Social Network Analysis and Mining (2020) 10:85
https://doi.org/10.1007/s13278-020-00697-w

ORIGINAL ARTICLE

GCNEXT: graph convolutional network with expanded balance theory for fraudulent user detection

Wataru Kudo[1] · **Mao Nishiguchi**[1] · **Fujio Toriumi**[1]

Received: 6 January 2020 / Revised: 22 September 2020 / Accepted: 23 September 2020 / Published online: 8 October 2020

Abstract
Rating platforms provide users with useful information on products or other users. However, fake ratings are sometimes generated by fraudulent users. In this paper, we tackle the task of fraudulent user detection on rating platforms. We propose GCNEXT (Graph Convolutional Network with Expended Balance Theory), an end-to-end framework based on graph convolutional networks (GCNs) and expanded balance theory, which properly incorporates both the signs and directions of edges. The experimental results on seven real-world datasets show that the proposed framework performs better, or even best, in most settings. In particular, this framework shows remarkable stability in inductive settings, which is associated with the detection of new fraudulent users on rating platforms. Furthermore, using expanded balance theory, we provide new insight into the behavior of users in rating networks that fraudulent users form a faction to deal with the negative ratings from other users. The owner of a rating platform can detect fraudulent users earlier and constantly provide users with more credible information by using the proposed framework.

Keywords Graph convolutional networks · Rating networks · Fraud detection · Balance theory

1 Introduction

With the recent spread of e-commerce platforms, user-generated content, such as ratings or reviews, is becoming more and more important in the decision-making process of consumers. Many online marketplaces or trading platforms (e.g., Amazon, Yelp, and Bitcoin Alpha) allow users to rate the quality of contents, or the trustworthiness of other users. The average ratings or posted comments help inform users before purchasing, visiting, or trading. Because users attach great weight to such information in personal consumption, fraudulent users have great financial motivation to generate fake ratings (Lappas et al. 2016; Luca et al. 2016; Kumar et al. 2018a). Identifying and understanding fraudulent users may help platform providers maintain their credibility. Our goal is to detect fraudulent users on rating platforms based on their social rating behavior.

Previous studies (Akoglu et al. 2015; Kumar et al. 2018b) have shown that the graph-based approaches are effective for the task of fraud detection. Rev2 (Kumar et al. 2018b), the state-of-the-art approach of this task appropriately models the interdependencies among users on rating platforms. Following the success of them, we also regard online rating data as a social graph. We call this graph a "rating network." Rating networks have two characteristic properties: 1) signed edges and 2) directed edges. Reviews are associated with rating values in most cases, and the meanings of edges greatly differ depending on the rating values. The edges of low ratings are associated with a negative relationship, while high ratings are associated with a positive one. Here, a rating network should be regarded as a kind of signed network, even if the rating values themselves are all positive. The direction of edges is also important, especially in user-to-user homogeneous rating networks, in which users can both rate and be rated. Rating others and being rated by others are different interactions; therefore, distinguishing between them could provide a better model of users behavior.

Recently, graph convolutional networks (GCNs) have performed remarkably well on several graph-related tasks,

✉ Wataru Kudo
kudo@torilab.net; kudo@crimson.q.t.u-tokyo.ac.jp

Mao Nishiguchi
nishiguchi@crimson.q.t.u-tokyo.ac.jp

Fujio Toriumi
tori@crimson.q.t.u-tokyo.ac.jp

[1] The University of Tokyo, Tokyo, Japan

including node classification (Kipf et al. 2016a; Veličković et al. 2017) and link prediction (Kipf et al. 2016b; Schlichtkrull et al. 2018). One of the appealing characteristics of GCNs is their end-to-end learning manner, which incorporates both the graph structures and attributes of each node. In this respect, GCNs appear to be suitable for the task of detecting fraudulent users on online rating platforms. However, in applying existing GCNs to rating networks, it is necessary to handle both the signs and directions of the edges.

As for handling signs, the signed graph convolutional network (SGCN) (Derr et al. 2018) is a successful approach. In SGCN, they effectively incorporate 2-hop relationships between nodes by applying balance theory (Heider 1946; Cartwright et al. 1956), a social theory for signed networks. However, SGCN is designed for undirected networks and cannot consider 2-hop relationships properly in directed networks. It is assumed that the appropriate expansion of balance theory is effective to settle this problem and capture the social behavior of users on rating networks.

The aim of this paper is to construct a detection model using expanded balance theory and GCNs. We conduct experiments on seven real-world datasets. Three of them (Bitcoin OTC, Bitcoin Alpha, and Epinions) are user-to-user homogeneous rating networks, and the others are use-to-product heterogeneous rating networks derived from Amazon.com.[1] In the experiments, we compare the existing state-of-the-art approach, network-embedding-based models, and GCN-based models including the proposed framework. Our main contribution in this paper is:

- We propose GCNEXT, a novel GCN-based end-to-end framework using expanded balance theory for fraudulent user detection that incorporates both the signs and directions of edges.
- From the experimental results, we show that the proposed framework performs well, or even best compared to the others we experimented. It also shows remarkable stability in inductive settings, which is associated with the detection of new fraudulent users on a rating platform.
- We present an analysis of a rating platform based on expanded balance theory and provide new insight into the behavior of fraudulent users in rating networks.

It is worth noting that the proposed method maintains high performance even when few edges are observed. In practical cases, the proposed framework is useful when the owner of a rating platform must cope with the detection of new fraudulent users quickly, as well as when applied to general detection tasks.

[1] https://www.amazon.com.

This paper is a revised and expanded version of Kudo et al. (2019). Compared to the previous version, we refer an additional work named Edgecentric (Shah et al. 2016) in Sect. 2, and we add SIDE (Kim et al. 2018) to the baselines described in Sect. 4. Furthermore, we conduct experiments on three additional datasets and a new setting to evaluate the robustness whose results are shown in Fig. 5.

The rest of this paper is organized as follows: In Sect. 2, we review existing studies related to our work. We then describe our proposed framework in Sect. 3, followed by Sect. 4, in which we discuss the experimental results and present a discussion. Finally, we conclude in Sect. 5.

