EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

ResGCN

Citation for published version (APA):
Pei, Y., Huang, T., van Ipenburg, W., & Pechenizkiy, M. (2022). ResGCN: attention-based deep residual modeling for anomaly detection on attributed networks. *Machine Learning*, *111*(2), 519-541. Advance online publication. https://doi.org/10.1007/s10994-021-06044-0

DOI:
10.1007/s10994-021-06044-0

Document status and date:
Published: 01/02/2022

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 05. Oct. 2023

Machine Learning (2022) 111:519–541
https://doi.org/10.1007/s10994-021-06044-0

ResGCN: attention-based deep residual modeling for anomaly detection on attributed networks

Yulong Pei[1] · Tianjin Huang[1] · Werner van Ipenburg[2] · Mykola Pechenizkiy[1]

Received: 30 September 2020 / Revised: 22 July 2021 / Accepted: 25 July 2021 /
Published online: 3 September 2021
© The Author(s) 2021

Abstract

Effectively detecting anomalous nodes in attributed networks is crucial for the success of many real-world applications such as fraud and intrusion detection. Existing approaches have difficulties with three major issues: sparsity and nonlinearity capturing, residual modeling, and network smoothing. We propose Residual Graph Convolutional Network (ResGCN), an attention-based deep residual modeling approach that can tackle these issues: modeling the attributed networks with GCN allows to capture the sparsity and nonlinearity, utilizing a deep neural network allows direct residual ing from the input, and a residual-based attention mechanism reduces the adverse effect from anomalous nodes and prevents over-smoothing. Extensive experiments on several real-world attributed networks demonstrate the effectiveness of ResGCN in detecting anomalies.

Keywords Anomaly Detection · Attention Mechanism · Graph Convolutional Network · Attributed Networks

Editors: João Gama, Alípio Jorge, Salvador García.

✉ Yulong Pei
y.pei.1@tue.nl

Tianjin Huang
t.huang@tue.nl

Werner van Ipenburg
werner.van.ipenburg@rabobank.nl

Mykola Pechenizkiy
m.pechenizkiy@tue.nl

[1] Department of Mathematics and Computer Science, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands

[2] Cooperatieve Rabobank U.A., Utrecht, The Netherlands

1 Introduction

Attributed networks are ubiquitous in a variety of real-world applications. Data from many real-world domains can be represented as attributed networks, where nodes represent entities with attributes and edges express the interactions or relationships between entities. Different from plain networks where only structural information exists, attributed networks also contain rich features to provide more details to describe individual elements of the networks. For instance, in social networks, user profiles contain important information to describe users. In citation networks, paper abstracts can provide complementary information to the citation structures. In gene regulatory networks, gene sequence expressions are the attributes beside the interactions between molecular regulators. Due to the ubiquity of attributed networks, various data mining tasks on attributed networks have attracted an upsurge of interest such as community detection (Falih et al., 2018; Li et al., 2018c; Pei et al., 2015), link prediction (Barbieri et al., 2014; Li et al., 2018a; Brochier et al., 2019), network embedding (Huang et al., 2017b, a; Meng et al., 2019), etc.

Anomaly detection is one of the most vital problems among these tasks on attributed networks because of its significant implications in a wide range of real-world applications including cyber attack detection in computer networks, fraud detection in finance and spammers discovery in social media, to name a few. It is more challenging to detect anomalies on attributed networks because both attributes and structures should be taken into consideration in order to detect anomalous nodes. We illustrate this with a toy example in Fig. 1. The anomalous node is different from others because of two reasons: (1) structurally it connects to all other nodes and (2) its attributes are significantly different from the majority.

Several approaches for anomaly detection on attributed networks have been proposed recently in the literature. Most of them aim at detecting anomalies in an unsupervised fashion because of the prohibitive cost for accessing the ground-truth anomalies (Ding et al., 2019a). They can be categorized into four types of methods that are based on: community analysis, subspace selection, residual analysis, and deep ing. Community analysis methods (Gao et al., 2010) detect anomalies by identifying the abnormality of current node with other nodes within the same community. Subspace selection approaches (Perozzi et al., 2014) first a subspace for features and then discover anomalies in that ed subspace. Residual analysis methods (Li et al., 2017; Peng et al., 2018)

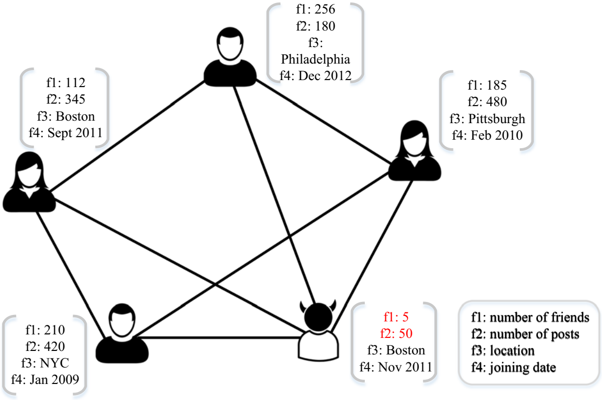


Fig. 1 An illustration of failure in previous message passing based anomaly detection approaches

explicitly model the residual information by reconstructing the input attributed network based on matrix factorization. Deep ing methods use deep neural networks to capture the nonlinearity of networks and detect anomalies in an unsupervised (Ding et al., 2019a) or supervised way (Liang et al., 2018).

However, there are three major issues in existing approaches: (1) sparsity and nonlinearity capturing, (2) residual modeling, and (3) network smoothing. *Capturing sparsity and nonlinearity* is important in anomaly detection on networks because real-world attributed networks are complex and non-linear. Previous shallow models such as nonnegative matrix factorization (Li et al., 2017; Peng et al., 2018) fail to detect anomalies because of the incapability of modeling nonlinearity. Although *residual modeling* has been explored in previous studies (Li et al., 2017; Peng et al., 2018), the residual is modeled from the reconstruction error and similarly only captures linear information because traditional matrix factorization frameworks have been used. Thus, they can neither be adaptively ed from the input networks nor capture more complex nonlinearity of real-world attributed networks. *Smoothing networks*, which is based on the homophily hypothesis (McPherson et al., 2001), is a commonly used strategy to detect anomalies on networks, e.g., (Ding et al., 2019a). However, those methods are not well-suited for anomaly detection because they might oversmooth the node representations, i.e., making anomalous nodes less distinguishable from the majority of normal nodes (Li et al., 2019), because the output representations may be oversmoothed to become similar to neighbors (Li et al., 2018b).

To tackle these issues, in this paper, we propose Residual Graph Convolutional Network (ResGCN), a novel approach for anomaly detection on attributed networks. ResGCN is capable of solving the above three problems as follows: (1) to *capture the sparsity and nonlinearity* of networks, ResGCN is based on GCN to model the attributed networks; (2) to *model residual information*, ResGCN s residual directly from the input using a deep neural network; and (3) to *prevent over-smoothing of node representations*, ResGCN incorporates the attention mechanism based on ed residual information, and different neighbors play different roles in passing messages according to the residual (attention). Thus, the information propagation of anomalous nodes can be reduced. The contributions of this paper are summarized as follows:

- We propose novel anomaly detection method named ResGCN. ResGCN captures the sparsity and nonlinearity of networks using GCN, s the residual information using a deep neural network, and reduces the adverse effect from anomalous nodes using the residual-based attention mechanism.
- We propose a residual information based anomaly ranking strategy and the residual information is ed from the input network instead of reconstruction errors.
- Results of our extensive experiments on real-world attributed networks demonstrate the effectiveness of ResGCN in the task of anomaly detection w.r.t. different evaluation metrics.

The rest of this paper is organized as follows. Section 2 formally defines the problem of anomaly detection on attributed networks. Section 3 introduces the proposed ResGCN model for anomaly detection. Section 4 provides empirical evidence of ResGCN performance on anomaly detection in real-world networks w.r.t. different evaluation metrics. Section 5 briefly discusses related work on anomaly detection on attributed networks. Finally, we conclude in Sect. 6.

