Higher-order Structure Based Anomaly Detection on Attributed Networks

1st Xu Yuan
School of Software Technology
Dalian University of Technology
Dalian, China
david@dlut.edu.cn

2nd Na Zhou
School of Software Technology
Dalian University of Technology
Dalian, China
zhouna824@mail.dlut.edu.cn

3rd Shuo Yu(✉)
School of Software Technology
Dalian University of Technology
Dalian, China
shuo.yu@ieee.org

4th Huafei Huang
School of Software Technology
Dalian University of Technology
Dalian, China
hfhuang@mail.dlut.edu.cn

5th Zhikui Chen
School of Software Technology
Dalian University of Technology
Dalian, China
zkchen@dlut.edu.cn

6th Feng Xia
School of Engineering, IT and Physical Sciences
Federation University Australia
Ballarat, Australia
f.xia@ieee.org

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671990

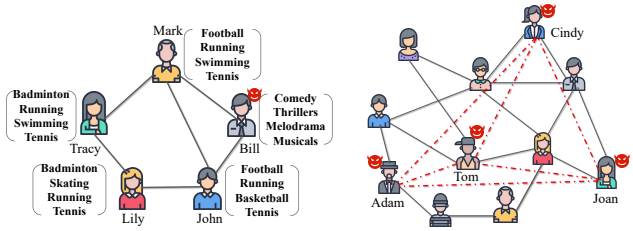
***Abstract*—Anomaly detection (such as telecom fraud detection and medical image detection) has attracted the increasing attention of people. The complex interaction between multiple entities widely exists in the network, which can reflect specific human behavior patterns. Such patterns can be modeled by higher-order network structures, thus benefiting anomaly detection on attributed networks. However, due to the lack of an effective mechanism in most existing graph learning methods, these complex interaction patterns fail to be applied in detecting anomalies, hindering the progress of anomaly detection to some extent. In order to address the aforementioned issue, we present a higher-order structure based anomaly detection (GUIDE) method. We exploit attribute autoencoder and structure autoencoder to reconstruct node attributes and higher-order structures, respectively. Moreover, we design a graph attention layer to evaluate the significance of neighbors to nodes through their higher-order structure differences. Finally, we leverage node attribute and higher-order structure reconstruction errors to find anomalies. Extensive experiments on five real-world datasets (i.e., ACM, Citation, Cora, DBLP, and Pubmed) are implemented to verify the effectiveness of GUIDE. Experimental results in terms of ROC-AUC, PR-AUC, and Recall@K show that GUIDE significantly outperforms the state-of-art methods.**

***Index Terms*—Anomaly Detection, Attributed Networks, Higher-order Structures, Autoencoder**

I. Introduction

With the increase of the amount of network data, detecting anomalies from network data has become a significant research problem of urgent societal concerns [1]. Moreover, anomaly detection has a wide range of applications in real life [2], such as financial fraud detection [3], [4], network intrusion detection [5], web spam detection [6], and industrial anomaly detection [7]. Therefore, the anomaly detection problem has attracted widespread attention from researchers [8].

Anomaly detection aims at finding the rare nodes whose behaviors are significantly different from other majority nodes. Furthermore, the abnormality of nodes on attributed networks depends on not only their abnormal situation of the network topology, but also the unusual condition of node attributes.



(a) Attribute anomaly (b) Structural anomaly

Fig. 1. The illustration about different types of anomalies on attributed networks.

Specifically, attribute abnormality mainly refers to the significant difference between the attributes of a node and its neighborhoods. For example, as shown in Fig 1 (a), Tracy, Mark, Bill, John, and Lily are in the same interest group, but John's hobbies are quite different from the other members. The abnormal structure mostly refers to small groups that are far away but too closely connected. For instance, in Fig 1 (b), Tom, Cindy, Joan, and Adam are members of a wire fraud organization. In order to facilitate the crime, they are closely related to each other. As it needs to simultaneously model the topological structure and node attributes, detecting anomalies of attributed networks is more challenging.

There have been many studies on detecting anomalies. Many studies attempt to find abnormal nodes through subspace selection of node feature [9], [10]. Some other methods consider exploiting residual analysis to detect anomalies [11], [12]. However, these methods are based on shallow learning mechanisms and have certain limitations. For example, they can not model the complex interaction of network attributes and structures. With the increasing development of deep learning technologies, the effectiveness of such kinds of methods has also been verified in addressing these problems [13]–[15]. Deep neural networks are employed to encode attributed networks, and

reconstruct the attributes and structures separately, which can utilize reconstruction errors to identify anomalies. However, previous studies still lack the ability of effectively utilizing complex interaction patterns among multiple entities to detect anomalies. When detecting anomalies, the significance of these complex interactions should be taken into consideration.

To address the above-mentioned problems, we present an unsupervised dual autoencoders framework, titled **GUIDE** (hi**G**her-order str**U**cture based anomaly detection on attr**I**bute**D** n**E**tworks). Different from previous methods, we use the higher-order structures to model complex interaction patterns between multiple entities for anomaly detection in the network. To better learn higher-order network structures, we propose a graph node attention layer, which can learn different weights according to structural differences between the node and its neighbors. Specifically, we first encode attributes and higher-order structures of nodes to obtain the corresponding latent representation, and then exploit the decoder to reconstruct it. Finally, the reconstruction errors from both higher-order structure and attribute perspectives are used to detect anomalies of attributed networks. We summarize the main contributions of this paper as follows:

- **Multiple Attributes-driven Anomaly Detection**: We propose a higher-order structure based anomaly detection method named GUIDE, which employs including node attributes and higher-order network structures to promote anomaly detection.
- **Higher-order Structure Attention Mechanism**: We design a higher-order structure attention mechanism, which utilizes structural differences between the node and its neighbors to generate attention weights. With this mechanism, our proposed GUIDE can better learn higher-order network structures.
- **Outperformance on Five Real-world Datasets**: Extensive experiments have been conducted on five real-world datasets, whose results show that GUIDE consistently outperforms all baseline methods significantly.

The rest of this paper is organized as follows. Section II generalizes the related work. Section III formally introduces the network motif and problem statement. In Section IV, we introduce the design of the proposed GUIDE. Section V empirically evaluates GUIDE on five real-world datasets. Section VI concludes the whole paper.

II. Related Work

Anomaly Detection on Attributed Networks. Compared with the plain (unlabeled) network, the attributed networks can model complex systems more effectively due to containing richer attribute information. Therefore, lots of researchers began to show interest in the problem of anomaly detection on attributed networks [16]–[18]. For example, Perozzi et al. [19] leveraged attributes and network structure to quantify the quality of neighborhoods so that find anomalous neighborhoods on attributed networks. Li et al. [11] found anomalous nodes by analyzing the residuals of attribute information and its coherence with network information. Moreover, Liu et al. [20] introduced a novel anomaly detection model, learning simultaneously node attributes and structural information to effectively detect local anomalies on attributed networks. Peng et al. [12] exploited CUR decomposition and residual analysis to filter out node attributes that are noisy and irrelevant, thereby avoiding their adverse effects for anomaly detection. Gutiérrez-Gómez et al. [21] explored all relevant contexts of anomalous nodes, and performed multiscalar anomaly detection on attributed networks. However, the above methods are limited by the shallow learning mechanism, so they cannot effectively learn the complex interactions between the node attribute and structure.

