Relevance-Aware Anomalous Users Detection in Social Network via Graph Neural Network

Yangyang Li*✉, Yipeng Ji†, Shaoning Li‡, Shulong He§, Yinhao Cao‡, Yifeng Liu*, Hong Liu¶‖, Xiong Li**, Jun Shi†† and Yangchao Yang††
*National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data, CAEIT, Beijing 100041, China;
†School of Computer Science and Engineering, Beihang University, Beijing 100191, China;
‡School of Cyber Science and Technology, Beihang University, Beijing 100191, China;
§National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China;
¶School of Software Engineering, East China Normal University, Shanghai 200062, China;
‖ Shanghai Trusted Industrial Control Platform Co., Ltd., Shanghai 200062, China;
**Beijing Bitcat Technology Co., Ltd., Beijing, China;
††Shenzhen CyberAray Network Technology Co., Ltd, CETC, Shenzhen, China.
Email: {liyangyang,liuyifeng3@cetc.com.cn}; {jiyipeng,19373215,caoyinhao}@buaa.edu.cn; hsl@cert.org.cn;
liuhong@ticpsh.com; li.xiong@foxmail.com; jshi@nscslab.net; forester@mail.ustc.edu.cn.

2021 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-3900-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IJCNN52387.2021.9534136

***Abstract*—Anomalous users detection in social network is an imperative task for security problems. Motivated by the great power of Graph Neural Networks(GNNs), many current researches adopt GNN-based detectors to reveal the anomalous users. However, the increasing scale of social activities, explosive growth of users and manifold technical disguise render the user detection a difficult task. In this paper, we propose an innovate Relevance-aware Anomalous Users Detection model (RAU-GNN) to obtain a fine-grained detection result. RAU-GNN first extracts multiple relations of all types of users in social network, including both benign and anomalous users, and accordingly constructs the multiple user relation graph. Secondly, we employ relevance-aware GNN framework to learn the hidden features of users, and discriminate the anomalous users after discriminating. Concretely, by integrating Graph Convolution Network(GCN) and Graph Attention Network(GAT), we design a GCN-based relation fusion layer to aggregate initial information from different relations, and a GAT-based embedding layer to obtain the high-level embeddings. Lastly, we feed the learned representations to the following GNN layer in order to consolidate the node embedding by aggregating the final users' embeddings. We conduct extensive experiment on real-world datasets. The experimental results show that our approach can achieve high accuracy for anomalous users detection.**

***Index Terms*—Social network, Abnormal social users detection, Heterogeneous graph neural network.**

I. Introduction

Nowadays, social networks have become a non-substitutable platform for people's daily interactions and socialization. However, along with the large scale of online activities, some abnormal social users, enormous 'bots' accounts like zombie users, spammers, social bots, etc. have also shown up, which lurk in around benign users and have a non-negligible impact on the security of social networks [1], IoT systems [2], IDC infrastructures [3]–[5], and the reliability of digital platforms [6], [7]. Abnormal users are essential tools for orchestrated manipulation of mass media events and actively involved in the discussion of important events, including the public opinion direction of various political events [8]. They are also responsible for disseminating less credible information or extreme ideologies, as well as increasing the level of confusion in some online discussions [9]. Therefore, detecting and filtering the anomalous users is essential for the security of social network [10].

Traditional approaches to anomalous users detection mainly concentrate on the explicit account information [11](nickname, head portrait etc.) collected from users' activities, and only treat users as isolated individuals. Furthermore, as a result of evolving techniques about generative adversarial networks(GANs) [12], [13], the anomalous users are able to imitate human-like behavior and disguise themselves. Hence the above static detection approaches are no longer adapted to the current situations, and the discrimination of anomalous users has become more ineffective [14]. Recently, due to the perfect performance of graph neural networks(GNNs) in capturing the hidden connectivity in graph structure, many GNN-based anomaly detectors have been applied to various fraud or anomaly detection scenarios [15], [16]. In contrast to the traditional detection methods, GNN-based approaches consider the neighborhood information to learn the node representations with neural modules. They can be trained in an end-to-end diagram [17], and their semi-supervised learning fashion dramatically decreases the labeling cost as well.

However, the existing GNN-based detection approaches still neglect the well-designed camouflage of the current newly anomalous users. They adjust their behaviors to alleviate the suspiciousness and would like to interact with normal users [18]. In other word, though users' external features and explicit relationships have been thoroughly exploited [19], they still lead to the serious loss of detection accuracy in consequence of failing to distinguish the disguised relation between normal

✉ Yangyang Li is the corresponding author

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

and anomalous users. As mentioned, anomalous users would build connection with some benign users to muddle through the feature-based detectors. Hence, extracting multiple relations from different users is of great significant to accurately classify the type of users most of which occupy semblable features [20], [21]. Though some recent works have noticed the similar challenges, their solutions still can not fit with the anomalous users detection problems.

