User-based Network Embedding for Collective Opinion Spammer Detection

Ziyang Wang\textsuperscript{a}, Wei Wei\textsuperscript{a, ∗}, Xian-Ling Mao\textsuperscript{b}, Guibing Guo\textsuperscript{c}, Pan Zhou\textsuperscript{a}, Shanshan Feng\textsuperscript{d}

\textsuperscript{a}Cognitive Computing and Intelligent Information Processing (CCHIP) Laboratory, School of Computer Science and Technology, Huazhong University of Science and Technology, China
\textsuperscript{b}School of Computer Science and Technology, Beijing Institute of Technology, China
\textsuperscript{c}Software College, Northeastern University, China
\textsuperscript{d}Inception Institute of Artificial Intelligence, UAE

Abstract

Due to the huge commercial interests behind online reviews, a tremendous amount of spammers manufacture spam reviews for product reputation manipulation. To further enhance the influence of spam reviews, spammers often collaboratively post spam reviewers within a short period of time, the activities of whom are called collective opinion spam campaign. As the goals and members of the spam campaign activities change frequently, and some spammers also imitate normal purchases to conceal identity, which makes the spammer detection challenging. In this paper, we propose an unsupervised network embedding-based approach to jointly exploiting different types of relations, \textit{e.g.}, direct common behaviour relation and indirect co-reviewed relation to effectively represent the relevances of users for detecting the collective opinion spammers.

The average improvements of our method over the state-of-the-art solutions on dataset AmazonCn and YelpHotel are \{14.09\%, 12.04\%\} and \{16.25\%, 12.78\%\} in terms of AP and AUC, respectively.

Keywords: Spam Detection, Collective Spammer, Network Embedding, Signed Network

\textsuperscript{∗}Corresponding author

\textit{Email address: weiw@hust.edu.cn} (Wei Wei)
1. Introduction

Online reviews are valuable source of reference for decision-making, and thus massive volume of spammers are attracted to post malicious generated reviews to promote/demote the target products. As opposed to individual spammers, collusive spammers are more harmful as they can easily propagate deceptive reviews for dominating the opinions of target products within a short period, and their activities are called collective opinion spam campaign. As such, it is highly valuable and desirable to develop an effective algorithm in accurately capturing collusion signals for collective opinion spammer detection. The review spam detection problem has attracted extensive attentions. Although these methods have greatly advanced the problem, they still easily fail due to the facts: Firstly, previous supervised-based methods formulate it as a classification task, whose performance often heavily rely on the scale of labeled data \[1, 2, 3, 4\]. Nevertheless, manual labeling spam reviews is extremely difficult \[5\]. Secondly, there exist several attempts for the spammer detection problem in an unsupervised fashion: (i) Previous linguistic-based methods typically have poor performance as the reviews are usually manipulated by spammers who would fine-tune their language \[6, 7\]; and (ii) Many behavior-based approaches focus on modeling highly-visible behaviors \[8, 9, 10, 11\] or temporal factors (e.g., co-bursting) \[12, 13, 14, 15, 16\] for detecting spammers. However, some spammers also make normal purchases to conceal identity, and thus such features are also not reliable for tackling the spammer detection problem.

Recently, exist several efforts have been dedicated to research on spammer detection by modeling the user-group-product relations over behavior features \[17, 18, 19\]. These efforts usually focus on extracting discriminative features (i.e., pointwise or pairwise features) from direct common behaviour relations (which are called direct relations). However, spammers are often required to balance workload within spam campaigns for evading detection, and thus it is insufficient to capture the collusion signals by modeling the direct relation alone. Intuitively, spammers are usually engaged repeatedly for different col-
lective opinion spam campaigns and colluders are likely to share more common
neighbors within the same spam-campaigns (even without direct collusion re-
lation), and thus the combination of the direct and indirect collusion relation
(which refers to indirect associations with multiple steps, e.g., k-steps, along
with the neighborhood structures of colluders, which are called indirect rela-
tion) is a relatively stationary collusion signal, which is not easily manually
manipulated.

To this end, in this work we propose a unsupervised network embedding-
based approach to learning the user embeddings by jointly exploiting the differ-
ent types of relations (i.e., direct relation and indirect relation) between pairwise
users for collective opinion spam detection.

The main contributions of our proposed method can be summarized as fol-
lows: i) To be our best knowledge, we are the first to jointly learning the
direct relevance and indirect relevance for collective spammers detection; ii) We
develop a signed network composed of positive and negative links, in which the
direct relevance measures the degree of being colluders in the signed network,
and the indirect relevance measures the potential possibility of being collud-
ers via a k-step neighborhood proximity over the signed network; iii) Extensive
experiments conducted on two real-world datasets demonstrate our proposed
method significantly outperforms the state-of-the-arts.

2. Related Work

We categorize the the existing work on the opinion spam detection into two
groups, i.e., linguistic-based, behavior-based.

Linguistic-based Spam Detection. Linguistic-based methods aim at ex-
tracting the discriminative linguistic features to differentiate the fake users from
normal ones. For example, these methods identify review spams according
to linguistic clue [5, 20, 11], writing-style feature [22], syntactic pattern [23],
LDA-based topic model [24], Bayesian generative model [17], positive-unlabeled
learning [24], frame-based model [25], and document-level features [2]. How-
ever, Mukherjee et al. [26] show that linguistic-based features are ineffective for spam detection problem. In contrast, our proposed model focuses on modeling spam-campaign characteristics (excluding textual features) within a unified unsupervised framework to explore the direct/indirect user-user relations of users to capture the collusion signals.

**Behavior-based Spam Detection.** Behavior-based spam detection aims at detecting a set of collective malicious manipulation of online reviews according to behavior-based features. Lim et al. [8] study the rating behavioral characteristics for review spammer detection. Jindal et al. [9] find unexpected rules to represent suspicious behaviors of reviews based on the unusual review patterns. Li et al. [10] propose a co-training method based on the extracted review-based/reviewer-based features for identifying spammers. Li et al. [4] propose to employ a coupled hidden Markov model to discover co-bursting behavior patterns.