Table 1 Table of notations

Symbol	Description
V	Node set
E	Edge set
m	Number of edges
n	Number of nodes
d	Number of attributes
$\boldsymbol{A}$	Adjacency matrix
$\boldsymbol{X}$	Attribute matrix
$\boldsymbol{W}^l$	The trainable weight matrix in the l^ th layer
$\boldsymbol{H}^l$	The latent representation matrix in the l^ th layer
$\boldsymbol{R}^l$	The residual matrix in the *lth* layer
α	The trade-off parameter for reconstruction error
λ	The residual parameter

2 Problem definition

We first summarize some notations and definitions used in this papers. Following the commonly used notations, we use bold uppercase characters for matrices, e.g., $\boldsymbol{X}$, bold lowercase characters for vectors, e.g., $\boldsymbol{b}$, and normal lowercase characters for scalars, e.g., c. The The *ith* row of a matrix X is denoted by $X_{i,:}$ and $(i,j)th$ element of matrix X is denoted as $X_{i,j}$. The Frobenius norm of a matrix is represented as $\|\cdot\|_F$ and $\|\cdot\|_2$ is the L_2 norm. In detail, the main symbols are listed in Table 1.

Definition 1 Attributed Networks. An attributed network $\mathcal{G} = \{V, E, \boldsymbol{X}\}$ consists of: (1) a set of nodes $V = \{v_1, v_2, ..., v_n\}$, where $|V| = n$ is the number of nodes; (2) a set of edges E, where $|E| = m$ is the number of edges; and (3) the node attribute matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, the *ith* row vector $\boldsymbol{X}_{i,:} \in \mathbb{R}^d,\ i = 1, ..., n$ is the attribute of node v_i.

The topological structure of attributed network $\mathcal{G}$ can be represented by an adjacency matrix $\boldsymbol{A}$, where $A_{i,j} = 1$ if there is an edge between node v_i and node v_j. Otherwise, $A_{i,j} = 0$. We focus on the undirected networks in this study and it is trivial to extend it to directed networks. The attribute of $\mathcal{G}$ can be represented by an attribute matrix $\boldsymbol{X}$. Thus, the attributed network can be represented as $\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}\}$. With these notations and definitions, same to previous studies (Li et al. 2017; Peng et al. 2018; Ding et al. 2019a), we formulate the task of anomaly detection on attributed networks:

Problem 1 Anomaly Detection on Attributed Networks. Given an attributed network $\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}\}$, which is represented by the adjacency matrix $\boldsymbol{A}$ and attribute matrix $\boldsymbol{X}$, the task of anomaly detection is to find a set of nodes that are rare and differ singularly from the majority reference nodes of the input network.

3 Proposed method

In this section we first introduce the background of GCN. Next, we present the proposed model ResGCN in details. Then we analyze the complexity of ResGCN.

3.1 Graph convolutional networks

GCN s node representations by passing and aggregating messages between neighboring nodes. Different types of GCN have been proposed recently (Kipf & Welling, 2016a; Hamilton et al., 2017), and we focus on one of the most widely used versions proposed in (Kipf & Welling, 2016a). Formally, a GCN layer is defined as

$$\boldsymbol{h}_i^{(l+1)} = f\Bigg(\sum_{j \in Ne(i)} \frac{1}{\sqrt{\widetilde{\boldsymbol{D}}_{i,i}\widetilde{\boldsymbol{D}}_{j,j}}} \boldsymbol{h}_j^{(l)} \boldsymbol{W}^{(l)} \Bigg), \tag{1}$$

where $\boldsymbol{h}_i^{(l)}$ is the latent representation of node v_i in layer l, $Ne(i)$ is the set of neighbors of node v_i, and $\boldsymbol{W}^l$ is the layer-specific trainable weight matrix. $f(\cdot)$ is a non-linear activation function and we select ReLU as the activation function following previous studies (Kipf and Welling 2016a) (written as $f_{ReLU}(\cdot)$ below). $\widetilde{\boldsymbol{D}}$ is the diagonal degree matrix of $\widetilde{\boldsymbol{A}}$ defined as $\widetilde{D}_{i,i} = \sum_j \widetilde{A}_{i,j}$ where $\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacency matrix of the input attributed network $\boldsymbol{G}$ with self connections $\boldsymbol{I}$. Equivalently, we can rewrite GCN in a matrix form:

$$\boldsymbol{H}^{(l+1)} = f_{ReLU}\Big(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{H}^{(l)} \boldsymbol{W}^{(l)} \Big). \tag{2}$$

For the first layer, $\boldsymbol{H}^{(0)} = \boldsymbol{X}$ is the attribute matrix of the input network. Therefore, we have

$$\boldsymbol{H}^{(1)} = f_{ReLU}\Big(\widetilde{\boldsymbol{A}} \boldsymbol{X} \boldsymbol{W}^{(0)} \Big). \tag{3}$$

The architecture of GCN can be trained end-to-end by incorporating task-specific loss functions. In the original study, GCN aims at semi-supervised classification task so the cross-entropy loss is evaluated by adding the softmax function as the output of the last layer.Formally, the overall cross-entropy error is evaluated on the graph for all the labeled samples:

$$\mathcal{L}_{cls} = - \sum_{i \in L} \sum_{c=1}^{C} \boldsymbol{Y}_{ic} \log \hat{\boldsymbol{Y}}_{ic} \tag{4}$$

where L is the set of nodes with labels, C is the number of classes, $\boldsymbol{Y}$ is the label and $\hat{\boldsymbol{Y}} = softmax(\boldsymbol{H})$ is the prediction of GCN passing the hidden representation in the final layer $\boldsymbol{H}^{(L)}$ to a softmax function.

Note that original GCN (Kipf & Welling, 2016a) is designed for semi-supervised ing, our target is to detect anomalies in an unsupervised way. Therefore, the cross entropy loss for (semi-)supervised ing is not suitable in our problem settings. We will introduce our proposed loss function which is based on network reconstruction errors in the following section.

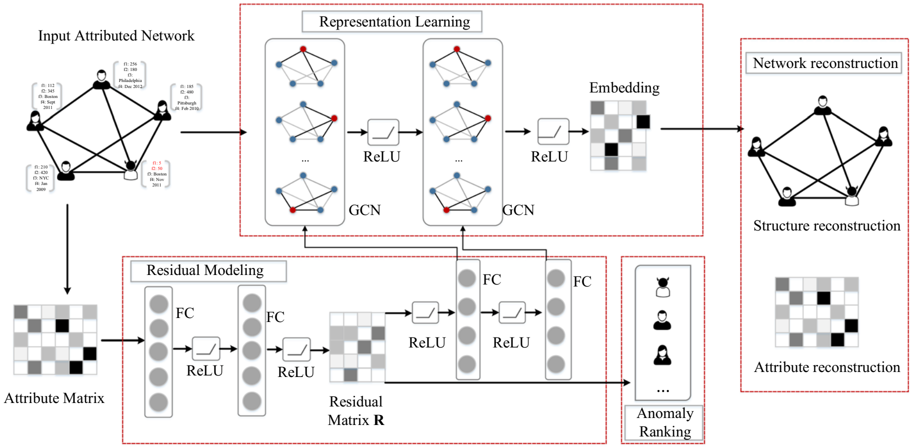


Fig. 2 The framework of our proposed ResGCN

3.2 ResGCN

In this section, we present the proposed framework of ResGCN in details. ResGCN consists of four components: residual modeling, representation ing, network reconstruction and anomaly ranking. The architecture of this model is illustrated in Fig. 2.

