Spatio-Temporal Graph Representation Learning for Fraudster Group Detection

Saeedreza Shehnepoor*, Roberto Togneri*, Senior Member, IEEE, Wei Liu, Member, IEEE, and Mohammed Bennamoun*, Senior Member, IEEE

Abstract—Motivated by potential financial gain, companies may hire fraudster groups to write fake reviews to either demote competitors or promote their own businesses. Such groups are considerably more successful in misleading customers, as people are more likely to be influenced by the opinion of a large group. To detect such groups, a common model is to represent fraudster groups’ static networks, consequently overlooking the longitudinal behavior of a reviewer, thus, the dynamics of coreview relations among reviewers in a group. Hence, these approaches are incapable of excluding outlier reviewers, which are fraudsters intentionally camouflaging themselves in a group and genuine reviewers happen to coreview in fraudster groups.

To address this issue, we propose “FGDT,” a framework for “fraudster group detection through temporal relations.” FGDT first capitalizes on the effectiveness of the HIN-recurrent neural network (RNN) in both reviewers’ representation learning while capturing the collaboration between reviewers. The HIN-RNN models the coreview relations of reviewers in a group in a fixed time window of 28 days. We refer to this as spatial relation learning representation to signify the generalizability of this work to other networked scenarios. Then, we use an RNN on the spatial relations to predict the spatio-temporal relations of reviewers in the group. In the third step, a graph convolution network (GCN) refines the reviewers’ vector representations using these predicted relations. These refined representations are then used to remove outlier reviewers. The average of the remaining reviewers’ representation is then fed to a simple fully connected layer to predict if the group is a fraudster group or not. Exhaustive experiments of FGDT showed a 5% (4%), 12% (5%), and 12% (5%) improvement over three of the most recent approaches on precision, recall, and F1-value over the Yelp (Amazon) dataset, respectively.

Index Terms—Fraudster group, graph convolution network (GCN) refinement, spatio-temporal modeling.

I. INTRODUCTION

Modern consumers’ significant uptake of online shopping has created a flourishing e-commerce environment, including the increasing reliance on product review

Fig. 1. Toy example of a coreview matrix demonstrating the behavior change of fraudsters in a fraudster group (green means genuine reviews and red means fraud ones). In time step $t_1$, reviewers $r_4$ and $r_3$ coreview an item with fraud reviews. In the next time step ($t_2$), the reviewers are camouflaged.

recommendations, such as those found on Yelp and Amazon. This has provided a fertile ground for fraud reviewers to deliberately mislead consumers through manipulated reviews. To increase their impression on users, fraudsters may work in teams, thus forming fraudster groups, to collectively attack (or promote) certain products or services. Group fraud may serve different purposes, such as manipulating the semantics of reviews, distributing the overall workload, and avoiding detection through temporal behavior manipulation [1], [2], [3]. The fraud related research area is currently dominated mainly by individual fraudster detection [4], [5], [6] and fraud review detection [7], [8], [9]. Recent years have seen an increased research effort in detecting fraudster groups [10], [11], [12], [13]. It is widely accepted that individual fraudsters can cause significant damage to businesses, fraudster groups may be even more damaging because of their coordinated and considerable volume of fraud reviews that they can collectively produce.

Fraudster groups also are much more difficult to detect compared with individual fraudsters. Each group fraudster can camouflage more easily by controlling his/her relation with other group members such that no single fraudster stands out [1], [3]. In other words, a group fraudster can escape detection by avoiding certain relations with other group reviewers or developing multiple relations with genuine reviewers. As a member of a community [14], group fraudsters can also manipulate their relations over time to appear to be genuine reviewers and, as such, the group appears authentic and avoids detection, as illustrated in Fig. 1.

The algorithms for fraudster group detection are dominated by two categories: frequent itemset mining (FIM)-based [10], [13] approaches or graph-based [11], [12] approaches. An FIM-based algorithm generally follows a two-step process: first, candidate groups are determined based on the same set of items (itemset) reviewed by the reviewers. Then, the candidate
groups are ranked based on the probability of being a fraudster or a genuine group. Note that the candidate group refers to a group of users with a possible fraudulent collaboration. Graph-based algorithms, on the other hand, employ graph partition or clustering algorithms [3] to provide a genuine group of the reviewers to determine the candidate groups’ probability of being either fraudsters or genuine. However, the approaches in both categories suffer from significant drawbacks.

First, the most recent approaches (Ji et al. [11], Zhang et al. [12], and Shehnepoor et al. [15]) overlooked the temporal nature of fraudster groups, as illustrated in Fig. 1. Fraudster groups can take advantage of such a limitation to manipulate relations and mislead detection algorithms.

Second, previous approaches are limited to a confined task of using either members’ relations or reviewers’ vector representations in the final classification. A better approach should consider all the available information to jointly learn both the semantic representation and the behavioral representation.

Third, current approaches on fraudster group detection are incapable of modeling “outlier” members of a group. In social review platforms, apart from purposely forming a fraudster group, reviewers may also form genuine groups due to similar interests. Note that the genuine groups are not necessarily formed and in this study, the genuine groups refer to genuine reviewers writing similar reviews for the same set of items. However, members in a fraudster group could make genuine reviews, and vice versa, members in a genuine group may write fake reviews. Such outliers, i.e., fraudster(s) in a genuine group or genuine reviewer(s) in a fraudster group, pollute the group, creating another level of challenges for fraudster group detection. The most recent approach of fraudster group detection by Shehnepoor et al. [15] excluded such outliers by removing the users with the least connection with other reviewers. However, such an approach overlooks the importance of joint representation in covering different aspects of a reviewer’s activities on a social platform. Joint representations have shown to be effective in different fraud detection tasks [7], [8], [16], and hence, are hypothetically helpful in removing the outliers, which has been verified effective by the results of this article (see Section IV-C4). Removing outliers with the least connection results in some improvement; however, such a removal approach is still not effective enough in removing outlier reviewers, resulting in a higher false positive (FP) rate and false negative (FN) rate.

Fig. 2 uses a toy example of a group with four reviewers to illustrate our proposed algorithm. Our proposed spatial–temporal architecture is a four-module pipeline, which continuously refines the initial reviewer representation for the downstream fraudster group detection task. The first module is responsible for the “spatial” relation between reviewers. This is not the spatial distance in the geographical sense, but a derived social closeness based on whether two reviewers have coreviewed. We define coreview as a relation between two reviewers when they reviewed the same item with the same rating and similar semantics in a specified window of time. For example, from Fig. 2(a), during $T_1$, reviewer $r_1$ and $r_2$ have a coreview relationship, whereas during $T_3$, $r_3$ and $r_4$ have coreviewed. A list of notations used in the article is given in Table I. We take advantage of the HIN-RNN [15] to model the coreview relation between reviewers, as it has been proven to be effective in utilizing spatial relations between reviewers. HIN-RNN [15] is an RNN used to model the relation between nodes in a graph considering the nodes’ heterogeneity. The heterogeneity is obtained through a fine-tuning of the semantics extracted using word embedding techniques based on the reviewer’s type (fraudster/genuine). Given the reviews, reviewer ids, item ids, ratings, and dates as the input, the proposed spatial modeling comprises three steps.

1) First, the candidate groups in each time window are determined.
2) With groups determined, reviews are used to extract the sum of word embeddings (SoWEs) for the reviewers in each time window as the semantic representation of the reviewer.
3) The HIN-RNN refines the candidate groups from the first step using the semantic representation extracted in the second step. The time windows ($t_i$) are then utilized...
in the reviewer representation step to extract the SoWE for the reviewer \( i \in \{1, 2, 3, 4\} \) in a corresponding time window \((V_{i,TW})\). The relations between reviewers in groups for different time windows are then refined with the reviewers’ representation by the HIN-RNN [15] to output the spatial relations. The HIN-RNN realizes a possible spatial correlation between \( r_2 \) and \( r_3 \) in the 1st and \( N \)th time window.