Li et al. [27] focus on the cold-start problem by using user behavior representation. Several efforts [11, 12, 13, 14, 15, 16] have also dedicated to research on temporal factors for spam detection. However, some spammers also make normal purchases to conceal identity, which affects the effectiveness of these spammer detection methods. Hence, the approaches that heavily reply on shallow behavior features or temporal factors are insufficient for accurately identifying the spammers.

There have been numerous attempts to detect spammers by modeling the user-group-product relations over behavior features. Mukherjee et al. [26] rank collusion spam groups over user-group-product relations for detection. Ye et al. [6] estimate the likelihood of products being spam campaign target for inferring spammers. Rayana et al. [28] utilize a loopy belief propagation model to infer spammers by extracting relational features, which is then extended by introducing active inference [29]. Wang et al. [3] learn the user embeddings generated by tensor decomposition for training a spam review classifier. Liu et al. [30] mainly focus on graph topology and temporal information for detecting fraud
users. Xu et al. [31] propose a regularized matrix factorization model to obtain reviewer behavior embeddings. Kaghazgaran et al. [32] pre-train a spammer classifier by modeling structurally similar users within a three-phase framework. Kumar et al. [19] iteratively calculate three quality metrics for spammer detection. Wang et al. [33] design a loopy belief propagation based algorithm to detect spammer groups. Additionally, You et al. [18] develop a unified deep learning architecture to tackle the cold-start problem in spam review detection. However, the spam campaigns might balance workloads of spammers to evade detection. These existing methods use shallow relational information, i.e., directly connected users (called direct relation) while neglecting indirect relations (i.e., indirectly connected users, called indirect relation), and they easily fail for the spammer detection task. In contrast, we focus on modeling user behavioral information within a unified unsupervised architecture to explore the direct/indirect user-user relations among reviewers for detection.

3. Problem Statement and Notations

Let \( P = \{p_j\}_{|P|} \) be the set of items over a set of product categories \( C = \{c_i\}_{|C|} \); \( U = \{u_i\}_{|U|} \) and \( X = \{x_{ij}\}_{|X|} \) be the set of users and the records of their reviews, where \( x_{ij} \in X \) denotes the reviews posted by \( u_i \) on product \( p_j \), since a user \( u_i \) might post different reviews for the same product \( p_j \) owing to multiple purchases. As such, a 4-tuple \((u_i, p_j, r, t)\) denotes the review generated by user \( u_i \) for product \( p_j \) with the rating \( r \) at time \( t \), in which the rating and the time-stamp are denoted by \( x_{r_{ijk}} \) and \( x_{t_{ijk}} \) for simplicity.

Then, given a set of users and the metadata of their posted reviews, i.e., \((u_i, p_j, r, t)\), the problem of collective opinion spammer detection is defined as identifying a set of spammers based on their participated spam campaigns, namely, which is regarded as a task that requires a set of spammers to collectively post malicious opinions (i.e., human-powered deceptive contents) on the target items, and outputs a ranking list of candidates with the spamicity scores, which can be regarded as the likelihood of candidates participating in opinion spam
Figure 1: **Overview of proposed approach.** First, a signed network is built based on the extraction of pairwise features from users’ reviews, then we jointly optimize two types of relevances, *i.e.*, direct relevance and indirect relevance for learning the user embedding in a low-dimensional space. Finally, the spamicity score of each user is calculated based on the learnt user embeddings for estimating the degree of being colluders.

### 4. The Proposed Approach

In this section, we present our proposed collective opinion spammer detection approach within an unified architecture (shown in Figure 1), named COSD, the objective of which is to jointly combine direct and indirect neighborhood exploration for learning the embedding representation of each user for more accurately identifying spam reviewers. The rationale behind is that **direct relevance embedding** controls the learning of user embeddings towards the pairwise users with a strong intensity of the collusive characteristics, while **indirect relevance embedding** tends to make pairwise users sharing more commonly co-rating neighbors closer. Indeed, we are only interested in **direct relevance** embedding of users to identify spammers, however we also consider **indirect relevance** embedding in our framework because such two types of embeddings will reinforce each other mutually to make the relevant users close over direct and indirect relations of users (the details will be elaborated later on). Next, we detail how to model the **direct relevance embedding** and **indirect relevance embedding** for
any pairwise users.

4.1. Modeling Spam-Campaign Characteristics

In this section, we mainly focus on how to characterize spam-campaigns on different dimensions for building the user-user weighted matrix $W$ to capture the intersections on the co-rated item sequences for each pairwise users. Many previous works are proposed for this task [26, 34, 31, 3]. By following the work [34], we adopt four different types of heterogeneous pairwise features as follows. Note we do not consider linguistic-based features from reviews.

- **Product Rating Proximity (PR)**, it measures the intensity of spammers’ agreements for pairwise users. Generally, spammers within a spam-campaign are instructed to post similar opinions with consistent ratings on target items. As such, given a pair of users $(u_i, u_j)$, we measure the intensity of spammers’ agreements as follows,

$$
\psi_{PR}(i, j) = \frac{2}{1 + \exp(\Gamma_{p,r}^{ij})}
$$

where $\Gamma_{p,r}^{ij}$ denotes the average rating deviation of pairwise users $(u_i, u_j)$ over their commonly reviewed items, $\Gamma_{p,r}^{ij} = \frac{1}{|P_i \cap P_j|} \sum_{p_k \in P_i \cap P_j} \left( \frac{1}{|x|} \sum_{x \in p_k} x_{i,k} - \frac{1}{|x|} \sum_{x \in p_k} x_{j,k} \right)$,

where $P_{i(j)}$ refers to the set of items reviewed by user $u_{i(j)}$. Eq. (1) favors to find pairwise spammers who have more consistent ratings on co-rated items, especially $\psi_{PR}(i, j) = 1$ when $\Gamma_{p,r}^{ij} = 0$.