3.2.1 Residual modeling

Although some previous studies explicitly model the residual information for anomaly detection on attributed networks, e.g., Radar (Li et al., 2017) and ANOMALOUS (Peng et al., 2018), these methods have two major limitations: (1) They are based on linear models, e.g., matrix factorization, so these shallow models are incapable of capturing the nonlinearity of networks. (2) The residual information has been modeled from the reconstruction error. Thus, they cannot be adaptively ed from the input networks. However, real-world networks are complex and residual information has different patterns in different datasets. Motivated by the study (Dabkowski & Gal, 2017), which proposes to the saliency map based on convolutional network, we propose to use a deep neural network to the residual by capturing the nonlinearity in ResGCN. Formally,

$$\boldsymbol{R}^{(l+1)} = f_{ReLU}(\boldsymbol{R}^{(l)} \cdot \boldsymbol{W}^{(l)}), \tag{5}$$

where $\boldsymbol{R}^{(l)}$ is the input for the fully connected (FC) layer l, and $\boldsymbol{W}^{(l)}$ is the layer-specific trainable weight matrix which needs to be ed during the training of the model. Note that R_i^l is the residual for node i in the lth layer. The output of this network is the residual matrix, denoted as $\boldsymbol{R}$.

Another aim of the residual modeling component is to the attention weights to control the message passing in network representation based on the residual information. Similarly, we use FC layer which takes the residual matrix $\boldsymbol{R}$ as input and the calculation is the same to Eq. 5. Each output of the FC layer corresponds to the attention weights for each

GCN layer shown in Figure 2. Therefore, the number of FC layers to the weights is equal to the number of GCN layers which will be presented below.

3.2.2 Representation ing

The second component of ResGCN aims at ing representations of the input attributed network. Our proposed representation ing method can not only capture the sparsity and nonlinearity of networks but also prevent the information propagating of anomalies. In this component, we adopt GCN with attention which is based on the residual information modeled in the first component to the embeddings of nodes. To make the computations tractable, we follow (Zhu et al., 2019) and assume all hidden representations of nodes are independent. Therefore, we can aggregate node neighbors as follows:

$$\boldsymbol{h}_i^{(l)} = \sum_{j \in Ne(i)} \frac{1}{\sqrt{\widetilde{\boldsymbol{D}}_{i,i}\widetilde{\boldsymbol{D}}_{j,j}}} \boldsymbol{h}_j^{(l)}. \tag{6}$$

To prevent the information propagation from the anomalous nodes, we propose an attention mechanism based on the residual information modeled by the first component to assign different weights to neighbors. The reason is that it is intuitive the nodes with larger residual errors are more likely to be anomalies (Li et al., 2017). We use the smooth exponential function to control the effect of residual information on weights. This is because this function follows graph attention networks (GAT) (Velickovic et al., 2017) which uses the normalized exponential function to model the attention and it also performs well in capturing the uncertainty of graph representation ing in (Zhu et al., 2019). Formally, the weight is defined as

$$\boldsymbol{\theta}_j^{(l)} = \exp(-\gamma \boldsymbol{R}_j^{(l)}), \tag{7}$$

where $\boldsymbol{\theta}_j^{(l)}$ are the attention weights of node v_j in the l^th layer and γ is a hyper-parameter. By taking the attention weights into account, the modified aggregated node neighbor representation can be written as:

$$\boldsymbol{h}_i^{(l)} = \sum_{j \in Ne(i)} \frac{1}{\sqrt{\widetilde{\boldsymbol{D}}_{i,i}\widetilde{\boldsymbol{D}}_{j,j}}} \boldsymbol{h}_j^{(l)} \circ \boldsymbol{\theta}_j^{(l)}, \tag{8}$$

where $\circ$ is the element-wise product. Then we apply able filters and non-linear activation function (ReLU used in this study) to $\boldsymbol{h}_{Ne(i)}^{(l)}$ in order to calculate $\boldsymbol{h}_i^{(l)}$. Formally the layer is defined as:

$$\boldsymbol{h}_i^{(l+1)} = f\Big(\sum_{j \in Ne(i)} \frac{1}{\sqrt{\widetilde{\boldsymbol{D}}_{i,i}\widetilde{\boldsymbol{D}}_{j,j}}} \Big(\boldsymbol{h}_j^{(l)} \circ \boldsymbol{\theta}_j \Big) \boldsymbol{W}^{(l)} \Big). \tag{9}$$

Equivalently, the matrix form is:

$$\boldsymbol{H}^{(l+1)} = f_{ReLU}\Big(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \Big(\boldsymbol{H}^{(l)} \circ \boldsymbol{\Theta} \Big) \boldsymbol{W}^{(l)} \Big), \tag{10}$$

where $\boldsymbol{\Theta} = \exp(-\gamma \boldsymbol{R}^{(l)})$. Similarly, for the first layer, we have

$$\boldsymbol{H}^{(1)} = f_{ReLU}\left(\widetilde{\boldsymbol{A}}\boldsymbol{X}\boldsymbol{W}^{(0)}\right). \tag{11}$$

The output of the last GCN layer is the node embedding matrix $\boldsymbol{Z}$.

3.2.3 Network reconstruction

The target of the third component of ResGCN is to reconstruct the network which consists of structure reconstruction and attribute reconstruction. Both reconstructions are based on the latent representation $\boldsymbol{Z}$ ed in the representation ing component.

Structure Reconstruction Let $\hat{\boldsymbol{A}}$ denote the reconstructed adjacency matrix. Following (Ding et al., 2019a; Kipf & Welling, 2016b), we use the inner product of the latent representations between two nodes to predict if an edge exists between them. Intuitively, if the latent representations of two nodes are similar, it is more likely that there is an edge between them. Formally, the prediction between two nodes v_i and v_j can represented as follows:

$$P(\hat{A}_{i,j} = 1|z_i, z_j) = f_{sigmoid}(z_i, z_j), \tag{12}$$

where $f_{sigmoid}$ function is to convert the prediction as a probability value. Accordingly, the whole reconstructed network structure based on the latent representations $\boldsymbol{Z}$ can be represented as follows:

$$\hat{\boldsymbol{A}} = sigmoid(\boldsymbol{Z}\boldsymbol{Z}^T). \tag{13}$$

Correspondingly, the reconstruction error for structure can be represented as:

$$E_S = \|\boldsymbol{A} - \hat{\boldsymbol{A}}\|_F^2. \tag{14}$$

Attribute Reconstruction To reconstruct the original attributes, DOMINANT (Ding et al., 2019a) uses another graph convolution layer as the decoder to reconstruct the attributes. However, considering that graph convolution is simply a special form of Laplacian smoothing and mixes the nodal features and its nearby neighbors (Li et al., 2018b), we adopt the multi-layer perceptron as our decoder instead. Formally, let $\hat{\boldsymbol{X}}$ be the reconstructed attributes and the reconstruction process can be formalized as follows:

$$\hat{\boldsymbol{X}} = \Phi^n(\boldsymbol{Z}), \tag{15}$$

where n denotes the number of FC layers and $\Phi^n(\cdot)$ denotes n-layer perceptron which is composed with linear functions followed by non-linear activation function. By taking the residual into consideration, the attribute reconstruction is:

$$E_A = \|\boldsymbol{X} - \hat{\boldsymbol{X}} - \lambda\boldsymbol{R}\|_F^2, \tag{16}$$

where λ is the residual parameter to control how much residual information we want to use in the attribute reconstruction error. This error is similar to (Li et al., 2017; Peng et al., 2018) which explicitly incorporate the residual information in attribute reconstruction.