Driven by the great success of deep learning, a mass of studies have been devoted to exploiting deep neural networks to detect anomalous nodes on attributed networks. For instance, Ding et al. [15] constructed a deep autoencoder using graph convolutional neural networks, and evaluated the abnormality of nodes through the reconstruction errors of node attributes and structure. Li et al. [22] utilized Laplacian sharpening to magnify the distance between representations of anomalous nodes and normal nodes, making it easier to find anomalies. Ding et al. [23] presented an adversarial graph differential network, utilizing generative adversarial ideas to detect anomalies on the attribute network, which can be naturally extended to newly observed data. Furthermore, Chen et al. [24] came up with a generative adversarial attributed networks anomaly detection model. By obtaining the sample reconstruction error generated by the generator and the discriminant loss of real node pairs, this algorithm can predict effectively abnormal nodes. Despite the above approaches achieving superior performance over other shallow methods, they cannot effectively utilize complex interaction patterns between multiple entities to detect anomalies.

Higher-order Network Representation Learning. The complex real network contains a wealth of higher-order structures (i.e., motifs), which reflect the internal relationships of nodes in the network [25]. Multiple studies have confirmed that it is effective to consider higher-order structures in network representation learning [26]–[28]. The framework for learning higher-order network embedding was proposed by Rossi et al. [29], aiming at utilizing various motif-based matrix formulations to learn effectively network embedding. Lee et al. [30] exploited a motif-based attention mechanism to learn higher-order interactions between nodes with their neighbours. In [31], Yu et al. chose proper motifs to strengthen the multivariate relationships, and improve the learning effect of higher-order graph representation. In addition, Xu et al. [25] aggregated the higher-order structure features and attribute features of nodes to obtain the final network embedding, and demonstrate superior performance in the node classification task. Liu et al. [32] simultaneously modeled the local higher-order structures and temporal evolution to learn node representation for dynamic attributed networks. Nevertheless, all the aforementioned methods focus on network representation learning, it is still not clear how to effectively utilize higher-order structures for anomaly detection.

(a) M31 (b) M32 (c) M41 (d) M42 (e) M43

Fig. 2. Network motifs used in this paper.

III. Preliminaries

A. Network Motif

Network motifs [33] refer to special subgraph structures frequently appearing in the network. Paranjape et al. [34] studied the timing network and found that network motifs help understand the crucial structure of the network. In addtion, network motifs [35] have specific practical significance. For instance, a three-order triangle motif can describe the collaboration relationship of three scholars in the academic network. Therefore, we can utilize the network motif to effectively model complex interaction patterns between multiple entities in the network.

Because network motifs comprised of five or more nodes are so complex and numerous that it is difficult to deal with them. In this paper, we adopt network motifs comprised of three or four nodes to analyze the network. In Fig 2, we list the motif types used in this paper.

We employ the node motif degree proposed by Yu et al. [31] to represent the higher-order structures of nodes in this paper. Specifically, the node motif degree is defined as follows.

Definition 1: **Node Motif Degree (NMD):** For the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a node $i \in V$, the node motif degree of i is $NMD(i)$, which represents node i is involved in the number of the motif M.

As shown in Fig 3, the node e contains the number of $\boldsymbol{M31}$ motifs is 3.

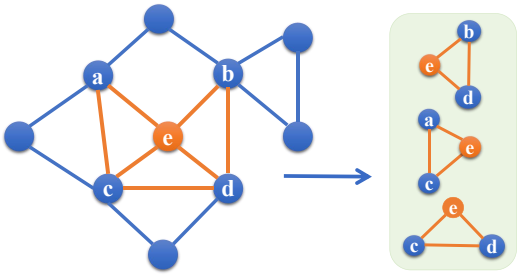


Fig. 3. A node motif degree calculation of M31 (For node e, it contains three M31 motifs, i.e., ebd, eac, ecd).

B. Problem Definition

In this paper, we use bold lowercase letters (e.g., $\mathbf{x}$) and bold uppercase letters (e.g., $\mathbf{X}$), to denote vectors and matrices, respectively. Besides, we use calligraphic fonts (e.g., $\mathcal{V}$) to represent sets. The i^{th} row of a matrix $\mathbf{X}$ is denoted by $\mathbf{x}_i$ and $(i,j)^{th}$ element of matrix $\mathbf{X}$ is denoted by $\mathbf{X}_{i,j}$. The notations mainly used in this paper are summarized in Table I.

TABLE I
Notations and Explanations Related to GUIDE.

Notation	Explanation
$\mathcal{G} = (\mathbf{A}, \mathbf{X})$	An attributed network
$\mathbf{A} \in \mathbb{R}^{n \times n}$	The adjacency matrix of $\mathcal{G}$
$\mathbf{X} \in \mathbb{R}^{n \times d}$	The attribute matrix of $\mathcal{G}$
$\mathbf{S} \in \mathbb{R}^{n \times m}$	The higher-order structures matrix of $\mathcal{G}$
$\mathbf{x}_i \in \mathbb{R}^d$	The attribute vector of the i^{th} node in $\mathcal{G}$
$\mathbf{s}_i \in \mathbb{R}^m$	The higher-order structure vector of the i^{th} node in $\mathcal{G}$
n	The number of nodes in $\mathcal{G}$
d	The dimension of attributes in $\mathcal{G}$
m	The dimension of higher-order structures in $\mathcal{G}$
$\lvert\cdot\rvert$	The number of elements of a set
$\sigma(\cdot)$	The non-linear activation function
$\mathbf{X}^T$	The transpose of a matrix $\mathbf{X}$
$\lVert\cdot\rVert_F$	The Frobenius norm of a matrix or vector
$\lVert\cdot\rVert_2$	The ℓ_2-norm of a matrix or vector

Definition 2: **Attributed Networks:** Give an attributed network $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the adjacency matrix, $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the attribute matrix. The i^{th} row vector $\mathbf{x}_i \in \mathbb{R}^d$ of the attribute matrix $\mathbf{X}$ represents the i^{th} node's attribute vector. Besides, $\mathbf{A}_{i,j} = 1$ if there is an edge between node i and node j, otherwise $\mathbf{A}_{i,j} = 0$.

Definition 3: **Structure Matrix:** The higher-order structures of $\mathcal{G}$ can be represented by a structure matrix $\mathbf{S}$. The i^{th} row vector $\mathbf{s}_i \in \mathbb{R}^m$ of the structure matrix $\mathbf{S}$ represents the i th node's structure vector, which is composed of the node motif degrees of $\boldsymbol{M31}$, $\boldsymbol{M32}$, $\boldsymbol{M41}$, $\boldsymbol{M42}$, $\boldsymbol{M43}$, and the original degree of the node.

Problem 1: **Anomaly Detection**. Given an attributed network $\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$, the task is to rank all the nodes according to their anomalous scores ($score(v_i)$), and the node that are significantly different from the majority nodes ($\geq 0.9n$) should obtain higher score and be ranked higher than other nodes.

