Understanding Fraudulent Behavior in Peer-to-Peer Lending

Yang Yang
Zhejiang University
yangya@zju.edu.cn

Yuhong Xu
Zhejiang University
xuyhhh@gmail.com

Chunping Wang
PPDAI Group Inc.
wangchunping02@ppdai.com

Yizhou Sun
University of California, Los Angeles
yzsun@cs.ucla.edu

Fei Wu, Yueting Zhuang
Zhejiang University
(wufei,yzhuang)@zju.edu.cn

Ming Gu
PPDAI Group Inc.
guming@ppdai.com

ABSTRACT

Emergence of microcredit, a hope to promote equality for impoverished people, is ushering a new chapter for financial industry. By taking the form of peer-to-peer lending over the web, it has the advantage of easy and fast online application process for borrowers. Of particular concern to P2P lending is the threat coming from fraudulent borrowers, who intend to deceive lenders for financial gain. Thus it is critical to identify frauds in P2P lending. Existing work of fraud detection in the context of traditional financial institutions are mainly based on users' historical loan records. However, statistics show that over 60% of P2P lending frauds happen at the first loan application, where no sufficient loan records are available.

In this paper, we study the problem of identifying fraudulent borrowers in P2P lending by considering their social information. More specifically, we employ a dataset provided by PPDai, which consists of over 11 million users and more than 1.5 billion call logs between them. We establish a mobile network and explore social factors that lead to users' fraudulent behaviors. Moreover, we disclose cheating agents, who benefit from inciting other users to cheat, providing false information, and faking application documents. Based on our observations, we propose a novel probabilistic framework to identify fraudulent borrowers and cheating agents simultaneously. Experimental results on real-world dataset demonstrate a significant improvement achieved by our model compared with several baselines. Furthermore, our model can effectively identify cheating agents without any supervised information.

CCS CONCEPTS

• **Computing methodologies → Artificial intelligence**;

KEYWORDS

Fraud detection, Social network, P2P lending

ACM Reference Format:
Yang Yang, Yuhong Xu, Chunping Wang, Yizhou Sun, Fei Wu, Yueting Zhuang, and Ming Gu. 2018. Understanding Fraudulent Behavior in Peer-to-Peer Lending . In *Proceedings of ACM SIGKDD Conference (KDD'18)*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

KDD'18, August 2018, London, United Kindom

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

"Even the poorest of the poor can work to bring about their own development." With this vision, microcredit was born as very small loans to impoverished borrowers, who typically lack collateral or a verifiable credit history and thus are highly likely to be rejected by traditional financial services. It is an endeavor to offer assistance and promote equality for this under-served population, which is huge in size especially in developing countries due to the lack of a mature credit system. In recent years, microcredit has grown rapidly by taking the form of peer-to-peer (P2P) lending over the web, which allows individuals to acquire uncollateralized loans from others without an intermediary financial institution. It is helped by easy and fast online application process. This allows many platforms to acquire massive number of users, such as PPDai[1], Zopa[2], Prosper[3], and LendingClub[4]. Taking PPDai, the first and one of the largest P2P lending sites in China, as an example, it has attracted more than 57 million users and funded over $11 billion loans by the end of September 2017.

Besides offering financial assistance, the key idea of P2P lending is to improve the credit of impoverished people, showing the world that many of them deserve trust. However, this idea is endangered seriously by some fraudulent borrowers, who intend to default and deceive lenders for financial gain. Therefore, it is critical to identify these fraudulent borrowers in advance in order to keep the healthy growth of P2P lending.

Traditional loan-fraud detection mainly employs users' historical loan records [4, 7, 16, 18, 31]. However, the major challenge in the P2P lending context is the lack of borrower's historical loan records. For instance, as Figure 1 shows, over 40% of borrowers (who applies for loan at least once) in PPDai only have one loan record. Meanwhile, around 61% of fraudulent loan requests happen at the first application. There has been little research in this area, as P2P lending is a relatively new financial phenomenon.

Inspired by the study that default behavior influences users with social relations [8], in this work, we propose to identify fraudulent borrowers by their social information. In particular, most P2P lending platforms require users to provide call logs (only metadata, no communication context) when they apply for a loan. We thereby construct a social communication network based on the user call logs, and study different social characteristics of fraudulent borrowers and others.

[1] http://http://www.ppdai.com/
[2] http://www.zopa.com/
[3] https://www.prosper.com/landing
[4] https://www.lendingclub.com/

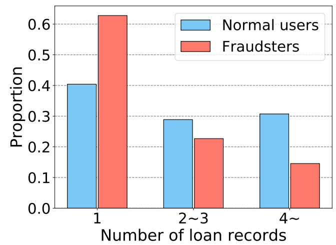


Figure 1: Distribution of number of loan requests applied by fraudsters or normal users.

The problem of identifying fraudulent borrowers based on social network is nontrivial. First of all, a user's social information is not intuitively correlated with her fraudulent behavior. How to discover the hidden correlation and design effective machine learning models to identify fraudulent borrowers is the major challenge. Secondly, through our study, we find a special type of users that do not default, but shall be responsible for many of fraudulent loans. We call these users as cheating agents, who benefit from inciting other users to cheat, providing false information, and faking personal information. We also find that many fraudulent borrowers will communicate with cheating agents, and vice versa. Thus the information of cheating agents could help in our task. Unfortunately, the ground truth data (i.e., label) of cheating agents are extremely hard to obtain as they will not default themselves. Therefore, identifying cheating agents without supervised information is a big challenge. Last but not least, users generate vast quantities of call logs everyday. How to efficiently process these data is our third challenge.

To address the first challenge, we conduct several exploratory analysis based on a dataset provided by PPDai, which consists of over 1.5 billion call logs between more than 10 million users. For example, we find that fraudulent borrowers tend to be more active in the network within the last week before applying for a loan. We also find that fraudulent borrowers connect with more cheating agents, and vice versa.