Based on the structure and attribute reconstruction errors, we can propose the objective function of our proposed ResGCN model. To jointly the reconstruction errors, the objective function of ResGCN is defined as the weighted combination of two errors:

$$\mathcal{L} = (1-\alpha)E_S + \alpha E_A \\ = (1-\alpha)\|\boldsymbol{A}-\hat{\boldsymbol{A}}\|_F^2 + \alpha\|\boldsymbol{X}-\hat{\boldsymbol{X}}-\lambda\boldsymbol{R}\|_F^2, \tag{17}$$

where α is the trade-off parameter to control the importance of errors from structure and attributed reconstruction. By minimizing the objective function, we aim to approximate the input attributed network based on the latent representations. Different from previous studies which rank reconstruction errors to detect anomalous nodes (Ding et al., 2019a), in our proposed model, we rank the residual matrix $\boldsymbol{R}$ for anomaly identification. Formally, the anomaly score for node v_i is

$$score(v_i) = \|\boldsymbol{R}_{i,:}\|_2. \tag{18}$$

Finally, the anomalies are the nodes with larger scores and we can detect anomalies according to the ranking of anomaly scores. This ranking strategy is superior to reconstruction error based methods because in our model the residual is explicitly ed from the data and implicitly updated by minimizing the reconstruction error. Therefore, it can better capture the anomaly of the data and less be adversely influenced by the noise from the model.

3.3 Complexity analysis

The computational complexity of GCN is linear to the number of edges on the network. For a particular layer, the convolution operation is $\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{W}$ and its complexity is $O(edf)$ (Ding et al., 2019a), where e is the number of non-zero elements in the adjacency matrix $\boldsymbol{A}$, d is the dimensions of attributes, and f is the number of feature maps of the weight matrix. For network reconstruction, we use link prediction to reconstruct the structure and multi-layer perceptron to reconstruct the attribute both of which are pairwise operations. Thus, the overall complexity is $O(ed\boldsymbol{F} + n^2)$ where $\boldsymbol{F}$ is the summation of all feature maps across different layers.

One issue in our model is that the complexity is quadratic with respect to the number of nodes in a network. Compared to anomaly detection methods only considering graph structures, most attributed network anomaly detection methods have higher computational complexity. A simple comparison is shown in Table 12 of Section 5. One reason is that these methods have to compute correlation between nodes w.r.t. attributes. However, for deep ing based methods, they can be efficiently conducted using advanced hardware such as GPUs and TPUs. We would like to leave other algorithmic improvements such as sampling strategies as future work.

4 Experiments

We evaluate the effectiveness of our proposed ResGCN model[1] on several real-world datasets and present experimental results in order to answer the following three research questions.

[1] The source code is available at https://bitbucket.org/paulpei/resgcn.

Table 2 Statistics of networks with ground-truth anomaly labels

	Amazon	Enron
# nodes	1418	13533
# edges	3695	176987
# attributes	28	20
# anomalies	28	5

Table 3 Statistics of networks without ground-truth anomaly labels

	BlogCatalog	Flickr	ACM
# nodes	5196	7575	16484
# edges	171743	239738	71980
# attributes	8189	12074	8337
# anomalies	300	450	600

- **RQ1**: Does ResGCN improve the anomaly detection performance on attributed networks?
- **RQ2**: Is deep residual matrix ranking strategy effective in identifying anomalies?
- **RQ3**: How do the parameters in ResGCN affect the anomaly detection performance?

4.1 Datasets

In order to evaluate the effectiveness of our proposed method, we conduct experiments on two types of real-world attributed networks: data with and without ground-truth anomaly labels. All networks have been widely used in previous studies (Li et al., 2017; Peng et al., 2018; Ding et al., 2019a; Gutiérrez-Gómez et al., 2019):

- Networks with ground-truth anomaly labels: Amazon and Enron[2]. Amazon is a co-purchase network (Müller et al., 2013). It contains 28 attributes for each node describing properties about online items including rating, price, etc. The anomalous nodes are defined as nodes having the tag *amazonfail*. Enron is an email network (Metsis et al., 2006) where each node is an email with 20 attributes describing metadata of the email including content length, number of recipients, etc, and each edge indicates the email transmission between people. Spammers are labeled as the anomalies in Enron data. The details of these attributed networks are shown in Table 2.
- Networks without ground-truth anomaly labels: BlogCatalog, Flickr and ACM[3]. BlogCatalog is a blog sharing website where users are the nodes and following relations between users are edges. Each user is associated with a list of tags to describe themselves and their blogs, which are used as attributes. Flickr is an image hosting and sharing website. Similarly, users and user following relations are nodes and edges, respectively. Tags are the attributes. ACM is a citation network where each node is a paper

[2] https://www.ipd.kit.edu/mitarbeiter/muellere/consub/.

[3] https://www4.comp.polyu.edu.hk/~xiaohuang/Code.html.

and each edge indicates a citation relation between papers. Paper abstracts are used as attributes. The details of these attributed networks are shown in Table 3.

For the networks with labels, we directly use these provided labels to evaluate our method. For the data without labels, we need to manually inject anomalies for empirical evaluation. To make a fair comparison, we follow previous studies for anomaly injection (Ding et al., 2019a). In specific, two anomaly injection methods have been used to inject anomalies by perturbing topological structure and nodal attributes, respectively:

- **Structural anomalies**: structural anomalies are generated by perturbing the topological structure of the network. It is intuitive that in real-world networks, small cliques are typically anomalous in which a small set of nodes are much more connected to each other than average (Skillicorn, 2007). Thus, we follow the method used in (Ding et al., 2019a, b) to generate some small cliques. In details, we randomly select s nodes from the network and then make those nodes fully connected, and then all the s nodes forming the clique are labeled as anomalies. t cliques are generated repeatedly and totally there are $s \times t$ structural anomalies.
- **Attribute anomalies**: we inject an equal number of anomalies from structural perspective and attribute perspective. Same to (Ding et al., 2019a; Song et al., 2007), $s \times t$ nodes are randomly selected as the attribute perturbation candidates. For each selected node v_i, we randomly select another k nodes from the network and calculate the Euclidean distance between v_i and all the k nodes. Then the node with largest distance is selected as v_j and the attributes $\boldsymbol{X}_j$ of node v_j is changed to $\boldsymbol{X}_i$ of node v_i. The selected node v_j is regarded as the attribute anomaly.

In the experiments, we set $s = 15$ and set t to 10, 15, and 20 for BlogCatalog, Flickr and ACM, respectively which are the same to (Ding et al., 2019a) in order to make the comparison with DOMINANT (Ding et al., 2019a). To facilitate the ing process, in our experiments, we follow (Ding et al., 2019b) to reduce the dimensionality of attributes using Principal Component Analysis (PCA) and the dimension is set to 20.

4.2 Evaluation metrics

In the experiments, we use two evaluation metrics to validate the performance of these anomaly detection approaches:

- **ROC-AUC**: we use the area under the receiver operating characteristic curve (ROC-AUC) as the evaluation metric for anomaly detection as it has been widely used in previous studies (Li et al., 2017; Peng et al., 2018; Ding et al., 2019a; Gutiérrez-Gómez et al., 2019). ROC-AUC can quantify the trade-off between true positive rate (TP) and false positive rate (FP) across different thresholds. The TP is defined as the detection rate, i.e. the rate of true anomalous nodes correctly identified as anomalous, whereas the FP is the false alarm rate, i.e. rate of normal nodes identified as anomalous (Gutiérrez-Gómez et al., 2019).
- **Precision@K and Recall@K**: Since we use the ranking strategy to detect anomalies, measures used in ranking-based tasks such as information retrieval and recommender systems can be utilized to evaluate the performance. Thus, we use Precision@K to

Table 4 Performance of different anomaly detection methods w.r.t. ROC-AUC with standard deviation. The bold indicates the best performance of all the methods

	Amazon	Enron
AutoEncoder	0.581 ± 0.004	0.311 ± 0.011
LOF (Breunig et al. 2000)	0.490 ± 0.006	0.440 ± 0.013
AMEN (Perozzi and Akoglu 2016)	0.470 ± 0.005	0.470 ± 0.008
Radar (Li et al. 2017)	0.580 ± 0.007	0.650 ± 0.009
ANOMALOUS (Peng et al. 2018)	0.602 ± 0.004	0.695 ± 0.007
DOMINANT (Ding et al. 2019a)	0.625 ± 0.005	0.685 ± 0.015
MADAN (Gutiérrez-Gómez et al. 2019)	0.680 ± 0.016	0.680 ± 0.009
ResGCN (Our Model)	0.710 ± 0.010	0.660 ± 0.007

measure the proportion of true anomalies that an approach discovered in its top K ranked nodes and Recall@K to measure the proportion of true anomalies that a method discovered in the total number of ground truth anomalies.

