# ComGA: Community-Aware Attributed Graph Anomaly Detection

Xuexiong Luo
Jia Wu
Amin Beheshti
Jian Yang
School of Computing, Macquarie University
xuexiong.luo@students.mq.edu.au
{jia.wu,amin.beheshti,jian.yang}@mq.edu.au

Xiankun Zhang
Yuan Wang
College of Artificial Intelligence, Tianjin University of Science and Technology
{zhxkun,wangyuan23}@tust.edu.cn

Shan Xue
School of Computing, Macquarie University
emma.xue@mq.edu.au

## ABSTRACT

Graph anomaly detection, here, aims to find rare patterns that are significantly different from other nodes. Attributed graphs containing complex structure and attribute information are ubiquitous in our life scenarios such as bank account transaction graph and paper citation graph. Anomalous nodes on attributed graphs show great difference from others in the perspectives of structure and attributes, and give rise to various types of graph anomalies. In this paper, we investigate three types of graph anomalies: local, global, and structure anomalies. And, graph neural networks (GNNs) based anomaly detection methods attract considerable research interests due to the power of modeling attributed graphs. However, the convolution operation of GNNs aggregates neighbors information to represent nodes, which makes node representations more similar and cannot effectively distinguish between normal and anomalous nodes, thus result in sub-optimal results. To improve the performance of anomaly detection, we propose a novel community-aware attributed graph anomaly detection framework (ComGA). We design a tailored deep graph convolutional network (tGCN) to anomaly detection on attributed graphs. Extensive experiments on eight real-life graph datasets demonstrate the effectiveness of ComGA[1].

## CCS CONCEPTS

• **Security and privacy → Intrusion/anomaly detection and malware mitigation**; • **Human-centered computing → Social network analysis**.

## KEYWORDS

Anomaly Detection, Community Structure, Attributed Graphs, Graph Neural Networks

**ACM Reference Format:**
Xuexiong Luo, Jia Wu, Amin Beheshti, Jian Yang, Xiankun Zhang, Yuan Wang, and Shan Xue. 2022. ComGA: Community-Aware Attributed Graph Anomaly Detection. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498389

[1]Code is available at https://github.com/DASE4/ComGA

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*WSDM '22, February 21–25, 2022, Tempe, AZ, USA*
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498389

## 1 INTRODUCTION

Graph anomaly detection aims to find rare behaviours that significantly deviate from the majority of nodes [25, 37], and has attracted growing attention in recent years due to the broad impact in various domains, including abuse monitoring in healthcare systems [22], financial fraud monitoring [3, 40, 41], and graph intrusion detection in cyber security [32]. In real-world life, various information data can be modeled attributed graphs [1, 12, 17] with structure relation and attribute information. For example, in Twitter, users are connected by various social relationships, and they carry rich profile information such as age, gender, and income. Anomalous nodes

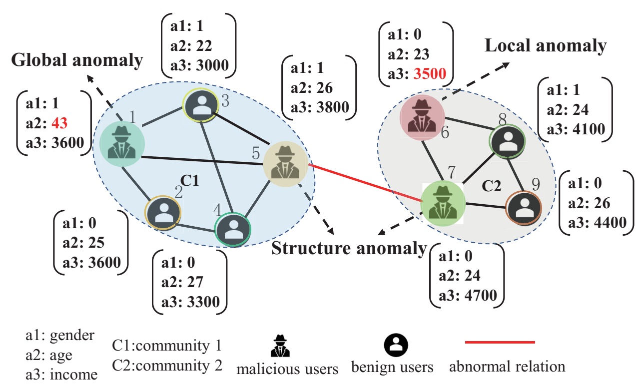



**Figure 1: The toy example is a Twitter graph with different types of graph anomalies, where the dense links of the graph form two different communities C1 and C2. When considering attribute information in the whole graph, user 1 is global anomaly since its attribute (a2) value is significantly higher than others. When considering attribute information within one community (e.g., C2), user 6 is local anomaly as its attribute (a3) value relatively deviates from other users within C2. When considering structure information across different communities, users 5 and 7 are structure anomaly because they have link with other communities while other users in their community do not have cross-community links.**

are commonly different from normal nodes in the perspectives of structure and attributes such as incorrect links and inconsistent attribute values with neighbors on attributed graphs. Furthermore, due to the interaction of graph structure and attribute information, this will cause the diversity of graph anomalies. As shown in Figure 1, graph anomalies can be divided into three categories: local, global, and structure anomalies. Besides, these graph anomalies are common in other life applications, and bring huge adverse impact for society. Thus, how to design an effective model to detect these anomalies is very important.

Graph neural networks (GNNs) [5, 10, 15, 24, 35, 36], as a powerful technique of capturing the interaction of structure and attributes on attributed graphs [21, 33], have been employed to anomaly detection application and achieve great success [8, 11, 19, 23]. However, how to analyze three kinds of graph anomalies above and improve the performance of anomaly detection remains a challenging task. First, the convolution operation of GNNs learns node representations by aggregating node's immediate neighbors information, which only captures local anomaly instead of global anomaly. Of course, if we roughly utilize more convolution layers to capture longer distance of neighbors for global anomaly, this will suffer from the over-smoothing of node representations. In other words, node representations between neighbors will become more and more similar to each other, which makes anomalous nodes undistinguishable from neighbors, thus result in sub-optimal results. Second, as analyzed above, GNNs only focus on neighborhood of structure information, and cannot discover community structure of graphs so methods based on GNNs will ignore the structure anomaly existing in graphs. How to effectively min community structure and capture local, global, and structure anomalies simultaneously in an unified framework is very challenging. Finally, these graph anomalies have strong relations with graph structure and attribute information. How to design a reasonable evaluation method to detect anomalous nodes with three graph anomalies from the perspectives of structure and attributes is worth considering.

To work out the challenges above, in this paper, we propose a Community-aware attributed Graph Anomaly detection framework (ComGA). In detail, we design a tailored deep graph convolutional network (tGCN) which propagates community-specific representation into its corresponding layers of GCN via multiple gateways. And, the hidden node representations of each layer in the tGCN module can fuse local and structure anomalies information. This can also respect community structure of the graph and alleviate over-smoothing of node representations when we build deeper tGCN to consider global anomaly. By doing so, the proposed method makes anomalous node representations more distinguishable and anomaly-aware node representations easier to be learned for multiple anomalies. To effectively capture community structure of graphs, inspired by [39] that extracts community structure information of the graph by deep neural networks, we introduce autoencoder to encode and decode modularity matrix of the graph to learn community-specific representation. Finally, to evaluate anomalous nodes from the perspectives of structure and attributes, we reconstruct graph structure and attribute information with corresponding decoders from anomaly-aware node representations, and the joint reconstruction errors of nodes are employed to detect anomalous nodes.