Inspired by our observations, we propose a novel probabilistic framework, dual-task factor graph. Generally, our model is semi-supervised and aims to identify fraudulent borrowers and cheating agents in a uniform framework. We build connections between these two roles of users, provide indirect supervised information for cheating agents from fraudulent borrowers' labels, and thereby handle the second challenge mentioned above. We also design an efficient approximate learning algorithm to handle large-scale data and train the model. Experimental results show that our model outperforms several state-of-the-art baseline methods. Furthermore, we demonstrate that our model can also effectively identify cheating agents, without any supervised information. We summarize our contributions as follows:

- Based on a large-scale dataset, we discover different characteristics of fraudsters, cheating agents, and normal users.
- We propose a novel semi-supervised framework to jointly model fraudulent borrowers and cheating agents.
- We construct sufficient experiments to validate the effectiveness of our model.

2 DATA AND PROBLEM

2.1 Dataset

Our dataset is provided by PPDai, one of the leading online consumer finance marketplaces in China, spanning June 2015 to May 2017. It consists of three types of data: *user call logs*, *user attributes*, and *loan records* (only used for labeling frauds) during that time.

More specifically, we have 1,563,368,539 telephone calls between 11,724,980 PPDai registered users. Each call log contains starting time, ending time, and masked user identity of caller and callee. For user attributes, we have each user's age, gender and educational level (desensitized). Due to privacy concerns, we only report overall statistics without revealing any identifiable information of individuals in this paper. A user may have multiple records of loan history which depends on the number of successful loans. Each record can be further composed of loan time, loan amount and repayment time. The loan history is used only to label fraudster identity. More specifically, we define a user who has 90 days overdue repayment as fraudster. In this way, among 3,900,906 users who have at least one record of loan history, we obtain 297,001 fraudsters in total.

2.2 Problem Definition

We extract a mobile communication network G from call logs in our dataset. Formally, a mobile communication network is a directed graph $G = (V, B, E)$, where V is the set of users, B is an attribute matrix with each element b_{ij} denoting the j-th attribute (e.g., age) of the user v_i, and each directed edge $e_{ij} \in E$ indicates that the user v_i calls the user v_j at least once ($v_i, v_j \in V$). Exiting work has concluded that the mobile network can roughly approximate one's social network [10] [33].

According to their historical loan records, we define an *identity label* y_i for each user v_i in G. For those who have defaulted a loan more than 90 days at least once, we define $y_i = 1$; for others who have loan history and never default a loan more than 90 days, we define their corresponding $y_i = 0$; for the remaining who have no loan history at all, we define an unknown identity label $y_i =?$, as we do not know if she will cheat yet. We then formulate our problem below.

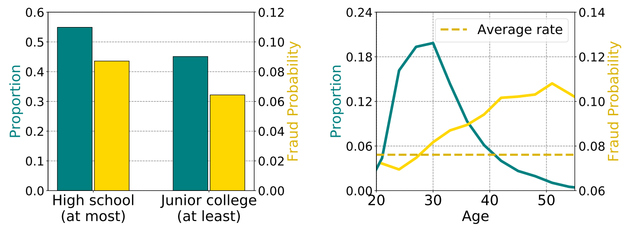
Definition 2.1. **Fraud borrower prediction.** Given a user v_i who has no loan history (i.e., $y_i =?$), a time t, and a mobile communication network $G = (V, B, E)$ extracted from all call logs before time t, and the identity vector Y, our goal is to predict, once the user v_i applies for a loan at time t, whether she will default more than 90 days.

Notice that our problem is different from existing work [26][9] [7] as we mainly consider the social network information and do not employ the historical loan records for the prediction task.

3 EXPLORATORY ANALYSIS

We categorize users in our dataset into three groups, which constitutes the basis for our analysis framework. We refer to users that default a loan for more than 90 days as *fraudulent borrowers*, or *frauds* in short [5]. People who benefit from encouraging and assisting

[5]We choose 90 days as to be consistent with PPDai's definition to fraudulent borrowers, used in their online operations.



(a) Education. (b) Age.

Figure 2: User attributes (education and age) of fraudulent borrowers and normal users.

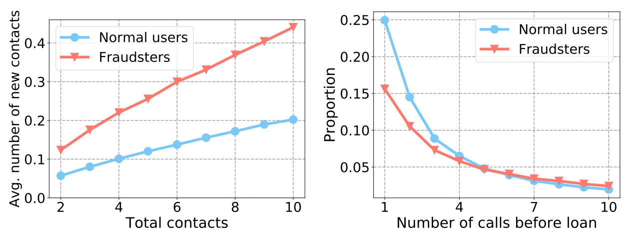


Figure 3: Calling behavior of fraudulent borrowers and normal users one week before they applying for loan.

other users to cheat by providing false information, faking application documents, eliminating uncredited records for fraudulent borrowers, etc., are referred to as *cheating agents*. In other words, cheating agents will influence some borrowers to become frauds. By our study, very few cheating agents themselves are frauds, to keep a low profile. The rest of the users who have applied at least one loan and kept paying their debts are *normal users*. In summary, we have 297,001 fraudulent borrowers, 12,985 cheating agents, and 3,603,905 normal users in our dataset. Our goal in this section is to explore the characteristics that differentiate frauds, cheating agents, and normal users.

3.1 Distinguishing Frauds from Normal Users

User attributes. We use the education level and user age as two examples to demonstrate how basic user attributes affect their fraudulent behavior (Figure 2). From Figure 2(a), we see that nearly 45.7% of PPDai users possess at least a junior college degree, while other users who are educated at most high school are more likely to be a fraudulent borrower. Meanwhile, as Figure 2(b) shows, the probability of a user being frauds increases as the user age grows.

Calling behavior. Comparing with normal users, fraudulent borrowers contact their friends more frequently in the last week before they applying for a loan. This phenomenon is consistently reflected on both the number of new contacts the user has (Figure 3(a)) and the number of calls made by the user (Figure 3(b)). It suggests that the network structure of fraudulent borrowers will vary more.

Social network. A person’s mobile network can reasonably approximate her social network. A user’s degree measures the number of other users she has called at least once. Degree and PageRank [22], a common metric of vertex importance, reflect the involvement of a user in her social network. Fraud borrowers present larger degree and higher PageRank score than normal users, shown in Figure 4(a) and Figure 4(b). Users with larger degree and higher importance are more likely to be a fraud.