The output of the spatial modeling step is subgraphs with refined relations between reviewers in a group in each time window.

The second module employs a simple recurrent neural network (RNN) to model the relations between the reviewers throughout different time windows, given the spatial relations from the previous spatial module. The RNN predicts a possible temporal relation between, say, \( r_1 \) and \( r_3 \) [from the possible similar temporal activities depicted in Fig. 2(b)]. This step addresses the previous approaches’ limitation in temporal relation modeling, by encoding the reviewers’ temporal relations. The output of this step is the collaboration matrix for each group at the \( N + 1 \)th time window. As mentioned, previous studies overlook the temporal relation between reviewers, which can be manipulated by fraudsters to avoid detection as a group. The second module of fraudster group detection through temporal relation (FGDT) models the temporal relation between the reviewers in order to overcome such a limitation. Note that the RNN in HIN-RNN models the relation between reviewers in a single time window. The employed autoregressive model in the HIN-RNN takes the history of the reviewers’ connection into account to predict the connection of the current reviewer to other reviewers in a graph. To use the temporal modeling, a second RNN is necessary to take the formation of the group at each time window and predict the reviewer connection based on previous spatio-temporal relations (captured in previous time windows).

In the third module, a graph convolutional network (GCN) is used to ensure that the reviewers’ effects on each other are captured. Hence, the GCN will refine the representation of the reviewers based on the captured spatio-temporal relations and the labels of each individual reviewer. For example, from Fig. 2(c), reviewers \( r_1, r_2, \) and \( r_4 \) have the type, while \( r_3 \) has a different one. So, the GCN provides further refinement based on the reviewers’ representation, the spatio-temporal relation, and the labels of the reviewers. The output of this step is the groups with the reviewers’ representation refined based on the spatio-temporal relations that were captured in the previous steps. As explained, in the current state of the art, the final classification considers either only the members’ relation or the reviewers’ vector space, not both. In this research, we use the behavioral [negative ratio (NR)] and linguistic [continuous bag of word (CBoW)] features as the semantic representation of the reviewers in the spatial modeling module. Using these combined features, the temporal modeling module predicts the temporal relationships between the reviewers. In the GCN refinement module, a GCN is applied to the semantic representation to: 1) integrate and 2) refine the representation to be used by the group classification module.

After the refinement in the third step, we apply the \( k \)-means clustering algorithm in the fourth step to recognize the outlier reviewers in each group based on the representations. As a result of such refinement, FGDT can exclude outlier reviewers in this step, although reviewer \( r_3 \) developed multiple relations with other reviewers to escape the detection [as shown by Fig. 2(d1)]. This ensures the inclusion of a joint representation in removing outliers (one of the key limitations of previous studies). Since previous approaches only consider static relations between the reviewers to eliminate the outliers, such a removal is not accurate. This is because reviewers develop different kinds of relations within different time frames. Our proposed FGDT takes the temporal information refined integrated reviewers’ representation into consideration to remove the outliers. This results in a more accurate outlier removal as the final outcome (Section IV-C4). The average of the remaining reviewers’ representation is then fed to a fully connected (FC) layer for the final classification of the group [Fig. 2(d2)].

We can summarize our contributions as follows.

1) For the first time, we propose an approach to model the spatio-temporal relations between reviewers in the groups. Such an approach not only produces more accurate modeling of the reviewers’ relations through spatial relations but also considers the possibility of fraudsters’ camouflage through the cover-up of reviews with a group of genuine reviewers or the unintentional genuine reviewers’ involvement in a fraudster group activity over time (see Section IV-C2). FGDT outperforms the state-of-the-art approaches (Ji et al. [11], Zhang et al. [12], and Shehnepoor et al. [15]) by 5% (4%), 12% (5%),
and 12 (5%) based on the precision, recall, and F1-value on the Yelp (Amazon) dataset, respectively (see Section IV-C).

2) Given the spatio-temporal relations between the reviewers in the groups, we use a GCN to refine the initial semantic-behavioral representation (SoWE + NR) of reviewers. This step discriminates further between genuine and fraudster reviewers in terms of their representations. The results show that the GCN effectively improves the performance of FGDT by 11% and 5% in terms of recall and F1-value on the Yelp dataset, respectively (see Section IV-C3).

3) With the refined representations, we propose a new approach to remove outlier reviewers from the group using the well-known k-means clustering algorithm. The results show that the proposed clustering algorithm outperforms the threshold-based removal approach proposed by Ji et al. [11] and the minimum connection removal approach proposed by Shehnepoor et al. [15] (see Section IV-C4).

The rest of this article is structured as follows. In Section II, we discuss the related work. In Section III, we introduce our methodology. In Section IV, we provide our experimental evaluation. We conclude this article with an outlook to future work in Section V.

II. RELATED WORKS

A. Spatio-Temporal Modeling

The concept of spatio-temporal modeling has been applied to various tasks and with different methods. Chen et al. [17] use the recurrent convolutional neural network (CRNN) to model human activities for a human activity recognition (HAR) task. The framework collects multimodal data from different sensors attached to different body parts and then extracts a salient representation of the features using a feature selection method (based on the Boltzmann machine). The features from each time step (called glimpse) are then fed to a convolutional neural network to imitate the brain process of converting the input information from the optic nerves to higher level information. The higher level information for each step is then fed to an RNN to predict the location of the body part and the corresponding action taken. The performance of the proposed network was evaluated on several datasets (e.g., MHEALTH, PAMAP2, and so on). The results show an accuracy of 73.38% for the best-case scenario on the PAMAP2 dataset.

In order to address the limitation of previous work in considering spatial relationships between different EEG signals, Zhang et al. [18] utilized a convolutional neural network. An RNN was also used to consider the chain-like relation between EEG signals at different times. Spatio-temporal modeling explores such information in both cascade and parallel manner to predict human intentions. The extensive experiments on a large-scale movement intention EEG dataset show an accuracy of 98.3%.

B. Graph Representation Learning

Various studies concentrated on graph representation learning as their focus. Leng et al. [19] proposed a graph regularized smooth nonnegative matrix factorization (GSNMF), which incorporates the regularization and smoothing constraint to consider the intrinsic geometric information of the dataset to discover the hidden semantics. The smoothing constraint is also added to nonnegative matrix factorization to produce a more optimized solution. The experiments demonstrate an improvement in the accuracy of the proposed approach by 8% as compared with the state-of-the-art solutions, reaching a clustering accuracy of 78.125%.

GCNs are among the most recent approaches employed to refine the representation of nodes in a graph. However, training a GCN requires high computation and storage costs for larger graphs. Liu et al. [20] examine the most recent sampling methods to reduce such costs and achieve efficiency and scalability. Different sampling methods, such as nodewise sampling methods, layerwise sampling methods, subgraph sampling methods, and heterogeneous sampling methods, are discussed and compared to provide better insights into the best sampling methods for each application.

Another joint representation learning is employed to remove outlier reviewers from a group using the well-known k-means clustering algorithm. The results show that the proposed clustering algorithm outperforms the threshold-based removal approach proposed by Ji et al. [11] and the minimum connection removal approach proposed by Shehnepoor et al. [15] (see Section IV-C4).

Another joint representation learning is employed to remove outlier reviewers from a group using the well-known k-means clustering algorithm. The results show that the proposed clustering algorithm outperforms the threshold-based removal approach proposed by Ji et al. [11] and the minimum connection removal approach proposed by Shehnepoor et al. [15] (see Section IV-C4).

