\mathbf{R}_S$

To update $\mathbf{R}_s$, we fix the other variables except $\mathbf{R}_s$ and remove the terms that are irrelevant to $\mathbf{R}_s$. Then from (9) we have following equation as:

$$\begin{aligned}\min_{\mathbf{R}_S}\ & \|\mathbf{R}_S\|_1+tr\left[\mathbf{Y}_1^T\left(\mathbf{S}-\mathbf{U}\mathbf{V}_S^T-\mathbf{R}_S\right)\right]\\ & +\frac{\mu}{2}\left\|\mathbf{S}-\mathbf{U}\mathbf{V}_S^T-\mathbf{R}_S\right\|_F^2\end{aligned} \tag{23}$$

The above subproblem can be rewritten as:

$$\min_{\mathbf{R}_S}\frac{\mu}{2}\left\|\mathbf{R}_S-\mathbf{E}\right\|_F^2+\left\|\mathbf{R}_S\right\|_1 \tag{24}$$

Where

$$\mathbf{E} = \mathbf{S}-\mathbf{U}\mathbf{V}_S^T+\frac{1}{\mu}\mathbf{Y}_1$$

The above equation has a closed form solution by following Lemma (Lin *et al.* 2010)

Lemma 2. Let $\mathbf{W}$ be a given matrix and λ is a positive scalar. If the optimal solution of

$$\min_{\mathbf{X}} \frac{1}{2} \|\mathbf{X} - \mathbf{W}\|_F^2 + \lambda \|\mathbf{X}\|_1 \tag{25}$$

is $\mathbf{X}^*$, then $\mathbf{X}_{ij}^*$ is given as:

$$\mathbf{X}_{ij}^* = S_\lambda \left(\mathbf{W}_{ij}\right) \tag{26}$$

where $\mathrm{S}_\lambda(\cdot)$ is called soft thresholding operator, which is defined as:

$$S_\lambda (x) = \begin{cases} x - \lambda & if\ x > \lambda \\ x + \lambda & if\ x < -\lambda \\ 0 & otherwise \end{cases} \tag{27}$$

Apparently, (24) has the solution given by the soft thresholding operator as:

$$\mathbf{R}_{sij} = S_{\frac{1}{\mu}} \left(\mathbf{E}_{ij}\right) \tag{28}$$

E. UPDATE $\mathbf{R}_f$

Similarly, to update $\mathbf{R}_f$, we remove terms that are not relevant with $\mathbf{R}_f$ and arrives at:

$$\min_{R_f} \alpha \left\|\mathbf{R}_f\right\|_1 + tr\left[\mathbf{Y}_2^T \left(\mathbf{F} - \mathbf{U}\mathbf{V}_f^T - \mathbf{R}_f\right)\right] + \frac{\mu}{2} \left\|\mathbf{F} - \mathbf{U}\mathbf{V}_f^T - \mathbf{R}_f\right\|_F^2 \tag{29}$$

The above function can be written as:

$$\min_{\mathbf{R}_f} \frac{\mu}{2} \left\|\mathbf{R}_f - \mathbf{M}\right\|_F^2 + \alpha \left\|\mathbf{R}_f\right\|_1 \tag{30}$$

Where $\mathbf{M} = \mathbf{F} - \mathbf{U}\mathbf{V}_f^T + \frac{1}{\mu}\mathbf{Y}_2$. Similar to the update of $\mathbf{R}_s$, the $\mathbf{R}_f$ can be updated as:

$$\mathbf{R}_{fij} = S_{\frac{\alpha}{\mu}} \left(\mathbf{M}_{ij}\right) \tag{31}$$

F. UPDATE $\mathbf{Y}_1$, $\mathbf{Y}_2$, μ

Once updating the variable, we now can update the ADMM parameters. Following the similar step in [Boyd *et al.* 2011], the parameters can be updated as:

$$\mathbf{Y}_1 = \mathbf{Y}_1 + \mu \left(\mathbf{S} - \mathbf{U}\mathbf{V}_S^T - \mathbf{R}_S\right) \tag{32}$$

$$\mathbf{Y}_2 = \mathbf{Y}_2 + \mu \left(\mathbf{F} - \mathbf{U}\mathbf{V}_f^T - \mathbf{R}_f\right) \tag{33}$$

$$\mu = \min\left(\rho\mu, \mu_{\max}\right) \tag{34}$$

Where $\rho > 1$ is a parameter to control the convergence speed and $u_{\max}$ is a larger number to prevent μ becomes too large.

G. ALGORITHM

Now, the framework of CoDetect can be summarized in Algorithm 1. We use K-means to initialize the variables $\mathbf{U}$, $\mathbf{V}_s$ and $\mathbf{V}_f$.

To accelerate the convergence speed, we follow the common way to initialize the $\mathbf{U}$, $\mathbf{V}_s$ and $\mathbf{V}_f$ with the k-means methods. To be specific, we apply k-means to cluster rows of $\mathbf{S}$, and set $\mathbf{V}_S = \mathbf{U}\mathbf{V}_S^T$ and $\mathbf{V}_f = \mathbf{U}\mathbf{V}_f^T$. μ is typically set in the range of to 10^{-6} to 10^{-3} initially depending on the data sets and is updated in each iteration. $\mu_{\max}$ is set to be a large value, 10^{10}. Parameter ρ is empirically set to 1.1 to give relatively stable converge speed.