- **Product Time Proximity (PT)**, it captures the temporal consistency for pairwise users. Intuitively, colluders are often asked to complete the task within a short-time frame (e.g., less than a week) for maximizing the influence, and thus the temporal traces of their reviews tend to be more intensive than normal users’. Hence, we use PT to capture the temporal consistency of pairwise users $(u_i, u_j)$,

$$
\psi_{PT}(i, j) = \frac{1}{C + \gamma \Gamma_{p,t}^{ij}}
$$

where $\Gamma_{p,t}^{ij}$ denotes the average time deviation of pairwise users $(u_i, u_j)$ over their commonly reviewed items, $\Gamma_{p,t}^{ij} = \frac{1}{|P_i \cap P_j|} \sum_{p_k \in P_i \cap P_j} \left( |x| \sum_{x \in p_k} x_{i,k} - |x| \sum_{x \in p_k} x_{j,k} \right)$,
where $\Gamma_{ij}^{p,t}$ denotes the average time deviation of pairwise user $(u_i, u_j)$ over their co-reviewed items, 
\[
\Gamma_{ij}^{p,t} = \frac{1}{|P_i \cap P_j|} \sum_{p_k \in P_i \cap P_j} \left( \frac{\sum_q x_{ikq}^t}{|x_{ik}|} - \frac{\sum_q x_{jkq}^t}{|x_{jk}|} \right);
\]
$C$ and $\gamma$ denote the smoothing factor and the trade-off parameter, which are empirically set at 1 and 20, respectively.

**Category Rating Proximity (CR)**, it measures the average category rating deviation between pairwise users. Intuitively, spammers from different spam-campaigns might have different rating distributions over reviewed categories, and thus the higher intersections of category rating distributions of two users are consistent, the more likely these two users are colluders. Hence, CR is computed based on the average category rating deviation $\Gamma_{ij}^{c,r}$ between pairwise users,
\[
\psi_{CR}(i,j) = \frac{2}{1 + \exp(\Gamma_{ij}^{c,r})} \quad (3)
\]
where $\Gamma_{ij}^{c,r} = \frac{1}{|C_i \cap C_j|} \sum_{c_k \in C_i \cap C_j} \left( \overline{c_{ik}} - \overline{c_{jk}} \right)$, and $C_{i(j)}$ is the set of categories reviewed by user $u_{i(j)}$; $\overline{c_{ik}}$ denotes the average rating of user $u_i$ in the $k$-th category, which is calculated by $\overline{c_{ik}} = \frac{1}{|c_k|} \sum_{p_j \in c_k} \frac{1}{|x_{ij}|} \sum_q x_{ijq}^r$.

**Category Time Proximity (CT)**, it measures the consistency of time distributions between pairwise users over co-reviewed categories. Analogous to CR, it is also a strong indicator to measure the degree of the collusive characteristics of pairwise users, which is estimated by,
\[
\psi_{CT}(i,j) = \frac{1}{C + \gamma \Gamma_{ij}^{c,t}} \quad (4)
\]
where $C$ and $\gamma$ are empirically set at 1 and 20, respectively; $\Gamma_{ij}^{c,t} = \frac{2}{|C_i \cap C_j|} \sum_{c_k \in C_i \cap C_j} \left( \overline{c_{ik}^t} - \overline{c_{jk}^t} \right)$, and $\overline{c_{ik}^t} = \frac{1}{|c_k|} \sum_{p_j \in c_k} \frac{1}{|x_{ij}|} \sum_q x_{ijq}^{tq}$.

**Pairwise Feature Combination.** To estimate the intensity of spammer agreement for any pairwise users, we follow the work [34] to employ a convex combination of the mentioned pairwise proximities with a weighting vector $\alpha$ for calculating $h_{ij}$,
\[
h_{ij} = \sum_k \alpha_k \psi_{(.)}(i,j), \quad (5)
\]
where each feature $\psi(i,j)$ (i.e., PR, PT, CR, CT) is normalized within $[0, 1]$, and $\sum_k \alpha_k = 1$ ($\alpha_k \geq 0$), $k$ is the index for each pairwise feature.

To verify the effectiveness of $h_{ij}$, we present an empirical data analysis on Amazon.cn, where we mainly focus on three different types of pairwise user relations according to their co-reviewed relations, i.e., colluder to colluder (C-C), non-colluder to colluder (NC-C), and non-colluder to non-colluder (NC-NC). Figure 2 shows the distribution of the spammer-agreement ($h_{ij}$) to the number of pairwise users over such 3 relations. For the sake of discussion, we mainly focus on C-C and NC-NC. As the spammer agreement increases, the change ratio of C-C and NC-NC is different, i.e., the number of NC-NC considerably decrease occurred within the range of $[0.4, 0.8]$. As opposed to NC-NC, C-C gradually decrease within such range. In particular, the number of NC-NC almost equals to 0 when $h_{ij} \geq 0.8$, and the number of C-C almost equals to 0 when $h_{ij} \leq 0.22$. Similar trends are also reported in [34]. Based on the observations, we measure the collusive characteristics of each pairwise users.
as follows, which is different from [34],

\[ w_{ij} = (h_{ij} - \zeta) * \eta_{PI}(i, j), \tag{6} \]

where \( \zeta \) is a hyper-parameter that can be set in different ways, e.g., \( \zeta = \overline{h_{(i,j)}} \), as most of pairwise users belong to NC-NC and thus tend to be firstly filtered by our method; \( \eta_{PI}(i, j) = \frac{|P_i \cap P_j|}{\sqrt{|P_i|} \sqrt{|P_j|}} \) is a confidence score induced by the proportion of items commonly reviewed by \((u_i, u_j)\).

### 4.2. Modeling of Direct Relevance Embedding

In this section, we mainly focus on how to learn the direct embeddings of users over direct co-rating relations for making a pair of users with a strong intensity of being colluders closer, otherwise the ones with a weak intensity of ones far away. As such, we first give a definition related to the direct relevance of a pair of users.

**Definition 1. Direct Relevance**, which is used to measure the degree of the spammer agreements based on their direct co-rating associations for any given pairwise users \((u_i, u_j)\).

Sequentially, we define a user-based signed network built based on direct relevance of users for embedding.