2 Related work

2.1 Fraudulent user detection on online rating platforms

Several efforts have been made in exploring the detection of fraudulent users. Existing research can be categorized into two major approaches: (1) feature-based approaches and (2) graph-based approaches.

2.1.1 Feature-based approaches

Feature-based approaches attempt to create a feature that represents each user's behavior. Features from texts (Fayazi et al. 2015; Sandulescu et al. 2015) and timestamps (Xie et al. 2012; Minnich et al. 2015) have been frequently studied. More related to our work, another work (Lim et al. 2010) modeled consensus, or the wisdom of crowds, from rating values on reviews. Although those approaches show good interpretability, they are not easy to generalize to a different platform or domain.

2.1.2 Graph-based

Most graph-based approaches use the relations among the entities in a rating network. Approaches based on belief propagation (Akoglu et al. 2013) or iterative algorithms (Wang and Guan 2012; Kumar et al. 2018b) have been common. Other than these approaches, there is an anomaly detection approach in unsupervised fashion leveraging edge information (Shah et al. 2016).

The most successful approach is Rev2 (Kumar et al. 2018b), an iterative algorithm with theoretical guarantees that outperforms the existing methods, including feature-based approaches. It deals with a rating network as a signed network.

Although Rev2 (Kumar et al. 2018b) can be used in both supervised and unsupervised settings, there appears to be great room for improvement in its performance with

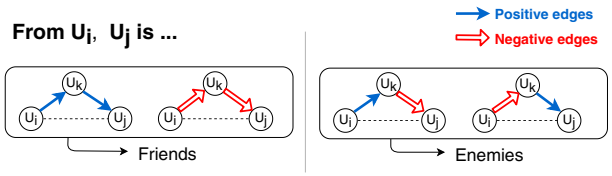

Fig. 1 Inference of relationship between the 2-hop neighbors in SGCN (Derr et al. 2018)

supervised settings. It learns the representation of entities from the relations among them in an iterative learning manner, which cannot be used to jointly learn the supervised task of fraudulent user detection. An end-to-end model that appropriately incorporates the relations among entities and can jointly learn the task of fraudulent user detection is considered a feasible means to improve performance in supervised settings.

2.2 Graph convolutional networks

The application of neural networks on graph data has seen recent rapid development. Taking rating networks as edge-attributed networks, the relational graph convolutional network (R-GCN) (Schlichtkrull et al. 2018) appears related to our task of detecting fraudulent users. R-GCN can incorporate different discrete types of edges in the graph. More recently, Jiang et al. (2019) proposed a GCN-based anomaly detection method and applied it to fraud detection.

As for the efforts on signed networks, the signed graph convolutional network (SGCN) (Derr et al. 2018) showed good performance in the link sign prediction task (Leskovec et al. 2010a). In SGCN (Derr et al. 2018), they claimed that special attention is needed to handle negative edges correctly in GCNs and then designed convolutional operations based on balance theory (Heider 1946; Cartwright et al. 1956), a social theory for signed networks that has shown its effectiveness in related tasks such as signed network embeddings (Leskovec et al. 2010; Kim et al. 2018). Below, we provide additional descriptions focusing on how they apply balance theory in SGCN.

2.3 Balance theory for SGCN Derr et al. (2018)

Balance theory formulates the intuition that "the friends of my friends are friends, and the enemies of my friends are enemies." In SGCN, they applied this theory to infer the relationship between entities without the direct edge and designed convolutional operations based on the inference. Figure 1 shows how the relationship of 2-hop neighbors is suggested in SGCN. Although SGCN is designed for undirected networks, we illustrate for convenience the case where it is applied to signed directed networks such as

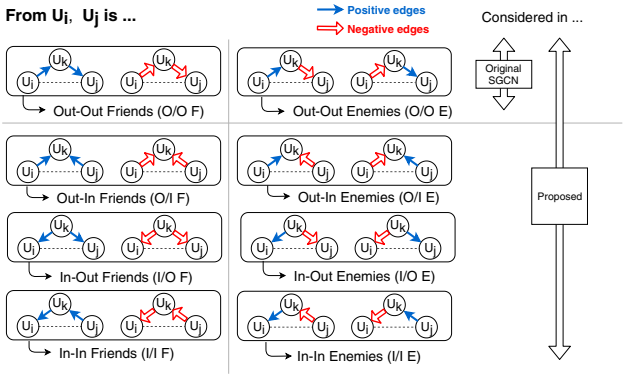

Fig. 2 Inference of relationships between 2-hop neighbors by proposed definition

rating networks. According to their equations, the considered directions are limited and the inferred relationship is one of two classes, "Friends" or "Enemies." Their implementation of balance theory makes it possible to incorporate signs of edges. However, there remains the problem of how to handle the edge directionality properly. In SGCN, the final representations of each node are informed by limited types of neighbors.

3 GCNEXT: graph convolutional network with expanded balance theory

One of our key motivations is to build a classification model that properly incorporates the sign and direction of edges. To achieve that, we propose an expansion of balance theory and design convolutional operations based on the expanded balance theory following the way in SGCN (Derr et al. 2018). In this section, we provide the basic theory behind the proposed framework, followed by the formulas in which the theory is incorporated.

3.1 Expanded balance theory for signed directed networks

We hypothesize that the basic idea of balance theory can be effectively applied to the task of node classification on online rating platforms. However, considering the properties of rating networks, where a different edge direction implies a different relation between nodes, it could be necessary to provide more detailed definitions of "Friends" and "Enemies" than that of SGCN (shown in Fig. 1). Below, we introduce our expansion of balance theory that establishes eight distinct types of relationships between 2-hop neighbors.

An edge has two binary attributes (sign and direction), so there are four possible patterns in an edge. Then, taking 2-hop neighbors into account, there are 16 patterns of

combinations. Consequently, as shown in Fig. 2, we can define eight different types of "Friends" and "Enemies."