Algorithm 1 CoDetect

```
Input: Similarity matrix $\mathbf{S} \in \mathbf{R}^{N\times N}$, feature matrix $\mathbf{F} \in \mathbf{R}^{N\times M}$; $\alpha$, low rank size $r$
Output: Similarity matrix residual $\mathbf{R}_s$, feature matrix residual $\mathbf{R}_f$
1.Initialize $\mu = 10^{-3}$; $\rho = 1.1$; $\mu_{\max} = 10^{10}$, $\mathbf{U}$, $\mathbf{V}_S$, $\mathbf{V}_f$, are initialized using K-means
2. repeat
    3. Calculate $\mathbf{Q}_S = \mathbf{S} - \mathbf{R}_S^T + \frac{1}{\mu}\mathbf{Y}_1^T$
    4. Update $\mathbf{V}_s$ using (17)
    5. Calculate $\mathbf{Q}_f = \mathbf{F} - \mathbf{R}_f^T + \frac{1}{\mu}\mathbf{Y}_2^T$
    6. Update $\mathbf{V}_f$ using (21)
    7. Calculate $\mathbf{E} = \mathbf{S} - \mathbf{U}\mathbf{V}_f^T + \frac{1}{\mu}\mathbf{Y}_1$
    8. Update $\mathbf{R}_s$ using (28)
    9. Calculate $\mathbf{M} = \mathbf{F} - \mathbf{U}\mathbf{V}_f^T + \frac{1}{\mu}\mathbf{Y}_2$
    10. Update $\mathbf{R}_f$ using (31)
    11. Calculate $\mathbf{A} = \mathbf{S} - \mathbf{R}_S$, $\mathbf{B} = \mathbf{F} - \mathbf{R}_f$
    12. Calculate $\mathbf{N} = \mathbf{Y}_1\mathbf{V}_S + \mathbf{Y}_2\mathbf{V}_f + \mu\mathbf{A}\mathbf{V}_S + \mu\mathbf{B}\mathbf{V}_f$
    13. Update $\mathbf{U}$ by Lemma 1
    14. Update $\mathbf{Y}_1$, $\mathbf{Y}_2$, $\mu$
15. until convergence
16. Output $\mathbf{R}_s$, $\mathbf{R}_f$ as anomaly in $\mathbf{S}$, $\mathbf{F}$
```

The convergence of the ADMM guarantee the convergence of our algorithm. As usual, we set $|J_{t+1} - J_t| / J_t$ as convergence criteria, where J_t is the object function value in (9). In our experiments, we also set another parameter maxIter to control the number of iterations for reducing the computational cost in special case. Experiments on two graph datasets find that our algorithm converges within 40 iterations.

H. COMPUTATION COMPLEXITY ANALYSIS

The computation cost for $\mathbf{V}_s$ is the computation of

$$\mathbf{V}_s = \left(\mathbf{S} - \mathbf{R}_s\right)^T \mathbf{U} + \frac{1}{\mu}\mathbf{Y}_1^T\mathbf{U}$$

Which is $O(N^2r)$ for each iteration. Similarity, the computation cost for $\mathbf{V}_f$ is the computation of

$$\mathbf{V}_f = \left(\mathbf{F} - \mathbf{R}_f\right)^T + \frac{1}{\mu}\mathbf{Y}_2^T\mathbf{U}$$

Which is $O(Ndr)$ for each iteration.

The computation cost for $\mathbf{R}_s$ is the computation of

$$\mathbf{E} = \mathbf{S} - \mathbf{U}\mathbf{V}_S^T + \frac{1}{\mu}\mathbf{Y}_1$$

and update of $\mathbf{R}_s$ using (28), which are $O(N^2r)$ and$O(N^2)$, respectively. Similarity, the computation cost for $\mathbf{R}_f$ involves the computation of $\mathbf{M}$ and update of $\mathbf{R}_f$, which are $O(Ndr)$ and $O(Nd)$, respectively.

The main computation cost of **U** involves the computation of N and its SVD decomposition, which is $O(Ndr)$ and $O(Nr^2)$. The computational cost for $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are both $O(Nd)$. Therefor, the overall time complexity is $O(Ndr + Nr^2 + N^2r)$. Since $d >> k$, the final computation cost is $O(Ndr + N^2r)$ for each iteration.

V. EXPERIMENTS

In this section, the synthetic data and real world data from *IKnow.com* are used to evaluate the effectiveness of CoDetect. We first perform qualitative analysis using synthetic data to demonstrate the detection result in an illustrative way. Then we evaluate CoDetect with other state-of-art matrix factorization methods and clustering methods in term of detection accuracy and detection time. Finally, we perform the model parameters analysis which prove the robustness of CoDetect.

A. FINANCIAL DATA SETS AND PREPROCESSING

1) SYNTHETIC DATA

Technically, the synthetic data is from small part of ICIJ Offshore Leaks Database. We only extract 100 financial entities and 2,000 transactions from this data set. Then we inject fraud patterns into this synthetic data. Under this setting, we have a sparse graph matrix, **S** with size of 100 × 100 and 2,000 points in matrix. And we also have a feature matrix, **F** with size of 100 × 30. Then we can perform qualitative analysis which provide a illustrative perspective for detection results.

2) MONEY LAUNDERING DATA

This data set is from ICIJ Offshore Leaks Database. We filter out uncompleted rows from the data set which leaves us a data set with 29,265 financial entities, and 571,113 transactions. We extract features from the transactions which is **F**, and build weighted graph **S** as described in previous section as: if two financial entities have trading history, there is an edge between them and the weight of the edge is calculated from the features of the two entities. Unfortunately, the fraud activates are not reported in this data sets. Any detected anomaly may not be considered as financial fraud. So we can't make these anomaly as ground-truth for evaluation. In our experiments, we randomly inject one of the fraud patterns into graph. we want to see if CoDetect can detect it from the residual matrix $\mathbf{R}_s$, at the same time, to see if CoDetect can reveal the anomaly feature from the residual matrix $\mathbf{R}_f$.