Next, we will describe in detail the GUIDE model which models node attributes and higher-order structures jointly to detect anomalies of the network.

IV. The Proposed Model - GUIDE

In this section, we introduce the proposed model GUIDE in detail. The framework of our approach is illustrated in Fig. 4. We design two essential components for GUIDE: attribute autoencoder and structure autoencoder, which are respectively responsible for reconstructing node attributes and higher-order structures. Then, we use reconstruction errors of node attributes and higher-order structures to calculate anomaly scores of nodes and rank them. Finally, anomalies in the network can be found by the ranking list.

A. Attribute Autoencoder

In this part, we aim at designing an effective autoencoder to reconstruct node attributes, thereby catching attribute anoma-

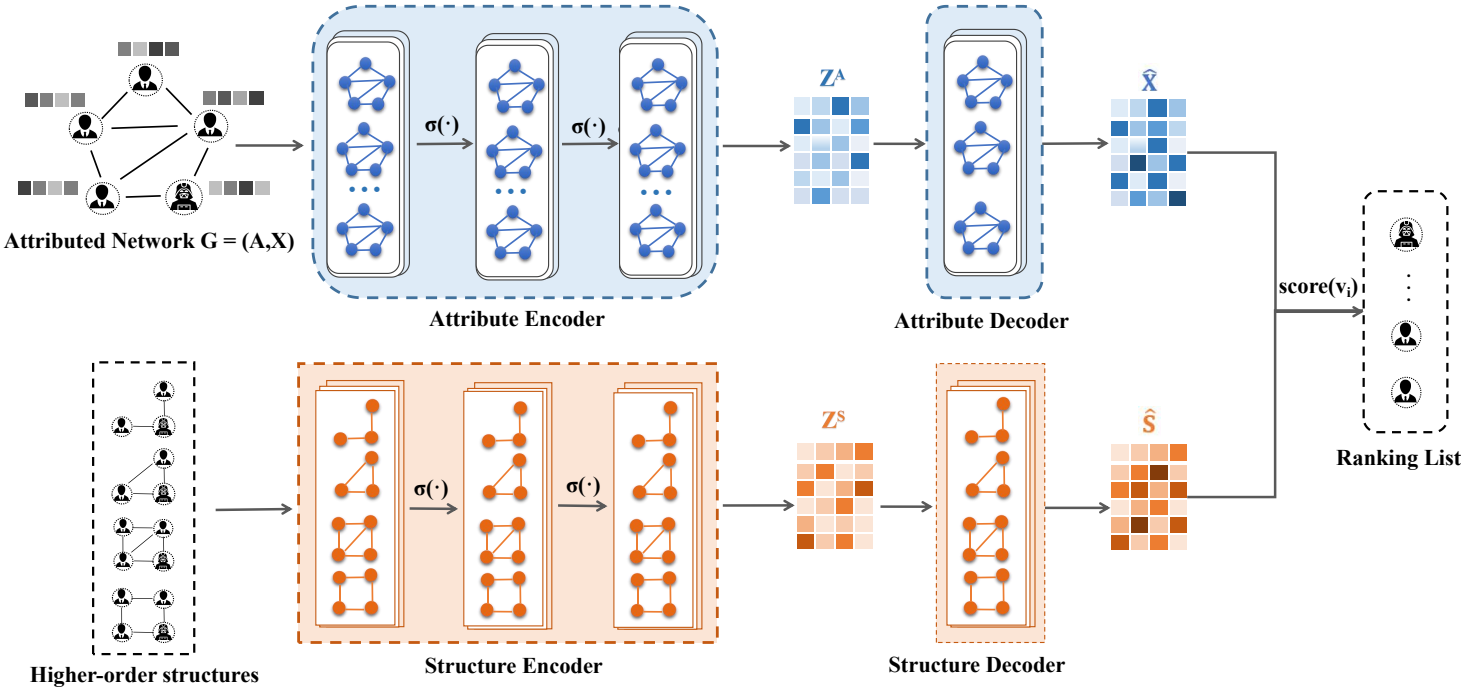


Fig. 4. The framework of the proposed GUIDE.

lies. Formally, an autoencoder network layer is described as:

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

where $\mathbf{H}^{(l)}$ is the latent representation of the input in layer l, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ denotes the adjacency matrix of the attributed network $\mathcal{G}$ with added self-connections and $\tilde{\mathbf{D}}_{i,i} = \sum_j \tilde{\mathbf{A}}_{i,j}$. $\sigma(\cdot)$ represents a non-linear activation function. We adopt $\mathbf{Relu}$ as activation function in this paper. $\mathbf{W}^{(l)}$ is the trainable weight matrix. We let $\bar{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$. Then the autoencoder network layer formula can be abbreviated as:

$$\mathbf{H}^{(l+1)} = f_{Relu}(\bar{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}). \tag{2}$$

In this paper, we encode the attributed networks using three autoencoder network layers, and the attribute matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ is regarded as the original input features:

$$\mathbf{H}^{(0)} = \mathbf{X}, \tag{3}$$

Therefore, the attribute encoder can be expressed as:

$$\mathbf{H}^{(1)} = f_{Relu}(\bar{\mathbf{A}}\mathbf{X}\mathbf{W}^{(0)}), \tag{4}$$

$$\mathbf{H}^{(2)} = f_{Relu}(\bar{\mathbf{A}}\mathbf{H}^{(1)}\mathbf{W}^{(1)}), \tag{5}$$

$$\mathbf{Z}^A = \mathbf{H}^{(3)} = f_{Relu}(\bar{\mathbf{A}}\mathbf{H}^{(2)}\mathbf{W}^{(2)}). \tag{6}$$

After using three autoencoder network layers, the encoder compresses the node attributes and network topology to get a low-dimensional latent representation $\mathbf{Z}^A$ of attributed networks. To reconstruct the node attributes, we exploit an autoencoder network layer to approximate the original attributes of nodes, which can be expressed as follows:

$$\hat{\mathbf{X}} = f_{Relu}(\bar{\mathbf{A}}\mathbf{Z}^A\mathbf{W}^{(3)}). \tag{7}$$

Here, $\hat{\mathbf{X}}$ represents the reconstructive attribute matrix. Therefore, the attribute reconstruction loss can be calculated:

$$\mathbf{R}_A = \mathbf{X} - \hat{\mathbf{X}}. \tag{8}$$

B. Structure Autoencoder

Considering the importance of higher-order structures to the anomaly detection on attributed networks, in this part, we use structure autoencoder to reconstruct the higher-order structures of nodes. And the calculated structure reconstruction loss can capture structural anomalies. Specifically, nodes with abnormal structures are usually too closely connected to some nodes on attributed networks. Their higher-order structures are very different from normal nodes, and cannot be reconstructed well.