With this new definition, we can distinguish four different meanings each for "Friends" and "Enemies." For example, we can explain each of the "Friends" with its corresponding intuition as follows:

- "Out-Out Friends" (hereinafter, "O/O F") is derived from the intuition "Someone liked by my favorite person is my friend" (as well as its opposite, replacing "liked" and "favorite" with their antonyms).
- "Out-In Friends" ("O/I F") is derived from the intuition "Someone who has the same opinion of the same entity as I do is my friend."
- "In-Out Friends" ("I/O F") is derived from the intuition "Someone who is rated in the same way by the same person as I am is my friend."
- "In-In Friends" ("I/I F") is derived from the intuition "Someone who likes someone who likes me is my friend" (as well as its opposite).

The newly defined "Enemies" is understood in the same way as above. Abbreviations used in the rest of the paper also follow those of "Friends" (e.g., "Out-Out Enemies" as "O/O E"). Note that the new "Friends" and "Enemies" are defined with direction because they contain asymmetric relations.

Determining who a user's "Friends" and "Enemies" are would appear to provide a great hint for detecting a fraudulent user on an online rating platform. We propose a framework that defines the convolutional operations based on the newly defined "Friends" and "Enemies." Below, we describe the details of the proposed framework.

3.2 Flow of model construction

In SGCN, the convolutional operations were defined based on the inference of the relationship between the nodes, as illustrated in Fig. 1. According to the original formulas, the representation of each node is updated only along with the node's outgoing edges (i.e., the edges from that node to others) in each layer. This could be a limitation, especially when the direction of the edges has a significant meaning in a network.

The proposed framework designs the convolutional operation based on the newly introduced definitions shown in Fig. 2, which enables the models to take both the signs and directions of the edges into account and to detect various types of interactions.

In SGCN, each layer has two aggregators with different roles. By contrast, the proposed framework has four aggregators in the first layer and eight aggregators in the second layer, each of which works differently to properly incorporate the relationship between nodes. We begin by providing the notations.

Table 1 Notations

Notation	Description
$\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$	Set of nodes
$sign \in \{+, -\}$	Sign of edge
$dir \in \{out, in\}$	Edge direction
x_i	Initialized vector of u_i
$N_i^{sign,dir}$	Neighbors of u_i along with Edges of sign and direction
$h_i^{sign,dir}$	Hidden representation of u_i After the 1st layer
$W_1^{sign,dir}$	Weight matrix of the 1st layer
$agg \in AGG$	Aggregator in the 2nd layer
$\mathcal{N}_i$, h'_i	Map functions
W_2^{agg}	Weight matrix of the 2nd layer
W_3	Weight matrix of fully Connected layer
Z	final embedding matrix

3.2.1 Notations

We use $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ as the set of n nodes, and $A \in \mathbb{R}^{n \times n}$ denotes the adjacency matrix. We use $sign \in \{+, -\}$ and $dir \in \{out, in\}$ to describe the edge's sign and direction, respectively. $X \in \mathbb{R}^{n \times d_0}$ denotes the initialized feature matrix of the nodes, where d_0 is the dimension of the initialized vectors and ith row x_i denotes the initialized vector of u_i. Table 1 shows the main notations of this paper.

In the rest of this section, we describe how we construct the initialized vectors of each node, followed by detailed explanations and the formulas of our framework.

3.2.2 Initialized vectors

We construct the initialized vector of each node from its rating distribution, which is one of the most naive ways representing users on rating platforms. We count the rating values that the nodes gave to others and were given by others separately and then normalize each, followed by concatenation. Figure 3 illustrates an example of how the initialized vector of user u_i would be constructed in a five-point scale user-to-user homogeneous rating network. Note that in the case of a user-to-item bipartite network such as Amazon.com, half of each node vector ("Rated-by-Others" part for user nodes and "Rate-Others" part for item nodes) is set to zero.

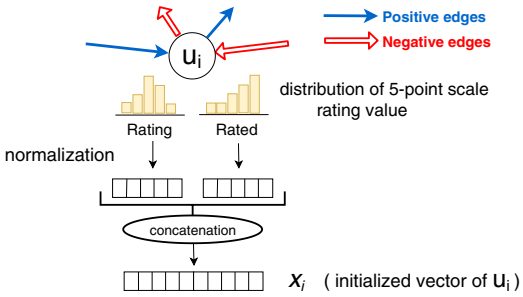

Fig. 3 Constructing the initialized vector of u_i. Note that there should be more edges around u_i but we omit them for simplification

3.2.3 Details and formulas

Following SGCN, our framework has two convolutional layers and one fully connected layer.

In the first convolutional layer, we construct four representation vectors for each node. Each vector is calculated by the aggregation of the neighbor nodes along with the four patterns of edges (derived from two binary attributes of edges: signs and directions). In each process, we first average the initialized vectors of the corresponding neighbors and then concatenate the averaged vector with the initialized vector of the node itself, followed by transformation by the corresponding weight matrix and nonlinear activation function. The output of this first layer is four vector representations of u_i, which are denoted as $h_i^{+,out}$, $h_i^{-,out}$, $h_i^{+,in}$, $h_i^{-,in}$. With $h_i^{sign,dir}$ as the notation of the four vectors in one statement, we formally define the 1st layer as follows:

$$h_i^{sign,dir} = \tanh\left(W_1^{sign,dir}\left[\sum_{j\in N_i^{sign,dir}} \frac{x_j}{|N_i^{sign,dir}|}, x_i\right]\right) \tag{1}$$

where tanh() is hyperbolic tangent function and $N_i^{sign,dir}$ denotes the set of u_i itself and the neighbors connected with u_i by the corresponding edges. Each $W_1^{sign,dir} \in \mathbb{R}^{d_1 \times 2d_0}$ is a linear transformation matrix to be learned, corresponding to the sign and direction of edges. Note that d_1 is the dimension of $h^{sign,dir}$.