In summary, our main contributions are three-fold:

- We analyze various types of graph anomalies on attributed graphs including local, global, and structure anomalies, which is benefit to spot existing anomalous nodes.
- We propose a tGCN model to improve the performance of anomaly detection on attributed graphs by explicitly capturing community structure of the graph and learning more distinguishable node representations for multiple anomalies.
- We conduct rich experiments on eight real-life graphs to verify its superior performance.

The framework of the paper consists of the following parts. Section 2 summarizes graph anomaly detection work and the difference of ComGA with others. We formally give related definitions and research problem in Section 3. In Section 4, we present the components of ComGA framework. Experimental evaluations on various attributed graph datasets are shown in Section 5. We conclude the whole paper and make a short representation for future research work in Section 6.

## 2 RELATED WORK

Recently, the research field of graph anomaly detection has attracted increasing attention. The previous methods [4, 38] only consider anomalous nodes from the structure or attributes perspective. Besides, some methods focus on constructing shallow models to spot anomalous nodes. CODA proposes an unified probabilistic model [9] to spot community anomaly of nodes. AMEN [29] constructs the ego-graph of each node to spot anomalous neighbors on attributed graphs. Some other shallow methods aim to find anomalies in attribute feature subspaces of nodes [26, 30, 31, 34]. In addition, some work utilize residual analysis to measure abnormal nodes [16, 42]. In particular, ANOMALOUS [42] fuses CUR decomposition and residual analysis to anomaly detection, and has the robustness for noisy features.

With the rocketing growth of GNNs [10, 15, 35], methods based on GNNs [7, 8, 11, 23] have been proposed for anomaly detection on attributed graphs. For example, DOMINANT [7] directly employs GCN [15] which represents nodes by aggregating neighbors' attribute information to learn node representations, then the reconstruction errors of node representations are leveraged for spotting anomalous nodes. To build self-supervised anomaly detection framework, HCM [11] models both local and global contextual anomalies by using hop counts prediction as a self-supervised task to train model. CoLA [23] designs a contrastive self-supervised learning framework to detect local anomaly within the graph. However, the real-world anomaly node labels are less on attributed graphs, so these methods might be not extremely effective.

Another related works [6, 19, 28] are to solve the problem that anomalous nodes are not distinguishable with other normal nodes due to the smoothing convolution operation of GNNs. For example, SpecAE [19] leverages Laplacian sharpening to alleviate the over-smoothing of node representations to detect global and community anomalies. ResGCN [28] utilizes the residual information based attention mechanism to learn anomalous node representations and prevent over-smoothing.

We summarize the difference between the existing GNNs based anomaly detection methods and our proposed model ComGA in

**Table 1: The summary of existing GNNs based anomaly detection methods and ComGA**

| Methods | LA | GA | SA | CS | DNR |
|---|---|---|---|---|---|
| DOMINANT [7] | ✓ | ✗ | ✗ | ✗ | ✗ |
| AnomalyDAE [8] | ✓ | ✗ | ✗ | ✗ | ✗ |
| HCM [11] | ✓ | ✓ | ✗ | ✗ | ✗ |
| CoLA [23] | ✓ | ✗ | ✓ | ✗ | ✓ |
| ResGCN [28] | ✓ | ✗ | ✗ | ✗ | ✓ |
| SpecAE [19] | ✓ | ✓ | ✗ | ✗ | ✓ |
| ComGA | ✓ | ✓ | ✓ | ✓ | ✓ |

[1] LA, GA, SA, CS, and DNR indicate local anomaly, global anomaly, structure anomaly, community structure, and distinguishable node representations, respectively.

Table 1. From the statistic results, our approach explicitly captures community structure of the graph and learns more distinguishable node representations for local, global, and structure anomalies.

## 3 DEFINITION AND PROBLEM STATEMENT

In this section, we firstly introduce two important definitions about attributed graph and modularity matrix, and then describe the research problem of this paper.

***DEFINITION 1.*** (Attributed Graph) Given an attributed graph as $\mathcal{G} = (V, E, X)$, with $V$ denotes the set of nodes $V = \{v_1, v_2, \cdots, v_n\}$, where $|V| = n$; $E$ denotes the set of edges, where $|E| = m$; the node attributes $\mathbf{X} \in \mathbb{R}^{n\times d}$, where the $i-th$ row vector $\mathbf{x}_i \in \mathbb{R}^d (i = 1, \cdots, n)$ is the attribute information for the node $v_i$. In addition, we define topology structure of the graph as the adjacency matrix $\boldsymbol{A}$, where $\boldsymbol{A}_{ij} = 1$ represents that there is a link between nodes $v_i$ and $v_j$. Otherwise, $\boldsymbol{A}_{ij} = 0$.

***DEFINITION 2.*** (Modularity Matrix) Modularity matrix $\mathbf{B}$ of the graph mainly describes the division of nodes in different communities [27, 39], and $\mathbf{B} = [b_{ij}] \in \mathbb{R}^{n\times n}$ can be written as:

$$b_{ij} = a_{ij} - \frac{k_i k_j}{2m}. \tag{1}$$

where $a_{ij}$ is the probability that nodes $v_i$ and $v_j$ generate edges belonging to the same community, $\frac{k_i k_j}{2m}$ is the expected number of edges between nodes $v_i$ and $v_j$ if edges are placed randomly, $k_i$ is the degree of node $v_i$ and $m = \frac{1}{2}\sum_i k_i$ represents the total number of edges.

Given an attributed graph $\mathcal{G}$, our model aims to detect nodes that significantly differ from other nodes from the perspectives of structure and attributes in the graph.

## 4 THE PROPOSED APPROACH

This section detailedly shows that how the proposed framework ComGA detects multiple anomalies from the perspectives of structure and attributes on attributed graphs, and it is illustrated in Figure 2.