Therefore, inspired by [23], we designed a graph node attention network (GNA) to encode the higher-order structures of the node. It can better learn the structural difference between the node and its neighbors by utilizing the higher-order structures attention mechanism and help identify structural anomalies. Specifically, a graph node attention layer can

Algorithm 1: The training process of GUIDE

Input : Attributed network $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, Training epochs $Epoch_{AE}$

Output: Well-trained GCN-AE and GNA-AE

$i \leftarrow 0$;

while $i < Epoch_{AE}$ **do**

  Compute the reconstructed node attributes $\hat{X}$ via Eq. (7);

  Compute the reconstructed higher-order network structures $\hat{S}$ via Eq. (11);

  Update GCN-AE and GNA-AE with the loss function Eq. (13);

end

learn the representation of node i in layer l , which can be formulated as follows:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{W}_1\mathbf{h}_i^{(l)} + \sum_{j \in N(i)\cup\{i\}} \alpha_{ij}\mathbf{W}_2\mathbf{h}_j^{(l)}\right), \quad (9)$$

where, $\mathbf{h}_i^{(l)}$, $\mathbf{h}_j^{(l)} \in \mathbb{R}^F$ is the input representation of node i and node j, respectively. $N(i)$ represents the neighborhood of node i. $\mathbf{h}_i^{(l+1)} \in \mathbb{R}^{F'}$ is the output representation of node i. $\mathbf{W}_1$,$\mathbf{W}_2 \in \mathbb{R}^{F' \times F}$ are two trainable weight matrices, and $\sigma(\cdot)$ denotes the **Relu** activation function. To determine the importance of different nodes, we calculate the normalized attention coefficient α_{ij} by:

$$\alpha_{ij} = \frac{\exp(\mathbf{a}^T\mathbf{W}_2(\mathbf{h}_i{}^{(l)} - \mathbf{h}_j{}^{(l)}))}{\sum_{k \in N(i)\cup\{i\}} \exp(\mathbf{a}^T\mathbf{W}_2(\mathbf{h}_i{}^{(l)} - \mathbf{h}_k{}^{(l)}))}. \quad (10)$$

where, $\mathbf{h}_i{}^{(l)} - \mathbf{h}_j{}^{(l)}$ represents the higher-order structures difference between node i and node j. $\mathbf{a} \in \mathbb{R}^{F'}$ is a parametrized weight vector. We generate the attentional weights based on the higher-order structures differences to facilitate the characterization of structural abnormality of node.

Similar to the attribute encoder, we use the three graph node attention layers to encode the higher-order structures of the node to obtain the corresponding latent representation $\mathbf{Z}^S$. In order to reconstruct structure matrix $\mathbf{S}$, we use another graph node attention layer to approximate the original higher-order structures of the node, which is expressed as follows:

$$\hat{\mathbf{S}} = \mathbf{graph\_node\_atten}(\mathbf{Z}^S), \quad (11)$$

Here, $\mathbf{graph\_node\_atten}(\cdot)$ represents the operation of the graph node attention network described earlier. $\hat{\mathbf{S}}$ represents the reconstructive structure matrix. Therefore, the structure reconstruction loss can be calculated:

$$\mathbf{R}_S = \mathbf{S} - \hat{\mathbf{S}}. \quad (12)$$

We can detect anomalous nodes of the network from the perspective of the higher-order structure.

TABLE II
DETAILS OF THE FIVE REAL-WORLD DATASETS WITH INJECTED ANOMALIES.

Network Name	ACM	Citation	Cora	DBLP	Pubmed
#nodes	9,360	8,935	2,708	5,484	19,717
#edges	15,556	15,098	5,278	8,117	44,338
#attributes	6,775	6,775	1,433	6,775	500
#anomalies	5%	5%	5%	5%	5%
#M31	3,898	3,716	1,630	1,788	12,550
#M32	66,214	97,839	47,411	41,536	661,782
#M41	678	723	220	414	3,291
#M42	6,575	8,174	2,468	3,680	53,407
#M43	7,002	5,825	1,536	3,177	100,440

C. Loss Function and Anomaly Detection

To jointly learn the reconstruction errors, we aim to minimize the loss function of both network higher-order structures and node attribute:

$$\begin{aligned}\mathcal{L} &= (1-\alpha)\mathbf{R}_S + \alpha\mathbf{R}_A \\ &= (1-\alpha)||\mathbf{S} - \hat{\mathbf{S}}||_F^2 + \alpha||\mathbf{X} - \hat{\mathbf{X}}||_F^2.\end{aligned} \quad (13)$$

where α is a balance parameter which controls the training weight of higher-order structures reconstruction errors and attribute reconstruction errors. We can utilize the reconstruction error to assess the anomaly degree of nodes. Specifically, the attributes or higher-order structures of a node cannot be reconstructed well, indicating that its behavior pattern deviates from the majority of other nodes, and the probability of it being an anomalous node is higher. Thus, we can use the reconstruction error from both higher-order structures and node attribute perspective to calculate the anomaly score of each node:

$$score(\mathbf{v}_i) = (1-\alpha)\,||\mathbf{s}_i - \hat{\mathbf{s}}_i||_F^2 + \alpha||\mathbf{x}_i - \hat{\mathbf{x}}_i||_F^2. \quad (14)$$

Note that the node with a higher score has a higher probability of being an abnormal node. So we can rank all nodes by their anomaly scores. And the detailed model training process is shown in Algorithm 1.

V. EXPERIMENTS

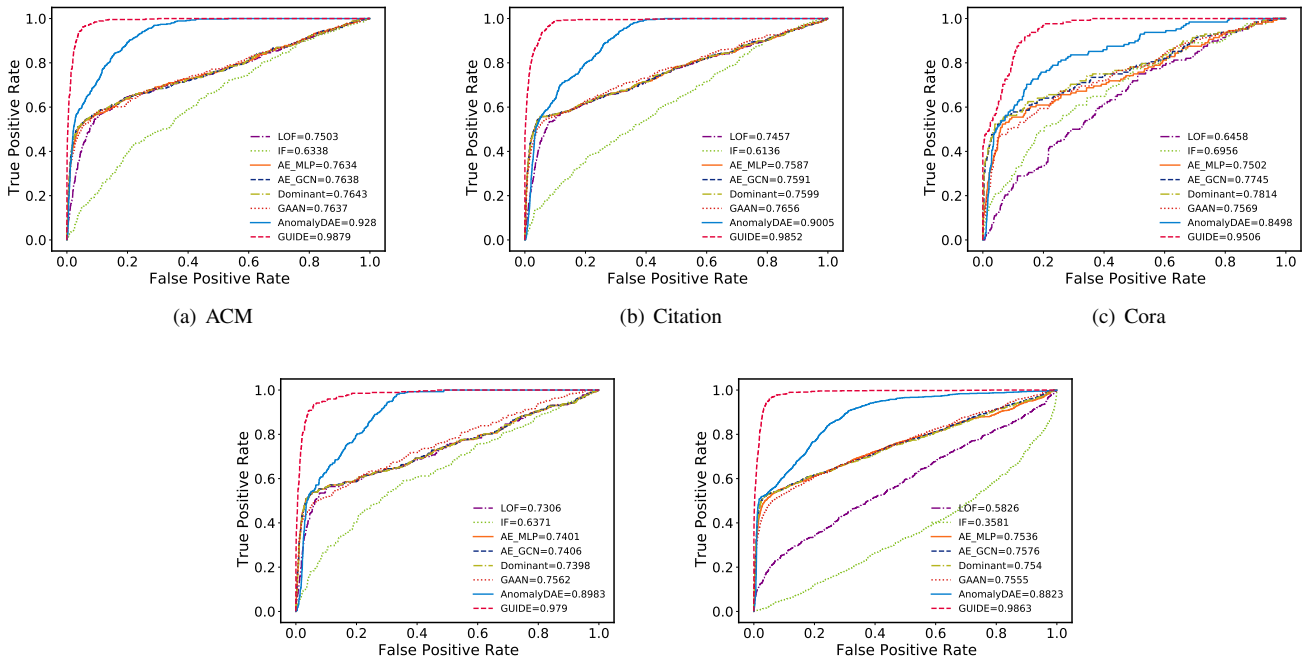
In this section, we perform extensive experiments on five real-world datasets to confirm the effectiveness of the GUIDE model.