In the second convolutional layer, with the input of hidden representations after the first layer $\{h_i^{sign,dir} | i = 1, 2, \ldots, n\}$ and adjacency matrix A, we get eight types of vectors for each node. Here, the eight types correspond to the all eight kinds of relations from a node to a 2-hop neighbor (shown in Fig. 2). In this layer, we have eight independent aggregators that work differently. Each is responsible for aggregating certain types of neighbors' output of the first layer, along with certain types of edges. Here, we define $AGG = \{agg\} = \{$O/O F, O/I F, I/O F, I/I F, O/O E, O/I E, I/O E, I/I E$\}$ as a set of eight aggregators. In this layer, the objectives of aggregation (i.e., along with what kind of edge and which representation of $h^{sign,dir}$ to select) differ depending on the aggregators. Furthermore, each aggregator has two different objectives to aggregate that are associated with the signs. Here, we define two map functions denoted by $\mathcal{N}_i$ and h'_i, both of which take agg and $sign$ as arguments, and return the type of neighbors and the representation to aggregate, respectively. The two functions are defined in Table 2.

Table 2 Definition of two map functions. $\overline{sign}$ means the complement of the sign

$agg \in$ AGG	Map functions	
	$\mathcal{N}_i(agg, sign)$	$h'_i(agg, sign)$
O/O F	$N_i^{sign,out}$	$h^{sign,out}$
O/I F	$N_i^{sign,out}$	$h^{sign,in}$
I/O F	$N_i^{sign,in}$	$h^{sign,out}$
I/I F	$N_i^{sign,in}$	$h^{sign,in}$
O/O E	$N^{sign,out}$	$h^{\overline{sign},out}$
O/I E	$N^{sign,out}$	$h^{\overline{sign},in}$
I/O E	$N^{sign,in}$	$h^{\overline{sign},out}$
I/I E	$N^{sign,in}$	$h^{\overline{sign},in}$

Note that we obtain the output representations of this layer z_i as the concatenation of vectors from the eight aggregators. Now we can formally define the second layer as follows:

$$z_i^{(agg,sign)} = \sum_{j\in\mathcal{N}_i(agg,sign)} \frac{h'_j(agg, sign)}{|\mathcal{N}_i(agg, sign)|} \tag{2}$$

$$z_i^{agg} = \tanh(W_2^{agg}[z_i^{(agg,+)}, z_i^{(agg,-)}, h'_i(agg, +), h'_i(agg, -)]) \tag{3}$$

$$z_i = \Big\|_{agg\in AGG} z_i^{agg} \tag{4}$$

where $W_2^{agg} \in \mathbb{R}^{d_2 \times 4d_1}$ is the transformation matrix to be learned in this layer.

Finally, at each node, we feed z_i to the fully connected layer, getting the probability that u_i is a fraudulent user. We then obtain the final output as follows:

$$\hat{y} = sigmoid(W_3 Z) \tag{5}$$

where $W_3 \in \mathbb{R}^{1\times 8d_2}$ is the weight matrix to be learned and Z is the final embedding matrix of nodes, composed of $\{z_i | i = 1, 2, \ldots, n\}$. The cross-entropy between $\hat{y}$ and the ground truth label is minimized. Following SGCN, we also use the stochastic gradient descent (SGD) style of updating for the parameters to be learned.

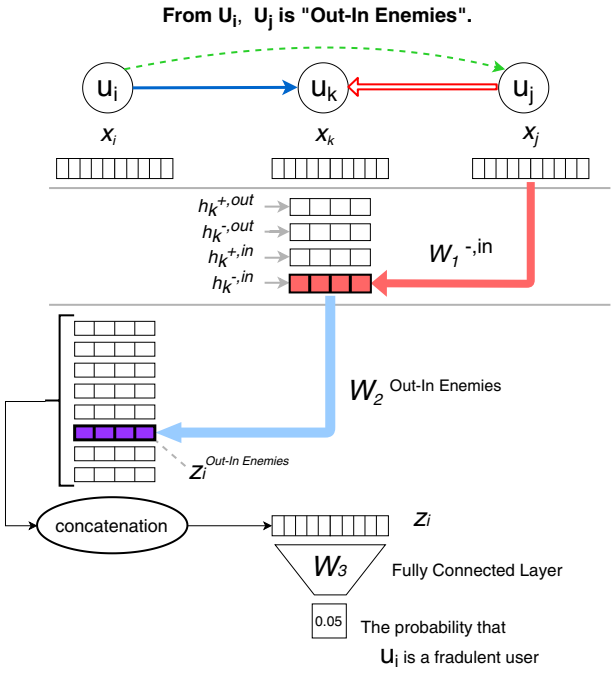

Fig. 4 Process of incorporating the features of a 2-hop neighbor during the classification of node u_i. Note that u_j and u_k are just examples of related nodes, while all 1-hop and 2-hop neighbors of u_i are considered in actual cases

With the above formulations, the proposed framework is expected to learn what kind of neighbors are important, as well as, how to transform the neighbor representations to be more informative. In Fig. 4, we show an example of how a user u_i is classified in our framework.

In comparing the proposed framework and SGCN, we can briefly summarize the difference: The final representation of a node can be informed by all eight types of 2-hop relationships in the proposed framework, while only two ("O/O F" and "O/O E") were used in SGCN.

Note that the proposed framework can incorporate all eight types of 2-hop relationships, but does not always require all of them. For example, in a heterogeneous bipartite rating network such as Amazon, only two types ("O/I F" and "O/I E") of 2-hop relationship are incorporated in representations of each user node. Consequently, the proposed framework can be applied generally to various types of rating networks, including user-to-product rating networks, as well as user-to-user trust networks.

4 Experiments

4.1 Datasets

We adopted the four real-world rating networks used in the experiments of Rev2 (Kumar et al. 2018b): Bitcoin OTC, Bitcoin Alpha, Amazon, and Epinions. The definition of ground truth (i.e., whether some users are fraudulent or benign) also follows their experiments. In addition, we also adopted three datasets to verify the generalization performance for bipartite rating networks. Ratings in all datasets have timestamps used in the experiments of inductive setting (Sect. 4.3.2). Below, we provide detailed descriptions of the seven datasets.