4.3 Baselines

To demonstrate the effectiveness of ResGCN in detecting anomalies, we compare it with the following anomaly detection methods:

- **LOF** (Breunig et al., 2000) measures how isolated the object is with respect to the surrounding neighborhood and detects anomalies at the contextual level. LOF only considers nodal attributes.
- **AMEN** (Perozzi & Akoglu, 2016) uses both attribute and network structure information to detect anomalous neighborhoods. Specifically, it analyzes the abnormality of each node from the ego-network point of view.
- **Radar** (Li et al., 2017) is an unsupervised anomaly detection framework for attributed networks. It detects anomalies whose behaviors are singularly different from the majority by characterizing the residuals of attribute information and its coherence with network information.
- **ANOMALOUS** (Peng et al., 2018) is a joint anomaly detection framework to optimize attribute selection and anomaly detection using CUR decomposition of matrix and residual analysis on attributed networks.
- **DOMINANT** (Ding et al., 2019a) utilizes GCN to a low-dimensional embedding representations of the input attributed network and then reconstruct both the topological structure and nodal attributes with these representations. Anomalies are selected by ranking the reconstruction errors.
- **MADAN** (Gutiérrez-Gómez et al., 2019) is a multi-scale anomaly detection method. It uses the heat kernel as filtering operator to exploit the link with the Markov stability to find the context for anomalous nodes at all relevant scales of the network.

In the experiments, for ResGCN, we optimize the loss function with Adam (Kingma & Ba, 2014) algorithm. Other parameter settings are shown in the Appendix. Besides, we use a two-layer autoencoder as a simple baseline. It disregards the structural information

Table 5 Performance of different anomaly detection methods w.r.t. precision@K on BlogCatalog. The bold indicates the best performance of all the methods

K	50	100	200	300
AutoEncoder	0.120	0.120	0.120	0.107
LOF (Breunig et al. 2000)	0.300	0.220	0.180	0.183
Radar (Li et al. 2017)	0.660	0.670	0.550	0.416
ANOMALOUS (Peng et al. 2018)	0.640	0.650	0.515	0.417
DOMINANT (Ding et al. 2019a)	0.760	0.710	0.590	0.470
MADAN (Gutiérrez-Gómez et al. 2019)	0.600	0.620	0.520	0.410
ResGCN (Our Model)	**0.848**	**0.860**	**0.670**	**0.483**

Table 6 Performance of different anomaly detection methods w.r.t. precision@K on Flickr. The bold indicates the best performance of all the methods

K	50	100	200	300
AutoEncoder	0.240	0.206	0.160	0.120
LOF (Breunig et al. 2000)	0.420	0.380	0.270	0.237
Radar (Li et al. 2017)	0.740	0.700	0.635	0.503
ANOMALOUS (Peng et al. 2018)	**0.790**	0.710	0.650	0.510
DOMINANT (Ding et al. 2019a)	0.770	0.730	0.685	0.593
MADAN (Gutiérrez-Gómez et al. 2019)	0.710	0.680	0.620	0.540
ResGCN (Our Model)	0.780	**0.830**	**0.875**	**0.680**

Table 7 Performance of different anomaly detection methods w.r.t. precision@K on ACM. The bold indicates the best performance of all the methods

K	50	100	200	300
AutoEncoder	0.030	0.030	0.024	0.020
LOF (Breunig et al. 2000)	0.060	0.060	0.045	0.037
Radar (Li et al. 2017)	0.560	0.580	0.520	0.430
ANOMALOUS (Peng et al. 2018)	0.600	0.570	0.510	0.410
DOMINANT (Ding et al. 2019a)	0.620	0.590	0.540	0.497
MADAN (Gutiérrez-Gómez et al. 2019)	0.580	0.540	0.560	0.420
ResGCN (Our Model)	**0.812**	**0.780**	**0.675**	**0.573**

and only reconstructs the attributes. Then the reconstruction errors are used as the ranking score to detect anomalies.

4.4 Experimental results

We conduct experiments to evaluate the performance of ResGCN by comparing it with several baselines on two different types of networks: networks with and without ground-truth anomaly labels. The experimental results w.r.t. ROC-AUC for networks with ground-truth labels are shown in Table 4. We observe from these results the following:

Table 8 Performance of different anomaly detection methods w.r.t. recall@K on BlogCatelog. The bold indicates the best performance of all the methods

K	50	100	200	300
AutoEncoder	0.026	0.028	0.033	0.036
LOF (Breunig et al. 2000)	0.050	0.073	0.120	0.183
Radar (Li et al. 2017)	0.110	0.223	0.367	0.416
ANOMALOUS (Peng et al. 2018)	0.107	0.217	0.343	0.417
DOMINANT (Ding et al. 2019a)	0.127	0.237	0.393	0.470
MADAN (Gutiérrez-Gómez et al. 2019)	0.105	0.215	0.375	0.380
ResGCN (Our Model)	**0.143**	**0.299**	**0.456**	**0.483**

Table 9 Performance of different anomaly detection methods w.r.t. recall@K on Flickr. The bold indicates the best performance of all the methods

K	50	100	200	300
AutoEncoder	0.016	0.024	0.056	0.082
LOF (Breunig et al. 2000)	0.047	0.084	0.120	0.158
Radar (Li et al. 2017)	0.082	0.156	0.282	0.336
ANOMALOUS (Peng et al. 2018)	0.087	0.158	0.289	0.340
DOMINANT (Ding et al. 2019a)	0.084	0.162	0.304	0.396
MADAN (Gutiérrez-Gómez et al. 2019)	0.078	0.150	0.306	0.356
ResGCN (Our Model)	**0.088**	**0.187**	**0.393**	**0.458**

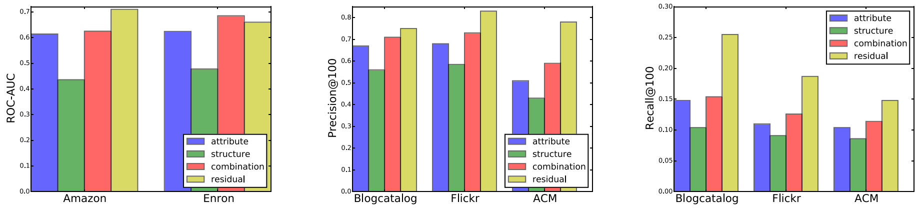
Table 10 Performance of different anomaly detection methods w.r.t. recall@K on ACM. The bold indicates the best performance of all the methods

K	50	100	200	300
AutoEncoder	0.003	0.007	0.010	0.013
LOF (Breunig et al. 2000)	0.005	0.010	0.015	0.018
Radar (Li et al. 2017)	0.047	0.097	0.173	0.215
ANOMALOUS (Peng et al. 2018)	0.050	0.095	0.170	0.205
DOMINANT (Ding et al. 2019a)	0.052	0.098	0.180	0.248
MADAN (Gutiérrez-Gómez et al. 2019)	0.052	0.086	0.210	0.225
ResGCN (Our Model)	**0.079**	**0.148**	**0.235**	**0.309**

- The proposed ResGCN model outperforms other baseline methods on Amazon data and achieves comparable result on Enron data. It demonstrates the effectiveness of ResGCN.
- Deep models such as DOMINANT and residual analysis based methods such as Radar and ANOMALOUS are superior to traditional approaches such as LOF and AMEN. It further validates the effectiveness of deep models and residual modeling.