3) INSURANCE FRAUD DATA

This data set is from insurance company benchmark (COIL2000) data set [45] which has 86 attributes for each customer records. Reviewing from attribute 65 to 85, we know that each customer can under subset of insurance policies. Then we form a bi-party graph for the representation that whether the customer is under certain insurance policies or not. This bi-party graph is **S**. And the rows of original data set is **F**. The last attribute can be used as target label for evaluation. In real life, the fraud data is accounting of small portion of data set. To fit this criterion, we filter out records with target label 1. The data set with target label 0 is consider to be normal. For each experiment we inject 10% records with target label 1. Then we construct S and F. We repeat the experiment 10 times for fully coverage of records with target label 1. And mean value of the performance is calculated.

4) CREDIT CARD FRAUD DATA

Statlog (German Credit Data) data set is used in our study. The preprocessing is similar to the preprocessing of COIL2000. In Statlog, attribute 4, qualitative is used to form the bi-party graph from data set where there is a connection if customer ran their credit card for the purpose in attribute 4. Then we have the matrix **S** and matrix **F**. We filter out record with label 2 and the remaining data set is considered to be normal. For each experiment we inject 10% records with label 2 as outliers. Then we construct S and F. We repeat the experiment 10 times for fully coverage of records with target label 2. And mean value of the performance is calculated.

B. QUALITATIVE ANALYSIS ON GRAPH MATRIX S

The outputs of algorithm 1 are graph similarity matrix residual $\mathbf{R}_s$ and graph feature matrix residual $\mathbf{R}_f$. With matlab sparse matrix toolbox, we can plot these two matrices as identifications of fraud patterns in graph. We can have a more illustrative way to spot the fraud activities in graph. CoDetect can perform fraud detection on graph matrix and identify the anomaly feature corresponding to this fraud simultaneously. From (1), we know that parameter α control the anomaly contribution from feature matrix. If we set $\alpha = 0$, CoDetect degenerate into a general matrix factorization method. This method is chosen as a baseline. Then we can compare CoDetect with other state-of-art detection methods based on matrix factorization, robust PCA(RPCA) [39] and Singular Value Decomposition(SVD) [40]. We follow a direct way to construct detectors by using original matrix minus low rank matrix as approximated by RPCA and SVD respectively. As a common practice, the parameters in detection models are tuned via cross-validation. We inject one type of fraud pattern into graph each time to build **S** and **F**.

In this set of experiments, we evaluate CoDetect in two scenarios: (1) with $\alpha = 0$, we observe how different detection model fare, and (2) with $\alpha \neq 0$, i.e., performing fraud detection with state-of-art models first, we examine how these detection models compare with CoDetect. For CoDetect, we set $\alpha = 0.1$ for synthetic data. Figure 6 depicts the experiments results on matrix **S**.

When $\alpha = 0$, the performance of CoDetect degrades. As the second column in Fig. 6 shown, CoDetect generates false positive detection. In short, when $\alpha \neq 0$, outlier detection on **F** helps. CoDetect consistently outperforms RPCA and SVD.

The main reason is: from algorithm 1, **U** transfers fraud knowledge between **S** and **F** which suppress the false positives. When $\alpha = 0$, there is no knowledge transferring.

Taking merge fraud patterns as an example, from Fig. 6(b), merge fraud pattern is a red line in similarity matrix.