- *Bitcoin OTC* is a homogeneous user-to-user trust network of Bitcoin users on OTC platform (Kumar et al. 2016). The set of rating values is $\{x \in \mathbb{Z} \mid -10 \leq x \leq 10, x \neq 0\}$. As for ground truth, the platform's founder and users he rated highly positively (≥ 5) are defined as benign, and the users whom these benign users uniformly rated negatively (≤ -5) are defined as fraudulent.
- *Bitcoin Alpha* is also a user-to-user trust network of Bitcoin users on the Alpha platform (Kumar et al. 2016). The set of rating values is the same as those of Bitcoin OTC. Ground truth is defined in a similar way to OTC, starting from the founder of this platform.
- *Epinions* is a user-to-post rating network (Massa et al. 2007) with integer rating values from 1 to 6. Ground truth is defined using a user-to-user trust network (Massa et al. 2007), which is independent of the user-to-post rating network. A user is defined as benign if its total trust rating is $\geq +10$ but as fraudulent if ≤ -10. Note that if a user rates multiple posts by another user, there will be multiple edges between those two users in the graph.
- *Amazon* is a user-to-product bipartite rating network (McAuley et al. 2013) with a five-point rating scale. The helpfulness vote, which can be used to indicate malicious behavior (Fayazi et al. 2015), is used to define ground truth. Benign users are those who receive at least 50 votes with a ratio of helpful-to-total votes of ≥ 0.75. Those who receive at least 50 votes with a ratio of helpful-to-total votes of ≤ 0.25 are defined as fraudulent.
- *Amazon App, Amazon Music, and Amazon Home* are also user-to-product bipartite rating networks from SNAP (Leskovec and Krevl 2014) with five-point rating scale, corresponding to "Apps for Android," "Digital Music," and "Home and Kitchen," respectively. The definition of ground truth follows Amazon dataset described above.

Note that we preprocess the Epinions dataset in our experiment by extracting a subgraph of random sampling nodes in terms of computational cost. In Table. 3, we show the properties of the seven datasets. Here, labeled nodes are a small fraction of all nodes.

Table 3 Properties of datasets used for the evaluation

Dataset	Nodes	Edges	Benign users	Fraudulent users	Average degree	Cluster coefficient
OTC	5881	35,592	136	180	12.10	0.15
Alpha	3783	24,186	138	102	12.79	0.16
Amazon	271,082	415,390	2358	241	3.06	0.00
Epinions	4180	70,227	726	70	33.60	0.18
Amazon App	98401	465,350	7998	194	9.46	0.00
Amazon Music	8901	37,836	816	115	8.50	0.00
Amazon Home	93,820	376,802	7339	52	8.03	0.00

4.2 Settings

4.2.1 Proposed framework

After calculating the initialized vector of each node, we need to convert raw rating networks (i.e., edges have raw rating values) to signed networks, since the proposed framework requires signed networks as input. We simply divide all edges of rating networks into two classes (positive and negative) using as a threshold the center of possible rating values in each dataset. In Amazon datasets (Amazon, Amazon App, Amazon Music, and Amazon Home), the set of possible ratings is {1,2,3,4,5}, so we truncated the edges with a rating value of 3.

For model selection, we train using two-thirds of the given training data and then select the best model with the remaining one-third. The performances reported below are for the test data held out from the training data. All other GCN-based models follow this procedure. As for hyperparameters, dimension of representation vectors for a relationship (denoted as d_1, d_2) is set to 32, following SGCN.

4.2.2 Baselines

We compare performances between the proposed framework and the following four methods in our experiments.

- Rev2 (Kumar et al. 2018b) is the state-of-the-art method for fraudulent user detection in all four datasets in this experiment.
- R-GCN (Schlichtkrull et al. 2018) is a GCN-based model for edge-attributed networks. Here, we conduct the experiments in three variants of R-GCN. The first variant is the most standard form of R-GCN, which learns different convolution kernels for different edge ratings. The second is similar to the first, but a hyperparameter named "base" is set to 3. Setting a smaller base value than the actual number of unique edge types is reported to be effective for avoiding overfitting and capturing the similarity of edge types. In the last variants, we preprocess edge types in the same way as the proposed framework. Here edges are recategorized into binary classes (positive and negative) with a threshold at the center of the possible rating values. We report the results from the best variants of each dataset.
- SIDE (Kim et al. 2018) is a random-walk-based network embedding method utilizing balance theory, which is designed for signed directed networks. The major difference from our proposed method is that SIDE learns in unsupervised fashion. The output of SIDE is a distributed representation, not a class label. So we use random forest to build a classification model with the distributed representation as input.
- SGCN (Derr et al. 2018) is a GCN-based model for signed networks. As described in Sect. 3.1, limited types of relationships between nodes are considered unlike the proposed framework. Although it is originally designed for link sign prediction, we use the final embedding to the node classification task where parameters are learned jointly.

4.3 Results

4.3.1 Transductive settings

The task is to classify the users on rating networks into two classes (fraudulent/benign) in transductive settings, where the nodes used to classify are involved in the graph of the learning phase. Following the settings of Kumar et al. (2018b), we adopt the metrics called Area Under the Curve (AUC) and calculate the average AUC score of tenfold cross-validation. Table 4 shows the results of our models and baselines described above. To evaluate the robustness of models, we also show the performances when the portion of training data is low (0.03) in Table 5. Also, we conduct the same experiments where the percentages of the training data are from 10 to 90% at 10% increments. Figure 5 shows the results. We observe that GCN-based models (R-GCN, SGCN, and ours) performed better than the existing state-of-the-art baseline (Rev2) in most cases. We also observe that models that utilize balance theory to convolutional operations (SGCN and ours) achieved better performance

Table 4 Average AUC values of tenfold cross-validation

Methods	OTC	Alpha	Amazon	Epinions	Amazon App	Amazon Music	Amazon Home
Rev2	0.893	0.84	0.857	0.854	0.737	0.962	0.730
R-GCN	0.960	0.926	0.818	0.767	0.616	0.824	0.713
SIDE	0.975	0.921	0.855	0.938	0.757	0.912	0.708
SGCN	0.994	0.959	0.871	0.959	0.704	**0.974**	0.751
GCNEXT	**0.996**	**0.97**	**0.875**	**0.973**	**0.766**	0.972	**0.756**

The best performing methods in each dataset are shown in bold

Table 5 Average AUC values of 30 random iterations when ratio of training data is 0.03