The experimental results w.r.t. Precision@K and Recall@K for networks without ground-truth labels are shown in Tables 5, 6, 7, 8, 9 and 10. From these evaluation results, we draw the following conclusions:

- The proposed ResGCN model outperforms other baseline methods on all three attributed networks except Precision@50 on Flickr. It demonstrates the effectiveness of our



(a) ROC-AUC **(b)** Precision@100 **(c)** Recall@100

Fig. 3 Comparison of different ranking strategies based on structure reconstruction, attribute reconstruction, combination of structures and attributes and residual information for anomaly detection: **a** ROC-AUC on Amazon and Enron, **b** Precision@100 on BlogCatalog, Flickr and ACM, and **c** Recall@100 on BlogCatalog, Flickr and ACM

method by combining residual modeling and deep representation ing using deep neural networks to detect anomalies.

- Superiority of ResGCN to other approaches in Precision@K and Recall@K indicates our proposed model can achieve higher detection accuracy and also find more true anomalies within the ranking list of limited length.
- Anomaly detection approaches using the deep architecture achieve better performance including ResGCN and DOMINANT. This verifies the importance of nonlinearity modeling for anomaly detection on attributed networks.
- The residual analysis based models, i.e., Radar and ANOMALOUS, although fail in capturing the nonlineariry of networks, achieve better performance than conventional approaches such as LOF. This demonstrates the rationality of explicit residual modeling in anomaly detection.

4.5 Ranking strategy analysis

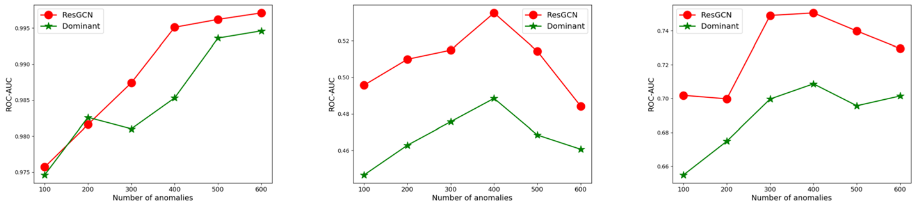
One of the advantages of our proposed ResGCN is the deep residual modeling to capture the anomalous information. Therefore, different from DOMINANT (Ding et al., 2019a), which ranks the weighted combination of attribute and structure reconstruction errors to select the anomalous nodes, we rank the residual information for anomaly detection. In this section, we compare different ranking strategies for anomaly detection: (1) ranking attribute reconstruction errors, (2) ranking structure reconstruction errors, (3) ranking the weighted combination of attribute and structure reconstruction errors, and (4) ranking based on the residual matrix. The first three strategies have been used in (Ding et al., 2019a) and the last one has been used in Radar (Li et al., 2017). The results of anomaly detection w.r.t. ROC-AUC on Amazon and Precision@100 and Recall@100 on BlogCatalog are shown in Fig. 3.

From the results, it can be observed that:

- ranking based on the residual matrix outperforms other ranking strategies on all the datasets w.r.t. different evaluation metrics except on Enron dataset. It demonstrates the effectiveness of residual modeling in ResGCN for anomaly detection.

Table 11 Comparison of residuals between normal and anomalous nodes

	Normal nodes	Anomalous nodes
Amazon	3.925	4.259
Enron	1.544	1.992
BlogCatalog	4.568	5.570



(a) Attributed anomalies **(b)** Structural anomalies **(c)** Mixed anomalies

Fig. 4 Results of different numbers of injected anomalies in BlogCatalog: **a** only injecting structural anomalies, **b** only injecting attributed anomalies, and **c** injecting both structural and attributed anomalies

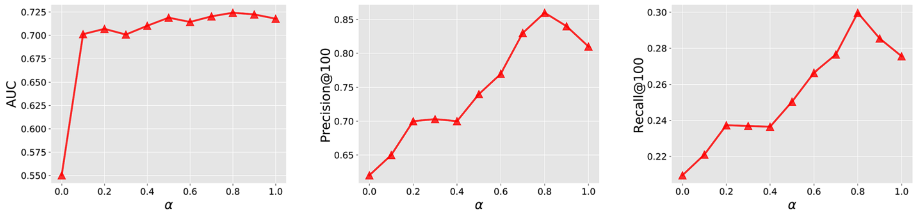
- By combining attribute and structure reconstruction errors, better detection performance can be achieved. This result indicates that both attributes and structures contain some useful information to detect anomalies.
- An interesting observation is that attributes play a more important role in detecting anomalies than structures as ranking attribute reconstruction errors performs better than structure construction errors.

In order to validate the effectiveness of the residual information, we compare the average residuals between normal and anomalous nodes in all these datasets. The results are shown in Table 11. From this comparison, it can be observed that the average residuals for anomalous nodes are larger than that of normal nodes across all datasets. It demonstrates that residual modelling in ResGCN results is an effective indicator helping to distinguish normal and anomalous nodes in attributed networks.

4.6 Attributed anomalies vs. structural anomalies

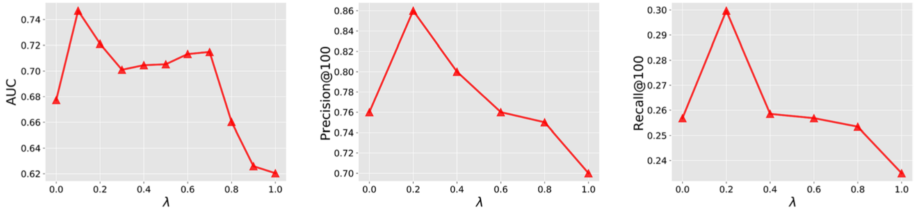
To further analyze the performance of our proposed ResGCN in detecting different types of anomalies, we conduct experiments in three different settings: (1) only injecting structural anomalies, (2) only injecting attributed anomalies, and (3) injecting both structural and attributed anomalies. To be consistent, we follow the same strategies introduced in Sect. 4.1 to inject these anomalies. Besides, we vary the numbers of injected anomalies from 100 to 600 to further investigate the effect of anomaly sizes on the performance of ResGCN.

We conduct the experiment on BlogCatalog. The ROC-AUC curves corresponding to different settings are shown in Fig. 4. We also compare ResGCN to Dominant in different settings because both methods share the same GCN component. From the results, the following conclusions can be drawn:



(a) ROC-AUC on Amazon **(b)** P@100 on BlogCatalog **(c)** R@100 on BlogCatalog

Fig. 5 Influence of the trade-off parameter α for structure and attribute reconstruction errors (ranging from 0.0 to 1.0): **a** ROC-AUC on Amazon, **b** Precision@100 on BlogCatalog, and **c** Recall@100 on BlogCatalog



(a) ROC-AUC on Amazon **(b)** P@100 on BlogCatalog **(c)** R@100 on BlogCatalog

Fig. 6 Influence of the residual parameter λ for loss function (ranging from 0.0 to 1.0): **a** ROC-AUC on Amazon, **b** Precision@100 on BlogCatalog, and **c** Recall@100 on BlogCatalog

- Generally, it is more difficult to detect structural anomalies than attributed anomalies; we can observe much higher AUC-ROC in detecting structural anomalies (Fig. 4a). This is due to the complex patterns of network data, e.g., non-IID distributions and subgraph structures.
- With more attributed anomalies injected, the detection rate increases. On the contrary, more injected structural anomalies first corresponds to an increasing detection rate, which then decreases when the number of anomaly is becoming larger than 400. This may happen because added structural anomalous nodes make the network even more difficult to analyze.
- Our proposed ResGCN outperforms Dominant in different setting which demonstrates the effectiveness of our method. Considering the key difference between ResGCN and Dominant, it demonstrates the effectiveness of residual modeling in guiding representation ing of GCN.