C. Fraudster Group Detection

Previous studies on fraudster group detection are generally categorized based on the strategy to determine the initial group formations and include two subcategories: FIM and graph-based algorithms.
1) Frequent Itemset Mining: Approaches in this category determine the initial groups based on an assumption that the reviewers with the same set of reviewed items (itemset) form a possible collaboration; thus, a possible group [24]. Allahbakhsh et al. [10] extended the FIM concept and defined a new detector called the biclique detector. The groups’ formation was initialized based on the biclique concept: reviewers writing reviews with the same rating on the same group of items. After the group formation initialization, handcrafted features, such as group rating value similarity (GVS) and group rating time similarity (GTS) were extracted. A scoring function was applied to the extracted features to determine the probability of a group being fraudster or genuine. FGDT yielded a precision of 75% for the fraudster group detection on the Yelp dataset. Xu et al. [13] also relied on the FIM concept to determine the candidate groups, where reviewers with at least two reviews on at least three coreviewed products formed a collaboration. A measure called homogeneity-based-collusive behavior measure (h-CBM) was proposed using the targeted item, rating, temporal traits, and reviewer activity. To score the groups, an unsupervised scoring model called the latent collusive model (LCM) was employed. The approach proposed by Xu et al. showed a precision of 85% on the Yelp dataset. To overcome the limitations of previous approaches, Shehnepoor et al. [15] proposed HIN-RNN, incorporating the semantic representation of the reviewers to refine the relations. For that purpose, Shehnepoor et al. first determined the candidate groups based on the FIM concept and then extracted the representation of the reviewers using CBoWs. Next, the HIN-RNN was proposed to refine the initial representation. Finally, to reduce the effects of outlier reviewers, the reviewers with minimum relations were removed. The representations of the remaining reviewers were used to predict the probability of the group being a fraudster or not. HIN-RNN achieved a precision of 81% on the Yelp dataset.

2) Graph-Based: Graph-based approaches use graph partition algorithms or clustering methods to determine the group formation based on the similarity between reviewers’ representation. Ji et al. [11] proposed an approach with a focus on products as the main target of the fraudster groups. Ji et al. [11] claimed that considering the products as the focal point will overcome the limitation of FIM-based approaches in concentrating only on reviewers. To have a representation for each group, seven handcrafted features (group rating deviation, group size, group review tightness, group one-day reviews, group extreme rating ratio, group coactiveness, and group collective review ratio) were computed. Next, to demonstrate the effectiveness of the item representation in fraud detection, three product-related features (product rating distribution, product average rating distribution, and suspicious score) were extracted. Finally, six individual fraudster features (ratio of extreme rating, rating deviation, the most reviews one-day, review time interval, account duration, and active time interval reviews) were used to obtain a single representation for each reviewer in a group. The targeted items were then scored based on item-related features. A kernel density estimation (KDE) was used to compute the burstiness for items. Outlier reviewers were also removed based on a threshold applied to each reviewer representation. The proposed approach by Ji et al. [11] showed a precision of 83% on the Yelp dataset. However, both temporal behavioral clues and reviewer representation refinement were overlooked in modeling the relations between reviewers. Zhang et al. [12] proposed a framework with three steps. In the first step, the similarity between reviewers’ ratings and their reviewed items was used to build a graph with reviewers as nodes and the similarities as the edges in the next step. Finally, to obtain the final prediction, a label propagation algorithm was proposed based on the propagation intensity and an automatic filtering mechanism. The proposed approach by Zhang et al. [12] provided a precision of 70% on the Yelp dataset. To demonstrate a comparison of FIM-based and graph-based approaches, an overall view of both approaches is shown in Fig. 3.

3) Research Gap: Although FIM-based approaches and the graph-based ones provide insight into fraudster group detection, they overlook the temporal clues of the reviewers’ behavior in a fraudster group. They also do not consider the reviewers’ relations in a group to determine the reviewer’s representation. Hence, such approaches also suffer from a limitation to consider the possibility of fraudster (genuine) reviewers’ involvement in genuine (fraudster) groups. In this research, we, therefore, devise a new approach to: 1) model temporal relations (coreviewing) between reviewers to predict the collaboration matrix; 2) refine the relations between the reviewers in a group based on their spatio-temporal relations; and 3) use the relations to refine the representations of the reviewers. In previous works, the outlier reviewers were also removed based on a static threshold or based on the number of relations, resulting in inaccurate pruning, easily manipulated by camouflage. We recognize this problem and utilize a clustering algorithm to remove the outlier reviewers.
III. FRAUDSTER GROUP DETECTION THROUGH TEMPORAL RELATION

Problem Definition: Let us consider a time-stamped review as a tuple \((r, t, i, d)\), where reviewer \(r \in R\) wrote text \(t \in T\) on item \(i \in I\), on date \(d \in D\). The goal of fraudster group detection is to obtain subgraphs \(R^x \subset R\) and classify them into either fraud or genuine.

A. Spatial Modeling

In this step, we aim to model the spatial relations between reviewers in the groups throughout different time windows with no overlaps.

1) Candidate Groups: To increase their impact, fraudster group members are likely to write coordinated fraud reviews in a shorter time frame as compared with genuine reviewers [1], [16]. Similar to previous studies, the reviewers with the same ratings in a period of 28 days establish an initial relation in a group [11], [12], [16]. Similar to previous studies, the reviewers with the same ratings in a period of 28 days establish an initial relation in a group [11], [12], [16].

Two reviewers will be connected as a tuple \((i, j)\) from a vector space with dimensionality \(F\). The SoWEs of a reviewer’s review text are used as a semantic representation of each reviewer. SoWE refers to the embedding of a sentence \(s\), as the embedding for word \(w_i\) from a vector space with dimensionality \(F\).

2) Reviewer Representation: In this step, reviews for each reviewer are aggregated and then split into sentences. With the promising performance of word embedding techniques in fraud detection [26], each word in the sentences is initialized with a pretrained word embedding, \(e_{w_i} \in \mathbb{R}^F\), as the embedding for \(w_i\) from a vector space with dimensionality \(F\). The SoWEs of a reviewer’s review text are used as a semantic representation of each reviewer. SoWE refers to the simple linear function of aggregating the embeddings of words to represent a sentence, a reviewer, or a group, shown to be effective in different domains, such as Ren and Ji et al. [26], White et al. [27], and Lyndon et al. [28]. Therefore, to obtain a SoWE for a sentence we applied an elementwise average to WEs of a sentence

\[
e_s = \frac{1}{n_{ws}} \sum_{i=1}^{n_{ws}} v_{ws,i}
\]

where \(n_{ws} \) is the number of words in the sentence, and the corresponding word embedding of word \(i\) in the sentence \(s\), respectively. Finally, we apply a max (as max-pooling) operator to obtain a vector representation for the reviewer. The process is depicted in Fig. 4.

Next, we extract a behavioral indicator for each reviewer, namely, the NR [8]

\[
NR = \frac{\sum_{r=1}^{N} R_{\text{rating}}}{N}
\]

where \(R_{\text{rating}}\) is the number of reviews of a reviewer with specific ratings (rating) in the range of 1–5 (5 is the highest rating), and \(N\) is the total number of reviews by the reviewer. The concatenation forms the final vector representation of reviewer \(r\) as \(v_r\).