Methods	OTC	Alpha	Amazon	Epinions	Amazon App	Amazon Music	Amazon Home
Rev2	0.691	0.722	0.79	0.624	0.669	0.947	0.565
R-GCN	0.886	0.732	0.738	0.682	0.552	0.654	0.608
SIDE	0.902	0.765	0.733	0.763	0.591	0.733	0.576
SGCN	0.934	0.777	**0.807**	0.933	0.682	0.945	0.705
GCNEXT	**0.947**	**0.818**	0.796	**0.949**	**0.685**	**0.953**	**0.712**

The best performing methods in each dataset are shown in bold

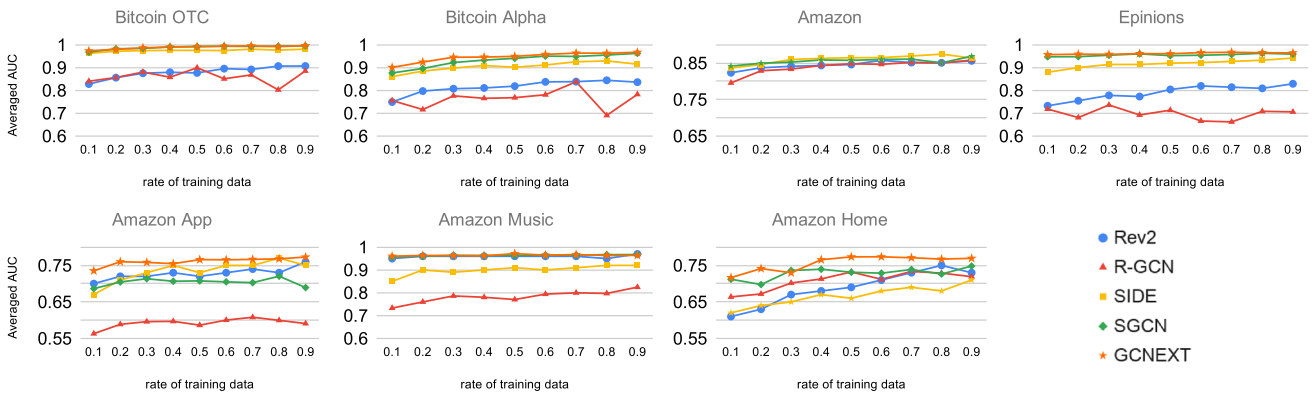

Fig. 5 AUC for several training data rates in transductive settings

than R-GCN. Furthermore, it is also notable that network-embedding-based approach utilizing balance theory (SIDE) performed better than R-GCN in most cases.

Note that the detection performance of Amazon App and Home is inferior to other datasets. This may be due to the fact that the sample size of fraudulent users is much smaller than that of benign users.

4.3.2 Inductive settings

We also conducted experiments in the inductive setting, which can be regarded as fraud detection towards newcomers to the rating platforms. Using timestamps on reviews in each dataset, we first extract a small percentage of reviews generated early and used them as training data. Then, in the test phase, the evaluation was conducted using all reviews. Note that GCN-based models do not require any retraining because the learned convolution kernel can be directly utilized for the inference task.

We compared the proposed framework and SGCN, which performed very well in the transductive setting. Figure 6 shows the results for the classification of the nodes that did not exist in the training phase. Since we randomly divide the nodes into the training set and the validation set in the training phase, we repeat the training and the evaluation 30 times and record the averaged AUC values in order to reduce the influence of randomness. We observe that the proposed framework performed better in most cases. In particular, there were significant improvements in the case with an extremely low ratio of training data.

4.4 Comparison of models

In this section, we interpret the above results by comparing models. We obtained the results for three criteria: (1)

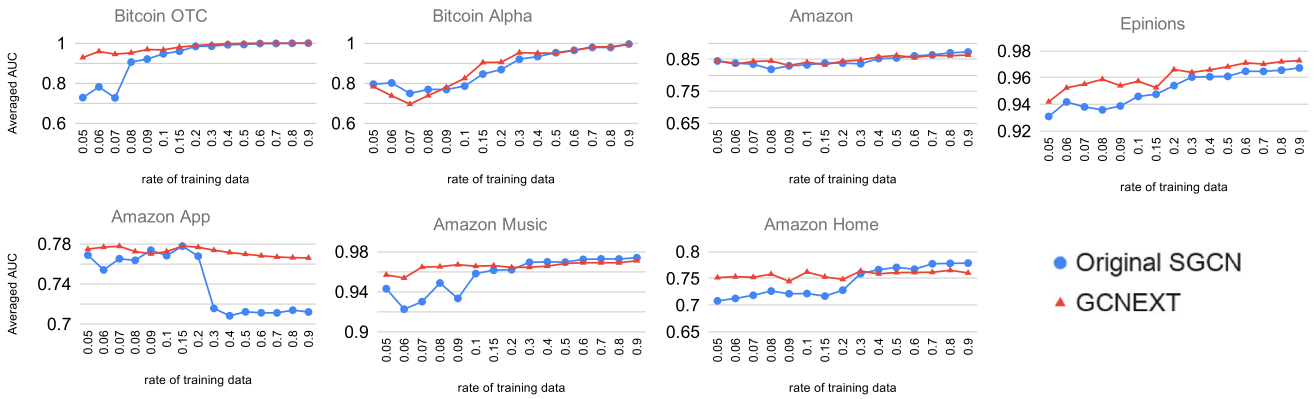

Fig. 6 AUC for the inductive settings

effectiveness of GCN-based end-to-end approaches, (2) effectiveness of utilizing balance theory, and (3) effectiveness of incorporating directions of edges.

4.4.1 Effectiveness of GCN-based end-to-end approaches

The results in the transductive settings indicate the effectiveness of GCN-based models in the semi-supervised fraudulent user detection task. Even simple implementation of GCN (R-GCN) achieves comparable performance to Rev2. Although Rev2 can capture the interaction among entities on rating network, extracted representations are not necessarily suitable for distinguishing fraudulent and benign users. This suggests that GCN-based models succeed in extracting distinguishing features from interactions among entities.