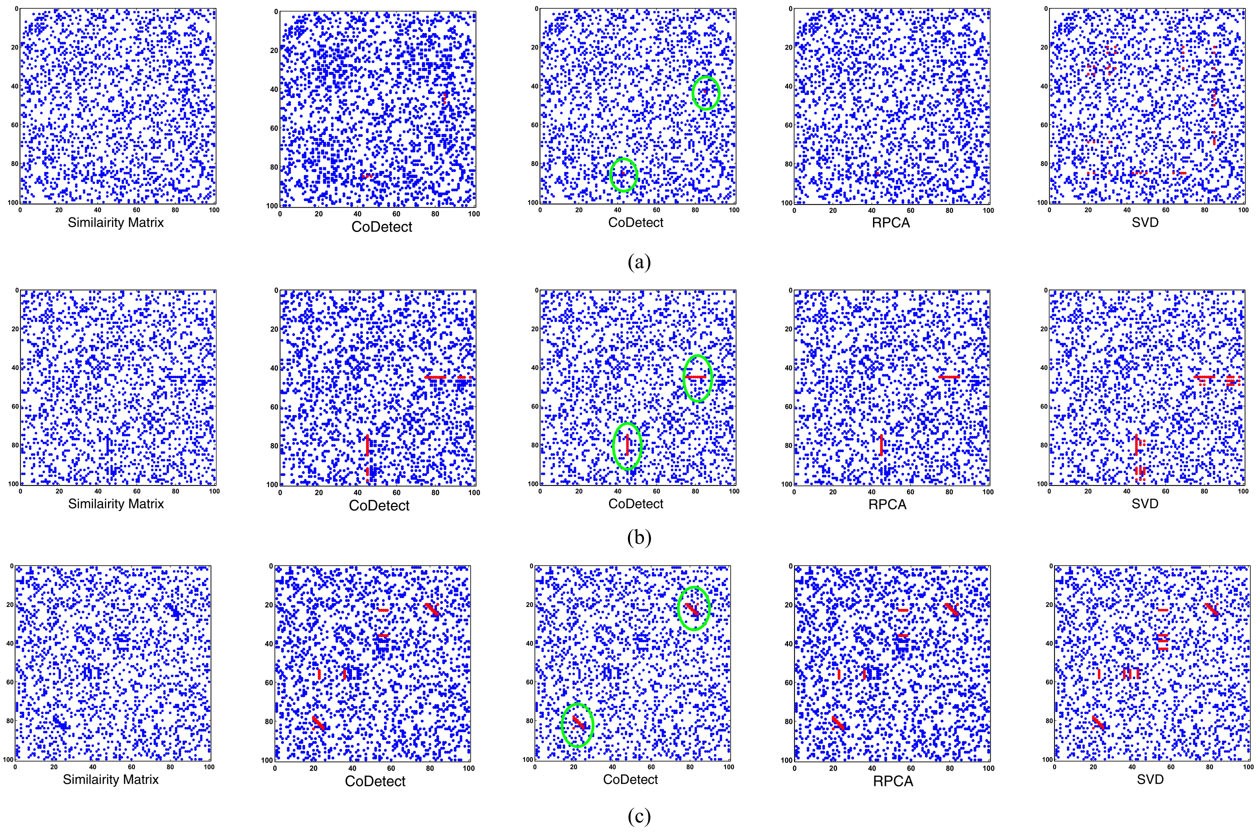


FIGURE 6. Fraud detection on synthetic graphs S with $\alpha = 0$ and$\alpha \neq 0$. Each blue dot in (a)–(c) represents an edge in graph. Red dots marked by green circles are detected anomaly patterns. (a) Outlier on similarity matrix. (b) Merge on similarity matrix. (c) Rings on similarity matrix.

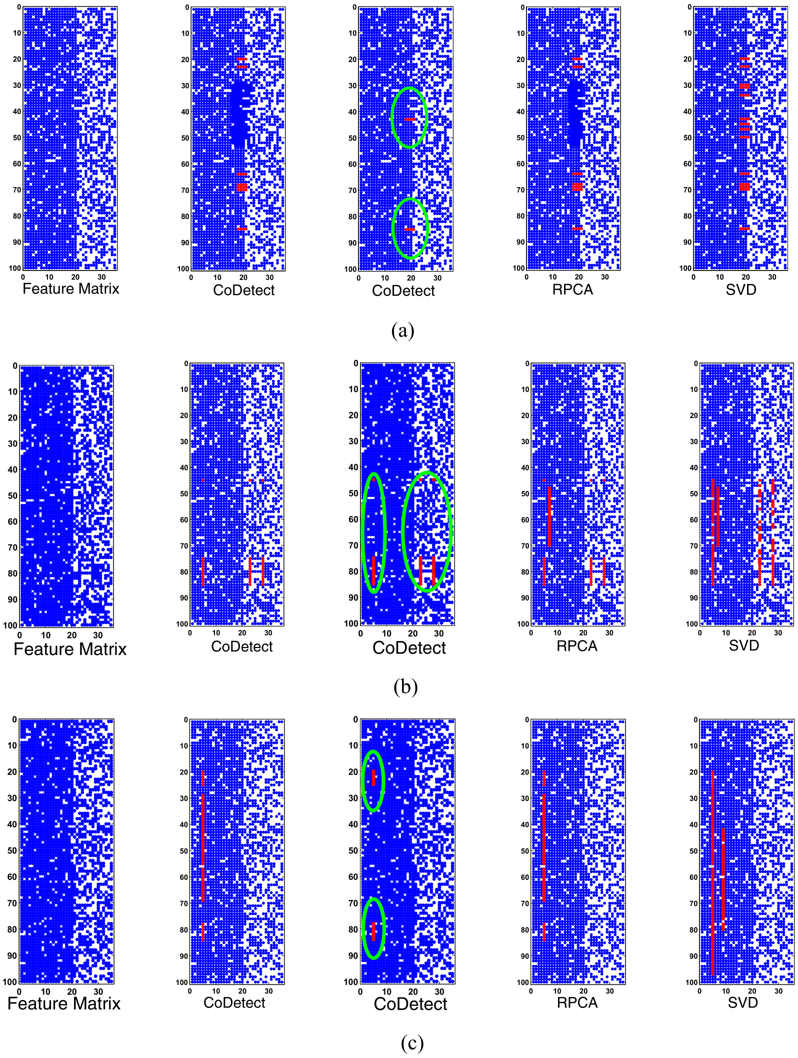
This means multi financial entities have business with one entity, as fraud scenario 2 shown (Fig. 4). RPCA detects all three fraud patterns as CoDetect does, but there are some false positive detection generated by RPCA and CoDetect with $\alpha = 0$. SVD generates more false positive detection as each sub-figure shown.

C. QUALITATIVE ANALYSIS ON FEATURE MATRIX F

The second set of experiments is to evaluate detection performance on feature matrix $\mathbf{F}$. We also evaluate CoDetect in two scenarios: (1) the first is to use objective function in Section 3.3 that is performing detection only on $\mathbf{F}$, and (2) with $\alpha \neq 0$, i.e., performing fraud detection with state-of-art models first, we examine how these detection models on $\mathbf{F}$ compare with CoDetect. For CoDetect, we set $\alpha = 0.1$ for synthetic data. We inject one type of fraud pattern into transaction each time to build $\mathbf{S}$ and $\mathbf{F}$. Figure 7 depicts the experiments results on matrix $\mathbf{F}$.

Similar to the results on matrix $\mathbf{S}$, the performance of CoDetect degrades when using objective function only. CoDetect generates false positives on $\mathbf{F}$. When $\alpha \neq 0$, CoDetect outperforms RPCA and SVD on $\mathbf{F}$ with no false positives. As we already known, $\mathbf{U}$ transfers fraud knowledge between $\mathbf{S}$ and $\mathbf{F}$. The benefit don't stop here. The pseudo class label $\mathbf{U}$ can be working as a indicator for tracing and forensic the fraud. Taking merge fraud as an example, from Figure(b), several columns are located as anomaly features. They are properties of Direction, Service and Value from SDLAT. These anomaly feature help executive to trace and forensic the fraud. We can easily locate the feature(s) which result in the fraud. From Figure 7(a) middle one, the red dot in green circle is detected an outlier fraud pattern by CoDetect. This means there is a common weight between two nodes (node 43 and node 85 in our experiments). Correspondingly, from Figure 7(c), the row 43 and 85 are detected as fraud nodes and column 18 to 21 are located as anomaly features. In SDLAT, feature 18 to 21 are all related to value of trading. Obviously, we can detect the fraud entities and anomaly feature simultaneously, the anomaly feature reveal the nature of the fraud.