3) Spatial Modeling Through HIN-RNN: Given the effectiveness of the HIN-RNN [15] in taking into account the reviewers’ long-range dependencies, we employ the HIN-RNN to model the relations between reviewers in different time windows. HIN-RNN takes the coreview networks as subgraphs, alongside the reviewers’ representation, and outputs the refined relations between the reviewers based on the reviewers’ representations. Therefore, first the subgraphs are mapped to corresponding adjacency matrices called “collaboration matrix” using a node ordering \(\pi (\pi (r_1), \ldots, \pi (r_i), \ldots, \pi (r_n))\), with \(r\) representing the reviewer and \(n\) is the number of reviewers. The \(\Pi\) is a set of all possible permutations of the reviewers which is \(n\). The corresponding collaboration matrix of an arbitrary ordering \(\pi \in \Pi\) is \(A^\pi \in \mathbb{R}^{n \times n}\) with \(A^\pi_{i,j} = 1(\pi(r_i), \pi(r_j)) \in E\), where \(E\) is the set of edges. We aim to learn a set of distributions \(p(G_T^i)\), \(G_T^1, \ldots, G_T^M\) for possible groups where each \(G_T^i\) represents a grouping of reviewers within a collaboration matrix.

Now, a mapping function \(f_S\) is applied to coreview networks to output the sequences, enabling us to use an autoregressive model

\[
S^\pi = f_S(G_T^\pi, \pi) = (S_{1}^\pi, S_{2}^\pi, \ldots, S_{n}^\pi).
\]

In (3), \(S^\pi \in [0, 1]^{i-1}, i \in [1, \ldots, n]\) represents the collaboration of the user \(i\) with other users called the “collaboration vector” \(S^\pi = (A_1^\pi, \ldots, A_i^{-1})^T, \forall i \in [2, \ldots, n]\) between reviewer \(\pi(r_i)\) and previous reviewers \(\pi(r_j), j \in [1, \ldots, i-1]\) in a group. To fully characterize \(p(G_T^i)\), \(p(S^\pi)\) is learned

\[
p(G_T^i) = \sum_{S^\pi} p(S^\pi) I[f_G(S^\pi) = G_T^i].
\]

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
We rewrite the $p(S^f)$ as an autoregressive conditional distribution

$$p(S^f) = \prod_{i=1}^{n+1} p(S^f_i | S^f_{i-1}, \ldots, S^f_1) = \prod_{i=1}^{n+1} p(S^f_i | S^f_{<i})$$

(5)

where $S^f_{<i} = \{S^f_{i-1}, S^f_{i-2}, \ldots, S^f_1\}$. To fully capture the relations, the HIN-RNN expands 5 as follows:

$$p(S^f_i | S^f_{<i}) = \prod_{j=1}^{i-1} p(S^f_{i,j} | v_i, S^f_{<i,j}, S^f_{<i})$$

(6)

where the $v_i$ is the current reviewer’s representation, $S^f_{<i,j} = \{S^f_{i,j-1}, S^f_{i,j-2}, \ldots, S^f_{i,1}\}$, and $S^f_{<i,j}$ is 1 if there is collaboration between reviewers $i$ and $j$ (the refined collaboration link between $r_2$ and $r_1$ in Fig. 2). The structure of the HIN-RNN model is displayed in Fig. 5.

To parameterize two autoregressive models, two RNNs are utilized. First

$$h_i = f_1(h_{i-1}, S^f_{<i-1}, v_i)$$

(7)

where the $h_1$ encodes the state of the groups (reviewers plus their collaboration matrix) up to the reviewer $i$ and the $f_1$ is the function learned by the RNN. Next, we obtain the collaboration matrix of the current reviewer

$$S^f_i = f_2(h_i)$$

(8)

where $S^f_i$ encodes the collaboration vector obtained from the function $f_2$ using the RNN.

B. Spatio-Temporal Modeling

Intuitively, group activities of fraudster groups increase gradually over time, and the final time windows encode richer temporal characteristics. In other words, as time goes forward, a better resolution of the fraudster groups’ temporal activities is obtained. Hence, acquiring the collaboration matrix of the final time window provides more accurate modeling. So, in this step, we aim to obtain the collaboration matrix of the groups. So far, we modeled the spatial relations between reviewers, as $p(G^{TW})$, $\forall TW \in \{1, \ldots, N\}$ [see (4)], where $N$ is the total number of time windows. Similar to Section III-A, first, the subgraphs for different time windows are mapped to a collaboration matrix of $C^{TW}, C \in \mathbb{R}^{M \times n \times n}$ with $C_{g,i,j}^{TW} = 1$, if reviewers $i$ and $j$ form a collaboration in group $g$.

To obtain the collaboration matrix of $g$ in $N + 1$th time window, we model the relation of the collaboration matrices in time windows as an autoregressive model

$$p(C_g) = p(C_g^1, \ldots, C_g^{N+1}) = \prod_{TW=1}^{N+1} p(C_g^{TW} | C_g^{TW-1}, \ldots, C_g^1).$$

(9)

The output of this step is the corresponding subgraph of the collaboration matrix in $N + 1$th time window. To parameterize the autoregressive model, we employed a simple one-to-many RNN to model the hidden states

$$h_g^{TW} = \hat{e}(W_h h_g^{TW-1} \oplus W_C C_g^{TW})$$

(10)

where $h_g^{TW}$ is the hidden state of the group $g$ at $TW$th time window, the $W_h$ is the weight matrix of hidden states and the $W_C$ is the weight matrix of the transition matrix. The collaboration matrix of $N + 1$th time window is obtained through

$$C_g^{N+1} = \tanh(h_g^N).$$

(11)

C. Representation Refinement Through Graph Convolutional Network

Given the refined spatio-temporal relations captured as a collaboration matrix in the previous steps, in this step, we employ the GCN to refine the representation of the reviewers in the collaboration matrix. As such refined representations are trained based on: 1) the collaboration matrix and 2) the labels for each reviewer, they bring different merits: first, the reviewers with a similar behavior are represented closer to each other in the feature space. Second, the outlier reviewers (a genuine reviewer in a fraudster group, or a fraudster in a genuine group) are distant from the majority of the reviewers (see Section IV-C3) in the feature space.

The GCN uses the concept of layerwise propagation for neural network models to encode both the adjacency matrix (collaboration matrix) and the labels of the nodes (reviewers). Therefore, the model is defined as $f(V_g, C_g)$, where $V_g$ is a 2-D vector representation of the reviewers in the group $g$, and $C_g$ is the collaboration matrix of group $g$ in the $N + 1$th time window. Conditioning $f(\cdot)$ on the collaboration matrix of each group helps the model to distribute the gradient information from the supervised loss and will enable the GCN to learn representations of the reviewers. With the introduction of a multilayer neural network to model $f(V_g, C_g)$, efficient information propagation is obtained, while the prediction is labeled on both the representations and the collaboration matrix. Therefore, in this study, we employ a two-layer GCN, with the assumption that the collaboration matrix for each group is symmetric. GCN defines a self-connected adjacency matrix (collaboration) matrix, $C_g = C_g + I$, where $I$ is the identity matrix. Then, a degree matrix is calculated for each
Algorithm 1 FGDT Algorithm

Output: The label of each group;
Input: T review texts of R reviewers and ratings rate, written on I items, N is the number of time windows, M number of groups;

% Step 1: spatial modeling;
for TW ← 1 to N do
  % candidate groups in time windows;
  if r, r coreview same item i ∈ I, ∀r, r ∈ R then
    % Link the possible collaborating reviewers;
    E(x, y) = 1;
    % reviewer representation;
    for r ← 1 to R do
      % Tokenize t ∈ T to S sentences;
      {s1, s2, ..., sT} ← tokenize(t);
      % Sentence Representation;
      for s ← 1 to S do
        % Tokenize St to n words;
        {w1, ..., wn} ← tokenize(S);
        % Word embeddings;
        {e1, ..., en} ← WE({w1, ..., wn});
        % The sentence representation;
        es ← average({e1, ..., en});
        % Negative Ratio of reviewer r;
        NR ← NR(rate1, rate2, ..., raten);
        % Final representation of r;
        vr ← max(concat(es, ∀s ∈ S)) + NR;
        % spatial modeling through HIN-RNN;
      for g ← 1 to M do
        for i ← 1 to R do
          hri ← RNN1(hri−1, E(0 : i − 1, 0 : i − 1), vi);
          % Collaboration matrix of reviewer i;
          C[i, ] ← RNN2(hri);