To further detect the well-disguised users in social network, in this paper, we propose a **GNN**-based **R**elevance-aware **A**nomalous **U**sers Detection (**RAU-GNN**) model to achieve fine-grained anomalous users detection results. We first extract the multiple relations from all users in social network. The relations between users could be roughly defined as an interaction, including retweets, comments or forwarding etc. [22]. As for anomalous users, they prefer to forwarding similar blogs and take action at the same time. All these features can be constructed into a unified multiple relation graph. Secondly, we leverage relevance-aware GNN-based framework to learn the hidden representation in the constructed relation graph from users. Concretely, we adopt GCN module to initially aggregate the structual information accross different relationships, and embedds the processed fusional features to the center nodes. Then we use multi-head GAT module to learn the high-level embeddings and we feed the final node embeddings to the following GNN layer, and aggregate all the users information from their neighbors, in order to consolidate the previous embeddings. Last we use the binary classification algorithm to classify the learned features, and discriminate the anomalous users. We evaluate our proposed model with real-world datasets and the experimental results show that RAU-GNN can achieve high accuracy for anomalous users detection, outperforms other comparable baseline models. The main contributions of RAU-GNN are summarized as follows:

- We extract different relations from users and accordingly establish a multiple relation users graph network as the basis of RAU-GNN, and explore the importance of different users and relations.
- We propose a anomaly detecter named RAU-GNN based on a relevance-aware GNN framework, which consists of a GCN-based relation fusion layer, a GAT-based embedding layer and a final GNN aggregator respectively. The integration of these GNN layers can better learn the high-level representations and see through the well-designed disguise.
- Extensive experiments with real-world datasets are conducted to validate the effectiveness of RAU-GNN on anomalous users detection. The results demonstrate that our approach can achieve high accuracy and outperforms other classic comparable baseline models.

The rest of this paper is organized as follows. In Sec. II, we introduce the background and related work for anomalous users detection. Sec. III demonstrates the necessary definition and models the detection problems. Sec. IV depicts the framework and components of our proposed RAU-GNN . We evaluate our proposed model with real-world datasets in Sec. V and analyze the experimental results with cases at length. Finally, we conclude this paper in Sec. VII.

II. Related Work

A. Graph neural network

GNNs [23] model is extended from the traditional neural network to implement graph-structured data. GNNs aim to learn dimensional vector representations for nodes in the networks. The recent diagram of GNNs is to generalize convolutional operation(GCN) [24], which generates nodes' representation by aggregating their own and neighbors' features. GCNs address the cyclic mutual dependencies architectually and can be classified into spectral-based GCN and spatial-based GCN. Spectral-based GCNs [25] apply a normalized graph Laplacian matrix and graph Fourier transform to make spectral convolutional operations. Spatial-based GCNs directly convolve the nodes' representations with their close neighbors' to derive and propagate the updated representations. Compared with the spectral-based GCNs, spatial-based GNNs are more flexible and scalable with the sizes and the structures of the graphs. Furthermore, spatial-based GCNs are more efficient and are preferred in recent works [26]. Graph attention network (GAT) [27] introduces the classical attention mechanism to GNNs. GAT outperforms in handling the long-term dependencies. It captures the content-based similarity between two entities and measures their distance of representations [24], [28], [29].

B. Anomalous users detection

Anomalous users detection mainly focuses on those who are mostly utilized as powerful tools to guide the direction of public opinion or carry out some crimes in social network [30]. The most common anomalous users around are 1)*Zombie user.* Zombie users are specifically fake followers who create an illusion of a high reputation and credibility via boosting the number of followers(similar to DDoS attack). Most of the time, zombie users remain silent without any interaction but suddenly show up for some purpose. 2)*Spammer.* Spammers aggressively post harmful content such as adult advertisements, e-magazines and links [31]. Different from zombie users, spammers interact with normal users by replying to comments, most of which are same in order to express the content and enhance their influence. 3)*Social bots.* Social bots are designed to disseminate a certain point of view that can lead the direction of public opinion by posting content with certain hashtags [32]. Social bots make disproportionate contribution in disseminating less credible information or extreme ideologies, as well as increasing the level of confusion on purpose.

To detect these anomalous users in social network, most approaches are proposed based on static personal features [33]. The basic Bayesian classification with user characteristics [34] and machine learning methods [35] have proven efficient for zombie users detection. Recently, many GNN-based detectors show better performance than traditional methods. Wang et

al. [9] leverage graph convolutional network for fraudster detection in the online app review system. Li [15] et al. design a GCN-based anti-Spam model which integrates a heterogeneous graph and a homogeneous graph to capture the local context and global context of a comment, and aggregate neighbor's information. Zhang et al. [36] leverage convolution mechanism to learn embeddings of each single-view attributed graph and attention mechanism to fuse different embeddings.

III. Problem Definition and Modeling

In this section, we will first give the necessary notations and definition of this paper in Table I, and model the anomalous users detection problems.

Definition 1. Multiple Relation Graph. A multiple relation graph is $\mathcal{G}$ that contains a set of relation graph $U^r = \{(\mathcal{V}, \{\mathcal{E}_r\}_{r=1}^R), Y\}$, where $\mathcal{V}$ stands for the collections of user nodes $\{v_1, v_2, \cdots, v_n\}$, and $\mathcal{E}_r$ stands for edges $e_{i,j}^r = (v_i, v_j)$ under relation $r \in R$. R is the total number of relations. Y is the node label.

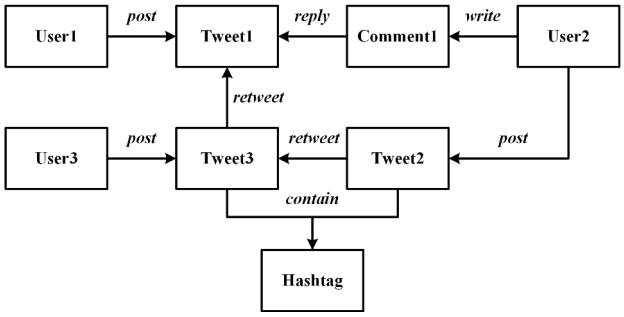

Fig. 1: An example of users interaction in multiple relation graph

Fig.1 gives an example of multiple relation network for users, including various relationships between users. Generally, users' interactions in social network can be categorized into *following, commenting/replying, posting/retweeting*, and *hashtagging*, which also represent different types of relation between nodes in constructed multiple relation graph. We denote these operations as $\{f, c, p, h\}$, respectively, which stand for the different types of relations in constructed multiple relation graph.