Furthermore, we define the fraudulent traffic of a vertex v as the maximal number of fraudulent neighbors v’s neighbors have. It reflects how much information between frauds can be diffused through v. As expected, from Figure 4(d), fraudulent borrowers have larger fraudulent traffic than normal users. The probability of a user being a fraud increases as her fraudulent traffic grows.

Interestingly, as Figure 4(c) shows, compared with normal users, fraudulent borrowers have a larger proportion of fraudulent second-degree-neighbors. Through some careful investigation, we find this result is caused by some abnormal vertexes, which bridges many fraudulent borrowers. Our next question is, who are these “abnormal bridges”?

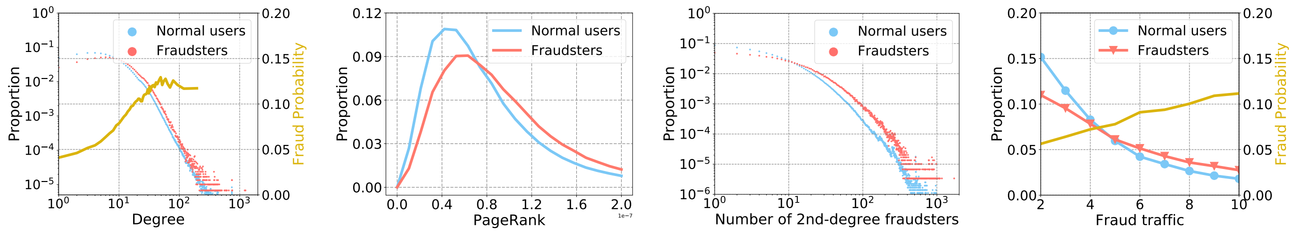
3.2 Study of Cheating Agents

Existence. To further confirm the existence of “abnormal bridges”, we create a null model based on the assumption that any vertexes in the mobile network uniformly connects to a fraudulent borrower or a normal user. We then compare how the number of fraudulent neighbors distributes in null model and in real data. Figure 5(a) shows a clear difference. Overall, compared with the null model, real-world network contains more vertexes connected with frauds.

By several case studies and interviews with business people of PPDai, we conclude that the above “abnormal bridges” are actually cheating agents, who connect with a lot fraudulent borrowers and benefit from providing false information, faking application documents, eliminating uncredited records, and so on.

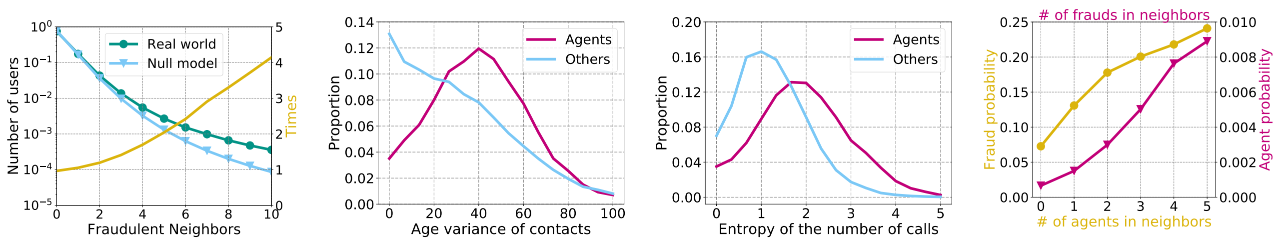
Identify cheating agents. We then explore factors that can help us identify cheating agents from the mobile network. Intuitively, cheating agents make calls to a more diverse population. Taking user age as an example to measure the population diversity, we validate the variance of the age distribution of a particular user’s neighbors. As Figure 5(b) shows, we see contacts of cheating agents have a larger variance. Moreover, the entropy of the number of phone calls over different neighbors tends to be larger for cheating agents, shown in Figure 5(c).

Connection between agents and frauds. Intuitively, users who connect with more cheating agents tend to default on their loans. On the other hand, users who have lots fraudulent neighbors are more likely to be cheating agents. We examine this in Figure 5(d), which shows that the probability of a user being fraudulent borrower increases as the number of her neighbors being cheating agents grows, and vice versa. This result also further confirms that the previously observed “abnormal bridges” are cheating agents. One thing worth to mention is that, fraudulent borrowers and cheating agents may overlap in theory (i.e., some cheating agents will default loans by themselves). However, our data show that there is nearly no such case.



(a) Degree. (b) PageRank score. (c) Second neighbors as frauds. (d) Fradulent traffic.

Figure 4: Distinguishing social-network characteristics between fraudulent borrowers and normal users. Figure 4(a) presents the degree distribution of fraudulent borrowers and normal users, and the probability of a user being as a fraud changes over her degree. Figure 4(b) and (c) are the comparison results of PageRank and the number of second-degree neighbors as frauds. Figure 4(d) shows the correlation between fraud traffic and fraud probability. We define the fraud traffic of a vertex as the maximal number of fraudulent neighbors among this vertex's neighbors.



(a) Existence of cheating agents. (b) Age diversity. (c) Call entropy. (d) Connection with frauds.

Figure 5: They study of cheating agents. Figure 5(a) examines the existence of cheating agents by constructing a null model. Figure 5(b) and (c) present features that can help to identify cheating agents. Figure 5(d) shows the correlation between cheating agents and fraudulent borrowers.

4 MODEL FORMULATION

4.1 Model Description

Overview. We develop a probabilistic model, Dual-Task Factor Graph (DTF), to jointly identify fraudulent borrowers, normal users, and cheating agents in a given mobile network. In general, the model itself can be thought of a factor graph over four types of random variables, which are introduced as follows:

- Identity of fraudulent borrowers. We define Y as a set of binary random variables to indicate whether a particular user is a fraudulent borrower or not. We denote identities that has been known as Y^L and unknown identities as Y^U.
- Identity of cheating agents. Similarly, we define Z as a set of binary random variables to indicate whether a particular user is a cheating agent or not. To be mentioned, as obtaining identities of cheating agents is hard, we put this part in an unsupervised setting and assume all elements in Z are unknown and need to be inferred.
- Fraud borrower features. Inspired by Section 3, we define random variable X to indicate user features extracted from personal attributes, calling behavior, and social network structure of users. We expect that a user v_i's feature X_i has correlation with her fraudulent borrower identity Y_i. We list details of how we define each feature in Table 1.
- Cheating agent features. Similarly, we define random variable $\tilde{X}$ to represent another set of user features that are correlated with identify of cheating agents. These features in $\tilde{X}$ are mainly defined based on the mobile network. Please see Table 2 for details.