A. Datasets

To comprehensively evaluate the GUIDE model, we choose five real-world datasets that have been used in many previous studies [15], [36], [37] in our experiments:

- **ACM**[1]: ACM is a citation network dataset extracted from the Association for Computer Machinery, which is composed of 16,484 scientific publications. And each edge denotes the citation relationship of papers in the network. The attributes of paper consist of sparse bag-of-words features extracted from the paper title.

[1]https://github.com/shenxiaocam/CDNE/blob/master/CDNE_codes/code/CDNE/data



(a) ACM (b) Citation (c) Cora

(d) DBLP (e) Pubmed

Fig. 5. ROC curves and AUC scores of all methods on five real-world datasets.

- **Citation**[1]: Citation is a citation network dataset composed of 8,935 nodes and 15,098 edges, where nodes denote scientific publications, edges represent citation links between publications. The attributes for nodes are the sparse bag-of-words features extracted from the article title.
- **Cora**[2]: Cora is a citation network dataset including 2,708 nodes, with 5,429 edges to indicate the citation relation. Each node is a scientific publication which is indicated by a binary feature vector.
- **DBLP**[1]: DBLP is a citation network dataset composed of 5,484 scientific publications collected from the DBLP Computer Science Bibliography. While the 8,117 edges are the citation relations among different papers. The node attributes are extracted from the article title.
- **Pubmed**[2]: Pubmed contains 19,717 scientific publications with 44,338 links indicating the citation relations between publications. The bag-of-words representations of documents are regarded as the node attributes.

Due to the absence of anomalies in the datasets, we refer to two widely used methods [16] to inject structural anomalies and attribute anomalies for each dataset, respectively. On one hand, we generate some small cliques to perturb the topological structure of the network. The intuition behind this method is that the nodes in a small clique are much more connected to each other than average, which are always an anomalous structure in many scenarios [38]. Therefore, we randomly choose p nodes from the attributed network as a small clique, and make them fully connected. Then all nodes in the small clique are considered as structural anomalies. We perform this process a total of q times and finally generate q small cliques. So there are total $p \times q$ structural anomalies. In experiments, the size of a small clique p is set to 15. The number of small cliques q is fine-tuned according to different datasets.

On the other hand, to inject attribute anomalies, which have the same number of structural anomalies, we first randomly pick another $p \times q$ nodes. Then we randomly choose another k nodes from the attributed network for each attribute perturbation node i and calculate the Euclidean distance between node i and all the k nodes. Finally, the node j among the k nodes whose Euclidean distance with node i is the largest exchanges the attributes with node i. We list the details of these five real-world datasets in Table II.

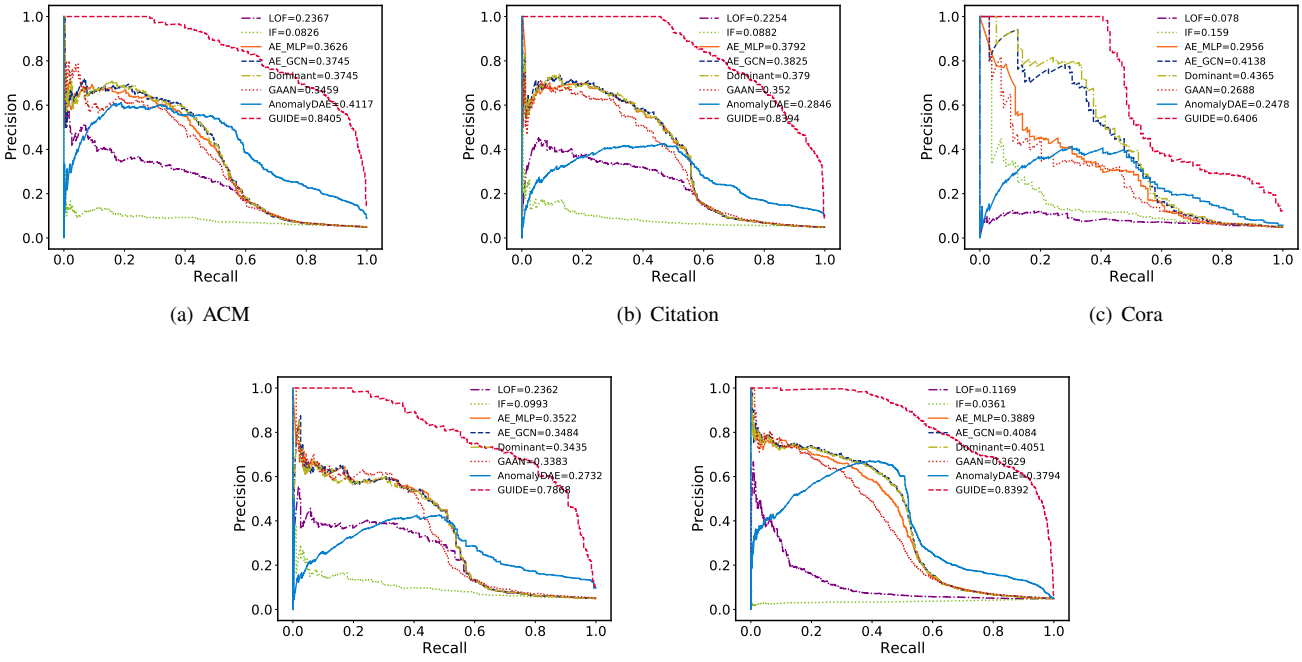
B. Experimental Settings

In this section, we describe the compared anomaly detection methods and common evaluation metrics in detail.

Baseline Methods. The GUIDE model is compared with the following baseline methods:

- **LOF (Local Outlier Factor)** [39] detects anomalies by comparing the local reachability density of the node with their neighbors.
- **IF (Isolation Forest)** [40] is an attribute based detection method, which detects anomalies utilizing their susceptibility to isolation.

[2]https://github.com/kimiyoung/planetoid/tree/master/data



(a) ACM (b) Citation (c) Cora

(d) DBLP (e) Pubmed

Fig. 6. PR curves and AUC scores of all methods on five real-world datasets.