### 4.1 Community Detection Module

Methods based on GNNs firstly build deep graph model to anomaly detection field but they only consider anomaly information of local neighbors by the aggregation operation of GNNs, and underestimate the importance of capturing community structure to anomalous nodes detection. Due to the nonlinear properties of real-world graphs, traditional reconstruction based community detection methods [13, 20, 27] that construct linear mapping process are less effective. Thus, we utilize deep neural network to capture community structure of graphs by avoiding of the above problem. We firstly construct the modularity matrix $\mathbf{B}$ of the graph to represent the division of nodes in different communities. To learn the community-specific representation, we employ a deep autoencoder to process $\mathbf{B}$. The autoencoder consists of two key components: encoder and decoder. We assume that there are $L$ layers and $l$ represents the $l-th$ layer in the autoencoder. In terms of the component of the encoder, we define the $\mathbf{H}_l$ as the representation learned by the $l-th$ layer.

$$\mathbf{H}_l = \phi(\mathbf{W}_e^l \mathbf{H}_{l-1} + b_e^l). \tag{2}$$

where $\phi$ indicates the activation function such as Relu or Sigmoid function, $\mathbf{W}_e^l$ and $b_e^l$ represent the weight matrix and bias of the $l-th$ layer in the encoder, respectively. And, the input of encoder $\mathbf{H}_0$ is the modularity matrix $\mathbf{B}$.

In terms of the component of the decoder, we aim to map the latent representation $\mathbf{H}$ from the output of the encoder back into the original data space, *i.e.,* reconstruct the original data from the latent representation:

$$\mathbf{H}'_l = \phi(\mathbf{W}_d^l \mathbf{H}'_{l-1} + b_d^l). \tag{3}$$

where we denote the weight matrix and bias of the $l-th$ layer in the decoder are $\mathbf{W}_d^l$ and $b_d^l$, respectively. The reconstruction loss function of the autoencoder is shown as follows:

$$L_{res} = \left\|\mathbf{B} - \hat{\mathbf{B}}\right\|_F^2 = \sum_{i=1}^{n} \left\|\mathbf{b}_i - \hat{\mathbf{b}}_i\right\|_2^2. \tag{4}$$

Where the $\hat{\mathbf{B}} = \mathbf{H}_L$ is the output of the decoder part, the $i-th$ column $\mathbf{b}_i$ of $\mathbf{B}$ represents node $v_i$.

### 4.2 tGCN Module

To effectively fuse community structure information to GCN model for structure anomaly, and learn more distinguishable anomalous node representations for local, global, and structure anomalies, we design a deep tGCN module. Firstly, we utilize GCN model to encode attributed graphs. GCN model[15] extends the convolution operation on graphs data to integrate structure and attribute information for graph embedding representation. Thus, the node representations of the $l-th$ layer in GCN model, $\mathbf{Z}_l$, can be learned by the following graph convolutional operation way:

$$\mathbf{Z}_l = \phi(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{Z}_{l-1}\mathbf{W}_{l-1}). \tag{5}$$

Where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$, $\mathbf{I}$ is the identity diagonal matrix of the adjacent matrix $\mathbf{A}$ for given graph, $\mathbf{W}$ is the weight matrix of graph convolutional network. In particular, the feature map parameters $\mathbf{W}_{l-1}$ are shared for all nodes in this model. Although the GCN model can learn great node embedding representations by aggregating neighbors' feature and effectively capture the complex interactions of structure and attributes on attributed graphs, it cannot directly apply anomaly detection task. Because the representation is not distinguishable for multiple anomalies due to the

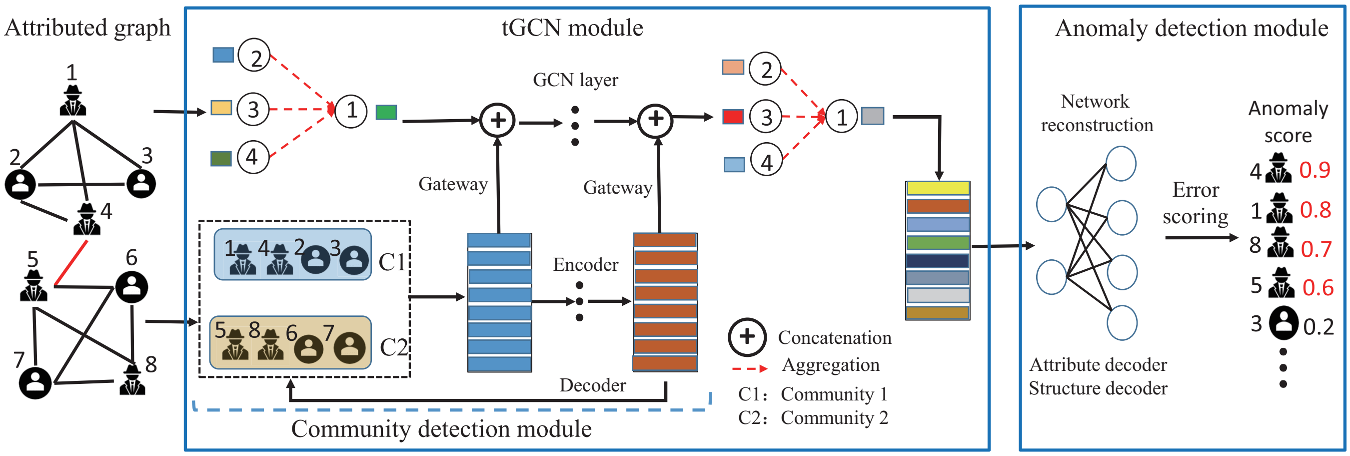



**Figure 2: The framework of ComGA consists of three major modules. For the community detection module, we utilize autoencoder to encode and decode modularity matrix of the graph to obtain community-specific representation of each node. For the tGCN module, we use the topology structure and nodal attributes as the input of GCN model, and aggregate neighbors information to capture local and global anomalies by GCN layer. Simultaneously, we introduce the gateway that propagates community-specific representation of each node in the autoencoder into the feature representations of its corresponding nodes in GCN model, which can fuse structure anomaly feature of each node and learn anomaly-aware node representations. For the anomaly detection module, we design structure decoder and attribute decoder to reconstruct the topology structure and nodal attributes from anomaly-aware node representations at the output of tGCN, respectively, and rank these anomalous nodes according to the corresponding anomaly score from the joint reconstruction errors.**

smoothing convolution operation, and ignore community structure of the graph for anomaly information detection. Thus, we design a gateway to combine the two representations $\mathbf{Z}_{l-1}$ and $\mathbf{H}_{l-1}$ generated by autoencoder together to get anomaly-aware node representations as follows:

$$\tilde{\mathbf{Z}}_{l-1} = \mathbf{Z}_{l-1} + \mathbf{H}_{l-1}. \tag{6}$$

Then we replace the input of $l-th$ layer in tGCN with $\tilde{\mathbf{Z}}_{l-1}$ to generate the new representation $\mathbf{Z}_l$:

$$\mathbf{Z}_l = \phi(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{Z}}_{l-1}\mathbf{W}_{l-1}). \tag{7}$$