4.7 Parameter analysis

ResGCN has two specific (hyper)parameters: (1) α setting the trade-off for structure and attribute reconstruction errors, and (2) λ corresponding to the residual in the loss function in Eq. 17.

We investigate the impact of these two parameters separately. Specifically, we test the anomaly detection performance by ranging α and λ from 0.0 to 1.0 on Amazon and

BlogCatalog datasets. The results are summarized in in Figures 5 and 6 from which we can observe that:

- The influence of α shows different trends on each of the networks. For Amazon, the performance becomes much better when $\alpha \geq 0.1$. For BlogCatalog, larger α achieves better performance. The commonnality is that ResGCN achieves the best performance on both networks when $\alpha = 0.8$.
- The impact of λ is similar on different networks, i.e., both Amazon and BlogCatalog prefer smaller α. Empirically, the best detection performance can be achieved when $\lambda = 0.1$ on Amazon and $\lambda = 0.2$ on BlogCatalog.

5 Related work

Anomaly detection is one of the active research areas in data mining and machine ing. There are different anomalies in different types of data, e.g., text (Kannan et al., 2017; Ruff et al., 2019), network (Bhuyan et al., 2013) and temporal (Gupta et al., 2013) data. Earlier studies of anomaly detection on graphs mainly focused on structural anomalies, e.g., (Noble & Cook, 2003) and (Eberle & Holder, 2007). However, compared to anomaly detection approaches on plain networks, anomaly detection on attributed networks is more challenging because both structures and attributes should be taken into consideration. In this section, we summarize the related work of anomaly detection on attributed networks.

Real-world networks often come with auxiliary attribute information, so recent years have witnessed an increasingly amount of efforts in detecting anomalies on attributed networks. Existing anomaly detection approaches on attributed networks can be categorized into several different types (Ding et al. 2019a): community analysis, subspace selection,residual analysis and deep ing methods.

CODA (Gao et al., 2010) focuses on community anomalies by simultaneously finding communities as well as spotting anomalies using a unified probabilistic model. AMEN (Perozzi & Akoglu, 2016) uses both attribute and network structure information to detect anomalous neighborhoods. Radar (Li et al., 2017) detects anomalies whose behaviors are singularly different from the majority by characterizing the residuals of attribute information and its coherence with network information. ANOMALOUS (Peng et al., 2018) is a joint anomaly detection framework to optimize attribute selection and anomaly detection using CUR decomposition of matrix and residual analysis on attributed networks. DOMINANT (Ding et al., 2019a) utilizes GCN to compress the input attributed network to succinct low-dimensional embedding representations and then reconstruct both the topological structure and nodal attributes with these representations. MADAN (Gutiérrez-Gómez et al., 2019) is a multi-scale anomaly detection method. It uses the heat kernel as filtering operator to exploit the link with the Markov stability to find the context for outlier nodes at all relevant scales of the network. For traditional anomaly detection methods on graphs, interested readers are referred to (Akoglu et al., 2015) for detailed discussion.

With the popularity of network embedding techniques, which assigns nodes in a network to low-dimensional representations and these representations can effectively preserve the network structure (Cui et al., 2018), ing anomaly aware network representations also attracts huge attentions. Recently, there are several studies taking both problems into

Table 12 Comparison of different anomaly detection methods.

Method	Struc Anomaly	Attr Anomaly	Complexity
LOF	√	×	$O(n \log n)$
OddBall	√	×	–
CatchSync	√	×	$O(e + cn)$
SCAN	√	×	$O(n)$
AMEN	√	√	$O(\|C\|^2 d + md)$
Radar	√	√	$O(kdn^2 + kn^3)$
ANOMOULOUS	√	√	$O(kdn^2)$
Dominant	√	√	$O(edH + n^2)$
MADAN	√	√	$O(n^2)$
AnomalyDAE	√	√	$O(ne + me + de + nd + n^2)$
ResGCN	√	√	$O(edH + n^2)$

Struc Anomaly and Attr Anomaly indicate for each method whether it can detect structural anomalies and attribute anomalies respectively. In the complexity, n, m and d are number of nodes, edges and attributes respectively. k and e are the number of iterations and dimension of embeddings respectively. In Dominant and our proposed ResGCN, H is the summation of all feature maps across different layers. In CatchSync, c is the number of grids and in AMEN, $|C|$ is the neighborhood size

consideration to anomaly aware network embedding in attributed networks (Liang et al., 2018; Zhou et al., 2018; Bandyopadhyay et al., 2019; Li et al., 2019; Bandyopadhyay et al., 2020). SEANO (Liang et al. 2018) is a semi-supervised network embedding approach which s a low-dimensional vector representation that systematically captures the topological proximity, attribute affinity and label similarity of nodes. SPARC (Zhou et al., 2018) is a self-paced framework for anomaly detection which gradually s the rare category oriented network representation. ONE (Bandyopadhyay et al., 2019) jointly align and optimize the structures and attributes to generate robust network embeddings by minimizing the effects of outlier nodes. DONE and AdONE (Bandyopadhyay et al., 2020) use two parallel autoencoders for link structure and attributes of the nodes respectively. By exploring the reconstruction errors for structures and attributes, the proposed methods can embedding and detect anomalies. Another related embedding methods aim to capture the uncertainties of ed representations, such as and DVNE (Zhu et al., 2018) *struc2gauss* (Pei et al., 2020), where each node is mapped to a Gaussian distribution and the variance can capture the uncertainties. Intuitively, nodes with higher uncertainties are more likely to be anomalous.

A brief comparison of some representative anomaly detection methods is presented in Table 12. Among these previous methods, Radar (Li et al., 2017) and ANOMALOUS (Peng et al., 2018) are most relevant to our proposed ResGCN. Both methods model residual information to identify anomalies. However, they can only capture the linear dependency because they use matrix factorization as the framework. Our method employs deep neural network which can effectively capture the non-linear residual information and is more applicable for real-world complex networks. Similarly to our method, Dominant (Ding et al., 2019a) makes use of GCN for network reconstruction. However it does not distinguish normal and anomalous nodes in message passing of GCN, unlike our ResGCN, which uses residual as attention to guide the message passing so that the negative influence of anomalies can be reduced. In comparison with previous approaches especially those that are able to detect both structural and attribute anomalies, the complexity

Table 13 Parameters used in the experiments

Parameter	Value
Learning rate	0.01
GCN layers	[64, 32]
FC layers (residual modeling)	[64, 64, 64]
FC layers (attention modeling)	[64, 64]
FC layers (attribute decoder)	[32, 64]

of ResGCN is not increased, i.e., the major part of the complexity is still $O(n^2)$, but its performance accuracy is.

Another related work is graph convolutional networks (GCNs). The original GCN (Kipf & Welling, 2016a) has been proposed to learn node representations by passing and aggregating messages between neighboring nodes. Different variants extending GCN have been proposed, e.g., by introducing attention (Velickovic et al., 2017), adding residual and jumping connections (Xu et al., 2018) and disentangling node representations (Ma et al., 2019).

6 Conclusions

In this paper, we proposed a novel graph convolutional network (GCN) with attention mechanism, ResGCN, to address the problem of anomaly detection on attributed networks. ResGCN can effectively address the limitations of previously proposed approaches. Its GCN component models the high-order node interactions with multiple layers of nonlinear transformations, thus capturing the sparsity and nonlinearity of networks. Its attention mechanism based on the explicit deep residual modeling can prevent anomalous nodes from propagating the abnormal information in the message passing process of GCN. Ranking the residual information is employed to detect anomalies. The experimental results demonstrate the effectiveness of ResGCN and its advantages over the previously proposed methods. In the future, we would like to investigate the extensions of ResGCN to dynamic and streaming networks.

Appendix: Implementation details

In the experiments, we used Adam (Kingma and Ba 2014) algorithm to optimize the loss function. The parameters settings we used are shown in Table 13.