TABLE III
EXPERIMENTAL RESULTS OF DIFFERENT ANOMALY DETECTION METHODS W.R.T. RECALL@K.

Recall@K															
	ACM			Citation			Cora			DBLP			Pubmed		
K	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150
LOF	0.054	0.094	0.132	0.048	0.092	0.134	0.031	0.078	0.125	0.073	0.139	0.205	0.023	0.043	0.063
IF	0.015	0.028	0.037	0.025	0.030	0.046	0.125	0.171	0.211	0.040	0.054	0.088	0.002	0.002	0.003
AE_MLP	0.074	0.148	0.213	0.078	0.159	0.235	0.180	0.305	0.383	0.109	0.208	0.322	0.038	0.076	0.116
AE_GCN	0.074	0.145	0.224	0.081	0.166	0.237	0.297	0.398	0.477	0.114	0.209	0.322	0.039	0.076	0.113
Dominant	0.070	0.152	0.222	0.083	0.157	0.237	0.313	0.414	0.484	0.110	0.212	0.315	0.039	0.077	0.112
GAAN	0.071	0.126	0.201	0.078	0.154	0.221	0.164	0.266	0.391	0.117	0.230	0.326	0.040	0.077	0.113
AnomalyDAE	0.044	0.118	0.194	0.016	0.060	0.106	0.109	0.313	0.453	0.029	0.095	0.176	0.017	0.038	0.066
GUIDE	**0.109**	**0.217**	**0.318**	**0.115**	**0.230**	**0.346**	**0.391**	**0.484**	**0.539**	**0.183**	**0.341**	**0.458**	**0.051**	**0.102**	**0.153**

- **AE_MLP (Autoencoder_Multilayer Perceptron Perceptron)** is a classic neural network model that can reconstruct node attributes to detect anomalies.
- **AE_GCN (Autoencoder_Graph Convolutional Network)** [41] is a deep learning model that can detect anomalies by reconstructing node attributes.
- **Dominant (Deep Anomaly Detection on Attributed Networks)** [15] detects anomalies from both the attribute and structural perspectives by calculating the reconstruction loss of attributes and structures, respectively.
- **GAAN (Generative Adversarial Attributed Network Anomaly Detection)** [24] is a generative adversarial anomaly detection model, which train jointly reconstruction loss and discriminator loss to detects anomalies.
- **AnomalyDAE (Deep Joint Representation Learning Framework for Anomaly Detection through A Dual Autoencoder)** [42] is a dual autoencoder learning model for anomaly detection. It can learn effectively the complex interactions between node attribute and network structure.

Evaluation Metrics. To evaluate the performances of each algorithm, three widely used metrics are adopted to compare different methods in this paper:

- **ROC-AUC:** The ROC curve can measure the relationship between true positive rate (TP) and false positive rate (FP) in different thresholds. AUC value is the area under the ROC curve, which approaches 1 indicating the better performance of the model.
- **PR-AUC:** The PR curve is the curve of precision against recall at different thresholds. AUC value is the area under the PR curve. The higher AUC value means the higher precision and recall.

- **Recall@K:** We utilize Recall@K to calculate the proportion of true anomalous nodes that an anomaly detection method found in all the ground truth anomalies:

$$Recall@K = \frac{|true\ anomalies\ in\ queried\ anomalies|}{|all\ true\ anomalies|}.$$

Parameter Settings. In the experiment, the Adam [43] algorithm is adopted to optimize the loss function on different datasets. We train the GUIDE model for 200 epochs and the learning rate is set to 0.001. We have also optimized the hyperparameters of the model on five real-world datasets through a parameter sensitivity experiment. For other baseline methods, we keep the settings described in the original papers.

C. Experimental Results

In our experiments, the performance of the GUIDE model is evaluated on multiple metrics by comparing it with the above baseline methods. We first show the experimental results w.r.t. ROC-AUC in Fig 5. While the experimental results in terms of PR-AUC are shown in Fig 6. Then the results w.r.t. Recall@K in Table III. According to these experimental results, we obtain the observations as follows:

- Our proposed GUIDE model significantly outperforms other baseline methods. The experimental results of the GUIDE model in terms of ROC-AUC, PR-AUC, and Recall@K have a great improvement compared to all baselines. It demonstrates the effectiveness of GUIDE for anomaly detection on attributed networks.
- The deep learning models (AE_MLP, AE_GCN, Dominant, GAAN, AnomalyDAE, and GUIDE) outperform the conventional anomaly detection methods (LOF and IF). For example, for ROC-AUC on Pubmed dataset, Dominant outperforms LOF by 17.14% and IF by 39.59% respectively, AnomalyDAE outperforms LOF by 29.97% and IF by 52.42% respectively, and GUIDE outperforms LOF by 40.37% and IF by 62.82% respectively. These methods break through the limitations of the shallow mechanism and can effectively address the key challenges of anomaly detection on attributed networks.
- AE_MLP, AE_GCN, and Dominant have similar performances in terms of ROC-AUC. Compared to Dominant, AE_MLP and, AE_GCN only considered reconstructing node attributes but ignored the reconstruction of the first-order structure. It demonstrates that node attributes are more important than the first-order structure of the network for anomaly detection on attributed networks. Therefore, it will not lead to too poor performance when ignoring the first-order structure of the network. This conclusion was also confirmed in [15].
- Although AnomalyDAE has shown its superior performance in terms of ROC-AUC, it cannot achieve satisfying results in terms of PR-AUC. For instance, the experimental results of the AnomalyDAE model in terms of ROC-AUC are only lower than those of our proposed model. But its experimental results in terms of PR-AUC are lower than those of most baseline methods. It confirms that AnomalyDAE is not suitable for scenarios with data imbalance. Compared with it, our GUIDE model has reached optimal performance in terms of both ROC-AUC and PR-AUC. Specifically, our ROC-AUC and PR-AUC scores increases by 5.99% and 42.88% on ACM, 8.47% and 55.48% on Citation, 10.08% and 39.28% on Cora, and 8.07% and 51.36% on DBLP, and 10.40% and 45.98% on Pubmed compared to AnomalyDAE. It shows that the outperformance of GUIDE in scenarios with data imbalance.
- Compared with other baseline methods, GUIDE can discover more true anomalous nodes within the ranking list of limited length according to the results of recall@K. Especially, compared with Dominant, our recall@K on Cora increases by 7.8% in top 50 ranked nodes, 7.0% in top 100 ranked nodes, and 5.5% in top 150 ranked nodes. Besides, compared with AnomalyDAE, our recall@K on ACM increases by 6.5% in top 50 ranked nodes, 9.9% in top 100 ranked nodes, and 12.4% in top 150 ranked nodes. It demonstrates the superiority of GUIDE for anomaly detection within the ranking list of limited length.

TABLE IV
IMPACT OF DIFFERENT STRUCTURAL AUTOENCODER W.R.T. AUC VALUES.