Essentially, from (8), we know that $\mathbf{U}$ guarantee the relation between anomaly points and anomaly feature and also suppress the false positive rate. These anomaly features are some important complementary information for anomaly points detected on graph matrix. In comparison, RPCA and SVD can only work on feature matrix, and generate much more false detection on feature matrix shown on Figure 7. Even we can use RPCA and SVD on graph matrix and feature matrix respectively, we hardly establish the relation between the detected anomaly. From perspective of security executive,



(a)

(b)

(c)

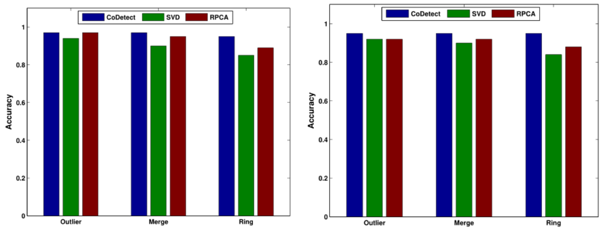
FIGURE 7. **Fraud detection on feature matrix S with $\alpha = 0$ and $\alpha \neq 0$. Each blue dot in (a)–(c) represents a feature. Red dots marked by green circles are detected anomaly features related to anomaly entities. (a) Outlier on feature matrix. (b) Merge on feature matrix. (c) Ring on feature matrix.**

we know fraud happened but we are not able to trace and forensic.

D. EVALUATION WITH MATRIX FACTORIZATION METHODS

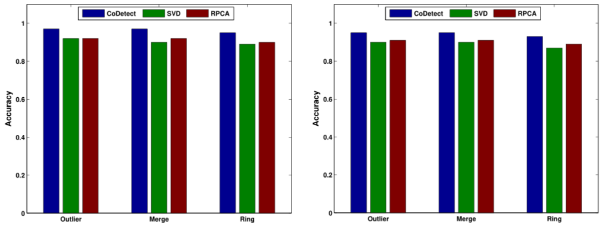
We evaluate the detection accuracy on similarity matrix and feature matrix respectively. We inject three fraud patterns into two dataset respectively. We first perform the experiments by CoDetect, Robust PCA and SVD for the comparison of accuracy on similarity. RPCA and SVD are used to extract top k rank components, then we obtain the residual matrix by original matrix minus top k rank components. Here k is set to 5. We omit the parameter analysis and only report the best performance on RPCA and SVD. We repeat the experiments 20 times and report the mean accuracy on similarity matrix. From Fig. 8 we see that CoDetect and RPCA achieves high detection accuracy on similarity matrix from synthetic data and real life data. We perform the experiments on feature matrix. From Fig. 9 we see that the RPCA and SVD detection accuracy drops dramatically. CoDetect achieve high detection accuracy on feature matrix.

Time Performance Analysis: We evaluate the time performance here. The experiments are all performed on machine with Intel(R) Core(TM) i7 CUP @ 2.60GHz and 32GB memory, running Windows 7. Each experiment is repeated 20 times and we report the mean time in second. We first evaluate the scalability of CoDetect with retune the size of graph. We tune the size of graph from 5,000 to 25,000 and tune the edge number from 5 × 10^5 to 15 × 10^5, then inject three fraud patterns into each graph. Then we evaluate the detection time performance in term of second. We find that CoDetect converge to threshold in 10 iteration mostly. So we set the iteration to 10 in order to reducing the computation cost. The result is presented in Fig. 10. It can be



(a) (b)

FIGURE 8. Detection accuracy on graph-based similarity matrix. CoDetect and Robust PCA achieve high detection accuracy on all fraud patterns. (a) Similarity matrix (synthetic data). (b) Similarity matrix (real life data).



(a) (b)

FIGURE 9. Detection accuracy on feature matrix. CoDetect achieve high detection accuracy of anomaly feature. (a) Feature matrix (synthetic data) (b) Feature matrix (real life data).

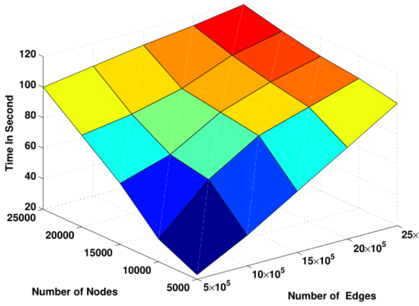


FIGURE 10. Detection time in second with different number of nodes and edges.

seen that CoDetect scales almost linearly with retune the graph size and number of edge. All the detection can be completed in acceptable time. The next experiments are performed using Iknow.com dataset with about 27,000 nodes and 5,600,000 edges. We compare the time performance of CoDetect, RPCA and SVD with different number of rank, r for computing the residual matrix. The result is presented in Fig. 11. Clearly, CoDetect achieves high time performance.

E. EVALUATION WITH SUBSPACE CLUSTERING METHODS

The fraud patterns can be represented as anomaly in subspace of graph matrix and feature matrix. Anomaly detection using subspace clustering base on the assumption that cluster in subspace with small samples means anomaly [24]. The next experiment is to evaluate CoDetect with three methods MAFIA [42], SCHISM [43] and DiSH [44], that have best subspace clustering performance reported in [41]. We follow the recommended parameters setting for three clustering methods. We change the number of cluster for anomaly detection and report the best.