Definition 2. Anomalous Users Detection on Graph. For the anomalous users detection problem, we aim to justify whether the given node v_i is abnormal. Hence we can treat the discrimination problem as a binary classification on the multiple relation graph. The label of nodes in multiple relation graph is $y_v \in \{0, 1\}$, where label 0 represents the benign user and label 1 represents the anomalous user that to be detected. The graph-based anomalous user detection algorithm learns the detection model $f(V_i, E_r; \theta)$ based on the labeled user nodes' information under each relations, and predict whether the unlabeled nodes in graph are anomalous users. θ denotes the parameters in model.

TABLE I: Glossary of Notations.

Notation	Description
$r; R$	Relation; Total number of relations
$\mathcal{G}; U^r$	Multiple relation graph; Homo user graph under relation r
$\mathcal{V}; \mathbf{E}; X,$	Node set; Edge set; Features
W	Sub-matrix of heterogeneous graph
$\tilde{D}$	Diagonal matrix
$\mathbf{H}_r^{(l+1)}$	Embedding at layer $(l+1)$ under relation r in GCN module
z_{v_i}	Normalized Embedding for node v_i
$\oplus$	Fusion operation
norm	Normalization
$\mathbf{h}_r^{(l+1)}$	Embedding at layer $(l+1)$ under relation r in GAT module
AGG	Aggregation operation
$\mathcal{Z}^{(l+1)}$	Embedding at layer $(l+1)$ in enhanced aggregator
$\mathcal{Z}_v$	Final representation for user v
$\mathcal{P}_v$	Possibility for node v to be an anomalous user
$\mathcal{L}_{\text{RAU-GNN}}$	Loss function for RAU-GNN model

Definition 3. GNN-based User Detection. A Graph Neural Network(GNN) is a layer-wise deep learning framework, aiming to embed and learn graph features by aggregating information from its neighbor nodes. Here we give the unified formulation of GNNs from perspective of neighbor aggregation to give a full picture of proposed GNN-based framework:

$$h_{v_i}^{(l+1)} = \sigma\left(\underset{\forall v_j \in N(v_i)}{AGG}\left(h_{v_i}^{(l)} \oplus h_{v_j}^{(l+1))}\right)\right). \tag{1}$$

For the center node v_i, $h_{v_i}^{(l+1)}$ is its hidden representation at the $(l+1)$-th layer, and we define $h_{v_i}^{(0)} = x_i$ is the input feature. $h_{v_i}^{(l)}$ is the input of $(l+1)$-th layer in GNN, that is to say, GNNs follow the propagation rule. AGG is the aggregation function such as mean aggregation or attention aggregation. $\oplus$ is the fusion operation (concatenation or summation) that combine the extracted features from its neighbor. The final purpose of GNN-based user detection is to make use of the advantage of GNNs on aggregating the information of neighborhood. The node embeddings at the last layer of GNN represent the kernel of each users, and we use these high-level embeddings to discriminate whether the users are anomalous.

IV. Relevance-Aware Anomalous Users Detection

A. Overall Framework

The overall framework of RAU-GNN proposed in this paper is shown in Fig.2. Concretely, our RAU-GNN contains three processes, including the construction of multiple relation graph for users, an integrating GNN-based framework to learn the high-level representations, and a discrimination layer to detect the anomalous users. The proposed GNN-based framework integrates three layers, including a GCN-based relation fusion layer, a GAT-based embedding layer and a final GNN aggregator. The following sections will introduce more details of RAU-GNN at length.