**Definition 2. User-based Signed Network (USN)**. A user-based signed network [39] is defined by a 2-tuple, i.e., \( \mathcal{G} = (\mathcal{U}, \mathcal{E}) \), which consists of a set of users \( \mathcal{U} = \{u_i\}_{|\mathcal{U}|} \), as well as a set of positive links \( \mathcal{E}^+ \) and a set of negative links \( \mathcal{E}^- \), and \( \mathcal{E} = \mathcal{E}^+ \cup \mathcal{E}^- = \{e_{ij}\}_{|\mathcal{E}|} \).

In **USN**, both positive and negative links are represented into a weighted matrix \( \mathbf{W} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{U}|} \), where each element \( w_{ij} \in \mathbf{W} \) indicates the intensity of being colluders for a pair of users, especially \( w_{ij} = 0 \) denotes the missing link between \( u_i \) and \( u_j \). Then, the direct relevance can be estimated by employing a likelihood function to minimize the negative log-likelihood of collusion possibility for any pairwise users \((u_i, u_j)\).
\[
L_d = \min \left( -\sum_{e_{ij} \in E} \log f(u_i, u_j; w_{ij}) \right),
\]

where \(u_{i(j)}\) denotes the \(d\)-dimensional user embedding vector. \(f(\cdot; \cdot; \cdot)\) is a likelihood function, and many approaches can be used to model it, here we define the function based on the principle of ReLu [37, 38], and Eq. (7) can be rewritten as,

\[
L_d = \min \left( \sum_{e_{ij} \in E} \max \left(0, w_{ij}\|u_i - u_j\|_F^2 + \delta \right) \right),
\]

where \(\|\cdot\|_F\) is the Frobenius norm of a vector, \(\delta\) indicates a smoothing parameter. As mentioned, the learnt user embeddings are expected to make a pair of users with a strong intensity of being colluders closer, otherwise the ones with a weak intensity of ones far away. More specifically, we consider the following cases regarding \(w_{ij}\).

**Case 1.** \(w_{ij} > 0\), which indicates \(e_{ij} \in E^+\) and the learnt user embeddings should be more closer, otherwise it should be penalized with a larger loss.

**Case 2.** \(w_{ij} < 0\), which means \(e_{ij} \in E^-\) and is expected to keep the pairwise users away from each other in the learning space, otherwise it should be penalized, and a hyper-parameter is used to control the penalty, i.e., the value of which should be set as 0 when \(\|u_i - u_j\|_F^2 > \frac{\delta}{w_{ij}}\).

According to the discussion, Eq. (8) can be rewritten by combining the above two cases,

\[
L_d = \min \left( \sum_{e_{ij} \in E} I_{ij} w_{ij} \|u_i - u_j\|_F^2 \right),
\]

where \(I \in \mathbb{R}^{|U| \times |U|}\) is an indicator matrix with \(I_{ij} = 0\) if \(w_{ij} < 0\) \& \(\|u_i - u_j\|_F^2 > \frac{\delta}{w_{ij}}\) and 1 otherwise.

Then, the remaining problem is how to estimate the intensity of being colluders, as well as model the indirect relevance embedding for each pairwise users. We will detail them in the following sections, respectively.
4.3. Modeling of Indirect Relevance Embedding

As mentioned, spammers within the spam campaigns might balance their workload to evade detection, and thus explicitly modeling the direct relevance via shallow relational information (i.e., direct co-rating relation) alone is insufficient for accurately capturing the collusion signals, which might make matrix $W$ sparse and result in a poor performance in spammer detection, even worse when no direct correlations. Intuitively, users sharing more common neighbors may be a strong indicator that they are colluders, and thus it is a natural way to combine the direct relation and indirect relation, which is a relatively stationary collusion signal for collective review spammer detection, and is also not easily manual manipulation. Hence, here we mainly focus on how to measure the indirect relevance for user embedding, via modeling indirect common neighbors for a pair of users.

4.3.1. Indirect Relevance Proximity

To measure the indirect relevance for pairwise users, we employ a truncated random walk \cite{39} to model the indirect neighborhood network structure of such users over USN. Note we solely consider to walk along with the positive links (i.e., $E^+$) for modeling the collusive characteristics of each pairwise users, and which is thus called positive-based random walk. Specifically, we obtain $r$ sequences with the maximum length of $k$ steps for user $u_i$, and thus totally obtain $r \times |U|$ sequences ($S^+$), where the indirect neighbors of user $u_i$ is collected from sequences containing the neighborhood structure of $u_i$, in which the interval between the points and $u_i$ is within the window size $\omega$.

Then, we learn the indirect relevance embeddings over their common $k$-step neighbors. Hence, we adopt skip-gram \cite{40} to compute the loss function of indirect relevance, which is defined by the sum of the negative log probability of any pairwise users based on the learnt indirect relevance embeddings within a window size $\omega$. 

12
\[ \mathcal{L}_{id} = \min \left( \sum_{i=1}^{\left| \mathcal{U} \right|} \sum_{u_i \in \mathcal{U} \setminus \mathcal{S}} \left( \sum_{i=\omega<j<i+\omega} - \log \Pr(u_j|u_i) \right) \right). \]  

where \( \Pr(u_j|u_i) \) is the co-occurrence probability parameterized using the inner product kernel with softmax, which is defined by

\[ \Pr(u_j|u_i) = \frac{\exp(u_i^T \Phi_j)}{\sum_{k=1}^{\left| \mathcal{U} \right|} \exp(u_i^T \Phi_k)} \]  

where \( u_i \) denotes the learnt direct relevance embedding of \( u_i \); and \( \Phi_j \) is \( u_j \)'s indirect relevance embedding of \( u_j \) when \( u_j \) is the indirect neighbor of \( u_i \), which is based on the principle of making the users sharing more \( k \) neighbors close in the learnt vector space.

However, Eq. (11) cannot be scalable due to the expensive computation overheads, as we need to finish the updates of all users when computing \( \Pr(u_j|u_i) \). To tackle the problem, we employ negative sampling for optimization \([40]\), we formulate the negative sampling function for pairwise user \((u_i, u_j)\) when computing Eq. (11),

\[ \log \sigma(u_i^T \Phi_j) + \kappa \mathbb{E}_{u_n \sim P_n(u_i)} [\log \sigma(-u_i^T \Phi_n)] \]  

where \( \sigma(x) = \frac{1}{1+\exp(-x)} \) is the sigmoid function; \( \kappa \) is the number of negative samples. We empirically set \( P_n(u_i) \propto d_{u_i}^{3/4} \) as \([40]\), and \( d_{u_i} \) is the degree of user \( u_i \).