% Step 2: temporal modeling;
for g ← 1 to M do
  for TW ← 1 to N do
    hWg ← ℓ(WhWgW − 1 + WCgW);
    Cg + 1 ← tanh hCg;
    Cg ← Cg + 1;

% Step 3: GCN refinement;
for g ← 1 to M do
  for r ← 1 to R do
    Vr ← vr + ∙ ∙ ∙ + vr;
    Zg ← softmax(ˆCg.ReLU(ˆCgVgW(0))W(1));

% Step 4: group classification;
for g ← 1 to M do
  g ← g[reviewers in the cluster with the minimum
  reviewers based on Zg] ;
  vg ← mean(vr) for r ∈ g;
  label(g) ← fc(vg);

node (reviewer) in group g as ˆDg.i,i = ∑j ˆCg.i,j. Given the

definitions, to obtain the final representation based on the

 collaboration matrix and representations, the GCN assumes

CGN learns. The model is then formulated as

Zg = f(Vg, Cg) = softmax(ˆCg.ReLU(ˆCgVgW(0))W(1))

(12)

where W(0) is the weight matrix of the input-to-hidden

layer and W(1) is the weight matrix of the hidden-to-output

layer. To output the label for each review GCN defines

softmax(x) = (exp(x)/∑i exp(xi)). To train the weight

matrices, we define a cross-entropy over the reviewers’ labels

loss = −∑i∈L ∑F Yf ln(Zf)

(13)

where F is the feature dimension, and L is the set of

labels (genuine, fraudster). This ensures further refinement of

the reviewers’ representation based on the reviewer type.

A schematic of GCN is displayed in Fig. 6.

D. Group Classification

Our proposed group classification method follows three

simple steps: clustering to remove the outlier reviewers, group-

level representation using SoWE, and an FN layer.

1) Clustering: In this step, we first use a simple k-means

algorithm to cluster the reviewers in a group based on their

representation. Intuitively, the reviewers in a group can be

categorized at most into two clusters (a mix of genuine and

fraudster reviewers). To determine the possibility of outliers’

existence in groups, we use the total sum of square (TSS),
as the sum of squared deviations from the overall mean, within

the sum of square (WSS), as the sum of the squared deviations

within a cluster, and between the sum of square (BSS), as the

sum of the squared deviations between the clusters.

For TSS, we calculate the elementwise squared between
each reviewer’s representation (Z) and the centroid

TSSg = ∑i=1n (Zg,i − c)2

(14)

where Zg,i is the value of the i'th element of the j'th reviewer

representation (with 1 ≤ i ≤ 100, being the representation

vector dimension), n is the number of reviewers in a group

and ci is the i'th element from the obtained centroid for the

group in the clustering algorithm.

For the WSS, we calculate the elementwise squared for each
reviewer’s representation in a cluster and the
corresponding centroid

\[ WSS_i = \sum_{m=1}^{2} \sum_{j=1}^{n_{cm}} \left( Z_{i,j}^{(m)} - c_i^{(m)} \right)^2 \]  

(15)

where \( n_{cm} \) is the number of reviewers in cluster \( m \) and \( Z_{i,j}^{(m)} \) is the value of the \( i \)th element of the \( j \)th reviewer in the \( m \)th cluster.

To obtain the final distance value we first calculate the elementwise BSS of the representation:

\[ BSS_i = TSS_i - WSS_i. \]  

(16)

Next, we calculate the second-norm of \( BSS_i \)

\[ |BSS| = \frac{1}{n_c} \sqrt{\sum_i BSS_i}. \]  

(17)

Then, if \(|BSS| < 0.5\), the group is considered as a group with no outliers. Otherwise, the group is considered a mixed group, and then the cluster with more reviewers is regarded as the dominant cluster and is selected for the next step. Note that with the normalized presentation for each reviewer, and the normalized TSS for all reviewers (divided by the total number of nodes), with 0.5 as the threshold, we assess the looseness/tightness of a group.

2) Fully Connected Layer: In this step, we first calculate the SoWE for each group through an elementwise average over the representations of the selected cluster from the previous step (Section III-D1) to obtain a final representation of each group. Finally, an FC layer is trained based on the obtained representation. FGDT Algorithm is presented in 1.

E. Limitations and Future Work

FGDT can be further improved if the following two challenges can be tackled. Reviewers who are involved in group fraud activities typically do not have long-term collusive relationships. Therefore, the major challenge imposed on our proposed framework, a temporal information encoding with a fixed-size window, is how to decide on the window-size to capture when the reviewers actually collude as a fraudster group. It is encouraging to see that even with fixed window size, we can observe some minor improvement. So, in future work, we plan to expand the temporal encoding to use a sliding window with multiple window sizes to address the limitation of the current framework’s fixed window size. Another challenge is the pipeline nature of FGDT. To train the whole framework, each step requires a separate training procedure. For the first step, a training procedure is required for each group for every single time window. This also applies to the second step, where different formations of each group are modeled for different time windows, requiring an RNN to be trained. Finally, in the third step, a GCN is trained for each group. For this study the training procedure took 110 h. Details of the platform used in the study are given in Table II. We will investigate an end-to-end joint training alternative to this in future work.

A. Experimental Setup

We trained a 100-dimension CBow embeddings on available datasets (Yelp and Amazon) with a window size of 2 and batch size of 512, as CBow has shown to be effective in fraud review detection [6], [15], [29]. To train the HIN-RNN, we used two RNNs in the spatial modeling step with gated recurrent units (GRUs), one learns the hidden state of the group with a hidden state size of 128, and the other learns the collaboration matrix with a hidden state size of 16. Two RNNs were trained jointly with a learning rate of 0.003 and in 3000 epochs to predict the coreview relations between reviewers. To model the temporal behavior, we used an RNN with \( 10^{-4} \) as the learning rate and binary cross-entropy as the objective function. For the training of the GCN, the learning rate was \( 10^{-5} \), the training epochs were 100, and the cross-entropy was used as the objective function.

B. Datasets

Similar to the previous studies [11], [12], we used the Yelp dataset and the Amazon dataset to demonstrate the scalability of FGDT. The Yelp dataset contains reviews ranging from “20 October, 2004” to “10 January, 2015” alongside the labels, reviewer id, item id, the rating, and the date of the review. The Amazon dataset provides the same metadata with reviews ranging from “01 February, 2000” to “10 October, 2010.” Fraudster groups are determined as described in Section III-A1. The details on datasets are provided in Table III. For evaluation, we used 80% of the data for training and 20% for testing.

C. Main Results

We used three well-known metrics to evaluate the performance of FGDT. First, precision

\[ \text{Precision} = \frac{TP}{TP + FP} \]  

(18)

Table II

| Resource   | Specs                                      |
|------------|--------------------------------------------|
| Memory     | 62GB System memory                         |
| Processor  | Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz   |
| GPU Processor | GeForce GTX 1060 6GB                      |

Table III

| Dataset | Reviewers | Items | Reviews | Candidate Groups |
|---------|-----------|-------|---------|------------------|
| Yelp    | 260,277   | 3,044 | 608,398 | 9,952            |
| Amazon  | 42,655    | 6,822 | 53,777  | 2,194            |

IV. EXPERIMENTAL EVALUATION

To demonstrate the effectiveness of FGDT, we compare our approach with the most recent approaches reviewed in Section II: Zhang et al. [12], Ji et al. [11], and the HIN-RNN by Shehnepoor et al. [15].