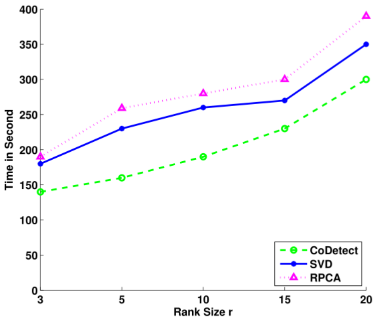
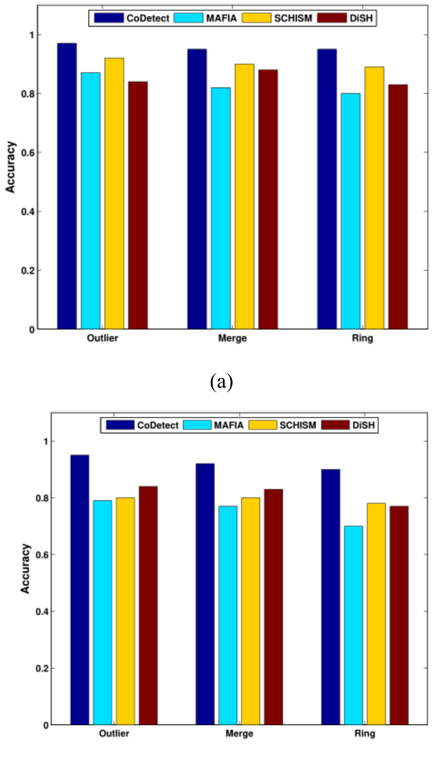


FIGURE 11. Comparison of time with different rank size.



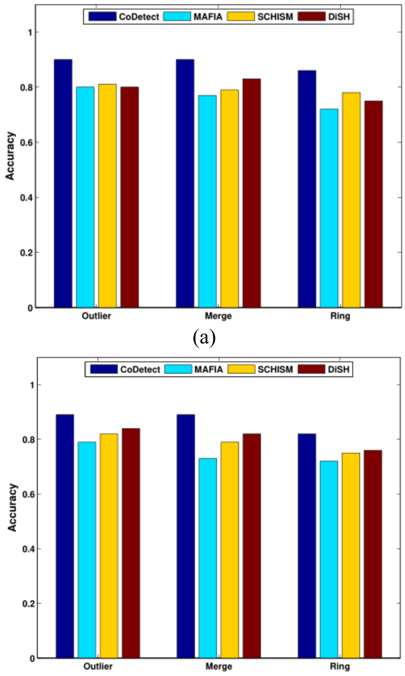
(a)

(b)

FIGURE 12. Detection accuracy on graph-based similarity matrix with subspace clustering. CoDetect achieve high detection accuracy on all fraud patterns. (a) Similarity matrix from synthetic data. (b) Similarity matrix from real life data.

From Fig. 12 we see that CoDetect achieves high detection accuracy on similarity matrix from synthetic data and real world data. We perform the experiments on feature matrix. As Fig. 13 show, CoDetect achieve high detection accuracy on feature matrix.

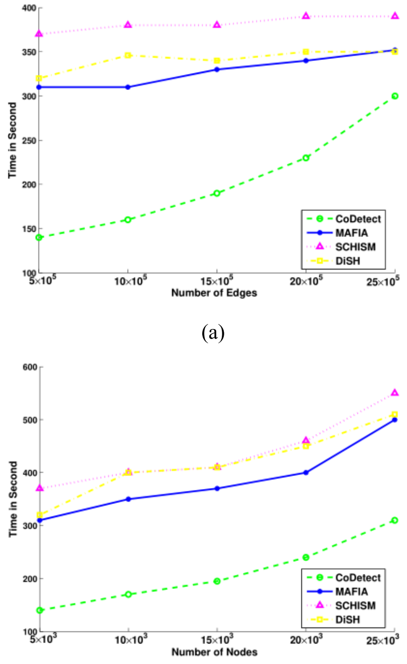
Time Performance Analysis: We set rank size $r = 5$ and fix iteration $= 20$ for CoDetect. We perform the time evaluation in two ways. The first one is to fix number of nodes, and evaluate the time performance with retune the number of edges. The second one is to fix number of edges, and evaluated time performance with retune the number of nodes. The result is presented in Fig. 14. Clearly, CoDetect achieves high time performance in all ways.



(a)

(b)

FIGURE 13. Detection accuracy on feature matrix with subspace clustering. CoDetect achieve high detection accuracy on all fraud patterns. (a) Feature matrix from synthetic data. (b) Feature matrix from real life data.



(a)

(b)

FIGURE 14. Detection time in second. (a) Fix nodes number to 5000. (b) Fix edges number to 5 × 10^5.

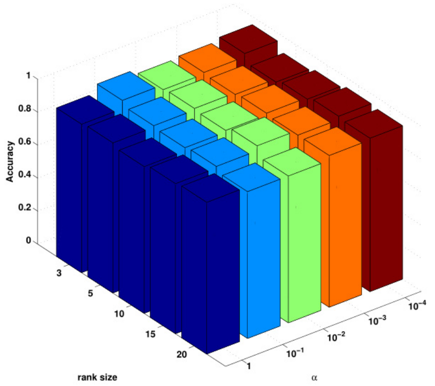


FIGURE 15. Detection performance with different rank size r and α.

F. MODEL PARAMETERS ANALYSIS

The last experiment is to evaluate the performance of CoDetect with respect to input parameters α and r, see Algorithm 1. We tune the parameter α by a "grid-search" strategy from $\{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. For parameter rank size r, we set $r = 3, 5, 10, 15, 20$ respectively. We evaluate the detection performance with each pair of input parameter and repeat the experiments 20 times for the average results.

From Fig. 15 we can see that CoDetect is not very sensitive to α. It makes the model robust to different datasets. We also find that CoDetect remains high detection accuracy with very low rank reconstruction rank.

VI. RELATED WORK

In this section, we first review the related work on financial fraud detection, and then we review anomaly detection which employs similar techniques or methods with fraud detection.

A. FINANCIAL FRAUD Detection

Financial fraud detection concerns about the detection of fraud in insurance, credit card, telecommunications and other financial crime activities such as money laundering [26], [36], [32], [34].