4.4. The Unified Model

In this section, we present an unified model to optimize our collective opinion spammer detection problem by minimizing a combination loss which consists of three different loss terms, i.e., two losses based on the two learnt relevance embeddings, i.e., direct relevance embeddings and indirect relevance embeddings, as well as a regularization loss, which is formalized as,
\[ L_{\text{mix}} = (1 - \beta)L_{\text{id}} + \beta L_{\text{d}} + \psi L_{\text{reg}} \]

\[ = -(1 - \beta) \left( \sum_{i=1}^{U} \sum_{u_i, u_j \in S} \left( \sum_{i - \omega < j < i + \omega} \sum_{k=1}^{V} \frac{\exp(u_i^T \Phi_j)}{\sum_{k=1}^{V} \exp(u_i^T \Phi_k)} \right) \right) \]

\[ + \beta \left( \sum_{e_{ij} \in E} I_{ij} w_{ij} \| u_i - u_j \|_F^2 \right) + \psi (\| U \|_F^2) \]

where \( \beta \) is a trade-off parameter for controlling the contributions of the learnt direct embeddings and indirect embeddings; \( \| \cdot \|_F^2 \) is the Frobenius norm of a matrix; and \( \psi \) is a regularization parameter. Note that, for each user \( u \), the direct user embedding \( u \) and the indirect user embedding \( \Phi \) are simultaneously learned during training.

**Spamnicity Score.**

The calculation of the final spamnicity score relies on the assumption that collective opinion spammer is based on jointly participated in a spam campaign via large-scale manipulation, and thus users within a same campaign group are more likely to have similar behaviors. To this end, we employ the Frobenius distance (similar to [31]) i.e., \( \text{Score}_F(u_i, u_j) = \exp(-\| u_i - u_j \|_F^2) \) to measure the intensity of a pair of users \( (u_i, u_j) \) being colluders. The higher the \( \text{Score}_F(u_i, u_j) \), the more likely the two users have collusion. For each user \( u_i \), we first calculate the \( \text{Score}_F \) between user \( u_i \) and all other users, and then the spamnicity score of user \( u_i \) is computed by averaging the sum of Frobenius-distance based (FD) scores of \( n \) users, whose FD scores are ranked in the top-\( n \) list of \( \text{Score}_F(u_i, \ast) \).

**4.5. Optimization**

In this section, we present the solution to the optimization problem stated in Eq. [13], which is optimized as follows,

**Optimize \( L_{\text{d}} \).** We first focus on the loss function of \( L_{\text{d}} \) and it can be rephrased as follow:
\[
L_d = \sum_{(i,j) \in E} I_{ij} w_{ij} \| u_i - u_j \|_2^2
\]

\[= 2 \text{tr}(U^T L U) \tag{14}\]

where \(L = D - \tilde{W}\), and \(\tilde{W} = I \odot W\), \(D \in \mathbb{R}^{|U| \times |U|}\) is a diagonal matrix and \(D_{i,i} = \sum_j \tilde{W}_{i,j}\), \(W\) is the weighted user-user matrix. For each iteration, we first update \(I\) and then update \(L\).

According to the Eq.\((14)\), we can utilize SGD to minimize \(L_d\), and the partial derivatives of \(L_d\) with respect to \(U\) can be computed by,

\[
\frac{\partial L_{\text{low}}}{\partial U} = 2(L + L^T) \cdot U \tag{15}\]

**Optimize \(L_{id}\).** Here, we first calculate the partial derivative of \(L_{id}\) with respect to user \(u_i\), namely,

\[
\frac{\partial L_{id}}{\partial u_i} = - \sum_{z \in u_i \cup \kappa_{N}(u_i)} (O(z, u_i) - \sigma(u_i^T \Phi_z)) \Phi_z \tag{16}\]

where \(O(z, u_i)\) is an indicator function, \(O(z, u_i) = 1\) if \(z\) is the neighbor of \(u_i\) and 0 otherwise; \(N^\kappa(u_i)\) is \(\kappa\) negative samples for \(u_i\). Then the partial derivative of \(L_{id}\) with respect to user \(u_i\) is updated by

\[
\frac{\partial L_{id}}{\partial \Phi_z} = - \sum_{z \in u_i \cup \kappa_{N}(u_i)} (O(z, u_i) - \sigma(u_i^T \Phi_z)) u_i \tag{17}\]

Subsequently, the regularization term can be updated by Eq. \((18)\),

\[
\frac{\partial L_{\text{high}}}{\partial U} = 2U. \tag{18}\]

For ease of reference, the learning process of our proposed method COSD is summarized in Algorithm 1. Specifically, it first builds the user-based signed network and generates the walk sequences using Eq. \((6)\) and Eq. \((7)\), and then iteratively update the user embedding matrix \(U\) and \(\Phi\) via Eq.\((15)\)-Eq.\((18)\). Finally, the spamicity scores are calculated via Frobenius Distance and output the spamicity score list \(S\) and the user embedding matrix \(U\).
Algorithm 1: The Learning Process of COSD

Input: $\forall < u_i, p_j, r_{ij}, t_{ij}, c_j >$

Output: $U$, $S$: User embedding matrix and spamicity scores list

1. Construct Signed Network based on Eq. (6) (7)
2. Obtain walk sequences by RandomWalk.
3. while not converged do
   4. Compute $\nabla L_{\text{low}}$ based on Eq. (15)
   5. Compute $\nabla L_{\text{reg}}$ based on Eq. (18)
   6. for $u_i \in U$ do
      7. Sample $\kappa$ negative nodes
      8. Compute $\nabla L_{\text{high}}$ based on Eq. (16) (17)
   9. end
10. Update $U$ and $\Phi$
11. end
12. for $u_i \in U$ do
    13. for $u_j \in U$ do
    14. $r_{i,j} = \exp(-\|u_i - u_j\|^2_F)$
    15. Insert $r_{i,j}$ into $R_i$
    16. end
    17. Accumulate top $k$ maximum values in $R_i$ to $s_i$
    18. Insert $(s_i, u_i)$ into $S$
19. end
20. RETURN $U$, $S$

Time Complexity. The procedure of our proposed method consists of the following sub-components, the computation costs of which are, (i) The construction of the user-based signed network ($O(|U|^2)$); (ii) The iteration of updating the user embedding matrix, $O(I \times |U| \times \kappa \times K)$, where $I$ is the number of iterations, $\kappa$ is the number of negative sampling, $K$ is the dimension of the representation vector; and (iii) The calculation of the spamicity scores list ($O(|U|^2 \times \text{log}(|U|))$).
Therefore, the time complexity of our approach is $O(I \times |U| \times \kappa \times K)$, as $O(I \times \kappa \times K) > O(|U| \times \log(|U|))$.