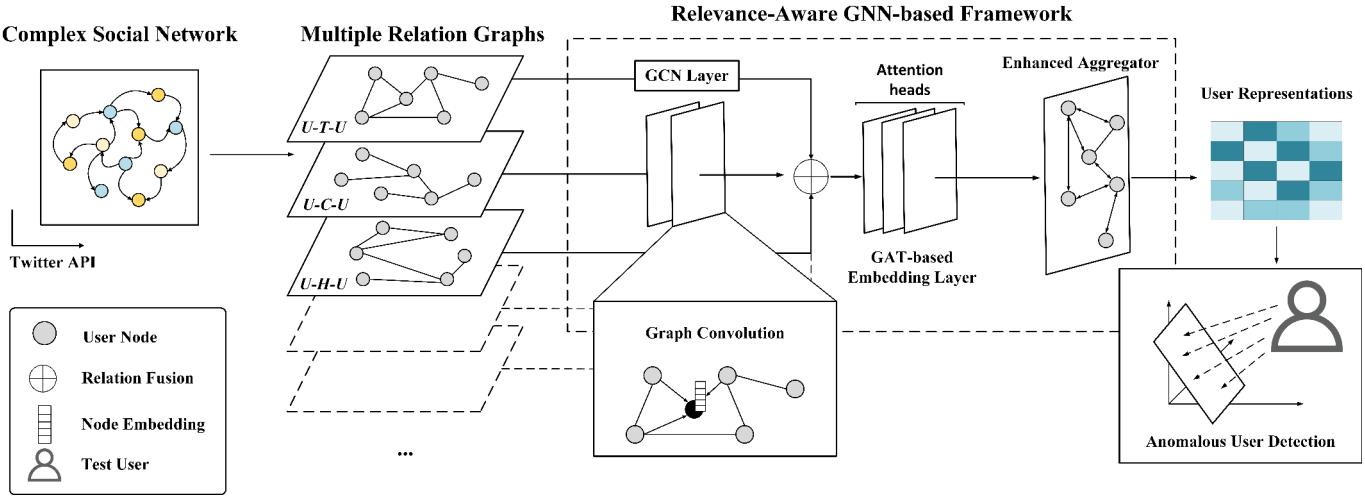

Fig. 2: Relevance-Aware Anomalous Users Detection

B. Construction of Multiple Relation Graph

To begin with, we extract the features of users from *user* object by Twitter API[1]. The rich extracted features serve as the initial node features in social network. During the preprocessing, we aim to unify all the users and their relations, and break them down to a set of relation-based homogeneous graphs. First, we extract the user-oriented elements, such as tweets, comments etc. and organize them in a unified manner to represent the different relations. For example, we extract a tweet that reveals User1 comments on tweet of other users. We denotes User1 as v_1 and add edge between v_1 and the extracted tweet comments. We repeat the same process for all the users in social network, and obtain a complex heterogeneous graph.

Next, we break down the initial graph to multiple relation graphs. The multiple relation graph only contains user nodes and with different edge attributes representing different types of relations. Specifically, the separating process is defined as follows:

$$U^r_{i,j} = \left[\sum_r W_{vr} \cdot W^T_{vr}\right], r \in \{f, c, p, h\}. \tag{2}$$

Here adjacency matrix $U^r_{N\times N}$ is the homogeneous user graph under relation r, where N is the total number of users in graph. $\cdot_{i,j}$ denotes the matrix element at the i^{th} row and j^{th} column, and r denotes the element type under the mentioned operations. W_{vr} is a sub-matrix denotes rows of user nodes and columns of type r. $\cdot$ is matrix transpose.

Lastly we give the extracted user feature vectors to the multiple relation graphs. The initial feature is denoted as $X = \{x_v \in R^d\}$, where x_v is the initial feature of user with d dimension. We collect all types of $\{U^r\}|^R_{r=1}$ and obtain the processed multiple relation graphs $\mathcal{G} = (X, \{U^r\}|^R_{r=1}, Y)$.

[1]https://developer.twitter.com/en/docs/twitter-api

C. Relevance-Aware GNN-based Framework

The proposed GNN-based framework consists of a GCN-based relation fusion layer, a GAT-based embedding layer and a final GNN aggregator, respectively.

1) GCN-based Relation Fusion Layer: We implement GCN module to learn the discriminating user node representation in each relation, and fuse the learned features in different relations to the center node. The layer-wise propagation rule of multi-layer GCN is defined as follows:

$$\mathbf{H}^{(l+1)}_r = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{U}_r\tilde{D}^{-\frac{1}{2}}H^{(l)}_r W^{(l)}\right), \tag{3}$$

where $\tilde{U}_r = U_r + I_N$, r denotes the relation type of graph U and N denotes the total number of users in this graph. $\tilde{D}$ is diagonal matrix with $\tilde{D}_{ii} = \sum_j \tilde{U}_{ij}$, I_N is the identity matrix, W is the parameter matrix and l is the number of layers. σ denotes the activation function, e.g., Sigmoid or ReLU. The input layer of GCN module is $H^{(0)} = X$, and X is the initial features constructed in Section IV-B. We use $H^r_{v_i}$ as the final representation of user node v_i under relation r.

Next we fuse the output representations from GCN layers in different relations, and obtain the final dense representations of the center node. We give the fusion formulation as follows:

$$z_{v_i} = \text{norm}\left(H^{(r=1)}_{v_i} \oplus H^{(r=2)}_{v_i} \oplus \cdots \oplus H^R_{v_i}\right), \tag{4}$$

where v_i is the center node. $H^{(r)}_{v_i}$ is the output representation under relation $r \in R$ with d' dimension. $\oplus$ denotes the operator that combines information of v_i across all its all relations, e.g., concatenation or summation. norm denotes the normalization process to avoid overfitting. It is noting that after the GCN-based fusion process, we unify the multiple relation graphs to one complete homogeneous user graph $\mathcal{A}$ with the combined representations, wherein all the users are involved. And we implement this unified graph to facilitate further processing.

2) GAT-based embedding layer: In order to prick camouflage of anomalous users, we leverage GAT module to study the correlations between users in social network, and embed the fusion features in user graph. To be specific, we define the GAT layer-wise propagation rule as follows:

$$\mathbf{h}_{v_i}^{(l+1)} \leftarrow \overset{heads}{\Big\|} \left(\mathbf{h}_{v_i}^{(l)} \oplus \underset{\forall v_j \in N(v_i)}{AGG} (\mathbf{h}_{v_j}^{(l)}) \right). \tag{5}$$

Here $\mathbf{h}_{v_i}^{(l+1)}$ is the representation of user v_i at the $(l+1)$th GAT layer, and the input $\mathbf{h}_{v_i}^{(0)}$ is the output of the fusion process z_{v_i}. $N(v_i)$ stands for a set of neighbors of user v_i according to graph $\mathcal{A}$. $\oplus$ denotes the operator that combines information of v_i with its neighbors. $\overset{heads}{\|}$ denotes head-wise concatenation. AGG represents the aggregation function that mapping neighborhood information into a vector. Here we adopt the attention aggregation. We use $\mathbf{h}_{v_i}$ as the final representation with dimension d' after GAT-based user embedding.