For previously proposed approaches included in our experimental study, we used the reference implementations by the authors if their source code have been publicly available. We set all hyper-parameters to the values recommended in the papers, in which the approaches were introduced:

- **AMEN**, we use the implementation in (Perozzi and Akoglu 2016). https://github.com/phanein/amen.
- **Radar** (Li et al. 2017), the official code: http://www.ece.virginia.edu/~jl6qk/code/Radar.zip.

- **ANOMOULOUS** (Peng et al. 2018), the official code: http://www.ece.virginia.edu/~jl6qk/code/ANOMALOUS.zip.
- **MADAM**, we use the implementation in (Gutiérrez-Gómez et al. 2019). https://github.com/leoguti85/MADAN.
- **Dominant** (Ding et al. 2019a), the official code: https://github.com/kaize0409/GCN_AnomalyDetection.

For LOF, we use the implementation of LOF in scikit-learn[4].

Acknowledgements This research has been partly funded by the Dutch Research Council (NWO) and the China Scholarship Council (CSC).

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Y. Pei and T. Huang. The first draft of the manuscript was written by Y. Pei and T. Huang and all authors commented on previous versions of the manuscript. All authors reviewed the results and approved the final version of the manuscript.

Availability of data and material The data is publicly available and from previous studies (Li et al., 2017; Peng et al., 2018; Ding et al., 2019a; Gutiérrez-Gómez et al., 2019).

Code availability The source code is available at https://bitbucket.org/paulpei/resgcn.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest/competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Akoglu, L., Tong, H., & Koutra, D. (2015). Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery, 29*(3), 626–688.

Bandyopadhyay, S., Lokesh, N., & Murty, M. N. (2019). Outlier aware network embedding for attributed networks. *Proceedings of the AAAI Conference on Artificial Intelligence, 33*, 12–19.

Bandyopadhyay, S., Vivek, S. V., & Murty, M. (2020). Outlier resistant unsupervised deep architectures for attributed network embedding. (pp. 25–33)

Barbieri, N., Bonchi, F., & Manco, G. (2014). Who to follow and why: link prediction with explanations. (pp. 1266–1275)

Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2013). Network anomaly detection: methods, systems and tools. *IEEE Communications Surveys & Tutorials, 16*(1), 303–336.

Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). Lof: identifying density-based local outliers. (pp. 93–104)

Brochier, R., Guille, A., & Velcin, J. (2019). Link prediction with mutual attention for text-attributed networks. (pp. 283–284)

[4] https://scikit-learn.org/.

Cui, P., Wang, X., Pei, J., & Zhu, W. (2018). A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering, 31*(5), 833–852.

Dabkowski P, Gal Y (2017) Real time image saliency for black box classifiers. In: Advances in Neural Information Processing Systems, pp 6967–6976

Ding, K., Li, J., Bhanushali, R., & Liu, H. (2019a). Deep anomaly detection on attributed networks. (pp. 594–602)

Ding, K., Li, J., & Liu, H. (2019b). Interactive anomaly detection on attributed networks. (pp. 357–365)

Eberle W, Holder L (2007) Discovering structural anomalies in graph-based data. In: Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), IEEE, pp 393–398

Falih, I., Grozavu, N., Kanawati, R., & Bennani, Y. (2018). Community detection in attributed network. (pp. 1299–1306)

Gao, J., Liang, F., Fan, W., Wang, C., Sun, Y., & Han, J. (2010). On community outliers and their efficient detection in information networks. (pp. 813–822)

Gupta, M., Gao, J., Aggarwal, C. C., & Han, J. (2013). Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering, 26*(9), 2250–2267.

Gutiérrez-Gómez L, Bovet A, Delvenne JC (2019) Multi-scale anomaly detection on attributed networks. arXiv preprint arXiv:191204144

Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, pp 1024–1034

Huang, X., Li, J., & Hu, X. (2017a). Accelerated attributed network embedding. (pp. 633–641)

Huang, X., Li, J., & Hu, X. (2017b). Label informed attributed network embedding. (pp. 731–739)

Kannan, R., Woo, H., Aggarwal, C. C., & Park, H. (2017). Outlier detection for text data. (pp. 489–497)

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980

Kipf TN, Welling M (2016a) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907

Kipf TN, Welling M (2016b) Variational graph auto-encoders. arXiv preprint arXiv:161107308

Li J, Dani H, Hu X, Liu H (2017) Radar: Residual analysis for anomaly detection in attributed networks. In: IJCAI, pp 2152–2158

Li, J., Cheng, K., Wu, L., & Liu, H. (2018a). Streaming link prediction on dynamic attributed networks. (pp. 369–377)

Li Q, Han Z, Wu XM (2018b) Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 32

Li, Y., Sha, C., Huang, X., & Zhang, Y. (2018c). Community detection in attributed graphs: An embedding approach.

Li, Y., Huang, X., Li, J., Du, M., & Zou, N. (2019). Specae: Spectral autoencoder for anomaly detection in attributed networks. (pp. 2233–2236)

Liang, J., Jacobs, P., Sun, J., & Parthasarathy, S. (2018). Semi-supervised embedding in attributed networks with outliers. (pp. 153–161)

Ma, J., Cui, P., Kuang, K., Wang, X., & Zhu, W. (2019). Disentangled graph convolutional networks. (pp. 4212–4221)

McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology, 27*(1), 415–444.

Meng, Z., Liang, S., Bao, H., & Zhang, X. (2019). Co-embedding attributed networks. (pp. 393–401)

Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam filtering with naive bayes-which naive bayes? *CEAS, Mountain View, CA, 17*, 28–69.

Müller, E., Sánchez, P. I., Mülle, Y., & Böhm, K. (2013). Ranking outlier nodes in subspaces of attributed graphs. (pp. 216–222)

Noble, C. C., & Cook, D. J. (2003). Graph-based anomaly detection. (pp. 631–636)

Pei, Y., Chakraborty, N., & Sycara, K. (2015). Nonnegative matrix tri-factorization with graph regularization for community detection in social networks.

Pei, Y., Du, X., Zhang, J., Fletcher, G., & Pechenizkiy, M. (2020). struc2gauss: Structural role preserving network embedding via gaussian embedding.

Peng Z, Luo M, Li J, Liu H, Zheng Q (2018) Anomalous: A joint modeling approach for anomaly detection on attributed networks. In: IJCAI, pp 3513–3519

Perozzi, B., & Akoglu, L. (2016). Scalable anomaly ranking of attributed neighborhoods. (pp. 207–215)

Perozzi, B., Akoglu, L., Iglesias Sánchez, P., & Müller, E. (2014). Focused clustering and outlier detection in large attributed graphs. (pp. 1346–1355)

Ruff, L., Zemlyanskiy, Y., Vandermeulen, R., Schnake, T., & Kloft, M. (2019). Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. (pp. 4061–4071)

Skillicorn, D. B. (2007). Detecting anomalies in graphs. (pp. 209–216)

Song, X., Wu, M., Jermaine, C., & Ranka, S. (2007). Conditional anomaly detection. *IEEE Transactions on knowledge and Data Engineering, 19*(5), 631–645.

Velickovic P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2017) Graph attention networks. arXiv preprint arXiv:171010903 1(2)

Xu, K., Li, C., Tian, Y., Sonobe, T., Ki, Kawarabayashi, & Jegelka, S. (2018). Representation learning on graphs with jumping knowledge networks. (pp. 5453–5462)

Zhou, D., He, J., Yang, H., & Fan, W. (2018). Sparc: Self-paced network representation for few-shot rare category characterization. (pp. 2807–2816)

Zhu, D., Cui, P., Wang, D., & Zhu, W. (2018). Deep variational network embedding in wasserstein space. (pp. 2827–2836)

Zhu, D., Zhang, Z., Cui, P., & Zhu, W. (2019). Robust graph convolutional networks against adversarial attacks. (pp. 1399–1407)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.