Generally, our goal is to model the joint probability of (Y, Z) conditioned on the observed user features $(X, \tilde{X})$, i.e., $P(Y, Z|X, \tilde{X})$. Factor graph provides us a way to factorize the "global" probability as a product of "local" factor functions [15], each presents the correlation between a particular set of random variables. This factorization makes the computation of the joint probability easy. The remaining key issue here is how to define each factors (i.e., the correlation between random variables).

Factors. According to previous analysis in Section 3, a user v_i's fraudulent borrower features $\boldsymbol{x}_i$ can reveal her fraudster identity y_i to some extent. Formally, we define factors Ψ^F to model the correlation between X and Y as

$$\Psi_i^F(\boldsymbol{x}_i, y_i) = \kappa_1 \exp(\boldsymbol{\alpha}_{y_i} \boldsymbol{x}_i) \tag{1}$$

where α_i is the model parameter as a $|\boldsymbol{x}_i|$-length vector, and κ_1 is a normalization term to ensure that the sum of factor equal to 1.

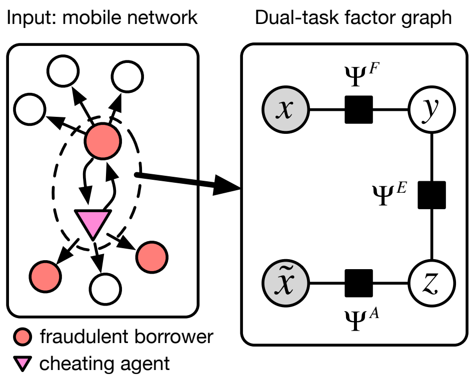


Figure 6: Graphical representation of our proposed model. Generally, it captures the connections between fraudulent borrowers and cheating agents.

Similarly, we also observe that user v_i's cheating agent identity can be reflected by her network structure. In particular, we define the factor Ψ^A to represent the correlation between $\tilde{X}$ and Z, which is instantiated as the following function:

$$\Psi_i^A(\tilde{\boldsymbol{x}}_i, z_i) = \kappa_2 \exp(\boldsymbol{\beta}_{z_i} \tilde{\boldsymbol{x}}_i) \tag{2}$$

where β_i is a $|\tilde{\boldsymbol{x}}_i|$-length vector with model parameters, and κ_2 is the normalization term.

From previous observation, we find that fraudulent borrower identity and cheating agent identity are correlated: users with more fraudulent neighbors are more likely to be cheating agents, and fraudulent borrowers have more cheating agents as their neighbors. Inspired by this phenomenon, we define factor Ψ^E to indicate the above correlation between Y and Z. More specifically, for each user pair v_i and v_j with an edge in the given mobile network G, we define a factor as follows:

$$\Psi_{ij}^E(y_i, z_j) = \begin{bmatrix} \gamma_{00} & \gamma_{10} \\ \gamma_{10} & \gamma_{11} \end{bmatrix} \tag{3}$$

where γ_{kl} captures the adaptability between $y = k$ and $z = l$. Intuitively, this factor bridges two otherwise disjoint model components that identify fraudulent borrowers and cheating agents respectively, and leads them to enhance the performance of each other.

Figure 6 presents the graphical presentation of our model. So far, we have defined three types of factors, i.e., Ψ^F, Ψ^A and Ψ^E, based on the insights obtained from our analysis in Section 3. By integrating all the factors together and according to the Hammersley-Clifford theorem [12], we obtain the following likelihood of a particular identity assignment:

$$\Pr(Y, Z|\boldsymbol{X}, \tilde{\boldsymbol{X}}; \boldsymbol{\theta}) = \frac{\prod_i \Psi_i^F(\boldsymbol{x}_i, y_i) \prod_j \Psi_j^A(\tilde{\boldsymbol{x}}_j, z_j) \prod_{i,j} \Psi_{ij}^E(y_i, z_j)}{\mathbb{Z}(\boldsymbol{X}, \tilde{\boldsymbol{X}})} \tag{4}$$

where $\mathbb{Z}(\boldsymbol{X}, \tilde{\boldsymbol{X}})$ is the partition function to ensure the sum of probability equal to 1, which takes the form as:

$$\mathbb{Z}(\boldsymbol{X}, \tilde{\boldsymbol{X}}) = \sum_{Y,Z} \prod_i \Psi_i^F(\boldsymbol{x}_i, y_i) \prod_j \Psi_j^A(\tilde{\boldsymbol{x}}_j, z_j) \prod_{i,j} \Psi_{ij}^E(y_i, z_j) \tag{5}$$

4.2 Model Inference and Learning

Inference. Suppose that C denotes all the random variables in our graph (i.e. $C = Y \cup Z$), one of the most typical inference problems are to predict the label (i.e. $c^* = argmax_c \Pr(c)$) given the mobile network G and user features. For discrete variables, the marginals could be computed by brute-force summation, but the time complexity is exponential. Another challenge here is that the graphical structure of our model may be arbitrary and contain cycles. To solve these issues, we adopt an approximate algorithm *Loopy Belief Propagation (LBP)* [21].

$$m_{as}(c_s) = \sum_{\boldsymbol{c}_a \setminus c_s} \Psi_a(\boldsymbol{c}_a) \prod_{t \in a \setminus s} m_{ta}(c_t) \tag{6}$$

$$m_{sa}(c_s) = \prod_{b \in N(s) \setminus a} m_{bs}(c_s) \tag{7}$$

The intuition behind LBP is that each of the neighboring factors of a given random variable would make a contribution (i.e. message) to its marginal, these messages can be iteratively updated by a propagation algorithm as shown in Equation 6 and Equation 7, where $N(s)$ denotes the adjacent factors of C_s, m_{as} denotes the message from factor Ψ_a to variable C_s and m_{sa} denotes the message in a reverse order. The approximate marginal $\Pr(c_s)$ is proportional to the product of all the incoming messages to variable C_s:

$$\Pr(c_s) \propto \prod_{a \in N(s)} m_{as}(c_s) \tag{8}$$

Learning. According to the previous definition, the log likelihood l of our model can be described as follows:

$$l(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}) = \log \Pr(Y^L|\boldsymbol{X}, \tilde{\boldsymbol{X}}; \boldsymbol{\theta}) = \log \mathbb{Z}(Y^L, \boldsymbol{X}, \tilde{\boldsymbol{X}}) - \log \mathbb{Z}(\boldsymbol{X}, \tilde{\boldsymbol{X}}) \tag{9}$$