The learned representation are detector's cognition about the users in social network, which are also a fusion of natural language semantics, temporal activities, and the structural information of the completed user graph. These high-level representations stand for the kernel of a user, and helps to identify the abnormal account.

3) Enhanced GNN Aggregator: Based on the relation fusion layer and embedding layer, we have obtained the hidden representations of each users in social network. In order to achieve a fine-grained detection result and be able to adaptive to the real-world scenario, we add a following enhanced GNN aggregator to consolidate the previous embeddings. We define the GNN aggregation as follows:

$$\mathcal{Z}^{(l+1)} = \sigma \left(\underset{\forall v_j \in N(v_i)}{AGG} (\mathcal{Z}_{v_i}^{(l)} \oplus \mathcal{Z}_{v_j}^{(l+1))}) \right), \tag{6}$$

where $\mathcal{Z}_{v_i}^{(l)}$ is the center node embedding at the l-th layer, $\mathcal{Z}_{v_j}^{(l+1))}$ is the neighbor node embedding at the $(l+1)$-th layer. $\oplus$ denotes the embedding summation. AGG represents the mean aggregation. σ denotes the activation function. Here we adopt ReLU non-linear function.

For the GNN aggregator is mostly applied to consolidate the previous learned embeddings instead of learning more high-level semantics, we fix the output dimension of GNN aggregator as d', which is the same as the output dimension of GAT-based embedding layer, and only adopt one layer GNN.

D. Optimization

After the enhanced GNN aggregator, for each center user node v, its final embedding is the output of aggregator $\mathcal{Z}_v$. Therefore, we accordingly design the loss of RAU-GNN as a cross-entropy loss function to minimize the ground-truth and predicted label, and achieve the anomalous detection:

$$\mathcal{L}_{\text{RAU-GNN}} = -\sum_{v \in \mathcal{V}} \log(y_v \cdot \mathcal{P}_v) + \lambda \|\Theta\|_2, \tag{7}$$

where $\mathcal{V}$ is all collection of node in user graph, y_v denotes the label of node v, and $\mathcal{P}_v$ denotes the probability of RAU-GNN's prediction. We define $\mathcal{P}_v = \sigma(\text{MLP}(\mathcal{Z}_v))$. σ is the activation function. We adopt Sigmoid in RAU-GNN. MLP denotes the multi-layer perception. λ is the weight parameter and we add $\|\Theta\|_2$, which represents the L_2-norm of all module parameters in GNN-based framework, to obtain the well generalization.

Algorithm 1: RAU-GNN: GNN-based Relevance-Aware Anomalous Users Detection

Input: A set of Multiple Relation Graphs with nodes features and labels: $\mathcal{G} = (X, \{U^r\}|_{r=1}^{R}, Y)$, Number of Layers, Mini-batches: L, B.
Output: User Representation $\mathcal{Z}_v$, $\forall v \in \mathcal{V}_{train}$.

$\mathbf{H}_v^{(0)} \leftarrow x_v$;
for $b = 1, 2, ..., B$ **do** `// Train in mini-batches`
  for $r = 1, 2, \cdots, R$ **do**
    for $l = 1, 2, \cdots, L$ **do**
      $\mathbf{H}_v^{(l+1)} \leftarrow$ Eq. (3);
  $\mathbf{H}_v^r \leftarrow \mathbf{H}_v^{(L)}$;
  $z_v \leftarrow$ Eq. (4);
  $\mathbf{h}_v^{(0)} \leftarrow z_v$;
  for $l = 1, 2, \cdots, L$ **do**
    $\mathbf{h}_v^{(l+1)} \leftarrow$ Eq.(5);
  $\mathbf{h}_v \leftarrow \mathbf{h}_v^{(L)}$;
  $\mathcal{Z}_v^{(0)} \leftarrow \mathbf{h}_v$;
  $\mathcal{Z}_v \leftarrow$ Eq. (6);
  $\mathcal{L}_{\text{RAU-GNN}} \leftarrow$ Eq. (7);
  Back-propagation to update parameters;

V. Experiments and Evaluation

A. Dataset and Graph Construction

We utilize the popular YelpChi review dataset, along with twitter datasets to study the anomalous user detection problems. The YelpChi dataset includes hotel and restaurant reviews filtered (spam) and recommended (legitimate) by Yelp. In Twitter dataset, we manually label users with more than 20% helpful entities from anomalous users. In this paper, we treat the spammers as anomalous users, and conduct the anomalous detection task on YelpChi and Twitter dataset, which can also be considered as a binary classification problem.

Similar to definition mentioned in Def. 1, we extract the reviews, products and time. In Twitter, we take tweets, comments and hashtags instead. We then construct the multiple relation graph for YelpChi and Twitter dataset. The details of construction is demonstracted in Section IV-B. The statistics are shown in Table II with some explanations:

YelpChi. 1) *R-U-R*: connects reviews sent by same users. 2) *R-P-R*: connects reviews under same product. 3) *R-T-R*:connects reviews that created in the same month. **Twitter**. 1) *U-T-U*: connects users that mention the same tweet. 2) *U-C-U*: connects users that comments the same content. 3) *U-H-U*:

TABLE II: Statistics in Dataset and Graph

	Nodes	Anomalous%	Relation	Edges
YelpChi	50,128	15.1%	R-U-R	51,715
			R-P-R	582,462
			R-T-R	4,402,892
			ALL	5,037,069
Twitter	12,384	24.3%	U-T-U	2,625,142
			U-C-U	516,706
			U-H-U	311,610
			ALL	4,206,916

connects users that involved in same hashtag. The number of edges belonging to each relations is also shown in Table II.