As we can see in Eq.(7), it is the community-specific representation learned from each encoder layer that is fused into a corresponding GCN layer, and the community structure information will be propagated to corresponding nodes by the normalized adjacency matrix $\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$. It is worth noting that we denote the attribute matrix $\mathbf{X}$ as the input of the first layer in tGCN module. Specifically, anomaly-aware node representations are easier to learn for local and structure anomalies through the above information passing way. Furthermore, when building deeper tGCN layer to global anomaly, it also learns more distinguishable anomalous node representations without the over-smoothing problem. In order to enhance the community-specific representation of the final layer in tGCN module, $\mathbf{Z}$, we design a community-guide module to closely integrate the autoencoder model and GCN model as follows:

$$L_{gui} = KL(\mathbf{Z}||\mathbf{H}). \tag{8}$$

## 4.3 Anomaly Detection Module

To determine multiple anomalous nodes from the structure perspective, we utilize structure decoder to take the learned latent representation $\mathbf{Z}$ as input to decode them for reconstruction of original graph structure:

$$\hat{\mathbf{A}} = sigmoid(\mathbf{Z}\mathbf{Z}^T). \tag{9}$$

Specifically, we perform inner product between two node representations to predict whether there is a link between two nodes:

$$p(\hat{\mathbf{A}}_{ij} = 1|\mathbf{z}_i, \mathbf{z}_j) = sigmoid(\mathbf{z}_i, \mathbf{z}_j^T). \tag{10}$$

Thus, we employ the structure reconstruction error to detect anomalous nodes on the graph. For a certain node, if its topology structure can be well reconstructed through the structure decoder, it has the low probability of being an anomalous node. On the contrary, if its structure information cannot be well reconstructed, it indicates that its structure patterns differ from the majority normal nodes. That is, it implies that the node has a higher probability of being an anomaly from the perspective of graph structure.

Similarly, to determine multiple anomalous nodes from the attributes perspective, we utilize attribute decoder to take the learned latent representation $\mathbf{Z}$ as input to decode them for reconstruction of original nodal attributes. Specifically, we use the graph convolutional layer as the attribute decoder as follows:

$$\hat{\mathbf{X}} = \phi(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{Z}\mathbf{W}_{ad}). \tag{11}$$

To learn the structure and attribute reconstruction errors from the perspectives of structure and attributes on the attributed graph, the objective function can be formulated as:

$$L_{rec} = (1-\alpha)\left\|\mathbf{A}-\hat{\mathbf{A}}\right\|_F^2 + \alpha\left\|\mathbf{X}-\hat{\mathbf{X}}\right\|_F^2. \quad (12)$$

Where $\alpha$ is the parameter which controls the balance between structure reconstruction and attributes reconstruction.

By minimizing the above objective function, the joint reconstruction errors can be employed to measure the abnormality of nodes. The anomaly score of node $v_i$ is determined according to:

$$Score_{v_i} = (1-\alpha)\left\|\mathbf{a}_i-\hat{\mathbf{a}}_i\right\|_2^2 + \alpha\left\|\mathbf{x}_i-\hat{\mathbf{x}}_i\right\|_2^2. \quad (13)$$

## 4.4 Loss Function

In order to learn the proposed model, different components of ComGA are jointly trained and each component requires dedicated training objective functions. Thus, we combine the above mentioned loss functions in an unified framework, and the joint loss function as follows:

$$L = L_{res} + L_{gui} + L_{rec}. \quad (14)$$

## 4.5 Complexity Analysis

In this work, for the autoencoder, the time complexity is $O(nd^2d_1^2\cdots d_L^2)$, where $d$ is the dimension of the input data and the dimension of each layer of the autoencoder is $d_1,\ldots,d_L$ according to the size of weight matrix such as $\mathbf{W}_e^1 \in \mathbb{R}^{d\times d_1}$. $n$ is the number of the input data. As for the tGCN module, the time complexity is linear as $O(mdd_1\cdots d_L)$, and $m$ is number of edges because the aggregation operation of GCN can be implemented on the sparse matrix. In addition, both the time complexity of constructing modularity matrix and the time complexity of inner product in anomaly detection module are $O(n^2)$. Thus, the overall time complexity of ComGA is $O(nd^2d_1^2\cdots d_L^2 + mdd_1\cdots d_L + 2n^2)$.

# 5 EXPERIMENTS

In this section, we conduct different types of experiments on eight real-life graph datasets, to verify the effectiveness of ComGA framework.

## 5.1 Datasets

There are two types of benchmark datasets such as ground-truth and injected anomaly graphs in the previous anomaly detection methods, where the ground-truth graphs are rare [11, 28, 42], and the majority of methods [6–8, 19, 23] only analyze the performance on injected anomaly graphs. To justly demonstrate the effectiveness of ComGA, we adopt eight real-world graph datasets including both ground-truth and injected anomaly graphs. Furthermore, considering the shortage of ground-truth anomalies, the injected artificially anomaly label as a common method is used widely. Thus, we inject a combined set of anomalies (*i.e.,* structure anomalies and attribute anomalies) for each dataset according to the perturbation scheme described in [7, 23], and the complete construction process can be found at GitHub[1]. The statistics of these eight datasets are shown in Table 2, and the detailed information is described as follows:

- **Ground-truth anomaly graphs** [11, 28, 42]: Amazon and Enron are two attributed graphs with ground-truth anomaly labels. Amazon is a co-purchase graph, and the nodes with the tag amazonfail are defined as anomalous nodes. Enron is an email graph. Attributes represent text information length, number of recipients, etc, and edges indicate the email is sent by the sender to the receiver. The anomalous nodes are defined as spammers.
- **Injected anomaly graphs** [6–8, 19, 23]: BlogCatalog, Flickr, ACM, Cora, Citeseer, and Pubmed are six attributed graphs with injected anomaly labels. BlogCatalog and Flickr are two popular social media graphs from the blog sharing website and the image hosting and sharing website, respectively. In these two datasets, nodes denote the existing users of websites, and links represent the relationships between users. Besides, a list of tags associated with each user is used as the attributes. ACM, Cora, Citeseer, and Pubmed are four typical paper citation graphs, which are composed of literature publications. In these four graphs, nodes denote the papers while edges represent one paper cites other papers or the paper is cited by others. The attribute information of each node is the title and abstract of papers.