We optimize the above objective function to estimate model parameters $\{\alpha, \beta, \gamma\}$. Unfortunately, Equation 9 is intractable as it is difficult for the exact computation of the partition function $\mathbb{Z}$. In practice, we train the model approximately. By employing *Bethe Approximation* [34], the negative log of partition function $\mathbb{Z}$ can be approximated by minimum of *Bethe free energy*:

$$O_{\text{BETHE}}(q) = -\mathcal{H}_{\text{BETHE}}(q) - \sum_a \sum_{\boldsymbol{c}_a} q(\boldsymbol{c}_a) \log \Psi_a(\boldsymbol{c}_a) \tag{10}$$

where q is a set of approximate marginal distributions generated by *LBP*, and $\mathcal{H}_{\text{BETHE}}(q)$ is *Helmholtz free energy*, which can be written as follows:

$$\mathcal{H}_{\text{BETHE}}(q) = -\sum_a \sum_{\boldsymbol{c}_a} q(\boldsymbol{c}_a) \log q(\boldsymbol{c}_a) + \sum_i \sum_{c_i} (d_i - 1) q(c_i) \log q(c_i) \tag{11}$$

Let O_{BETHE} and $\hat{O}_{\text{BETHE}}$ represent the *Bethe free energy* of two graphical models: one excludes the observed values in Y and is

Algorithm 1: Learning algorithm of the proposed model.

Data: A mobile network G, two fully observed user attribute matrices $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$, a partially labeled fraudulent borrower identity vector $\boldsymbol{Y}$, an unlabeled cheating agent identity vector $\boldsymbol{Z}$, and the learning rate λ.
Result: Estimated parameter $\boldsymbol{\theta}$, convergent q, $\hat{q}$

Initialization θ and q, $\hat{q}$ randomly;
while *not converge* **do**
  repeat
    Perform Equation 6, 7 and 8 in graphical model, where only $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$ are observed;
  until q *converge*;
  repeat
    Perform Equation 6, 7 and 8 in graphical model, where $\boldsymbol{X}$, $\tilde{\boldsymbol{X}}$, and $\boldsymbol{Y}^L$ are observed;
  until $\hat{q}$ *converge*;
  Calculate $\frac{\partial l}{\partial \theta}$ by Equation 13;
  Update $\theta_{new} = \theta_{old} + \lambda * \frac{\partial l}{\partial \theta}$ by equation 13;
end

only given by X and $\tilde{X}$; and another one regards X, $\tilde{X}$ and Y^L as observed. We then further yield the objective function:

$$l(\boldsymbol{\theta}) \approx l'(\boldsymbol{\theta}, q, \hat{q}) = \min_{q} O_{BETHE}(q) - \min_{\hat{q}} \hat{O}_{BETHE}(\hat{q}) \tag{12}$$

The parameter learning procedure can be viewed as a coordinate ascent. More specifically, we run LBP for two graphical models to get optimal q and $\hat{q}$ with $\boldsymbol{\theta}$ fixed, and then take gradient decent to partially maximize $l'(\boldsymbol{\theta}, q, \hat{q})$, which take the form as

$$\frac{\partial l}{\partial \theta} \approx \frac{\partial}{\partial \theta} \sum_{a} \sum_{\boldsymbol{c}_a} (\hat{q}(\boldsymbol{c}_a) - q(\boldsymbol{c}_a)) \log \Psi_a(\boldsymbol{c}_a) \tag{13}$$

See details of our learning procedure in Algorithm 1.

Time complexity It takes $O(T|E|)$ to perform LBP in our algorithm, where $|E|$ is the number of edges in the given mobile network, and T is the number of iterations of LBP. The gradient computation takes $O(|E|+|V|)$, where $|V|$ is the number of variables in our model. Thus in turn, our model has a time complexity of $O(RT|E|)$, where R is the number of iterations. Empirically, LBP converges quickly in our dataset (i.e. $T \approx 8$), and R is around 350.

5 EXPERIMENTS

In this section, we present the results from a series of experiments to evaluate the effectiveness of our proposed method. All the experiment are implemented in Python 2.7.6 on a 1.2GHz Intel Cores server with 56 CPUs and 396GM RAM, running Ubuntu 14.04.5.

5.1 Experimental Setup

Dataset. To conduct experiments and validate the effectiveness of our model, we sample a network with around from the dataset we introduced in Section 2.1. In particular, we perform random walk on the complete mobile network, and in turn obtain a graph G with 205,824 vertexes, 1,252,741 edges between them, and involved with

Table 1: List of features correlated with fraudulent borrowers and used in Ψ^F.

Feature	Description
demographics	Age and gender of v_i.
education level	Educational level of v_i.
indegree & outdegree	The number of v_i's neighbors that have made calls to(from) v_i.
fraudulent degree	The number of v_i's fraudulent neighbors before v_i applying to loan.
#2nd-degree neighbors	The number of users who have common neighbor with v_i.
#2nd-degree fraudsters	The number of fraudsters who have common neighbor with v_i before v_i applying to loan.
fraudulent traffic	$\max_{j \in N(i)} \sum_{k \in N(j) \setminus i} \mathbb{1}_{\{y_k=1\}}$, The maximal fraudulent degree of v_i's neighbors before v_i applying to loan.
clustering coefficient	$\frac{\lvert e_{jk}: v_j, v_k \in V, e_{jk} \in E \rvert}{d_v(d_{v_i}-1)}$, where v_j and v_k are v_i's neighbors, and d_{v_i} is v_i's degree.
PageRank	The PageRank value of v_i in graph.
#new contacts	The number of new contacts that user v_i contact within a week(day) before loan.
#calls before loan	The number of phone calls that user v_i make within a week(day) before loan.
peak of call	The maximal number of phone calls that user v_i make within a week(day).
contacts similarity	The cosine similarity of v_i contacts vector before and within a week.

