ANEMONE: Graph Anomaly Detection with Multi-Scale Contrastive Learning

Ming Jin
Monash University
ming.jin@monash.edu

Yixin Liu
Monash University
yixin.liu@monash.edu

Yu Zheng
La Trobe University
yu.zheng@latrobe.edu.au

Lianhua Chi
La Trobe University
l.chi@latrobe.edu.au

Yuan-Fang Li
Monash University
yuanfang.li@monash.edu

Shirui Pan
Monash University
shirui.pan@monash.edu

ABSTRACT

Anomaly detection on graphs plays a significant role in various domains, including cybersecurity, e-commerce, and financial fraud detection. However, existing methods on graph anomaly detection usually consider the view in a single scale of graphs, which results in their limited capability to capture the anomalous patterns from different perspectives. Towards this end, we introduce a novel graph anomaly detection framework, namely ANEMONE, to simultaneously identify the anomalies in multiple graph scales. Concretely, ANEMONE first leverages a graph neural network backbone encoder with multi-scale contrastive learning objectives to capture the pattern distribution of graph data by learning the agreements between instances at the patch and context levels concurrently. Then, our method employs a statistical anomaly estimator to evaluate the abnormality of each node according to the degree of agreement from multiple perspectives. Experiments on three benchmark datasets demonstrate the superiority of our method.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Anomaly detection**; • **Mathematics of computing** → **Graph algorithms**.

KEYWORDS

Anomaly Detection, Graph Neural Networks, Contrastive Learning

ACM Reference Format:
Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan. 2021. ANEMONE: Graph Anomaly Detection with Multi-Scale Contrastive Learning. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM'21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3482057

1 INTRODUCTION

Recently, anomaly detection on graphs has received increasing attention in the community of data mining [9] due to the wide applications of graph-structured data in modeling real-world systems, including e-commerce, and finance [16]. Taking e-commerce fraud detection as an example, an anomaly detection algorithm can help to identify fraudulent sellers by analysing the properties (i.e., attributes) and connection (i.e., structure) of users.

Unlike conventional anomaly detection methods which merely consider the attributive information of each sample and ignore their underlying correlations, graph anomaly detection, on the other hand, takes sample (i.e., node) attributes as well as the topological information (i.e., node adjacency) into consideration simultaneously [5]. Earlier methods leverage shallow mechanisms like ego-network analysis [11], residual analysis [4] or CUR decomposition [10] to detect anomalous nodes, which fail to learn informative knowledge from high-dimensional attributes. Recently proposed methods [1, 5] exploit deep graph autoencoder to anomaly detection and make significant performance improvements. Very recently, by introducing graph self-supervised learning [7], CoLA [6] integrates contrastive learning into graph neural network (GNN) [14] to detect graph anomalies effectively.

Despite their success, these methods mainly detect anomalies from the perspective of a single scale, ignoring the fact that node anomalies in graphs often occur in different scales. For instance, some e-commence cheaters may directly trade with a small number of unrelated items/users (i.e., local anomalies), while other cheaters tend to hide themselves in large communities of underground industry (i.e., global anomalies). Such heterogeneity of scales leads to the sub-optimal performance of existing methods.

To bridge this gap, we propose a graph **AN**omaly d**E**tection framework with **M**ulti-scale c**ON**trastive l**E**arning (**ANEMONE** for abbreviation) to detect anomalous nodes in graphs. First, to capture anomalous patterns in different scales, our proposed framework simultaneously performs patch- and context-level contrastive learning via two GNN-based models. Moreover, ANEMONE employs a novel anomaly estimator to predict the abnormality of each node by leveraging the statistics of multi-round contrastive scores. The main contributions of this work are summarized as follows:

- We propose a multi-scale contrastive learning framework, ANEMONE, for graph anomaly detection, which captures the anomalous patterns in different scales.
- We design a novel statistics-based algorithm to estimate node abnormality with the proposed contrastive schema.
- We conduct extensive experiments on three benchmark datasets to demonstrate the superiority of ANEMONE in detecting node-level anomalies on graphs.