## 5.2 Baselines

We compare our ComGA framework with the following seven popular anomaly detection methods:

- **AMEN** [29] uses both attributes and graph structure information to detect anomalous neighbors from the ego-graph.
- **Radar** [16] detects anomalies by analyzing the residuals of attribute information and its coherence with graph information.
- **ANOMALOUS** [42] integrates CUR decomposition and residual analysis to detect anomalous nodes on attributed graphs.
- **DOMINANT** [7] develops a principled graph convolutional autoencoder to learn node representations for anomaly detection.
- **AnomalyDAE** [8] employs two separate autoencoders to interactively learn structure representation and attributes representation to analyze anomalous nodes. Here, in order to make a fair experimental comparison with other baselines, we all use the final reconstruction errors to measure abnormal nodes. Thus, we select the variants of AnomalyDAE as baseline, which does not have penalty term for loss function and reconstruction errors.
- **ResGCN** [28] uses the residual-based attention mechanism and prevents over-smoothing for anomalous node representations.
- **CoLA** [23] designs a contrastive learning framework based on GNNs to learn the representative information for anomalous nodes.

**Parameter Settings** In the ComGA experiments, we set the number of layers to 3 for encoder and the dimension of each layer in the encoder to $d-2000-500-128$, where $d$ is the dimension of the input data. The dimension of decoder is inverse. The dimension of the layers in GCN model is $d-2000-500-128-128$. We train our model with 60, 100, 150, 150, 60, 100, 100, and 80 iterations for Amazon, Enron, Cora, Citeseer, Pubmed, BlogCatalog, Flickr, and ACM, respectively. We utilize the Adam [14] algorithm as the optimizer of our model with learning rate as 0.001 and 0.0001 for Enron and

**Table 2: The statistics of the datasets**

| | ground-truth anomaly | | injected anomaly | | | | | |
|---|---|---|---|---|---|---|---|---|
| Datasets | Amazon | Enron | Cora | Citeseer | Pubmed | BlogCatalog | Flickr | ACM |
| nodes | 1,418 | 13,533 | 2,708 | 3,327 | 19,717 | 5,196 | 7,575 | 16,484 |
| edges | 3,695 | 176,897 | 5,429 | 4,732 | 44,338 | 171,743 | 239,738 | 71,980 |
| attributes | 28 | 20 | 1,433 | 3,703 | 500 | 8,189 | 12,074 | 8,337 |
| anomalies | 28 | 5 | 150 | 150 | 600 | 300 | 450 | 600 |

**Table 3: AUC values (%) on eight datasets**

| Methods | Amazon | Enron | Cora | Citeseer | Pubmed | BlogCatalog | Flickr | ACM |
|---|---|---|---|---|---|---|---|---|
| AMEN [29] | 47.0 | 47.0 | 62.66 | 61.54 | 77.13 | 63.92 | 65.73 | 56.26 |
| Radar [16] | 58.0 | 65.0 | 65.87 | 67.09 | 62.33 | 74.01 | 73.99 | 72.47 |
| ANOMALOUS [42] | 60.2 | 69.5 | 57.70 | 63.07 | 73.16 | 72.37 | 74.34 | 70.38 |
| DOMINANT [7] | 62.5 | 68.5 | 81.55 | 82.51 | 80.81 | 74.68 | 74.42 | 76.01 |
| AnomalyDAE [8] | 53.61 | 61.2 | 76.27 | 72.71 | 81.03 | 78.34 | 75.08 | 75.13 |
| ResGCN [28] | **71.0** | 66.0 | 84.79 | 76.47 | 80.79 | 78.5 | 78.0 | 76.8 |
| CoLA [23] | 47.26 | 34.97 | 87.79 | 89.68 | **95.12** | 78.54 | 75.13 | 82.37 |
| ComGA | 63.07 | **72.9** | **88.4** | **91.67** | 92.2 | **81.4** | **79.9** | **84.96** |

Amazon. The learning rate is 0.00001 for other six datasets. The parameter $\alpha$ is empirically set as 0.6, 0.2, 0.2, 0.1, 0.3, 0.4, 0.4, 0.2 for Amazon, Enron, Cora, Citeseer, Pubmed, BlogCatalog, Flickr, and ACM, respectively. For other baselines, we retain to the settings described in the corresponding papers to report optimized results .

## 5.3 Performance Evaluation

We evaluate anomaly detection performance by comparing ComGA with the seven baseline methods. AUC value is utilized as the evaluation metric, which indicates that an anomalous node chosen randomly has higher probability of ranking than a normal node. We run our method 10 times and report the average results to prevent extreme cases. Then, the results are reported in Table 3. From the evaluation results, we make the following observations:

- For graphs with ground-truth anomaly labels, ComGA achieves the best among the baselines on Enron. However, ComGA achieves comparable results on Amazon because the size of Amazon is small and graph structure information is simple. ComGA cannot effectively capture rich multiple anomalies information on Amazon. For graphs with injected anomaly labels, ComGA outperforms all the baselines on all the five datasets except for Pubmed. It verifies the effectiveness of learning more distinguishable anomalous node representations for multiple anomalies. Due to the large anomaly labels on Pubmed, CoLA, as a self-supervised anomaly detection framework, achieves the best results compared with ComGA.
- Compared to the shallow methods (AMEN, Radar, and ANOMALOUS), and GNNs based methods (DOMINANT and AnomalyDAE ), our method achieves better performance. The main reason is that ComGA successfully prevents over-smoothing of anomalous node representations and makes them more distinguishable.
- For the same method ResGCN about preventing over-smoothing, ComGA also achieves better performance. The reason is that our method designs a tailored community-aware anomaly detection framework, which utilizes community structure information of the graph to alleviate the problem of over-smoothing of node representations for different types of anomalies.
- Despite the advanced contrastive learning based semi-supervised learning framework CoLA, ComGA also achieves good results. The main reason is that ComGA considers more types of anomalies from local, global, and structure anomalies than local and structure anomalies.

## 5.4 Ablation Studies

To demonstrate the importance of capturing community structure of the graph for anomaly detection, we conduct ablation studies experiment comparing ComGA with two variants on six datasets. Specifically, we define the following variants:

- **ComGA-S**: This variant replaces the input of autoencoder with adjacency matrix, and it only considers the pure graph structure information.
- **ComGA-A**: This variant replaces the input of autoencoder with attribute information matrix, and it incorporates more nodal attribute information.