37,454,890 call logs. Among all users, we have 20,010 fraudulent borrowers and 185,814 normal users (around 1 : 9.3) . Notice that the ratio of frauds here is slightly higher than that in the complete dataset, as our sampling strategy aims to provide a relatively complete mobile network. There are 594 cheating agent labels, which are only used as the ground truth for test.

Given the mobile network G and an identity vector Y, the task in our experiment is to determine the unknown values in Y (i.e. Y^U). We conduct 5-fold cross validation to train and test with Precision, Recall, F1-score and AUC as metrics for evaluation.

Baselines. We consider the following comparative methods in our experiment:

- *Logistic Regression(LR):* We apply logistic regression which use all features listed in Table 1 to train a classification model, and determine whether a specific user is a fraudster or not.
- OddBall: It is a fast and unsupervised method[3] to detect anomalous nodes in weighted graph. In practice, we construct a undirected graph where each vertex correspond to a user. We create a weighted link between two users if there exist any call log between them, and the weighted value is equal to the number of call logs.
- HITS: Due to the correlation between fraudsters and agents that we analyzed in Section 3, we apply HITS algorithm in the graph that is same as what we introduced in OddBall method. We use

Table 2: List of features correlated with cheating agents and used in Ψ^A.

Feature	Description
age diversity	variance of the distribution of ages that v_i's neighbors belong.
degree	The number of v_i's neighbors that have made calls to or from v_i.
clustering coefficient	$\frac{\|e_{jk}:v_j,v_k \in V, e_{jk} \in E\|}{d_v(d_{v_i}-1)}$, where v_j and v_k are v_i's neighbors, and d_{v_i} is v_i's degree.
degree growth	The increasing rate of v_i's degree in dynamic graph.
entropy	entropy of the number of phone calls over different neighbors of v_i

Table 3: Performance of detecting fraudsters.

Method	Precision	Recall	F1	AUC
LR	0.187	0.549	0.279	0.710
HITS	0.114	0.591	0.191	0.561
OddBall	0.120	0.587	0.199	0.575
DeepWalk	0.126	0.417	0.194	0.567
DTF	**0.215**	**0.580**	**0.317**	**0.757**

the authority value of each user to determine whether she is a fraudster or not.

- DeepWalk: It uses local information obtained from random walks in communication network to learn the latent representation vector for each user [25]. We use these vectors as features to train a logistic regression to classify users.
- DTF: It is our proposed model. We empirically set the parameter $\lambda = 0.1$ and $|Y^L|/|Y^U| = 7/3$. To be mentioned, because we do not introduce any ground truth about cheating agents in our model, we empirically fix the parameter in Ψ^E as $\begin{bmatrix} 1.00 & 0.73 \\ 0.97 & 1.20 \end{bmatrix}$. In this setting, we enhance adaptability of $y = 1, z = 1$ and reduce the adaptability of $y = 1, z = 0$ and $y = 0, z = 1$ due to previous observation. This manual adjustment can be think of a prior to our model. Notice that after accumulating sufficient labels for cheating agents, Ψ^E can be estimated automatically according to the learning algorithm in Section 4.

5.2 Identifying Fraudulent Borrowers

Table 3 lists performances of all comparative methods. Overall, our method outperforms all baselines in terms of F1-score and AUC (e.g., +50.6% in terms of F1). We also test the significance of this result to further confirm the improvement of our method ($p \ll 10^{-9}$).

Due to lack of supervised information, HITS and OddBall perform worse than our model.

HITS, OddBall, and DeepWalk mainly consider network structural characteristics of fraudulent borrowers. Among them, HITS only measures vertex importance and performs worse than others. OddBall only uses neighbors-related features but do not explore 2nd-degree neighbor's properties, which are considered useful according to our previous analysis in Section 3. DeepWalk aims to learn sufficient structural features automatically from the given mobile network. The significant difference between its performance with that of ours suggests that non-structure features like calling behavior are further required in our task.

LR considers both structural and behavior features just like our method does. Comparing with LR, our model (DTF) yields an improvement of 12.5% on F1-score and 6.5% on AUC. The major difference between these two methods is that DTF jointly models fraudulent borrowers and cheating agents, the latter in turn helps to improve the performance of identifying fraudulent borrowers.

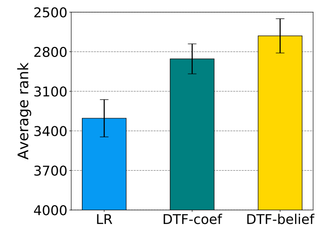


Figure 7: Performance of detecting cheating agents.

5.3 Identifying Cheating Agents

The major challenge for detecting cheating agents is that the ground truth data (or the label) is extremely difficult to obtain. In practice, staff of PPDai will call people suspected to be cheating agents, pretend to be a potential client, see if the other side will commit as a cheating agent, and collect the labels. The above time-consuming process is the only way for obtaining labels. Fortunately, PPDai kindly provides 594 labels obtained in such way, based on which we design two experiments to examine the effectiveness of the DTF model we proposed in the cheating agent detection task.

Feature effectiveness. In the first experiment, to verify the effectiveness of our features (Table 2) in agent detection task, we utilize a linear model that adopts these features to score each user. Then we fine tune the weight in this linear model by evaluating the presence of labeled agent among the top 1000 scored users. We report 100 suspicious users to PPDai through this way, and they evaluate the results by calling these suspicious. Eventually, 50 calls successfully get through, and the very preliminary method with the features we discovered hits 18 cheating agents (36%), achieving an over 2 times improvement compared with PPDai's previous strategy.

In spite of the performance improvement, exhausted searching parameters is inadvisable. We then perform another experiment to demonstrate the ability of our model in this task.

Comparison results. In order to make comparison, we apply logistic regression (*LR*) as our baseline that uses features described in Table 2. More specifically, we include all positive instances and randomly sample 20000 data from remainders as negative instances. We further separate these data into training/test set with a ratio of 7:3, use the trained LR to score and rank the data in test set. Please notice that as the proportion of cheating agents is very low, the sampled negative instances are trustful.