B. Baseline Algorithm

To verify the effectiveness of RAU-GNN, we compare our proposed model against baselines, including general GNN models and new GNN-based methods. Details of the selected baseline algorithm are demonstrated as follows:

GCN and *GAT* are semi-supervised homogeneous graph models that use convolution or attention mechanism to aggregate neighborhood information of graph nodes. *HGT* [37] characterize the heterogeneous attention over different types of nodes, and it performs cross-layer messages from different types of neighbors for higher-order aggregation. *GraphSAGE* is an inductive framework that leverages node attribute information to efficiently generate representations on previously unseen data. *FDStar* [9] is a graph convolutional network approach for fraudster detection in review system. *GAS* [15] is a GCN-based Anti-Spam model, capturing the local and global context of a comment to detect spammers. *SemiGNN* is a semisupervised attentive graph neural network that utilizes the multi-view labeled and unlabeled data for fraud detection.

Furthermore, to inspect the validation of relevance-aware GNN-based framework, we decompose the RAU-GNN into a plain relation RAU-GNN(PR-RAU-GNN), which directly sum up the initial embedding across relations without GCN module, and a plain aggregation RAU-GNN(PA-RAU-GNN), which removes the last enhanced GNN aggregator.

C. Experimental Setting

Due to the small percentage of anomalous users in dataset, we adopt mini-batch training techinique to efficiently train RAU-GNN. During the detection process, we set the output dimension of final embeddings 64, batch size 1024 for YelpChi and 256 for Twitter, learning rate 0.005, L_2 regularization weight λ 0.001. The parameters of RAU-GNN are initialized with Xavier and we adopt Adam optimizer. All experiments are conducted on a 64 core Intel Xeon CPU E5-2680 v42.40GHz with 256GB RAM and 1×NVIDIA Tesla P100-PICE GPU. We implement RAU-GNN and other comparison models with Python 3.7.1 and Pytorh 1.6.0.

VI. Overall Evaluation

In this section, we demonstrate the experimental results in detection effectiveness, accuracy and the impact of parameters of RAU-GNN.

We adopt Accuracy score and Recall to quantitatively evaluate the effectiveness of detection. In our experiments, the performances are reported with the best results. We choose different percent of the data samples for training RAU-GNN and the residual are organized for testing. Table III illustrates the Accuracy scores and Recall of each model. It is obvious that RAU-GNN outperforms all other baselines and obtains better classification accuracy. Generally, models adapted for heterogeneous graph perform better compared with other homograph-based models. In summary, RAU-GNN achieves dramatic improvement in the accuracy and effectiveness of anomalous users detection. We evaluate the performance of models at length in the following sections.

A. Single Relation and Multiple Relation

As for all the baseline models in Table III, GCN, GAT, GraphSage and PR-RAU-GNN are implemented on single relation graph that all relations are merged together, wherein PR-RAU-GNN runs on the simple combination of multiple relation graph. HGT, FDStar, GAS and Semi-GNN are implemented on the constructed multiple relation graph. Compared with the performances of single relation models, GNN-based model on multiple relation graph generally obtain better results on accuracy score and recall. Among all the multi-relation GNNs baselines, GAS outperforms all other models, for GAS additionally consider the local contents to enhance the classifier. Better than GAS, RAU-GNN aggregates information from different relations, and consolidate the node embedding on a homogeneous user graph. The experimental results show that RAU-GNN can better filter the anomalous users in social network. It also demonstrates the significance of relations between users when there are more anomalous users lucks around, and verifies the behavior preference of anomalous users that widely connects with benign users. It is noting that the performance of PR-RAU-GNN is similar to other single relation GNN model, which proves the effectiveness of multiple relation aggregation.

B. Training Percentage Analysis

From Table III, we can find there are little difference among the training percentages. Even if the training percentage is increasing, the fluctuation of accuracy score and recall is small, and maintain at a certain level. The experimental results of gapped training percentages demonstrate the advantage of semi-supervised learning, where a small amount of labeled nodes is enough for training a model, and would achieve better classification results.

C. Variants of RAU-GNN

As mentioned above, we decompose the RAU-GNN into two plain model variants. PA-RAU-GNN directly sum up the initial embedding across relations without GCN module. We

TABLE III: Anomalous detection performance (%) on two datasets under different percentage of training data.

	Metric	Train%	GCN	GAT	HGT	Graph-Sage	GAS	FDStar	Semi-GNN	PR-RAU	PA-RAU	RAU-GNN
YELP	Accuracy	10%	52.12	54.06	58.83	53.52	61.54	66.82	52.12	55.63	69.16	**70.04**
		20%	53.21	57.43	60.04	54.37	62.85	65.76	51.86	56.14	70.03	**70.62**
		30%	53.41	57.26	59.43	55.08	61.28	65.43	51.65	55.82	70.42	**70.96**
		40%	52.34	56.48	58.61	54.24	62.04	65.27	51.69	55.75	70.94	**71.20**
	Recall	10%	51.32	53.45	56.12	52.35	56.10	54.33	52.37	54.41	63.51	**63.60**
		20%	53.41	53.95	56.13	52.84	55.62	55.14	52.41	54.36	64.82	**64.78**
		30%	52.76	54.11	55.84	52.63	55.68	54.73	52.18	54.69	65.44	**65.52**
		40%	51.92	54.36	55.92	52.82	55.71	54.58	51.59	54.58	65.43	**65.73**
Twitter	Accurary	10%	68.54	67.72	71.57	65.86	75.41	72.18	68.20	69.12	78.38	**78.42**
		20%	69.23	68.57	70.83	66.93	76.27	73.45	66.53	69.75	78.92	**79.71**
		30%	69.14	67.21	71.46	67.05	73.56	74.37	65.94	70.11	79.58	**79.82**
		40%	68.43	68.13	70.92	67.51	72.32	74.65	66.82	69.67	79.26	**79.64**
	Recall	10%	60.42	61.24	65.16	64.73	68.31	69.55	63.35	65.13	70.49	**70.53**
		20%	59.03	63.41	66.69	64.65	68.65	69.64	64.43	66.29	70.38	**71.49**
		30%	59.75	62.87	65.86	65.69	69.29	68.82	63.87	66.08	71.06	**71.84**
		40%	59.58	62.13	66.73	66.56	68.45	68.57	63.31	65.74	71.35	**71.73**