Statistical models have been used for detection of financial fraud [35]. Bahnsen *et al.* [38] improve the detection performance by calibrating probabilities before establishing Bayes model. HMM model is used to model the customers' credit card shopping patterns for detection of credit card fraud [27]. The shopping items indicate the hidden state and the corresponding prices from certain ranges are the observation. LR(Logistic Regression), Support Vector Machines(SVMs) and Random Forest(RF) are evaluated for credit card detection. The detection models are built on primary features and derived features from transaction [30]. Whitrow *et al.* [28] proposed a new preprocessing strategy for better fraud detection with SVMs and KNN classification. Transactions aggregated in term of time window, then data with new features is used to model the pattern. Wei *et al.* [29] addressed the problem of unbalanced financial data and employed cost-sensitive neural network to punish the misclassification of fraud transaction. Sahin *et al.* [33] incorporate cost function into decision tree to boost performance on unbalanced data. Following the general procedure of classification, feature selection is proceed to boost the detection performance of credit card fraud [31]. Perols [35] performed a systematic analysis of financial fraud detection with popular statistical and machine learning models. The evaluation is under the supervised manner. All these methods rely on accurate identification of fraud patterns from data set and these methods also suffer from the problem of unbalanced data. Bolton and David [37] perform fraud detection with clustering methods. This unsupervised manner is under the assumption that small cluster indicates the anomaly in data. CoDetect is an unsupervised model which is based on matrices co-factorization. The matrices from graph represent the genuine proprieties(features and connections) of financial data. The detection results give a better understanding of fraud patterns and furthermore, help to trace the originate of fraud groups.

B. ANOMALY DETECTION

Financial fraud detection only focuses on a particulars domain: financial activities. Anomaly detection tries to nd patterns in data that is unusual seen or out of expectation [11]. So anomaly detection can be seen as a general form of fraud detection. Fraud detection is one application of anomaly detection [4]. Two techniques are most related to fraud detection. One is one-class classification [16]. Another one is clustering based outlier detection [19]. One-class classification usually based on the assumption that the detection model is built on data which is generated from one or several statistical distributions [17], [14]. This assumption might not hold when encountering high dimensional data with bit portion of corrupted items [23], [24], [22]. There is lot of work on graph-based outlier detection [6]. Akoglu *et al.* [3] proposed a new algorithm on graph-based anomaly detection. Eberle and Holder [5] discovered structural information for anomaly detection from graph-based data. Sun *et al.* [10] segment the bi-parties graph for the anomaly detection. Tong and Lin [7] proposed a novel algorithm for better detection and interpretation of anomaly in graph-based data. Henderson *et al.* [13] proposed a new way to construct feature for better mining performance from graph-based data. More recently, much attentions have been payed to time-involving graph [15], [18], [12], [20]. There are lots of work on social mining from graph-based data [21].

VII. CONCLUSION

We propose a new framework, CoDetect, which can perform fraud detection on graph-based similarity matrix and feature matrix simultaneously. It introduces a new way to reveal the nature of financial activities from fraud patterns to suspicious property. Furthermore, the framework provides a more interpretable way to identify the fraud on sparse matrix. Experimental results on synthetic and real world data sets show that the proposed framework (CoDetect) can effectively detect the fraud patterns as well as suspicious features. With this co-detection framework, executives in financial supervision can not only detect the fraud patterns but also trace the original of fraud with suspicious feature.

Financial activities are involving with time. We can represent these activities into similarity tensor and feature tensor. So we would like to study how to integrate tensor into co-detect framework for fraud detection.

REFERENCES

[1] C. Sullivan and E. Smith. "Trade-Based Money Laundering: Risks and Regulatory Responses," Social Sci. Electron. Publishing, 2012, p. 6.

[2] United Press International. (May 2009). *Trade-Based Money Laundering Flourishing*. [Online]. Available: http://www.upi.com/Top News/2009/05/11/Trade-based-money-laundering-flourishing/UPI-17331242061466

[3] L. Akoglu, M. McGlohon, and C. Faloutsos, "OddBall: Spotting anomalies in weighted graphs," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2010, pp. 410–421.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, 2009, Art. no. 15.

[5] W. Eberle and L. Holder, "Mining for structural anomalies in graph-based data," in *Proc. DMin*, 2007, pp. 376–389.

[6] C. C. Noble and D. J. Cook, "Graph-based anomaly detection," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 631–636.

[7] H. Tong and C.-Y. Lin, "Non-negative residual matrix factorization with application to graph anomaly detection," in *Proc. SIAM Int. Conf. Data Mining*, 2011, pp. 1–11.

[8] S. Wang, J. Tang, and H. Liu, "Embedded unsupervised feature selection," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 470–476.

[9] Z. Lin, M. Chen, and Y. Ma. (2010). "The Augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices." [Online]. Available: https://arxiv.org/abs/1009.5055.

[10] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos, "Neighborhood formation and anomaly detection in bipartite graphs," in *Proc. 15th IEEE Int. Conf. Data Mining*, Nov. 2005, p. 8.

[11] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Netw.*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007.

[12] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, p. 18–32, Jan. 2014.

[13] K. Henderson *et al.*, "It's who you know: Graph mining using recursive structural features," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 663–671.

[14] F. Keller, E. Müller, and K. Bohm, "HiCS: High contrast subspaces for density-based outlier ranking," in *Proc. ICDE*, Apr. 2012, pp. 1037–1048.

[15] D. Koutra, E. Papalexakis, and C. Faloutsos, "Tensorsplat: Spotting latent anomalies in time," in *Proc. PCI*, Oct. 2012, pp. 144–149.

[16] J. H. M. Janssens, I. Flesch, and E. O. Postma, "Outlier detection with one-class classifiers from ML and KDD," in *Proc. ICMLA*, Dec. 2009, pp. 147–153.