5. EXPERIMENTS

5.1. Data Sets

In this section, we use two public real-world datasets for evaluation, i.e., AmazonCn, YelpHotel, as compared to the state-of-the-art methods. The statistics of the datasets are summarized in Table 1: (i) AmazonCn is a collection of real consumers’ reviews from Amazon.cn, which already has gold standard of collective spammers whose reviews are filtered by AmazonCn, and then gather those tagged users with their co-rating behaviors; and (ii) YelpHotel contains real reviews about hotel on Yelp.com, which is collected by Yelp’s own filtering mechanism.

Note that there are not labels of collective spammer in YelpHotel, and thus we follow the work to form spammer groups via mining user common behaviors with Frequent Itemset Mining (FIM).

5.2. Evaluation Metrics

As mentioned, the output of collective opinion spammer detection problem is a ranking list of candidates with the spamicity scores, which can be deemed to the likelihood of candidates participating in opinion spam. As such, to evaluate the detection performance of different approaches, we follow the work in.
to adopt four well-known metrics, \textit{i.e.}, (i) \textbf{Average Precision (AP)}, which measures the average precision of collective spammer retrieving over the interval of $r$ (recall value) from 0 to 1; (ii) \textbf{Area Under ROC Curve (AUC)}, which measures the accuracy based on \textit{False Positive Ratio} (FPR) against \textit{True Positive Ratio} (TPR) for binary classification; (iii) \textbf{Precision@k (P@k)}, which measures the percentage of true spammers in the top-$K$ returned candidates; and (iv) \textbf{Normalized Discounted Cumulative Gain@k (NDCG@k)}, the performance of the detection model based on the ground-truth (\textit{i.e.}, spammer (1)/non-spammer (0)) of the returned spammers, which is the normalization of Discounted Cumulative Gain (DCG) at each position for a chosen value of $k$.

5.3. Baseline Methods

We compare our model with the following state-of-the-art spam detection methods:

\textbf{GsRank} \cite{26}: This method employs a ranking model for spotting spammer groups by extracting group behavior features and individual behavior features over user-product-group relationships, which is iteratively computed for the spamicity scores of users in the reviewer group.

\textbf{FraudInformer} \cite{34}: This method extends the Markov Random Walk model to obtain a ranking of reviewers’ spamicity scores by exploring multiple heterogeneous pairwise features from reviewers’ rating behaviors and linguistic patterns.

\textbf{HoloScope} \cite{30}: This method detects fraud users by using a scalable dense block detection method to explore graph topology and temporal spikes.

\textbf{FRAUDSCAN} \cite{31}: This method utilizes the regularized matrix factorization to learn the fraud campaign embedding for detecting fraud campaign.

\textbf{FraudNE} \cite{41}: It is a deep structure embedding framework to capture the highly non-linear structure information between the vertexes, the different types of which are jointly embedded into a latent space via multiple non-linear layers. The spammers are detected through a clustering algorithm by averaging the degrees of each cluster to find a high-density area in the latent space.
Figure 3: Effectiveness comparison between COSD and state-of-the-art approaches on the two datasets.

ColluEagle. [33]: A powerful Markov random field (MRF)-based model which builds the user network over co-review behaviors, and then employs the loopy belief propagation model to calculate the spamicity scores of users under a pairwise based Markov Random Field (MRF) framework.

Our method. Our proposed collective opinion spammer detection method is called COSD, which simultaneously learn the direct relevance embeddings and indirect relevance embeddings over the co-reviewed correlations for detecting spammers. To analyze the different impacts of the two types of relevance embeddings, we also evaluate two variants of our model, that is, (i) COSD-D, this is a variant of our COSD model and we only optimize direct relevance loss; and (ii) COSD-ID, this is a variant of our COSD model that we only optimize indirect relevance loss.

5.4. Implementation Details

For GsRank, we utilize FIM to extract candidate spammer groups as mentioned in [26]. For FraudInformer, we treat all features fairly as mentioned in [34], and the computation ends when $\epsilon = 10^{-6}$. For HoloScope, we use the implementation provided by the shared source code. For FraudInformer, we keep the hyper-parameters (i.e., $\alpha_1, \alpha_2, \alpha_3, k$) consistent with the original paper and con-
duct grid search for the learning rate in \{0.1, 0.01, 0.001, 0.0001\}. For FraudNE, the learning rate is searched in \{0.01, 0.001, 0.0001\}, the trade-off parameters \(\alpha\) is searched in \{0.1, 0.5, 1, 2\} and the scale factor \(\beta\) is searched in \{5, 10, 15, 20\}. For ColluEagle, we conduct grid search for the variance of a normal distribution \(\sigma_1, \sigma_2\) in \{1, 10, 30, 60, 90, 120\} and the prior rate in \{0, 0.2, 0.4, 0.6, 0.8\}.

The parameters for our proposed method are empirically set as follows: The embedding size \(K\) is set as 64 on AmazonCn and 128 on YelpHotel, respectively. For the setting of the random-walk of each node, \(\gamma = 30\), \(t = 8\), \(\omega = 5\), \(\kappa = 8\) for two datasets; and for \(\{\beta, n\}\), we set is as \{0.6, 25\} for AmazonCn, and \{0.4, 40\} for YelpHotel.