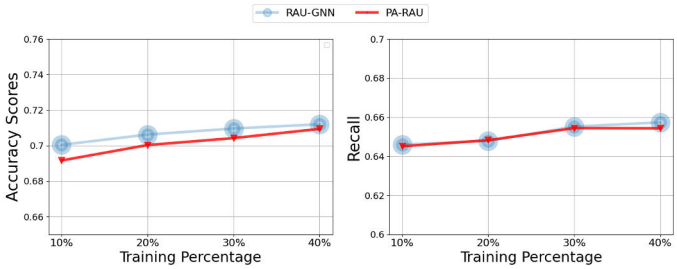

Fig. 3: Model Performance under Different Training Percentage

use PA-RAU-GNN to verify the advantage of aggregation in multiple relation. PA-RAU-GNNremoves the last enhanced GNN aggregator, and use the output from GAT module as the final embeddings. The last 3 columns in Table III shows the performance of RAU-GNN and proposed variants. It is obvious that RAU-GNN outperforms than all other baselines. The performance of PR-RAU-GNN is mentioned in multiple relation analysis. It proves the feasibility of cross-relation aggregation. PA-RAU-GNN achieves better performance, for it retains the GCN-based relation fusion layer. The improvement of RAU-GNN compared with PA-RAU-GNN verifies the effect of enhanced aggregator. Due to that the last aggregator is used to improve the robustness and generalization of model, it only brings little promotion on accuracy score and recall.

D. Hyperparameter Sensitivity

We analyze the hyperparameter in this subsection. Figure 4 shows the testing performance of RAU-GNN. To analyze the number of layers in GCN-based fusion layer, we observe the results by increasing the number of layers. Figure 4(a) shows the performance of different layer numbers on YelpChi dataset. We can see a peak when the number of layer is set to 2. For the 3-layer GCN, RAU-GNN confronts with the overfitting problem, and obtain a worse result. Figure 4(b) shows the effect of different embedding size. We set the output embedding size to 16, 32, 64 and 128. It is obvious that embedding szie with 64 would achieve better results.

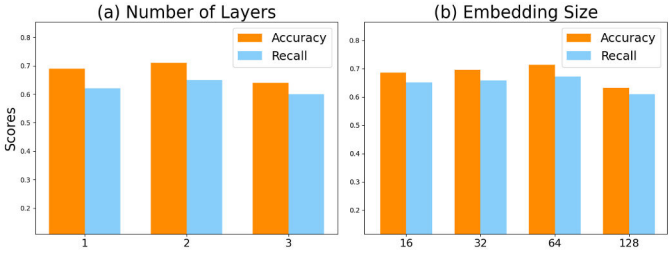

Fig. 4: Hyperparameter Sensitivity Analysis

VII. Conclusion

In this paper, we develop a new GNN-based relevance-aware anomalous user detection model, named RAU-GNN, to effectively discriminate the wel-disguised anomalous users in social network. Firstly, RAU-GNNextracts multiple relations between users in social network, and accordingly constructs the multiple user relation graph. Secondly, we design relevance-aware GNN framework to learn the high-level of users, and discriminate the anomalous users through discriminating. Concretely, we design a GCN-based relation fusion layer to aggregate initial information from different relations through convolutional operation, and a GAT-based embedding

layer to represent the hidden embeddings of users. Lastly, we feed the learned representations to the following GNN aggregator in order to get the node embedding by aggregating the final users' embeddings, and develop the robustness and generalization of RAU-GNN. The experimental results show that our approach can achieve better accuracy for anomalous users detection.

ACKNOWLEDGMENT

The authors of this paper were supported by NSFC through grant U20B2053, and S&T Program of Hebei through grant 20310101D, Major Science and Technology Plan Projects of Hainan through grant ZDKJ2019008, This work was also supported by the Opening Project of Shanghai Trusted Industrial Control Platform.

REFERENCES

[1] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proceedings of the 25th international conference on world wide web*, 2016, pp. 145–153.

[2] B. Qian *et al.*, "Orchestrating the development lifecycle of machine learning-based iot applications: A taxonomy and survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–47, 2020.

[3] C. Hu *et al.*, "Toposch: Latency-aware scheduling based on critical path analysis on shared yarn clusters," in *IEEE Conference on Cloud Computing (Cloud)*, 2020.

[4] Z. Wen *et al.*, "Ga-par: Dependable microservice orchestration framework for geo-distributed clouds," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2019.

[5] R. Yang *et al.*, "D^2ps: A dependable data provisioning service in multi-tenant cloud environment," in *17th IEEE International Symposium on High Assurance Systems Engineering*. IEEE Computer Society, 2016, pp. 252–259.