[17] N. A. Heard, D. J. Weston, K. Platanioti, and D. J. Hand, "Bayesian anomaly detection methods for social networks," *Ann. Appl. Statist.*, vol. 4, no. 2, pp. 645–662, 2010.

[18] J. Tang and H. Liu "CoSelect: Feature selection with instance selection for social media data," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 1–9.

[19] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognit. Lett.*, vol. 24, nos. 9–10, pp. 1641–1650, 2003.

[20] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, *Outlier Detection for Temporal Data* (Synthesis Lectures on Data Mining and Knowledge Discovery). San Rafael, CA, USA: Morgan & Claypool, 2014.

[21] J. Tang, Y. Chang, and H. Liu "Mining social media with social theories: A survey," *ACM SIGKDD Explorations Newslett.*, vol. 15, no. 2, pp. 20–29, 2013.

[22] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 89–98.

[23] Q. Gu and J. Zhou, "Co-clustering on manifolds," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 359–368.

[24] K. Sim, V. Gopalkrishnan, A. Zimek, and G. Cong, "A survey on enhanced subspace clustering," *Data Mining Knowl. Discovery*, vol. 26, no. 2, pp. 332–397, 2013.

[25] S. Mcskimming, "Trade-based money laundering: Responding to an emerging threat," *Deakin Law Rev.*, vol. 15, no. 1, 2010.

[26] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decision Support Syst.*, vol. 50, no. 3, pp. 559–569, 2011.

[27] A. Srivastava, A. Kundu, S. Sural, and A. K. Majumdar, "Credit card fraud detection using hidden Markov model," *IEEE Trans. Depend. Sec. Comput.*, vol. 5, no. 1, pp. 37–48, Jan./Mar. 2008.

[28] C. Whitrow, D. J. Hand, P. Juszczak, D. J. Weston, and N. M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," *Data Mining Knowl. Discovery*, vol. 18, no. 1, pp. 30–55, 2009.

[29] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, "Effective detection of sophisticated Online banking fraud on extremely imbalanced data," *World Wide Web*, vol. 16, no. 4, pp. 449–475, 2013.

[30] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Syst.*, vol. 50, no. 3, pp. 602–613, 2011.

[31] P. Ravisankar, V. Ravi, G. R. Rao, and I. Bose, "Detection of financial statement fraud and feature selection using data mining techniques," *Decision Support Syst.*, vol. 50, no. 2, pp. 491–500, 2011.

[32] D. Zhang and L. Zhou "Discovering golden nuggets: Data mining in financial application," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 4, pp. 513–522, Nov. 2004.

[33] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," *Expert Syst. Appl.*, vol. 40, no. 15, pp. 5916–5923, 2013.

[34] M. E. Edge and P. R. F. Sampaio, "A survey of signature based methods for financial fraud detection," *Comput. Secur.*, vol. 28, no. 6, pp. 381–394, 2009.

[35] J. Perols, "Financial statement fraud detection: An analysis of statistical and machine learning algorithms," *Auditing, J. Pract. Theory*, vol. 30, no. 2, pp. 19–50, 2010.

[36] C. Phua, V. Lee, K. Smith, and R. Gayler. (2010). "A comprehensive survey of data mining-based fraud detection research." [Online]. Available: https://arxiv.org/abs/1009.6119

[37] R. J. Bolton and J. H. David, "Unsupervised profiling methods for fraud detection," in *Proc. Credit Scoring Credit Control 7*, 2001, pp. 1–16.

[38] A. C. Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten, "Improving credit card fraud detection with calibrated probabilities," in *Proc. SIAM Int. Conf. Data Mining*, 2014, pp. 1–9.

[39] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 1, 2011, Art. no. 11.

[40] M. Aharon, M. Elad, and A. Bruckstein, "*K*-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

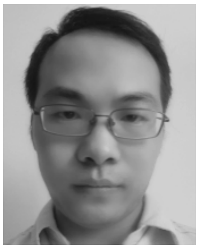
[41] G. Moise, A. Zimek, P. Kröger, H. P. Kriegel, and J. Sander, "Subspace and projected clustering: experimental evaluation and analysis," *Knowl. Inf. Syst.*, vol. 21, p. 299, Dec. 2009.

[42] H. Nagesh, S. Goil, and A. Choudhary, "Adaptive grids for clustering massive data sets," in *Proc. SIAM Int. Conf. Data Mining*, 2001, pp. 1–17.

[43] K. Sequeira and M. Zaki, "SCHISM: A new approach for interesting subspace mining," in *Proc. ICDM*, Nov. 2004, pp. 186–193.

[44] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, I. Müller-Gorman, and A. Zimek, "Detection and visualization of subspace cluster hierarchies," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2007, pp. 152–163.

[45] L. Akoglu, H. Tong, J. Vreeken, and C. Faloutsos, "Fast and reliable anomaly detection in categorical data," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 415–424.



DONGXU HUANG received the B.S. and M.S. degrees in network security from Northwestern Polytechnical University, where he is currently pursuing the Ph.D. degree with the School of Automation. His current research interests include social network, data mining, and big data.

DEJUN MU is currently a Professor with the School of Automation, Northwestern Polytechnical University. His current research interests include computer network, social network, and big data.

LIBIN YANG received the Ph.D. degree from Northwestern Polytechnical University, China, in 2009. He was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. He is currently an Assistant Professor with the School of Automation, Northwestern Polytechnical University. His current research interests include information retrieval, computer network, and game theory.

XIAOYAN CAI received the Ph.D. degree from Northwestern Polytechnical University, China, in 2009. She was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. She is currently an Associate Professor with the School of Automation, Northwestern Polytechnical University. Her current research interests include citation recommendation, document summarization, information retrieval, and machine learning.