5.5. Evaluation Results and Analysis

**Comparison of the Spammer detection Performance.** This experiment is to evaluate the effectiveness of identifying the spammers by our approach with the baseline methods on AmazonCn and YelpHotel. Figure (3a) and (3b) shows \(AP\) and \(AUC\) of each method on such two datasets. Each method is performed 10 times, respectively.

From Figure (3a) and (3b), we can observe that: COSD consistently outperforms all baselines on all metrics. For example, COSD outperforms GsRank by [18.8\%, 14.5\%] in terms of \(AP\) and \(AUC\) on Amazon and [12.9\%, 1.6\%] on YelpHotel, respectively. This is because COSD is capable of learning the information from pairwise user relations with behavioral information, while GsRank is based on the group information, which might omit several subtle effective information from users. In addition, COSD outperforms FraudInformer by [17.9\%, 16.6\%] on Amazon and [26.5\%, 21.8\%] on YelpHotel in terms of \(AP\) and \(AUC\), respectively. However, COSD and FraudInformer both make use of pairwise user features for collective spammers detection, which demonstrate that COSD is more effective and robust in exploiting the relationships between users from different-levels, i.e., direct relevance and indirect relevance, which can effectively distinguish users with low likelihood of collusion and thereby avoiding heaps of noise. Comparing with FraudNE, our method outperforms it on both
datasets, as FraudNE neglects the rating information and time information in the datasets, besides it is hard for FraudNE to capture the potential collusion between pair of users. Additionally, we also observe that COSD outperforms FRAUDSCAN by [7.9%, 9.2%] on AmazonCn, and [4.8%, 1.6%] on YelpHotel in terms of AUC and AP, respectively. And comparing with state-of-the-art method ColluEage, the performance is improved by [3.4%, 5.1%] on on AmazonCn, and [11.3%, 7.7%] on YelpHotel in terms of AUC and AP, respectively. The reason might be COSD effectively captures the possibility of collusion for pairwise users who do not have common co-reviewed records, and COSD can model users’ neighbor structure over the indirect relations for detection.

From Figure (4) we can observe that most of methods perform worse when
rank $k$ increases, and our method outperforms other methods consistently on AmazonCn and Yelp datasets in terms of Precision@k and NDCG@k, which also demonstrates the effectiveness of our method. More importantly, COSD can combine the direct relevance and indirect relevance over user behavioral features for jointly optimizing the losses of learning such two relevance embedding, and thus the learnt user embeddings might preserve not only the behavioral information but also the topology information, which makes the users with direct and indirect collusive connections close in the learnt low-dimensional vector space.

The Impact of Direct Relevance Embedding. From Figure (3), we can observe that COSD-D and COSD-ID achieve good results. COSD-D outperforms all baselines on AmazonCn in terms of AP, which demonstrate the effectiveness of using a signed network with positive and negative links, and it also shows that optimizing the direct relevance is capable of more accurately reflecting the realistic relationships in low-dimensional space.

The Impact of indirect Relevance. From Figure (3), we can observe that COSD-ID does not perform as well as COSD-D, however it still outperforms other baselines on AmazonCn in terms of AP, which demonstrates that the effectiveness of performing a random walk along with the positive links on a built user-based signed network while discarding the negative links with weak collusion signals (even noise) of being colluders, and thus it can focus on the analysis of the filtered neighbor structure for more accurately capturing the collusion signals, and it also shows that using the indirect relations is useful for exploring the implicit associations for any pairwise uses in these fraud campaigns.

5.6. On the Sensitivity of Parameters

In this section, we study the impact of three parameters, i.e., $\beta, \zeta, K$, in our method.

\footnote{We only perform parameter sensitivity experiment on AmazonCn dataset as the scale of the YelpHotel dataset is too small to observe significant changes.}
For parameter $\beta$ (used in Eq. (13)), which controls the contributions of direct relevance embeddings and indirect relevance embeddings in our model, and As shown in Figure (5a), our proposed method achieves the best result when $\beta = 0.6$, and simultaneously making use of both direct relevance embedding and indirect relevance embedding outperforms the extreme cases when only considering the indirect relevance embedding ($\beta = 0$) or the direct relevance embedding ($\beta = 1$), which is consistent with the former analysis, i.e., direct relevance embedding is more important than indirect relevance embedding.

For parameter $\zeta$ (used in Eq. (6)), which is a thresholds to distinguish high-probability and low-probability collusion signal. As shown in Figure (5b), our
proposed method achieves the best result within the range of \([0.3, 0.4]\), which demonstrate that when the value of \(\zeta\) is small, it might result in more noise involved, while a large one would lead to the insufficiently learning of indirect relevance embedding due to neglecting more indirect collusive relationships.

For parameter \(K\), which is the dimensional (same for direct and indirect) of the representation embedding vector. As shown in Figure (5c), when varying \(K\), the performance first increases and then decreases, and the best perform is achieved when \(K = 64\), which demonstrate that our proposed method can well work with a low dimension vector space.

6. Conclusion

In this paper, we propose a novel unsupervised network embedding-based approach to jointly combine direct and indirect neighborhood exploration for learning the user embeddings for more accurately identifying spam reviewers. Experiments on two real-world datasets demonstrate that our proposed approach significantly outperforms all of baselines on all metrics, by learning the user representation via jointly optimizing the direct relevance and indirect relevance.

For future work, we plan to investigate the following issues: 1) trying to incorporate more pairwise features and prior knowledge information into the model to enhance the representation ability of direct relevance modeling; 2) considering an unsupervised graph neural network algorithm to enhance the representation ability of user embedding, via modeling the indirect relevance of users in the latent space; and 3) conducting more comprehensive studies to investigate the sensitivity of the parameters in the model.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No.61602197 and Grant No.61772076, and in part
References

[1] T. Fornaciari, M. Poesio, Identifying fake amazon reviews as learning from crowds.

[2] Y. Ren, Y. Zhang, Deceptive opinion spam detection using neural network, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 140–150.

[3] X. Wang, K. Liu, S. He, J. Zhao, Learning to represent review with tensor decomposition for spam detection, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 866–875.

[4] H. Li, G. Fei, S. Wang, B. Liu, W. Shao, A. Mukherjee, J. Shao, Bimodal distribution and co-bursting in review spam detection, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 1063–1072.