A. Experimental Setup

W
TABLE IV
COMPARISON BETWEEN THE RESULTS ON FGDT AND THREE STATE-OF-THE-ART STUDIES [11], [12], [15]

| Metrics                      | Yelp        | Amazon     |
|------------------------------|-------------|------------|
|                              | Precision   | Recall     | F1-value   | Precision | Recall | F1-value   |
| Existing Approaches          |             |            |            |           |        |            |
| Zhang et al. [12]            | 0.70        | 0.20       | 0.32       | 0.50      | 0.45   | 0.55       |
| Ji et al. [11]               | 0.83        | 0.60       | 0.69       | 0.82      | 0.92   | 0.86       |
| Shehnepoor et al. [15]       | 0.81        | 0.82       | 0.81       | 0.85      | 0.90   | 0.87       |
| Spatial modeling             |             |            |            |           |        |            |
| Spatial modeling + temporal modeling | 0.74    | 0.70       | 0.72       | 0.85      | 0.90   | 0.87       |
| Spatial modeling + temporal modeling + GCN | 0.76     | 0.80       | 0.78       | 0.83      | 0.85   | 0.84       |
| Spatial modeling + temporal modeling + clustering | 0.83 | 0.93       | 0.86       | 0.87      | 0.93   | 0.90       |
| Spatial modeling + temporal modeling + GCN + clustering (FGDT) | 0.86    | 0.94       | 0.93       | 0.89      | 0.95   | 0.92       |

where TP is the number of true positive samples and FP is the number of false positive samples. We also use recall

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (19)
\]

where FN is the number of false negative samples. Finally, we also use the F1-value

\[
\text{F1-value} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (20)
\]

1) Comparison With Baselines: We compared FGDT against three state-of-the-art approaches. The results are displayed in Table IV. We devised different configurations to show the effectiveness of each step in this study.

Spatial Modeling: In this configuration, we obtain the collaboration matrix of a static network (without temporal modeling). Next, the average representation of the reviewers in the group is used to calculate the final label for each group.

Spatial Modeling + Temporal Modeling: In this configuration, after the spatiomodeling step, the collaboration matrix of the \( N + 1 \)th time window is obtained. Similar to the previous configuration, we then calculate the average of the reviewers’ representation of the group to predict the label of each group.

Spatial Modeling + Temporal Modeling + GCN: Given the spatio-temporal modeling of the groups, in this configuration, the GCN is employed to refine the representation of the reviewers. The refined representation of the reviewers is used to predict the group’s label.

Spatial Modeling + Temporal Modeling + Clustering: This is the complete framework with the GCN refinement module removed, which is to evaluate the performance of the FGDT when the temporal modeling is incorporated.

Spatial Modeling + Temporal Modeling + GCN + Clustering: This is the whole framework.

As shown in Table IV, FGDT outperforms markedly the most recent state-of-the-art approach (Shehnepoor et al. [15]) by 12% for recall and 9% for F1-value on the Yelp dataset. Additionally, FGDT improves the performance by 5% for all metrics on the Amazon dataset. Previous studies overlooked the temporal clues, which are effective in capturing the true behavior of reviewers in social media as group fraudsters change their behavior instantly to camouflage. However, the approach by Shehnepoor et al. [15] is still superior to FGDT with the spatial modeling alone, and the spatial modeling + temporal modeling. The main reason is that in such configurations, i.e., spatial and spatial + temporal modeling, outlier reviewers’ removal is not considered. On the other hand, Shehnepoor et al. [15] removed the outlier reviewers based on the minimum connections, resulting in better performance. The GCN refinement improved the spatio-temporal relations captured in the reviewers’ representation. Previous studies also suffered from a limitation in their inability to exclude genuine reviewers from a fraudster group or a fraudster from a genuine group. With the representations refined through GCN, we can exclude the outlier reviewers. Outlier reviewers are known to be responsible for the increase in \( FN \) and \( FP \). As shown in Table IV, FGDT outperforms all previous approaches. As shown in Table IV, the temporal modeling
slightly improves the performance of FGDT on both the Yelp and Amazon datasets. However, due to users’ short-term activities, temporal modeling is not as effective as spatial modeling with HIN-RNN.

Intuitively, as the reviewers’ representation is refined, the purpose of adding the clustering step is only to achieve a more uniform representation (i.e., a representation of the group with all reviewers of the same type, namely, genuine or fraudster). As a result, adding the clustering step does not necessarily improve the performance any further.

2) Temporal Modeling Effectiveness: In this section, we first explain the ablative study in Table IV, and then we provide explanations related to temporal trends in the datasets.

a) Ablative study: As shown in Table IV, the temporal modeling successfully improves the performance of the FGDT on the Yelp dataset for all three metrics, compared with the spatial modeling. On the other hand, the performance is reduced for the Amazon dataset. This is mostly because the Amazon dataset is considerably smaller than the Yelp dataset. Additionally, the Amazon dataset does not contain groups for almost the first half of the time windows (Fig. 8), while in the Yelp dataset (Fig. 7) groups are available from the first time window. As such, the Yelp dataset provides more information on the relations between reviewers in a group to the RNN. Given the long-range temporal activity of the reviewers in a group, an RNN improves the performance of FGDT, while the lack of temporal activities of reviewers from different groups results in a decrease in the performance on the Amazon dataset.

b) Observation analysis: Fig. 9 shows the number of interactions between genuine (fraudster) reviewers with fraudster (genuine) groups over different time windows. (a) Number of genuine reviewers’ interactions with fraudster groups in the Yelp dataset. (b) Number of genuine reviewers’ interactions with fraudster groups in the Amazon dataset. (c) Number of fraudster reviewers’ interactions with genuine groups in the Yelp dataset. (d) Number of fraudster reviewers’ interactions with genuine groups in the Amazon dataset.

Fig. 9. Number of interactions between genuine (fraudster) reviewers in a fraudster (genuine) group over different time windows. (a) Number of genuine reviewers’ interactions with fraudster groups in the Yelp dataset. (b) Number of genuine reviewers’ interactions with fraudster groups in the Amazon dataset. (c) Number of fraudster reviewers’ interactions with genuine groups in the Yelp dataset. (d) Number of fraudster reviewers’ interactions with genuine groups in the Amazon dataset.
Fig. 10. Refinement effectiveness on reviewers representation on both datasets. (a) Two-dimensional representation of the reviewers in a batch with 512 samples before refinement on the Yelp dataset. (b) Two-dimensional representation of the reviewers in a batch with 512 samples after refinement on the Yelp dataset. (c) Two-dimensional representation of the reviewers in a batch with 512 samples before refinement on the Amazon dataset. (d) Two-dimensional representation of the reviewers in a batch with 512 samples after refinement on the Amazon dataset.

3) Effectiveness of Refinement: In this study, we employed the GCN to refine the representations based on spatio-temporal relations captured between reviewers. To demonstrate the effectiveness of the GCN refinement, we first provide experimental results from the ablative study (Table IV) and then analyze the observations to analyze how the GCN refines the reviewers’ representation on two datasets.

a) Ablative study: As shown in Table IV, the GCN refinement significantly improves the performance of FGDT. The GCN refinement is the most effective step to improve the performance of FGDT, as an improvement of 5%–10% on the Yelp dataset for three metrics is obtained. Similarly, the improvement on the Amazon dataset is 4%–8%. To refine the representations, in the training phase, the GCN takes the representations and the collaboration matrix and fine-tunes the parameters based on the reviewers’ labels. The main reason behind such a performance gain is because of the GCN is able to consider both labels and the collaboration matrix in the refinement. This helps FGDT better utilize the labels of outlier reviewers in a group, thus preventing fraudsters from escaping detection, and genuine reviewers from being mistaken as fraudsters.

b) Comparative analysis of refinement: To observe how the GCN refinement affects the reviewers’ representation, we devised an experiment that first transforms the reviewers’ representation from a 100-D feature space to a 2-D feature space using principal component analysis. For this purpose, we used test batches of 512 reviewers with the maximum number of fraudsters in each dataset (102 fraudsters in the test batch from the Yelp dataset and 39 fraudsters in the test batch from the Amazon dataset). After the transformation, the

| Dataset | Metric | 8 days | 28 days | 112 days |
|---------|--------|--------|---------|----------|
| Yelp    | Precision | 0.84   | 0.86    | 0.80     |
|         | Recall   | 0.90   | 0.94    | 0.84     |
|         | F1-value | 0.87   | 0.93    | 0.82     |
| Amazon  | Precision | 0.82   | 0.89    | 0.75     |
|         | Recall   | 0.85   | 0.95    | 0.78     |
|         | F1-value | 0.83   | 0.92    | 0.76     |
two most significant dimensions are considered to represent the feature space. Fig. 10 shows the effectiveness of the GCN refinement on the representations of the reviewers.