Model	ACM	Citation	Cora	DBLP	Pubmed
GUIDE_GCNEN	0.9784	0.9797	0.9414	0.9531	0.9762
GUIDE_GCNDE	0.9727	0.9665	0.9263	0.9534	0.9591
GUIDE_GCN	0.9763	0.9596	0.9368	0.9577	0.9501
GUIDE	**0.9879**	**0.9852**	**0.9506**	**0.9790**	**0.9863**

D. Ablation Experiment

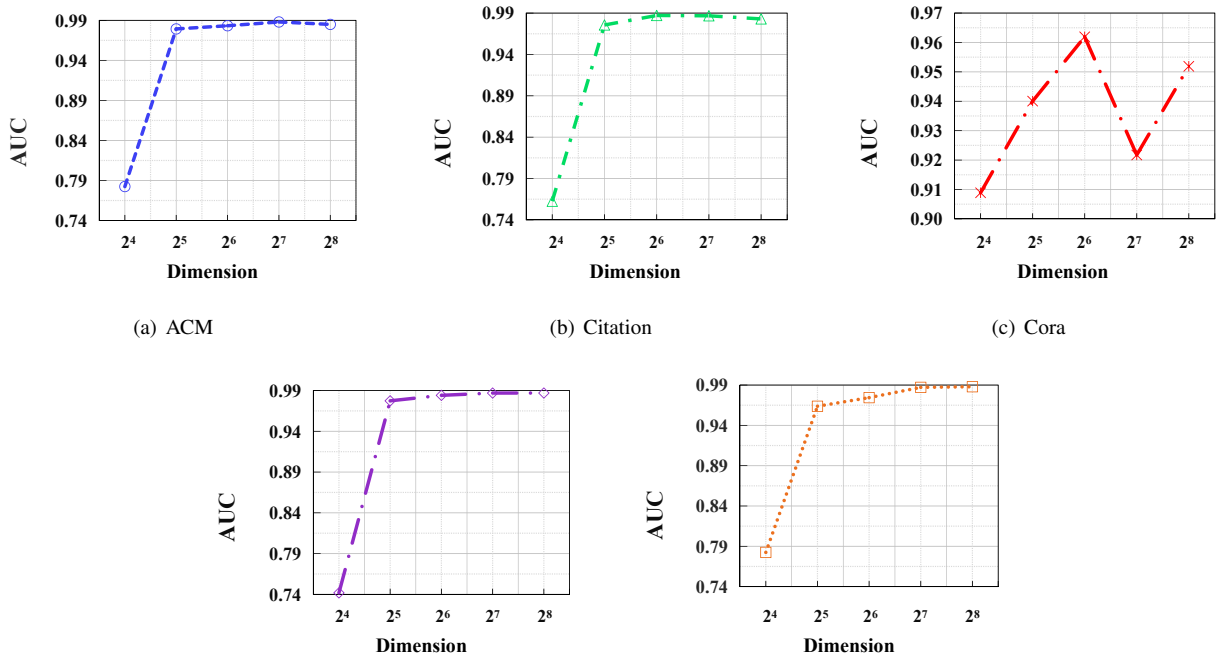
To confirm the effectiveness of considering higher-order structures in our framework, we replace the graph node attention network in the structural autoencoder with a graph convolutional network. The specific operations are as follows:

- **GUIDE_GCNEN:** The structure encoder is replaced with GCN, and the structure decoder is constant.
- **GUIDE_GCNDE:** The structure encoder is constant, and the structure decoder is replaced with GCN.
- **GUIDE_GCN:** Both structure encoder and structure decoder are replaced with GCN.

We perform evaluations for these methods on five real-world datasets, respectively. The experimental results are shown in Table IV. We found that although the GUIDE model achieved the best performance on all five datasets, the performance do not decrease significantly after replacing the structure autoencoder. It proves that our framework is effective and the higher-order structures play a key role in anomaly detection on attributed networks.

E. Parameter Analysis

In this section, we study the parameter sensitivity of different numbers of the embedding dimension and the balance parameter α for anomaly detection. The experiment results are



(a) ACM (b) Citation (c) Cora

(d) DBLP (e) Pubmed

Fig. 7. Impact of different embedding dimension size w.r.t. AUC values.

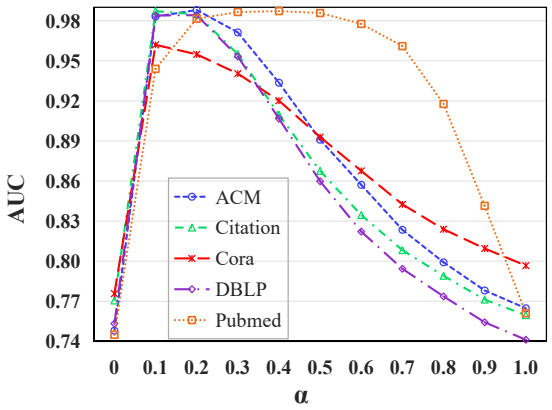


Fig. 8. Impact of different α w.r.t. AUC values.

presented in Fig. 7 and Fig. 8, respectively. We can observe that too low dimension would degrade the performance of the model in Fig. 7, which is caused by underfitting. Moreover, in Fig. 8, we can see that if GUIDE only consider the higher-order structures reconstruction errors ($\alpha = 0$) or the attribute reconstruction errors ($\alpha = 1$), which will lead to poor performance. It demonstrates that the anomaly detection on attributed networks should pay attention to the attributes and the higher-order structures of the node simultaneously. Meanwhile, we find that GUIDE can achieve the best performance when α is around 0.1 to 0.3 on five datasets. It adequately verified the importance of higher-order structures for anomaly detection on attributed networks.

VI. Conclusion

In this paper, we design a higher-order structure based unsupervised learning framework GUIDE for anomaly detection. Specifically, we employ higher-order network structures when detecting anomalies, and address the limitations of existing methods in modeling complex interaction patterns between multiple entities in the real world. We use dual autoencoders to detect anomalies from both the attribute and higher-order structure perspectives, respectively. To further improve the ability to learn the higher-order structures, we introduce a graph node attention layer, which utilizes the higher-order structure attention mechanism to effectively capture the structural difference between the node and its neighbors. Finally, We can calculate the anomaly score of nodes by the node attribute and higher-order structure reconstruction loss, and sort to find the anomaly. The experiment results on five real-world datasets show that GUIDE outperforms all baseline methods in terms of ROC-AUC, PR-AUC, and Recall@K.

Different higher-order structures generally correspond to various interaction patterns in the real world. For example, the motif M32 can represent the citation relationship between three papers in citation networks. Motif M41 can represent the four-person collaboration relationship in collaboration networks. The significance of different motifs in a certain network should be further evaluated and thus promoting anomaly detection.

ACKNOWLEDGMENT

This work is partially supported by National Natural Science Foundation of China under Grant No. 62102060. And the authors would like to thank Chen Cao from Dalian University of Technology for many helpful comments with this manuscript.

REFERENCES

[1] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, "Graph learning: A survey," *IEEE Transactions on Artificial Intelligence*, 2021.

[2] S. Yu, J. Liu, H. Wei, F. Xia, and H. Tong, "How to optimize an academic team when the outlier member is leaving?," *IEEE Intelligent Systems*, vol. 36, no. 3, pp. 23–30, 2020.