[5] M. Ott, Y. Choi, C. Cardie, J. T. Hancock, Finding deceptive opinion spam by any stretch of the imagination, arXiv preprint arXiv:1107.4557.

[6] J. Ye, L. Akoglu, Discovering opinion spammer groups by network footprints, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2015, pp. 267–282.

[7] A. Mukherjee, V. Venkataraman, B. Liu, N. Glance, What yelp fake review filter might be doing?, in: Seventh international AAAI conference on weblogs and social media, 2013.

[8] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, H. W. Lauw, Detecting product review spammers using rating behaviors, in: Proceedings of the 19th ACM international conference on Information and knowledge management, 2010, pp. 939–948.
[9] N. Jindal, B. Liu, E.-P. Lim, Finding unusual review patterns using unexpected rules, in: Proceedings of the 19th ACM international conference on Information and knowledge management, 2010, pp. 1549–1552.

[10] F. H. Li, M. Huang, Y. Yang, X. Zhu, Learning to identify review spam, in: Twenty-second international joint conference on artificial intelligence, 2011.

[11] S. Xie, G. Wang, S. Lin, P. S. Yu, Review spam detection via temporal pattern discovery, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012, pp. 823–831.

[12] S. Günneemann, N. Günneemann, C. Faloutsos, Detecting anomalies in dynamic rating data: A robust probabilistic model for rating evolution, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 841–850.

[13] N. Günneemann, S. Günneemann, C. Faloutsos, Robust multivariate autoregression for anomaly detection in dynamic product ratings, in: Proceedings of the 23rd international conference on World wide web, 2014, pp. 361–372.

[14] H. Li, Z. Chen, A. Mukherjee, B. Liu, J. Shao, Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns, in: Ninth international AAAI conference on web and social Media, 2015.

[15] S. KC, A. Mukherjee, On the temporal dynamics of opinion spamming: Case studies on yelp, in: Proceedings of the 25th International Conference on World Wide Web, 2016, pp. 369–379.

[16] J. Ye, S. Kumar, L. Akovali, Temporal opinion spam detection by multivariate indicative signals, in: Tenth International AAAI Conference on Web and Social Media, 2016.

[17] J. Li, M. Ott, C. Cardie, E. Hovy, Towards a general rule for identifying deceptive opinion spam, in: Proceedings of the 52nd Annual Meeting of
the Association for Computational Linguistics (Volume 1: Long Papers), 2014, pp. 1566–1576.

[18] Z. You, T. Qian, B. Liu, An attribute enhanced domain adaptive model for cold-start spam review detection, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1884–1895.

[19] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, V. Subramanian, Rev2: Fraudulent user prediction in rating platforms, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 333–341.

[20] S. Shehnepoor, M. Salehi, R. Farahbakhsh, N. Crespi, Netspam: A network-based spam detection framework for reviews in online social media, IEEE Transactions on Information Forensics and Security 12 (7) (2017) 1585–1595.

[21] Y. Liu, B. Pang, X. Wang, Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph, Neurocomputing 366 (2019) 276–283.

[22] C. G. Harris, Detecting deceptive opinion spam using human computation, in: Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.

[23] S. Feng, R. Banerjee, Y. Choi, Syntactic stylometry for deception detection, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2012, pp. 171–175.

[24] H. Li, B. Liu, A. Mukherjee, J. Shao, Spotting fake reviews using positive-unlabeled learning, Computación y Sistemas 18 (3) (2014) 467–475.

[25] S. Kim, H. Chang, S. Lee, M. Yu, J. Kang, Deep semantic frame-based deceptive opinion spam analysis, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 1131–1140.
[26] A. Mukherjee, B. Liu, N. Glance, Spotting fake reviewer groups in consumer reviews, in: Proceedings of the 21st international conference on World Wide Web, 2012, pp. 191–200.

[27] Q. Li, Q. Wu, C. Zhu, J. Zhang, W. Zhao, Unsupervised user behavior representation for fraud review detection with cold-start problem, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2019, pp. 222–236.

[28] S. Rayana, L. Akoglu, Collective opinion spam detection: Bridging review networks and metadata, in: Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining, 2015, pp. 985–994.

[29] S. Rayana, L. Akoglu, Collective opinion spam detection using active inference, in: Proceedings of the 2016 SIAM International Conference on Data Mining, SIAM, 2016, pp. 630–638.

[30] S. Liu, B. Hooi, C. Faloutsos, Holoscope: Topology-and-spike aware fraud detection, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1539–1548.

[31] C. Xu, J. Zhang, Z. Sun, Online reputation fraud campaign detection in user ratings., in: IJCAI, 2017, pp. 3873–3879.

[32] P. Kaghazgaran, J. Caverlee, A. Squicciarini, Combating crowdsourced review manipulators: A neighborhood-based approach, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 306–314.

[33] Z. Wang, R. Hu, Q. Chen, P. Gao, X. Xu, Collueagle: collusive review spammer detection using markov random fields, Data Mining and Knowledge Discovery 34 (6) (2020) 1621–1641.
[34] C. Xu, J. Zhang, Combating product review spam campaigns via multiple heterogeneous pairwise features, in: Proceedings of the 2015 SIAM International Conference on Data Mining, SIAM, 2015, pp. 172–180.

[35] C. Xu, J. Zhang, K. Chang, C. Long, Uncovering collusive spammers in chinese review websites, in: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, 2013, pp. 979–988.

[36] J. Leskovec, D. Huttenlocher, J. Kleinberg, Predicting positive and negative links in online social networks, in: Proceedings of the 19th international conference on World wide web, 2010, pp. 641–650.

[37] R. H. Hahnloser, H. S. Seung, Permitted and forbidden sets in symmetric threshold-linear networks, in: Advances in neural information processing systems, 2001, pp. 217–223.

[38] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, H. S. Seung, Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit, Nature 405 (6789) (2000) 947–951.

[39] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 701–710.

[40] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.

[41] M. Zheng, C. Zhou, J. Wu, S. Pan, J. Shi, L. Guo, Fraudne: a joint embedding approach for fraud detection, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.