As shown, the GCN effectively refines the genuine and fraudster representation based on the relations in the group on the Yelp dataset. With the refinement, the reviewers are significantly discriminated based on their label [Fig. 10(a) versus (b)]. Similarly, the reviewers’ representation is also refined after applying the GCN to their representation on the Amazon dataset [Fig. 10(c) versus (d)]. However, the refinement on the Amazon dataset is not as discriminative as on the Yelp dataset. Intuitively, much fewer outlier reviewers (and generally genuine/fraudster groups) exist in the Amazon dataset. As a result of fewer contacts by outlier reviewers in a group, the refinement effectiveness also reduces. This results in lower discrimination on the Amazon dataset, accordingly.

4) Effects of Clustering in Determining Outlier Reviewers: As explained in Section III-D1, to reduce the effects of genuine (fraudster) reviewer(s) in a fraudster (genuine) group, we used the $k$-means clustering algorithm to remove the outlier reviewers.

a) Ablative study: The clustering removes fraudster reviewer(s) from a genuine group, and genuine reviewers from a fraudster group, thus reducing the $FP$. This, in turn, results in better precision. On the other hand, with the removal of the genuine reviewer(s) from a fraudster group, the fraudster group’s probability to be a fraudster is accordingly increased, thus reducing the $FN$. As a result of a decrease in $FN$, recall is also improved. However, the improvement in the F1-value for the Yelp dataset is significantly higher than the F1-value improvement on the Amazon dataset. This is mainly because the camouflage activity (the number of interactions between fraudster reviewers and genuine groups) in the Yelp dataset is higher.

b) Comparative analysis of clustering: To demonstrate the effectiveness of the proposed removal strategy, we also used two different strategies from previous studies.

1) Removing reviewers if their distances (Euclidean) from the centroid are above a certain threshold [11] in the 1-means algorithm (only one centroid).

2) Removing reviewers with minimum connections as suggested by Shehnepoor et al. [15].

Figs. 11 and 12 display the performance of FGDT on different removal strategies. As shown the performance is improved against two other strategies. Using a threshold to remove the reviewers based on a representation distance from the centroid, likely removes some reviewers from a group, but not all groups are necessarily a mix of fraudsters and genuine reviewers.

Removing reviewers based on a minimum number of connections results in inaccurate removals. This is because some fraudsters make connections with multiple genuine reviewers in a group to escape detection. Similarly, genuine reviewers are also unintentionally involved in fraudster activities, because they happen to have reviewed the same item with the same rating and similar semantics. FGDT, on the other hand, utilizes the encoded spatio-temporal relations between reviewers and then refines the representations based on the given relations through the GCN. The proposed strategy for the clustering step will remove the outlier reviewer(s) in the feature space. If no outlier reviewers are recognized in a group, then all the reviewers remain in the group.

c) Observations: To compare the effectiveness of different clustering versions, we used three different clustering algorithms: first, we used the Gaussian mixture model (GMM) as the clustering algorithm. In this algorithm, the reviewers with a distance greater than one standard deviation (Mahalanobis distance > 1) were removed. As for the second algorithm we used the $k$-medians [30], with the median as the centroid. This, intuitively better represents the centroid, as the average of representations may not be a suitable centroid of a group,
due to the existence of outlier reviewers. The second version utilizes the $k$-means [31], as FGDT. Figs. 13 and 14 show the performance of FGDT based on $k$-means, $K$-medians, and GMM. Using the mean to calculate the centroid of groups improves the performance of FGDT compared with the version, which employed the median to obtain the centroid. Similarly using only the centroid from $k$-means demonstrated a better performance than using the GMM’s Mahalanobis distance. As shown, the improvement is the same almost for both datasets.

V. CONCLUSION

Previous studies incorporated static networks to model a collaboration matrix on the fraudster group detection task, overlooking temporal features, while suffering from limitations in incorporating reviewers’ influence on each other.

In this study, we proposed a four-step framework to address these limitations: spatial modeling of the reviewers’ coreview behaviors in different time windows, temporal modeling of spatial relations in a sequence of time windows, the reviewers’ representation refinement through GCN, and the outlier removal from the groups through $k$-means clustering before the final classification of groups. FGDT effectively improved the performance of FGDT by $5\%$ (4%), $12\%$ (5%), and $12\%$ (5%) on the Yelp (Amazon) dataset, respectively.

For future work, the stochastic modeling of the reviewers’ relations can be considered to provide a better behavioral representation of the reviewers, as most fraudsters randomly change their grouping relations to escape detection.

REFERENCES

[1] A. Mukherjee, B. Liu, and N. Glance, “Spotting fake reviewer groups in consumer reviews,” in Proc. 21st Int. Conf. World Wide Web, Apr. 2012, pp. 191–200.

[2] L. Y. Xiang, Y. Li, W. Hao, P. Yang, and X. B. Shen, “Reversible natural language watermarking using synonym substitution and arithmetic encoding,” Comput. Mater. Continua, vol. 55, no. 3, pp. 541–559, 2018.

[3] J. Ye and L. Akoglu, “Discovering opinion spammer groups by network footprints,” in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases. Cham, Switzerland: Springer, 2015, pp. 267–282.

[4] N. Jindal and B. Liu, “Opinion spam and analysis,” in Proc. Int. Conf. Web Search Web Data Mining, 2008, pp. 219–230.

[5] A. Mukherjee, B. Liu, and N. Glance, “What Yelp fake review filter might be doing?” in Proc. Int. AAAI Conf. Web Social Media, 2013, vol. 7, no. 1, pp. 1–10.

[6] S. Shenhnpoor, R. Togneri, W. Liu, and M. Bennamoun, “DFraud: Multi-component fraud detection free of cold-start,” IEEE Trans. Inf. Forensics Security, vol. 16, pp. 3456–3468, 2021.

[7] S. Shenhnpoor, M. Salehi, R. Farahbakhsh, and N. Crespi, “NetSpam: A network-based spam detection framework for reviews in online social media,” IEEE Trans. Inf. Forensics Security, vol. 12, no. 7, pp. 1585–1595, Jul. 2017.

[8] A. Rayana and L. Akoglu, “Collective opinion spam detection: Bridging review networks and metadata,” in Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2015, pp. 985–994.

[9] S. Shenhnpoor, R. Togneri, W. Liu, and M. Bennamoun, “ScoreGAN: A fraud review detector based on multi task learning of regulated GAN with data augmentation,” 2020, arXiv:2006.06561.

[10] M. Allahbakhsh, A. Ignjatovic, B. Benatallah, S.-M.-R. Beheshi, E. Benamou, and N. Foo, “Collusion detection in online rating systems,” in Proc. Asia-Pacific Web Conf. Berlin, Germany: Springer, 2013, pp. 196–207.