From Figure 3, we can see that the AUC values of ComGA are better than DOMINANT, ComGA-S, and ComGA-A on six datasets. This verifies that learning community-specific representation on graphs data is more effective to characterize anomalous nodes and alleviate the over-smoothing of node representations. However, the performance of ComGA and ComGA-S is similar on Cora and ACM. The reason is that Cora and ACM as citation graphs have no obvious community structure. Besides, ComGA-S and ComGA-A achieve better anomaly detection results than DOMINANT on most datasets. This shows that designing the gateway is beneficial to learn more distinguishable and anomaly-aware node representations, and improves the performance of anomaly detection.

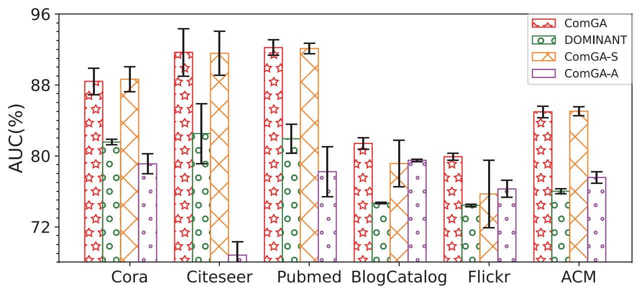



**Figure 3: Ablation analysis for ComGA model on six datasets**

## 5.5 Parameter Analysis

In the section, we investigate the effects of parameters $\alpha$ and $d$ in our model for anomaly detection, these are the trade-off parameter to balance the impact of structure reconstruction and attributes reconstruction, and the dimension of attributed graph embedding representations, respectively.

**Effect of Trade-off Parameter $\alpha$.** To explore the impact of the parameter $\alpha$ on six datasets for anomaly detection, we vary it from 0 to 1 to present the AUC values. As shown in Figure 4, when $\alpha$ is set 0, and merely considers the structure reconstruction, the performance is poor. When $\alpha$ is set 1, and merely considers the attributes reconstruction, the performance is better than the former case. When $\alpha$ is around 0.1 to 0.3, ComGA has the best performance on Cora, Citeseer, Pubmed, and ACM. Furthermore, when $\alpha$ is around 0.3 to 0.5, ComGA has better results on BlogCatalog and Flickr. This shows that structure reconstruction carries more weight than attributes reconstruction in ComGA. In other words, our method emphasizes anomalous nodes from structure information more. The reason is that the gateway of tGCN propagates learned community-specific representation into GCN, which combines more anomalies information including local, global, and structure anomalies.

**Effect of Embedding Dimension Parameter $d$.** To explore the impact of the parameter $d$ on six datasets for anomaly detection, we vary it from 8 to 512 to present the AUC values. As shown in Figure 4, the performance of our method is relative stable with different dimension values on BlogCatalog, Flickr, and ACM apart from the volatile performance on Cora, Citeseer, and Pubmed. Simultaneously, when $d$ is 128 or 256, ComGA has the best performance on these datasets. The main reason is that our method constructs a deeper graph model to learn anomaly-aware node representations.

## 5.6 Analysis of Gateway

To further analyze the effect of gateway to alleviate over-smoothing and learn more distinguishable anomalous node representations in tGCN module, we vary the depth of tGCN module while doing corresponding change of autoencoder layers. We do experiments on six datasets and present the AUC values in Figure 5. The figure is a violin analysis diagram, which mainly shows the distribution of experimental data. The white dot represents the median of experimental data, and the black box ranges from the lower quartile to the upper quartile for the data. The thin black line represents the maximum and minimum values of the data. Besides, the external shape is the kernel density estimation, which means the more centralized the data, the fatter it is. As we can see that 4-layers module or 5-layers module achieve better performance than 2-layers module. This demonstrates the effectiveness of gateway to alleviate the problem of over-smoothing, and make anomalous node representations more distinguishable. As it has been proved in [2, 18], as the number of GCN layers increases, the representations learned will be increasingly over-smoothing. This situation makes anomalous nodes less distinguishable from the majority nodes. However, in our model, the deeper of tGCN, the better the performance; and it is worth noting that 2-layers module also has competitive performance because this module does not cause serious over-smoothing problem. Besides, community-specific representation is propagated into tGCN, which enhances the performance for anomaly detection.

## 5.7 Discussion

In the anomaly detection module, we design the joint reconstruction errors to evaluate the abnormality of nodes from the perspectives of structure and attributes on attributed graphs. However, the number of zero elements on real-life graph datasets is larger than that of non-zero elements, which means that the above reconstruction error way causes more likely to reconstruct the zero elements rather than non-zero ones. Thus, this way of reconstruction error will degenerate the performance of anomaly detection. To address this problem, we impose two offset coefficient matrixes to adjust the weight values of different elements for structure reconstruction and attributes reconstruction. Thus, the reconstruction objective function in Eq.(12) can be reformulated as follows:

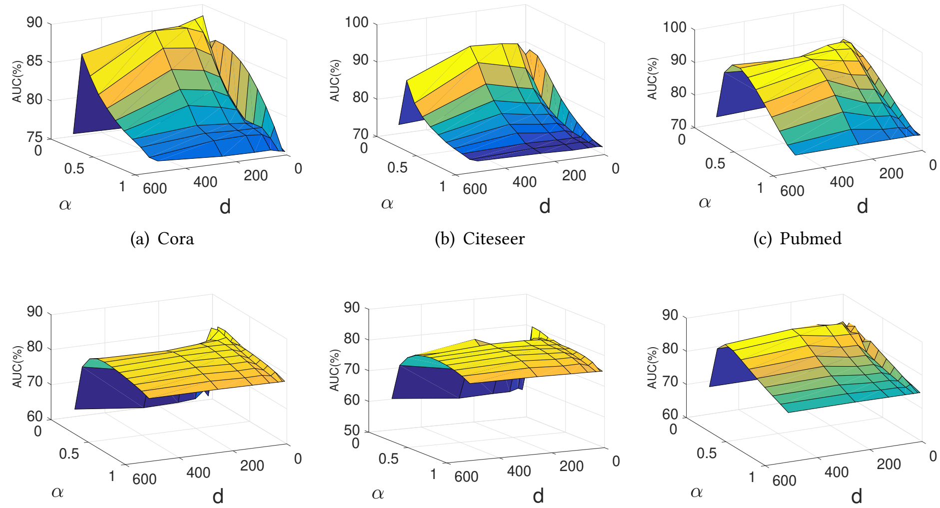
$$\bar{L}_{rec} = (1-\alpha)\left\|(\mathbf{A}-\hat{\mathbf{A}})\odot \mathbf{S}^1\right\|_F^2 + \alpha\left\|(\mathbf{X}-\hat{\mathbf{X}})\odot \mathbf{S}^2\right\|_F^2. \quad (15)$$

Where $\mathbf{S}^1$ and $\mathbf{S}^2$ are two offset coefficient matrixes, $s^1_{ij} = 1$ and $s^2_{ij} = 1$ when $\boldsymbol{A}_{ij} = 0$ and $\boldsymbol{X}_{ij} = 0$, otherwise, $s^1_{ij} = \theta > 1$ and $s^2_{ij} = \gamma > 1$, and $\odot$ means the Hadamard product.