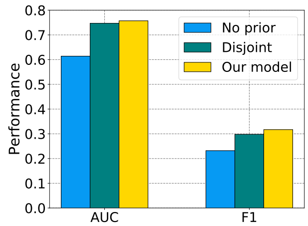


Figure 8: Performance of our model on detecting fraudulent borrowers with different factors.

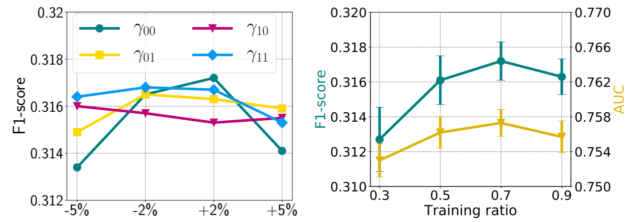


Figure 9: Performance of our model on detecting fraudulent borrowers under different settings.

For our model, we use *beliefs* of Z, which indicates the cheating agent label in our model, as final scores (*DTF-belief*). Additionally, we alter coefficients of trained LR to the coefficients we obtained from Ψ^A as another comparative method (*DTF-coef*).

We evaluate the result by sorting scored users in descending order and then calculating the average rank of labeled agents. The smaller average rank stands for a better performance. Figure 7 examines the performances of agent detection under these different approaches, we can see that our model (DTF-belief) yields the best result where the average rank of agents have a drop of 18.9%. In addition, DTF-coef can also outperform LR significantly (i.e. $p \ll 0.01$) by using a different set of coefficients obtained from Ψ^A.

To be mentioned, in the learning and inference phases of our model that we introduced in Section 4, we did not involve any label of agent identity and did not even tell our model the physical meaning of Z. Instead, the model can infer it and capture agent identity by bridging and utilizing the supervised information of fraudulent borrowers and the correlation between Y and Z.

5.4 Model Structure Analysis

To validate the necessity of the local structure and training phase of our model, we further design two experiments. In the first experiment ("*No prior*" in Figure 8), we remove all individual factors (i.e. Ψ^F and Ψ^A) from our model and only preserve Ψ^E, which models the correlation between Y and Z, to demonstrate the necessity of user features (i.e., user attributes, calling behavior, and social network structural features). In the second experiment ("*Disjoint*" in Figure 8), we aim to examine if the idea of bridging fraudulent borrowers and cheating agents contributes in our model. In particular, we first use logistic regression to train Ψ^F and Ψ^A respectively and merge these two parts into our model without further training. It is worth noting that we use negative sampling to create negative labels of non-agent while we training the parameter of Ψ^A.

Figure 8 shows the results of above two experiments. We can see that removing individual factors (i.e. Ψ^F and Ψ^A) from our model causes a 26.1% drop on F1-score and a 18.4% drop on AUC. This result, along with the result of LR, demonstrates the fact that personal attributes and structural information are both essential to the final performance. In addition, DTF yields an improvement of 5.4% on F1-score and 1.3% on AUC compared with the Disjoint model (i.e. train Ψ^F and Ψ^A separately). It reveals the capability that DTF can better understand the correlation between two kinds of identities by the training step.

5.5 Parameter Sensitivity

We finally examine how the model parameters influence its performance. We conducted experiment on fraudulent borrower prediction task with all other parameters fixed except Ψ^E. More specifically, we gradually increase or decrease a single parameter of Ψ^E in the range of 5% and check its effect. From Figure 9(a), we find that the performance is basically stable varying γ_{kl}, which reflects the robustness of our model. We also test the performance of DTF under different ratio of training instances in Figure 9(b). Initially, increasing the ratio has some effect in the results, but this effect quickly fade away when the ratio exceeds 0.5.

6 RELATED WORK

In this section, we briefly review the various methods that proposed for fraud detection which is widely applied in many fields [5, 28, 32].

Loan fraud detection is most relevant to our work. Many researchers have formulated this task as a typical classification problem. Individuals are classified into default and nondefault groups based on the observed attributes, which are historical loan requests and income information in most cases [4, 7, 16, 18, 31]. For example, Byanjankar et al. [7] proposes a credit scoring model using artificial neural networks. Different from existing work, in this paper, we propose a framework to identify frauds by employing social network information of users.

Since frauds usually behave differently from others, outlier detection [11, 20, 24] and anomaly user detection [2, 29] methodologies can also be adopted. Based on the insights that fraudsters may be reflected by the relationships between objects, some work utilize relational classification methods [17, 19]. For example, Akoglu et al. [1] proposed a framework to spot fraudsters and fake reviews in online review datasets. Another type of works use decomposition-based algorithm [6, 14, 23, 27, 30]. For example, Sun et al. [30] propose a method that use a low-rank approximation to evaluate the level of anomalous. Hooi et al. [13] propose a camouflage-resistant method to detect fraudsters in a bipartite graph and provided its upper bounds on the effectiveness. Most of the existing work are either supervised or unsupervised. In this paper, we discover a group of special identities(i.e. cheating agents) and develop a semi-supervised framework to detect fraudulent borrowers and cheating agents simultaneously.

7 DEPLOYMENT

The proposed model is deployed as an important part of PPDai's anti-fraud system, where the data and features are mainly supported by a Hadoop platform (150 servers, each with a CPUs, 256 GB RAM) with scientific computation empowered by Spark. This system keeps automatically pushing suspicious cases to staff for manual investigation and recording the investigation results as labels for future model improvement.

In a real application, a model is deployed whenever it could produce values. In our case, the separate cheating agent model (Ψ^A) and fraudulent borrower model (Ψ^F) are both in the stage of deployment during the development of the DTF model. More specifically, the cheating agent model has been working online to help PPDai identify cheating agents much more efficiently. There are several challenges for the deployment of fraudulent borrower model, mostly due to some time consuming features like PageRank on huge user population (around 10 million). We handle this issue by updating such features for the whole network in an incremental way instead of a re-calculation. By jointly learning the two parts, the proposed DTF model brings additional improvement as mentioned and is planned to be deployed next step.