[11] S.-J. Ji et al., “A burst-based unsupervised method for detecting review spammer groups,” Inf. Sci., vol. 536, pp. 454–469, Oct. 2020.

[12] F. Zhang, X. Hao, J. Chao, and S. Yuan, “Label propagation-based approach for detecting review spammer groups on E-commerce websites,” Knowl.-Based Syst., vol. 193, Apr. 2020, Art. no. 105520.

[13] G. Xia, M. Hu, C. Ma, and M. Danelo, “GSCPM: CPM-based group spamming detection in online product reviews,” in Proc. IEEE Int. Conf. Commun. (ICC), May 2019, pp. 1–6.

[14] W. Chen, Z. Liu, X. Sun, and Y. Wang, “A game-theoretic framework to identify overlapping communities in social networks,” Data Mining Knowl. Discovery, vol. 21, no. 2, pp. 224–240, 2010.

[15] S. Shenhnpoor, R. Togneri, W. Liu, and M. Bennamoun, “HIN-RNN: A graph representation learning neural network for fraudster group detection with no handcrafted features,” IEEE Trans. Neural Netw. Learn. Syst., early access, Nov. 9, 2021, doi: 10.1109/TNNLS.2021.3123876.

[16] A. Mukherjee et al., “Spotting opinion spammers using behavioral footprints,” in Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2013, pp. 632–640.

[17] K. Chen, L. Yao, D. Zhao, Y. Wang, X. Chang, and F. Nie, “A semi-supervised recurrent convolutional attention model for human activity recognition,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 5, pp. 1747–1756, May 2020.

[18] D. Zhang, L. Yao, K. Chen, S. Wang, X. Chang, and Y. Liu, “Making sense of spatio-temporal preserving representations for EEG-based human intention recognition,” IEEE Trans. Cybern., vol. 50, no. 7, pp. 3033–3044, Jul. 2020.

[19] C. Leng, H. Zhang, G. Cai, I. Cheng, and A. Basu, “Graph regularized $L_p$ smooth non-negative matrix factorization for data representation,” IEEE/CAA J. Autom. Sinica, vol. 6, no. 2, pp. 584–595, Mar. 2019.

[20] X. Liu, M. Yan, L. Deng, G. Li, X. Ye, and D. Fan, “Sampling methods for efficient training of graph convolutional networks: A survey,” IEEE/CAA J. Autom. Sinica, vol. 9, no. 2, pp. 205–234, Feb. 2022.

[21] Q. Lian, W. Yan, X. Zhang, and S. Chen, “Single image rain removal using image decomposition and a dense network,” IEEE/CAA J. Autom. Sinica, vol. 6, no. 6, pp. 1428–1437, Nov. 2019.

[22] Z. Liu, X. Luo, and Z. Wang, “Convergence analysis of single latent factor-dependent, nonnegative, and multiplicative update-based nonnegative latent factor models,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 4, pp. 1737–1749, Apr. 2021.

[23] D. Wu, X. Luo, M. Shang, Y. He, G. Wang, and X. Wu, “A data-characteristic-aware latent factor model for web services QoS prediction,” IEEE Trans. Knowl. Data Eng., vol. 34, no. 6, pp. 2525–2538, Jun. 2020.

[24] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in Proc. Int. Conf. Very Large Data Bases, vol. 1215, 1994, pp. 487–499.

[25] D. Wang, S. Gu, X. Zhao, and X. Xu, “Graph-based review spammer group detection,” Knowl. Inf. Syst., vol. 55, no. 3, pp. 571–597, 2018.

[26] Y. Ren and D. Ji, “Neural networks for deceptive opinion spam detection: An empirical study,” Inf. Sci., vol. 385–386, pp. 213–224, Apr. 2017, doi: 10.1016/j.ins.2017.01.015.

[27] L. White, R. Togneri, W. Liu, and M. Bennamoun, “Generating bags of words from the sums of their word embeddings,” in Proc. 17th Int. Conf. Intell. Text Process. Comput. Linguistics, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, A. Gelbukh, Ed. Berlin, Germany: Springer-Verlag, 2018, pp. 91–102.
[28] L. White, R. Togneri, W. Liu, and M. Bennamoun, “How well sentence embeddings capture meaning,” in Proc. 20th Australas. Document Comput. Symp., New York, NY, USA, Dec. 2015, doi: 10.1145/2838931.2838932.

[29] Y. Ren and Y. Zhang, “Deceptive opinion spam detection using neural network,” in Proc. 26th Int. Conf. Comput. Linguistics, 2016, pp. 140–150.

[30] H.-S. Park and C.-H. Jun, “A simple and fast algorithm for k-medoids clustering,” Expert Syst. Appl., vol. 36, no. 2, pp. 3336–3341, Mar. 2009.

[31] G. Hamerly and C. Elkan, “Learning the $k$ in $k$-means,” in Proc. Adv. Neural Inf. Process. Syst., vol. 16, 2003, pp. 281–288.

Saeedreza Shehnepoor received the B.S. degree in information technology from the Sharif University of Technology, Tehran, Iran, in 2014, and the M.S. degree in computer science and information networks from the University of Tehran, Tehran, in 2016. He is currently pursuing the Ph.D. degree with The University of Western Australia, Perth, WA, Australia.

His research interests include machine learning and specifically issues related to the fraud contents in social media using graph theory and deep learning.

Roberto Togneri (Senior Member, IEEE) received the Ph.D. degree from The University of Western Australia, Perth, WA, Australia, in 1989.

In 1988, he joined the School of Electrical, Electronic and Computer Engineering, The University of Western Australia, where he is currently an Associate Professor and also leads the Signal Processing and Recognition Laboratory. He is the Chief Investigator on three Australian Research Council (ARC) Discovery Project research grants. His research interests include signal processing and pattern recognition, including feature learning, detection and enhancement of audio signals, statistical and neural network models for speech and speaker recognition, audio–visual recognition and biometrics, and related aspects of language modeling and understanding.

Dr. Togneri is an Area Editor of IEEE Signal Processing Magazine Columns and Forums.

Wei Liu (Member, IEEE) received the Ph.D. degree from The University of Newcastle, Callaghan, NSW, Australia, in 2003.

She is currently a full-time Teaching and Research Academic with the Department of Computer Science and Software Engineering, The University of Western Australia, Perth, WA, Australia. Her current industry-related research projects include knowledge graph refinement for geological survey reports, incident log analysis and visualization, short-term traffic prediction, and cognitive computing for asset management. She has received three Australian Research Council (ARC) Grants and managed several industry grants. Her research interests include knowledge discovery from natural language text, deep learning methods for knowledge graph construction and analysis, and sequential data mining and forecasting in traffic and water consumption domains.

Mohammed Bennamoun (Senior Member, IEEE) received the M.Sc. degree in control theory from Queen’s University, Kingston, ON, Canada, in 1992, and the Ph.D. degree in computer vision from the Queensland University of Technology (QUT), Brisbane, QLD, Australia, in 1996.

In 1993, he lectured robotics at Queen’s University and then joined QUT as an Associate Lecturer. From 2007 to 2012, he served as the Head of the School of Computer Science and Software Engineering, The University of Western Australia, Perth, WA, Australia, where he is currently a Winthrop Professor. His research interests include computer vision (particularly 3-D), such as object recognition and biometrics, machine/deep learning, robotics, such as obstacle avoidance and robot grasping, signal/image processing, and control theory.

Dr. Bennamoun served as a member of the Australian Research Council (ARC) College of Experts from 2012 to 2015 and from 2019 to 2021. He also served as a member of the ARC Excellence Research for Australia in 2018. He is also a member of the College of Assessors for the Ministry of Business, Innovation and Employment, New Zealand.