Similarity, the anomaly score objective function in Eq.(13) can be reformulated as follows:

$$\bar{Score}_{v_i} = (1-\alpha)\left\|(\mathbf{a}_i-\hat{\mathbf{a}}_i)\odot \mathbf{s}^1_i\right\|_2^2 + \alpha\left\|(\mathbf{x}_i-\hat{\mathbf{x}}_i)\odot \mathbf{s}^2_i\right\|_2^2. \quad (16)$$

To demonstrate the effectiveness of the offset coefficient matrixes for anomaly detection, we put ComGA and DOMINANT which all employ the reconstruction errors to characterize anomalous nodes on the offset coefficient matrixes, and present the AUC values in Table 4. Besides, we do the same contrastive analysis for AnomalyDAE. As we can see, most methods with the offset coefficient matrixes achieve great performance on six datasets. Furthermore, ComGA or ComGA-O achieve the best performance among almost all the other methods. This demonstrates the effectiveness of capturing community structure and learning distinguishable node representations for multiple anomalies.




(a) Cora (b) Citeseer (c) Pubmed

(d) BlogCatalog (e) Flickr (f) ACM

**Figure 4: Parameter analysis for anomaly detection w.r.t different values for the parameters $d$ and $\alpha$.**

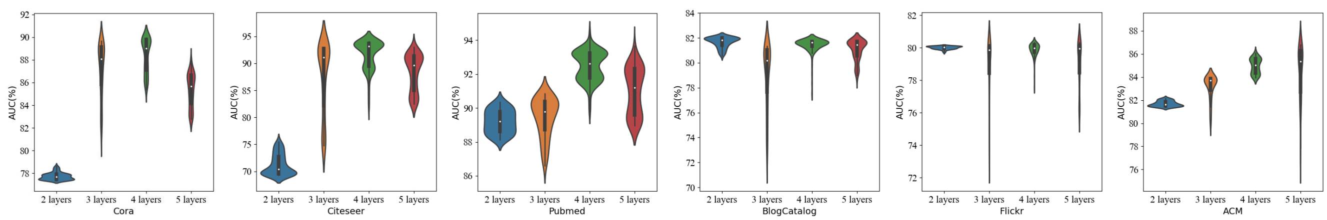



**Figure 5: Effect of gateway for anomaly detection on six datasets**

**Table 4: Effect of offset coefficient matrix for anomaly detection on six datasets**

| Methods | Cora | Citeseer | Pubmed | BlogCatalog | Flickr | ACM |
|---|---|---|---|---|---|---|
| DOMINANT | 81.55 | 82.51 | 80.81 | 74.68 | 74.42 | 76.01 |
| DOMINANT-O | 86.41 | 85.24 | 90.51 | 96.72 | 96.08 | 88.30 |
| AnomalyDAE | 76.27 | 72.71 | 81.03 | 78.34 | 75.08 | 75.13 |
| AnomalyDAE-O | 87.11 | 73.72 | 84.35 | 96.46 | 95.91 | 89.61 |
| ComGA | 88.4 | **91.67** | **92.2** | 81.4 | 79.9 | 84.96 |
| ComGA-O | **89.84** | 89.64 | 91.75 | **97.26** | **97.25** | **90.03** |

## 6 CONCLUSION

To improve the performance of anomaly detection, in this paper, we propose a novel community-aware attributed graph anomaly detection framework (ComGA). Specifically, we design the tGCN to propagate the community-specific representation into the GCN via multiple gateways. Thus, the tGCN module can respect the community structure of graphs and effectively learn more distinguishable node representations for local, global, and structure anomalies. Besides, this also extremely alleviates the over-smoothing of anomalous node representations. The experimental results demonstrate the superiority of the proposed ComGA method over the state-of-the-art methods in various real-world datasets. In the future, we plan to design more effective community detection methods to replace the modularity matrix, and extend ComGA for heterogeneous attributed graphs.

## 7 ACKNOWLEDGMENTS

This work is supported by the DECRA Project under DE200100964, the Natural Science Foundation of Tianjin under 19JCYBJC15300, the National Natural Science Foundation of China under 61702367 and 61976156, the Tianjin Science and Technology Commissioner project under 20YDTPJC00560, and the Research Project of Tianjin Municipal Commission of Education under 2017KJ033. X. Zhang and Y. Wang are the corresponding authors.

## REFERENCES


[1] Leman Akoglu, Hanghang Tong, Brendan Meeder, and Christos Faloutsos. 2012. PICS: Parameter-free Identification of Cohesive Subgroups in Large Attributed Graphs. In *SDM*. 439–450.

[2] Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. 2020. Structural Deep Clustering Network. In *WWW*. 1400–1410.

[3] Bernardo Branco, Pedro Abreu, Ana Sofia Gomes, Mariana SC Almeida, João Tiago Ascensão, and Pedro Bizarro. 2020. Interleaved sequence rnns for fraud detection. In *KDD*. 3101–3109.

[4] Markus M Breunig, Hanspeter Kriegel, Raymond T Ng, and Jorg Sander. 2000. LOF: identifying density-based local outliers. In *ACM SIGMOD Rec*, Vol. 29. 93–104.

[5] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. In *ICLR*.

[6] Kaize Ding, Jundong Li, Nitin Agarwal, and Huan Liu. 2020. Inductive Anomaly Detection on Attributed Networks. In *IJCAI*. 1288–1294.

[7] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep Anomaly Detection on Attributed Networks. In *SDM*. 594–602.

[8] Haoyi Fan, Fengbin Zhang, and Zuoyong Li. 2020. AnomalyDAE: Dual autoencoder for anomaly detection on attributed networks. In *ICASSP*. 5685–5689.

[9] Jing Gao, Feng Liang, Wei Fan, Chi Wang, Yizhou Sun, and Jiawei Han. 2010. On community outliers and their efficient detection in information networks. In *KDD*. 813–822.