8 CONCLUSIONS

In this paper, we study the problem of identifying fraudsters in P2P lending platforms by employing social network information. Based on a real-world dataset provided by PPDai with over 1.5 billion call logs between more than 11 million users, we conduct several exploratory analysis. We demonstrate several different characteristics between normal users and fraudulent borrowers. Moreover, we unearth a special type of users, named as cheating agents, from the network. Based on our observations, we propose a novel probabilistic framework to uniformly model fraudulent borrowers and cheating agents. We further formulate prediction tasks to validate the effectiveness of our model. Experimental results show that our model outperforms several baselines. Furthermore, our model can effectively identify cheating agents without any supervised information. by bridging the information of fraudulent borrowers and cheating agents.

REFERENCES

[1] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. 2013. Opinion Fraud Detection in Online Reviews by Network Effects. *ICWSM* 13 (2013), 2–11.

[2] Leman Akoglu and Christos Faloutsos. 2010. Event detection in time series of mobile communication graphs. In *Army science conference*. 77–79.

[3] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. 2010. Oddball: Spotting anomalies in weighted graphs. In *PAKDD'10*. Springer, 410–421.

[4] Arash Bahrammirzaee. 2010. A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications* 19, 8 (2010), 1165–1195.

[5] Richard J Bolton and David J Hand. 2002. Statistical fraud detection: A review. *Statistical science* (2002), 235–249.

[6] Horst Bunke, Peter J Dickinson, Miro Kraetzl, and Walter D Wallis. 2007. *A graph-theoretic approach to enterprise network dynamics*. Vol. 24. Springer Science & Business Media.

[7] Ajay Byanjankar, Markku Heikkila, and Jozsef Mezei. 2015. Predicting Credit Risk in Peer-to-Peer Lending: A Neural Network Approach. *IEEE Symposium Series on Computational Intelligence* (2015).

[8] Andrew T Carswell and Douglas C Bachtel. 2009. Mortgage fraud: A risk factor analysis of affected communities. *Crime, law and social change* 52, 4 (2009), 347–364.

[9] Chuang-Cheng Chiu and Chieh-Yuan Tsai. 2004. A web services-based collaborative scheme for credit card fraud detection. In *e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on*. IEEE, 177–181.

[10] Yuxiao Dong, Yang Yang, Jie Tang, Yang Yang, and Nitesh V. Chawla. 2014. Inferring user demographics and social strategies in mobile social networks. In *KDD'14*. 15–24.

[11] Jing Gao, Feng Liang, Wei Fan, Chi Wang, Yizhou Sun, and Jiawei Han. 2010. On community outliers and their efficient detection in information networks. In *KDD'10*. ACM, 813–822.

[12] John M Hammersley and Peter Clifford. 1971. Markov fields on finite graphs and lattices. *Unpublished manuscript* (1971).

[13] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. 2016. FRAUDAR: Bounding Graph Fraud in the Face of Camouflage. In *KDD'16*. 895–904.

[14] Tsuyoshi Idé and Hisashi Kashima. 2004. Eigenspace-based anomaly detection in computer systems. In *KDD'04*. ACM, 440–449.

[15] Frank R Kschischang, Brendan J Frey, and H-A Loeliger. 2001. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on* 47, 2 (2001), 498–519.

[16] Tian-Shyug Lee and I-Fei Chen. 2005. A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications* 28, 4 (2005), 743–752.

[17] Zhongmou Li, Hui Xiong, Yanchi Liu, and Aoying Zhou. 2010. Detecting blackhole and volcano patterns in directed networks. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 294–303.

[18] Rashmi Malhotra and Davinder K Malhotra. 2003. Evaluating consumer loans using neural networks. *Omega* 31, 2 (2003), 83–96.

[19] Mary McGlohon, Stephen Bay, Markus G Anderle, David M Steier, and Christos Faloutsos. 2009. Snare: a link analytic system for graph labeling and risk detection. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1265–1274.

[20] Emmanuel Muller, Patricia Iglesias Sánchez, Yvonne Mulle, and Klemens Bohm. 2013. Ranking outlier nodes in subspaces of attributed graphs. In *Data Engineering Workshops (ICDEW), 2013 IEEE 29th International Conference on*. IEEE, 216–222.

[21] Kevin P Murphy, Yair Weiss, and Michael I Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *UAI'99*. 467–475.

[22] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report SIDL-WP-1999-0120. Stanford University.

[23] Mitchell A Peabody. 2002. *Finding groups of graphs in databases*. Ph.D. Dissertation. Drexel University.

[24] Bryan Perozzi, Leman Akoglu, Patricia Iglesias Sánchez, and Emmanuel Müller. 2014. Focused clustering and outlier detection in large attributed graphs. In *KDD'14*. ACM, 1346–1355.

[25] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD'14*. 701–710.

[26] S Benson Edwin Raj and A Annie Portia. 2011. Analysis on credit card fraud detection methods. In *Computer, Communication and Electrical Technology (ICCCET), 2011 International Conference on*. 152–156.

[27] Peter Shoubridge, Miro Kraetzl, WAL Wallis, and Horst Bunke. 2002. Detection of abnormal change in a time series of graphs. *Journal of Interconnection Networks* 3, 01n02 (2002), 85–101.

[28] H Lookman Sithic and T Balasubramanian. 2013. Survey of insurance fraud detection using data mining techniques. *arXiv preprint arXiv:1309.0806* (2013).

[29] Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip S Yu. 2007. Graphscope: parameter-free mining of large time-evolving graphs. In *KDD'07*. ACM, 687–696.

[30] Jimeng Sun, Yinglian Xie, Hui Zhang, and Christos Faloutsos. 2008. Less is more: Sparse graph mining with compact matrix decomposition. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 1, 1 (2008), 6–22.

[31] Zhe Sun, Marco A Wiering, and Nicolai Petkov. 2014. Classification system for mortgage arrear management. In *Computational Intelligence for Financial Engineering & Economics (CIFEr), 2104 IEEE Conference on*. 489–496.

[32] Shiguo Wang. 2010. A comprehensive survey of data mining-based accounting-fraud detection research. In *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, Vol. 1. IEEE, 50–53.

[33] Yang Yang, Chenhao Tan, Zongtao Liu, Fei Wu, and Yueting Zhuang. 2018. Urban Dreams of Migrants: A Case Study of Migrant Integration in Shanghai. In *AAAI'18*.

[34] Jonathan S Yedidia, William T Freeman, and Yair Weiss. 2003. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium* 8 (2003), 236–239.