[3] J. Vanhoeyveld, D. Martens, and B. Peeters, "Value-added tax fraud detection with scalable anomaly detection techniques," *Applied Soft Computing*, vol. 86, p. 105895, 2020.

[4] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo, "Fraud detection: A systematic literature review of graph-based anomaly detection approaches," *Decision Support Systems*, vol. 133, p. 113303, 2020.

[5] E. Viegas, A. Santin, V. Abreu, and L. S. Oliveira, "Enabling anomaly-based intrusion detection through model generalization," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, pp. 934–939, 2018.

[6] A. Makkar and N. Kumar, "An efficient deep learning-based scheme for web spam detection in iot environment," *Future Generation Computer Systems*, vol. 108, pp. 467–487, 2020.

[7] D. Ramotsoela, A. Abu-Mahfouz, and G. Hancke, "A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study," *Sensors*, vol. 18, no. 8, p. 2491, 2018.

[8] S. Yu, F. Xia, Y. Sun, T. Tang, X. Yan, and I. Lee, "Detecting outlier patterns with query-based artificially generated searching conditions," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 134–147, 2020.

[9] P. I. Sánchez, E. Müller, F. Laforet, F. Keller, and K. Böhm, "Statistical selection of congruent subspaces for mining attributed graphs," in *2013 IEEE 13th International Conference on Data Mining*, pp. 647–656, 2013.

[10] P. I. Sánchez, E. Müller, O. Irmler, and K. Böhm, "Local context selection for outlier ranking in graphs with multiple numeric node attributes," in *Proceedings of the 26th International Conference on Scientific and Statistical Database Management*, no. 16, pp. 1–12, 2014.

[11] J. Li, H. Dani, X. Hu, and H. Liu, "Radar: Residual analysis for anomaly detection in attributed networks," in *IJCAI*, pp. 2152–2158, 2017.

[12] Z. Peng, M. Luo, J. Li, H. Liu, and Q. Zheng, "Anomalous: A joint modeling approach for anomaly detection on attributed networks," in *IJCAI*, pp. 3513–3519, 2018.

[13] D. Zhu, Y. Ma, and Y. Liu, "Anomaly detection with deep graph autoencoders on attributed networks," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, 2020.

[14] S. Bandyopadhyay, S. V. Vivek, and M. Murty, "Outlier resistant unsupervised deep architectures for attributed network embedding," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 25–33, 2020.

[15] K. Ding, J. Li, R. Bhanushali, and H. Liu, "Deep anomaly detection on attributed networks," in *Proceedings of the 2019 SIAM International Conference on Data Mining*, pp. 594–602, 2019.

[16] K. Ding, J. Li, and H. Liu, "Interactive anomaly detection on attributed networks," in *Proceedings of the twelfth ACM international conference on web search and data mining*, pp. 357–365, 2019.

[17] L. Zhang, J. Yuan, Z. Liu, Y. Pei, and L. Wang, "A robust embedding method for anomaly detection on attributed networks," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2019.

[18] L. Xue, M. Luo, Z. Peng, J. Li, Y. Chen, and J. Liu, "Anomaly detection in time-evolving attributed networks," in *International Conference on Database Systems for Advanced Applications*, pp. 235–239, 2019.

[19] B. Perozzi and L. Akoglu, "Scalable anomaly ranking of attributed neighborhoods," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, pp. 207–215, 2016.

[20] N. Liu, X. Huang, and X. Hu, "Accelerated local anomaly detection via resolving attributed networks.," in *IJCAI*, pp. 2337–2343, 2017.

[21] L. Gutiérrez-Gómez, A. Bovet, and J.-C. Delvenne, "Multi-scale anomaly detection on attributed networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 678–685, 2020.

[22] Y. Li, X. Huang, J. Li, M. Du, and N. Zou, "Specae: Spectral autoencoder for anomaly detection in attributed networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2233–2236, 2019.

[23] K. Ding, J. Li, N. Agarwal, and H. Liu, "Inductive anomaly detection on attributed networks," in *29th International Joint Conference on Artificial Intelligence, IJCAI 2020*, vol. 2021-January, pp. 1288–1294, 2020.

[24] Z. Chen, B. Liu, M. Wang, P. Dai, J. Lv, and L. Bo, "Generative adversarial attributed network anomaly detection," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1989–1992, 2020.

[25] J. Xu, S. Yu, K. Sun, J. Ren, I. Lee, S. Pan, and F. Xia, "Multivariate relations aggregation learning in social networks," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, pp. 77–86, 2020.

[26] R. A. Rossi, R. Zhou, and N. K. Ahmed, "Deep inductive network representation learning," in *Companion Proceedings of the The Web Conference 2018*, pp. 953–960, 2018.

[27] J. Piao, G. Zhang, F. Xu, Z. Chen, and Y. Li, "Predicting customer value with social relationships via motif-based graph attention networks," in *Proceedings of the Web Conference 2021*, pp. 3146–3157, 2021.

[28] C. Yang, M. Liu, V. W. Zheng, and J. Han, "Node, motif and subgraph: Leveraging network functional blocks through structural convolution," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 47–52, 2018.

[29] R. A. Rossi, N. K. Ahmed, and E. Koh, "Higher-order network representation learning," in *Companion Proceedings of the The Web Conference 2018*, pp. 3–4, 2018.

[30] J. B. Lee, R. A. Rossi, X. Kong, S. Kim, E. Koh, and A. Rao in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 499–508, 2019.

[31] S. Yu, F. Xia, J. Xu, Z. Chen, and I. Lee, "Offer: A motif dimensional framework for network representation learning," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 3349–3352, 2020.

[32] Z. Liu, C. Huang, Y. Yu, and J. Dong, "Motif-preserving dynamic attributed network embedding," in *Proceedings of the Web Conference 2021*, pp. 1629–1638, 2021.

[33] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[34] A. Paranjape, A. R. Benson, and J. Leskovec, "Motifs in temporal networks," in *Proceedings of the tenth ACM international conference on web search and data mining*, pp. 601–610, 2017.

[35] S. Yu, Y. Feng, D. Zhang, H. D. Bedru, B. Xu, and F. Xia, "Motif discovery in networks: A survey," *Computer Science Review*, vol. 37, p. 100267, 2020.

[36] D. Zhu, Y. Ma, and Y. Liu, "Deepad: A joint embedding approach for anomaly detection on attributed networks," in *International Conference on Computational Science*, vol. 12138, pp. 294–307, 2020.

[37] S. Bandyopadhyay, N. Lokesh, and M. N. Murty, "Outlier aware network embedding for attributed networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 12–19, 2019.

[38] D. B. Skillicorn, "Detecting anomalies in graphs," in *2007 IEEE Intelligence and Security Informatics*, pp. 209–216, 2007.

[39] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

[40] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*, pp. 413–422, 2008.

[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR*, 2016.

[42] H. Fan, F. Zhang, and Z. Li, "Anomalydae: Dual autoencoder for anomaly detection on attributed networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5685–5689, 2020.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR*, 2015.