4.4.2 Effectiveness of utilizing balance theory

In transductive settings, SGCN-based models (SGCN and ours) and even network-embedding with balance theory (SIDE) outperformed R-GCN in most cases. Significantly, this implies that considering the sign of edges by applying balance theory greatly improves the performance of fraudulent user detection on rating networks.

4.4.3 Effectiveness of incorporating directions of edges

Although our expansion provides no significant improvement in transductive settings, the proposed framework does outperform SGCN in most cases of inductive settings. To uncover the reason for this, we focus on the Bitcoin OTC dataset, in which the difference of AUC in inductive settings was the greatest between SGCN and the proposed framework.

We show how fraudulent and benign users have relationships shown in Fig. 2 with their 2-hop neighbors in (1) the entire Bitcoin OTC network (Fig. 7) and (2) the earliest 5% edges of Bitcoin OTC network (Fig. 8). Note that legends indicate the permutation of the first edges' sign (from source to 1-hop) and the second edges' sign (from 1-hop to 2-hop). For example, "P_N" indicates that the first edge is positive and the second edge is negative. As an example of interpretations, we observe that fraudulent users in entire networks (Fig. 7) have approximately 80,000 2-hop neighbors categorized into "I/O F." About 60,000 of them are derived from two negative edges ("N_N"), while remaining others are from two positive edges ("P_P").

We observe that edges in the early days (Fig. 8) are quite limited in variety compared to the complete network shown in Fig. 7. In particular, it is significant that relationships

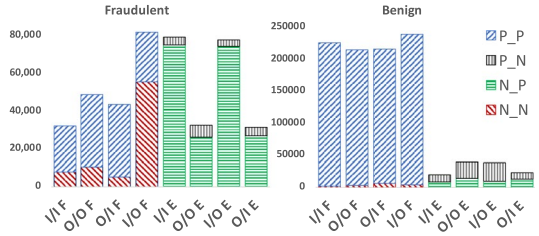

Fig. 7 Number of 2-hop relationships in Bitcoin OTC. These are separately counted depending on whether the source user is fraudulent or benign

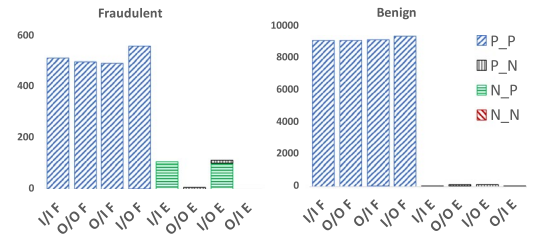

Fig. 8 Number of 2-hop relationship in **the earliest 5 % edges** of Bitcoin OTC. These are separately counted depending on whether the source user is fraudulent or benign

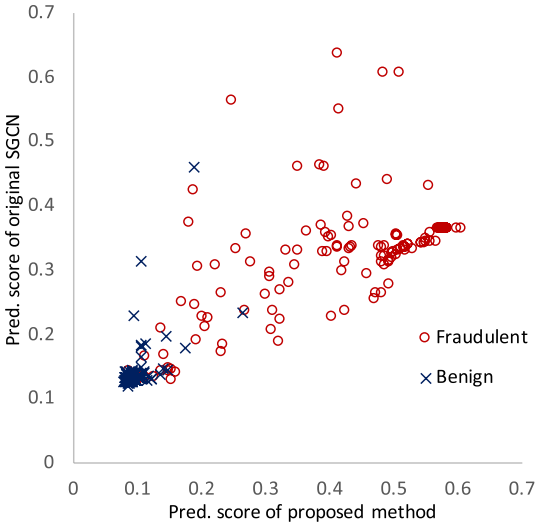

Fig. 9 Scatter plot for each instance of test sets in inductive settings when training edge rate is 5%

containing negative edges (such as "Enemies" relationships or "N_N" relationships in "Friends") do not appear.

In terms of model comparison, SGCN considers only "O/O F" and "O/O E," while the proposed framework can incorporate all types. In transductive settings, where all nodes and edges can be used in the training phase, interactions with "O/O F" and "O/O E" already appear characteristic enough, as shown in Fig. 7. We can assume that is the reason why there is no significant difference between SGCN and the proposed framework in transductive settings.

In inductive settings with a low training edge ratio, where observable relationships in training data can be quite limited as shown in Fig. 8, it may be necessary to incorporate interactions exhaustively, including those that cannot be captured by SGCN. Figure 9 shows a scatter plot of predicted scores for each instance of test sets in inductive settings when the training edge ratio is 5%. We observe that there are many fraudulent users whom the proposed method predicts to be fraudulent with confidence but SGCN does not. This is probably because the fraudulent and benign users can be distinguished by considering all 2-hop relationships but not by using a limited part of them, such as "O/O F" and "O/O E." The results suggest that fraudulent users have distinctive features, even in the early stage (i.e., before increasing negative edges), and that the proposed framework succeeds in extracting them while SGCN fails.

In general, the proposed framework can be effectively used when observable relationships are limited and distinctive relationships among users are unknown. In a practical case, the owner of a rating platform can detect new fraudulent users quickly. Since the proposed framework uses the learned parameter in classifying and requires only a rating

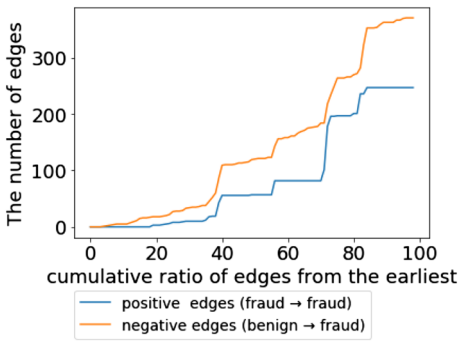

Fig. 10 Transition graph of the positive edges from fraudulent to fraudulent and negative edges from benign to fraudulent

network as input, there is no need for retraining of the detection task, even for the newcomers. That is a promising advantage compared to the existing methods such as Rev2, which require an additional iterative learning process to evaluate newcomers.