[6] R. Yang, Y. Zhang *et al.*, "Reliable computing service in massive-scale systems through rapid low-cost failover," *IEEE Trans. Serv. Comput.*, vol. 10, no. 6, pp. 969–983, 2017. [Online]. Available: https://doi.org/10.1109/TSC.2016.2544313

[7] R. Yang *et al.*, "Performance-aware speculative resource oversubscription for large-scale clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 7, pp. 1499–1517, 2020.

[8] X. Yang, Y. Lyu *et al.*, "Rumor detection on social media with graph structured adversarial learning," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*, 2020, pp. 1417–1423.

[9] J. Wang *et al.*, "Fdgars: Fraudster detection via graph convolutional networks in online app review system," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 310–316.

[10] Y. Gao, L. Xiaoyong *et al.*, "Hincti: A cyber threat intelligence modeling and identification system based on heterogeneous information network," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[11] D. M. Beskow and K. M. Carley, "Its all in a name: detecting and labeling bots by their name," *Comput. Math. Organ. Theory*, vol. 25, no. 1, pp. 24–35, 2019.

[12] S. Huang, J. Xie, X. Dai, C. Jiajun *et al.*, "A reinforced generation of adversarial examples for neural machine translation," in *Proceedings of the ACL*, 2020, pp. 3486–3497.

[13] S. Wang, J. Cao *et al.*, "Seqst-gan: Seq2seq generative adversarial nets for multi-step urban crowd flow prediction," *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 6, no. 4, pp. 1–24, 2020.

[14] S. Zhu, J. Li *et al.*, "Adversarial directed graph embedding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[15] A. Li, Z. Qin *et al.*, "Spam review detection with graph convolutional networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2703–2711.

[16] L. Sun, B. Cao, J. Wang, W. Srisa-an, P. Yu, A. D. Leow, and S. Checkoway, "Kollector: Detecting fraudulent activities on mobile devices using deep learning," *IEEE Transactions on Mobile Computing*, 2020.

[17] Z. Liu, Y. Dou *et al.*, "Alleviating the inconsistency problem of applying graph neural network to fraud detection," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieva*, 2020.

[18] Y. Dou, Z. Liu *et al.*, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 315–324.

[19] C. Li, X. Peng, S. Zhang, H. Peng, S. Y. Philip, M. He, L. Du, and L. Wang, "Modeling relation paths for knowledge base completion via joint adversarial training," *Knowledge-Based Systems*, 2020.

[20] J. Zhao *et al.*, "Automatically predicting cyber attack preference with attributed heterogeneous attention networks and transductive learning," *Computers & Security*, vol. 102, p. 102152, 2021.

[21] J. Zhao, X. Liu *et al.*, "Multi-attributed heterogeneous graph convolutional network for bot detection," *Information Sciences*, vol. 537, pp. 380–393, 2020.

[22] H. Peng, J. Li *et al.*, "Streaming social event detection and evolution discovery in heterogeneous information networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2021.

[23] F. Scarselli *et al.*, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[24] H. Peng, J. Li *et al.*, "Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[25] H. Peng, J. Li, Q. Gong, Y. Song, Y. Ning, K. Lai, and P. Yu, "Fine-grained event categorization with heterogeneous graph convolutional networks," in *IJCAI International Joint Conference on Artificial Intelligence*, 2019, pp. 32–38.

[26] H. Peng, R. Yang, Z. Wang *et al.*, "Lime: Low-cost and incremental learning for dynamic heterogeneous information networks," *IEEE Transactions on Computers*, 2021.

[27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proceedings of the ICLR*, 2018.

[28] Y. Cao *et al.*, "Knowledge-preserving incremental social event detection via heterogeneous gnns," in *Proceedings of The Web Conference 2021*, 2021.

[29] H. Peng, J. Li, Q. Gong, Y. Ning, S. Wang, and L. He, "Motif-matching based subgraph-level attentional convolutional network for graph classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5387–5394.

[30] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1285–1298.

[31] P. Kaghazgaran, M. Alfifi, and J. Caverlee, "Wide-ranging review manipulation attacks: Model, empirical study, and countermeasures," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 981–990.

[32] A. Hernandez-Suarez, G. Sanchez-Perez, K. Toscano-Medina, V. Martinez-Hernandez, H. Perez-Meana, J. Olivares-Mercado, and V. Sanchez, "Social sentiment sensor in twitter for predicting cyber-attacks using l1 regularization," *Sensors*, vol. 18, no. 5, p. 1380, 2018.

[33] Q. Sun *et al.*, "Pairwise learning for name disambiguation in large-scale heterogeneous academic networks," in *2020 IEEE International Conference on Data Mining (ICDM)*, 2020, pp. 511–520.

[34] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "Fame for sale: Efficient detection of fake twitter followers," *Decision Support Systems*, vol. 80, pp. 56–71, 2015.

[35] A. Khalil, H. Hajjdiab, and N. Al-Qirim, "Detecting fake followers in twitter: A machine learning approach," *International Journal of Machine Learning and Computing*, vol. 7, no. 6, pp. 198–202, 2017.

[36] Y. Zhang, Y. Fan, Y. Ye, L. Zhao, and C. Shi, "Key player identification in underground forums over attributed heterogeneous information network embedding framework," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 549–558.

[37] Z. Hu, Y. Dong *et al.*, "Heterogeneous graph transformer," in *Proceedings of the WWW*, 2020, pp. 2704–2710.