[10] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*. 1024–1034.

[11] Tianjin Huang, Yulong Pei, Vlado Menkovski, and Mykola Pechenizkiy. 2021. Hop-count based self-supervised anomaly detection on attributed networks. *arXiv preprint arXiv:2104.07917* (2021).

[12] Xiao Huang, Jundong Li, and Xia Hu. 2017. Label Informed Attributed Network Embedding. In *WSDM*. 731–739.

[13] Di Jin, Zheng Chen, Dongxiao He, and Weixiong Zhang. 2015. Modeling with node degree preservation can accurately find communities. In *AAAI*.

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[16] Jundong Li, Harsh Dani, Xia Hu, and Huan Liu. 2017. Radar: Residual Analysis for Anomaly Detection in Attributed Networks. In *IJCAI*. 2152–2158.

[17] Jundong Li, Ruocheng Guo, Chenghao Liu, and Huan Liu. 2019. Adaptive Unsupervised Feature Selection on Attributed Networks. In *KDD*. 92–100.

[18] Qimai Li, Zhichao Han, and Xiaoming Wu. 2018. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In *AAAI*. 3538–3545.

[19] Yuening Li, Xiao Huang, Jundong Li, Mengnan Du, and Na Zou. 2019. SpecAE: Spectral AutoEncoder for Anomaly Detection in Attributed Networks. In *CIKM*. 2233–2236.

[20] Fanzhen Liu, Zhao Li, Baokun Wang, Jia Wu, Jian Yang, Jiaming Huang, Yiqing Zhang, Weiqiang Wang, Surya Nepal, and Quanzheng Sheng. 2022. eRiskCom: An e-commerce risky community detection platform. *VLDBJ* (2022).

[21] Fanzhen Liu, Shan Xue, Jia Wu, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Jian Yang, and Philip S. Yu. 2020. Deep learning for community detection: progress, challenges and opportunities. In *IJCAI*. 4981–4987.

[22] Juan Liu, Eric A Bier, Aaron Wilson, John Alexis Guerragomez, Tomonori Honda, Kumar Sricharan, Leilani H Gilpin, and Daniel Davies. 2016. Graph Analysis for Detecting Fraud, Waste, and Abuse in Healthcare Data. *Ai Magazine* 37, 2 (2016), 33–46.

[23] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. 2021. Anomaly Detection on Attributed Networks via Contrastive Self-Supervised Learning. *IEEE TNNLS* (2021), 1–15.

[24] Xuexiong Luo, Jia Wu, Chuan Zhou, Xiankun Zhang, and Yuan Wang. 2020. Deep Semantic Network Representation. In *ICDM*. 1154–1159.

[25] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. 2021. A comprehensive survey on graph anomaly detection with deep learning. *IEEE TKDE* (2021).

[26] Emmanuel Müller, Patricia Iglesias Sánchez, Yvonne Mülle, and Klemens Böhm. 2013. Ranking outlier nodes in subspaces of attributed graphs. In *ICDE Workshops*. 216–222.

[27] M E J Newman. 2006. Modularity and community structure in networks. *PNAS* 103, 23 (2006), 8577–8582.

[28] Yulong Pei, Tianjin Huang, Werner van Ipenburg, and Mykola Pechenizkiy. 2020. ResGCN: Attention-based Deep Residual Modeling for Anomaly Detection on Attributed Networks. *CoRR* abs/2009.14738 (2020).

[29] Bryan Perozzi and Leman Akoglu. 2016. Scalable Anomaly Ranking of Attributed Neighborhoods. In *SDM*. 207–215.

[30] Bryan Perozzi, Leman Akoglu, Patricia Iglesias Sánchez, and Emmanuel Müller. 2014. Focused clustering and outlier detection in large attributed graphs. In *KDD*. 1346–1355.

[31] Patricia Iglesias Sanchez, Emmanuel Muller, Fabian Laforet, Fabian Keller, and Klemens Bohm. 2013. Statistical Selection of Congruent Subspaces for Mining Attributed Graphs. In *ICDM*. 647–656.

[32] Wei Song, Heng Yin, Chang Liu, and Dawn Song. 2018. DeepMem: Learning Graph Neural Network Models for Fast and Robust Memory Forensic Analysis. In *ACM CCS*. 606–618.

[33] Xing Su, Shan Xue, Fanzhen Liu, Jia Wu, Jian Yang, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Di Jin, Quan Z. Sheng, and Philip S. Yu. 2021. A Comprehensive Survey on Community Detection with Deep Learning. *IEEE TNNLS* (2021).

[34] Patricia Iglesias Sánchez, Emmanuel Müller, Oretta Irmler, and Klemens Böhm. 2014. Local context selection for outlier ranking in graphs with multiple numeric node attributes. In *SSDBM*. 16:1–16:12.

[35] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[36] Teng Xiao, Zhengyu Chen, Donglin Wang, and Suhang Wang. 2021. Learning How to Propagate Messages in Graph Neural Networks. In *KDD*. 1894–1903.

[37] Hongzuo Xu, Yijie Wang, Songlei Jian, Zhenyu Huang, Yongjun Wang, Ning Liu, and Fei Li. 2021. Beyond Outlier Detection: Outlier Interpretation by Attention-Guided Triplet Deviation Network. In *WWW*. 1328–1339.

[38] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas A. J. Schweiger. 2007. SCAN: A Structural Clustering Algorithm for Networks. In *KDD*. 824–833.

[39] Liang Yang, Xiaochun Cao, Dongxiao He, Chuan Wang, Xiao Wang, and Weixiong Zhang. 2016. Modularity based community detection with deep learning. In *IJCAI*. 2252–2258.

[40] Ge Zhang, Zhao Li, Jiaming Huang, Jia Wu, Chuan Zhou, and Jian Yang. 2022. eFraudCom: An E-commerce Fraud Detection System via Competitive Graph Neural Networks. *ACM TOIS* (2022).

[41] Ge Zhang, Jia Wu, Jian Yang, Amin Beheshti, Shan Xue, Chuan Zhou, and Michael Sheng. 2021. FRAUDRE: Fraud Detection Dual-Resistant to Graph Inconsistency and Imbalance. In *ICDM*.

[42] Peng Zhen, Minnan Luo, Jundong Li, Huan Liu, and Qinghua Zheng. 2018. ANOMALOUS: A Joint Modeling Approach for Anomaly Detection on Attributed Networks. In *IJCAI*. 3513–3519.