4.5 Insight into the behavior of fraudulent users

Analysis based on 16 types of 2-hop relationships derived from our expanded balance theory reveals that fraudulent users early on (Fig. 8) have few negative edges from their 1-hop neighbors, while they have a lot in the grown network (Fig. 7). Furthermore, it is counterintuitive that they have many "I/I F" relationships, which correspond to the case when a fraudster is trusted by someone who is trusted by another, even in the grown network. In this section, we aim to uncover how and when those relationships increase as the network grows.

Following the previous section, we focus on the Bitcoin OTC network and analyze how fraudulent users are rated by others. In Fig. 10, we show transitions in the number of (1) positive edges from fraudulent users to fraudulent users and (2) negative edges from benign users to fraudulent users, as the network grows. It is indicated that the positive edges between fraudulent users increase in a short time corresponding to the increase in negative edges from benign users to fraudulent ones. This implies a behavior of fraudulent users that they form a faction to deal with the negative ratings from other users. This results imply that incorporating time series with interactions among users is likely to improve performance.

As indicated above, expanded balance theory proposed in this paper is helpful as an analytical framework and can lead to a useful insight. Applying the theory to provide more detailed analyses on rating networks is one of our future directions.

5 Conclusion

In this paper, we investigated the task of fraudulent user detection on rating networks. We proposed GCNEXT, an end-to-end GCN-based framework using expanded balance theory, which effectively incorporates both the signs and directions of edges. The experimental results show that the proposed framework performs better, or even best, in most settings. In particular, this framework shows remarkable stability in inductive settings, which is associated with the detection of new fraudulent users on rating platforms. In practical cases, our framework helps the owner of a rating platform detect fraudulent users earlier and constantly provides users with more credible information. We also analyzed a rating network using our proposed theory and provided a new insight into fraudulent users on rating platforms that fraudulent users form a faction to deal with the negative ratings from other users.

For our future work, we will explore the way to incorporate time series with interactions among users in a GCN-based framework. Furthermore, we will tackle more detailed analyses using expanded balance theory for a better understanding of rating. In addition, the design and implementation of online assessments is one of our future tasks. For example, the system operator could evaluate the precision by checking for fraudulent users detected by the proposed method in order to reduce false positives.

Another direction to consider in the future is the fairness of the fraud detection system. Since the current features are purely based on the behavioral history of each user, it is not considered to be a big problem. However, we need to be quite cautious when we reflect the user's attributes in the initial vector.

References

Akoglu L et al (2013) Opinion fraud detection in online reviews by network effects. In: Seventh international AAAI conference on weblogs and social media

Akoglu L et al (2015) Graph based anomaly detection and descriptio: a survey. Data Min Knowl Discov 29(3):626–688

Cartwright D et al (1956) Structural balance: a generalization of Heider's theory. Psychol Rev 63(5):277

Derr T et al (2018) Signed graph convolutional networks. In: 2018 IEEE international conference on data mining (ICDM). IEEE

Fayazi A et al (2015) Uncovering crowdsourced manipulation of online reviews. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. ACM

Heider F (1946) Attitudes and cognitive organization. J Psychol 21(1):107–112

Jiang J et al (2019) Anomaly detection with graph convolutional networks for insider threat and fraud detection. In: MILCOM 2019–2019 IEEE military communications conference (MILCOM)

Kim J et al (2018) Side: representation learning in signed directed networks. In: Proceedings of the 2018 world wide web conference on world wide web. International world wide web conferences steering committee

Kipf TN et al (2016) Variational graph auto-encoders. arXiv preprint arXiv:1611.07308

Kipf TN et al (2016a) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907

Kudo W, Nishiguchi M et al (2019) Fraudulent user detection on rating networks based on expanded balance theory and GCNs. In: 2019 IEEE/ACM international conference on advances in social networks analysis and mining

Kumar S et al (2016) Edge weight prediction in weighted signed networks. In: 2016 IEEE 16th international conference on data mining (ICDM). IEEE

Kumar S et al (2018) Rev2: Fraudulent user prediction in rating platforms. In: Proceedings of the eleventh ACM international conference on web search and data mining. ACM

Kumar S, et al. (2018) False information on web and social media: a survey. arXiv preprint arXiv:1804.08559

Lappas T et al (2016) The impact of fake reviews on online visibilit: a vulnerability assessment of the hotel industry. Inf Syst Res 27(4):940–961

Leskovec J et al (2010) Signed networks in social media. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM

Leskovec J et al (2010) Predicting positive and negative links in online social networks. In: Proceedings of the 19th international conference on World wide web. ACM

Leskovec J, Krevl A (2014) SNAP datasets: Stanford large network dataset collection. http://snap.stanford.edu/data. Accessed 30 Dec 2019

Lim E-P et al (2010) Detecting product review spammers using rating behaviors. In: Proceedings of the 19th ACM international conference on Information and knowledge management. ACM

Luca M et al (2016) Fake it till you make I: reputation, competition, and Yelp review fraud. Manag Sci 62(12):3412–3427

Massa P et al (2007) Trust-aware recommender systems. In: Proceedings of the 2007 ACM conference on recommender systems. ACM

McAuley JJ et al (2013) From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In: Proceedings of the 22nd international conference on world wide web. ACM

Minnich AJ et al (2015) Trueview: harnessing the power of multiple review sites. In: Proceedings of the 24th international conference on world wide web. International world wide web conferences steering committee

Sandulescu V et al (2015) Detecting singleton review spammers using semantic similarity. In: Proceedings of the 24th international conference on world wide web. ACM

Schlichtkrull M et al (2018) Modeling relational data with graph convolutional networks. In: European semantic web conference. Springer, Cham

Shah N et al (2016) Edgecentric: anomaly detection in edge-attributed networks. In: 2016 IEEE 16th international conference on data mining workshops (ICDMW). IEEE

Veličković P et al (2017) Graph attention networks. arXiv preprint arXiv:1710.10903

Wang G et al (2012) Identify online store review spammers via social review graph. ACM Trans Intell Syst Technol (TIST) 3:61

Xie S et al (2012) Review spam detection via temporal pattern discovery. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining. ACM