CIKM'21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482057

2 PROBLEM STATEMENT

In this paper, we focus on the anomaly detection problem for attributed graph. Let $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ be an attributed graph with a node set $\mathcal{V} = \{v_1, \cdots, v_n\}$. $\mathbf{A} \in \mathbb{R}^{n \times n}$ denotes the binary adjacency matrix where $\mathbf{A}_{i,j} = 1$ indicates that there is a link between v_i and v_j otherwise $\mathbf{A}_{i,j} = 0$. $\mathbf{X} \in \mathbb{R}^{n \times f}$ denotes the attribute matrix where the i-th row $\mathbf{X}[i,:] \in \mathbb{R}^f$ indicates the attribute vector of v_i. With the aforementioned notations, we formalize the graph anomaly detection problem as follows:

Definition 2.1 (Graph Anomaly Detection). Given an attributed graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, the target is to learn a function $\mathcal{Y}(\cdot) : \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times f} \rightarrow \mathbb{R}^n$, which takes the graph as input data and outputs a vector of anomaly scores $\mathbf{y}$ to measure the degree of abnormality of each node. Specifically, the i-th element $y^{(i)}$ in the output scoring vector $\mathbf{y}$ expresses the abnormality of the v_i, where a larger score means a higher abnormality.

It is worth noticing that the graph anomaly detection is performed under an unsupervised scenario, meaning that the ground-truth labels are inaccessible during the training stage.

3 PROPOSED ANEMONE FRAMEWORK

We present a framework, namely ANEMONE, based on the multi-scale contrastive learning [2] for graph anomaly detection. The overall pipeline of our method is illustrated in Figure 1. For a selected target node , ANEMONE calculates the anomaly score of this node by capitalizing on two main components: *multi-scale contrastive learning model* and *statistical anomaly estimator*. In *multi-scale contrastive learning model*, two GNN-based contrastive networks learn the patch-level (i.e., node versus node) agreement and context-level (i.e., node versus ego-net) agreement respectively. After that, *statistical anomaly estimator* aggregates the patch- and context- level scores acquired by multiple augmented ego-nets and calculates the final anomaly score of the target node via statistical estimation. We introduce the two components in the following sections.

3.1 Multi-Scale Contrastive Learning Model

3.1.1 Augmented Ego-nets Generation. In *multi-scale contrastive learning model*, we first generate two ego-nets of the target node with data augmentation as the networks' input. The motivation behind ego-nets generation is to capture the surrounding substructures of the target node (which is proved to be highly related to the node's abnormality [6, 8]) as well as provide sufficient diversity of input data for model training and statistical estimator. Taking the above into consideration, we employ a random walk-based algorithm, RWR [12], as our data augmentation strategy. To be concrete, taking a target node v_i as center, we sample two ego-nets with a fix size K which are denoted as $\mathcal{G}_p^{(i)} = (\mathbf{A}_p^{(i)}, \mathbf{X}_p^{(i)})$ and $\mathcal{G}_c^{(i)} = (\mathbf{A}_c^{(i)}, \mathbf{X}_c^{(i)})$. In each ego-net, we set the first node in the node set as the center (target) node.

To prevent information leakage in the following contrastive learning step, a pre-processing named *target node masking* should be implemented in the ego-nets before we feed them into the contrastive networks. Concretely, we replace the attribute vector of target node with a zero vector: $\mathbf{X}_p^{(i)}[1,:] \leftarrow \vec{0}, \mathbf{X}_c^{(i)}[1,:] \leftarrow \vec{0}$.

3.1.2 Patch-level Contrastive Network. The target of *patch-level contrastive network* is to learn the agreement between the embedding of masked target node within ego-net $\mathcal{G}_p^{(i)}$ and embedding of original target node v_i. Firstly, the node embeddings $\mathbf{H}_p^{(i)}$ of ego-net are obtained by the GNN module:

$$\begin{aligned} \mathbf{H}_p^{(i)} &= GNN_\theta\left(\mathcal{G}_p^{(i)}\right) = GCN\left(\mathbf{A}_p^{(i)}, \mathbf{X}_p^{(i)}; \mathbf{\Theta}\right) \\ &= \sigma\left(\widetilde{\mathbf{D}_p^{(i)}}^{-\frac{1}{2}} \widetilde{\mathbf{A}_p^{(i)}} \widetilde{\mathbf{D}_p^{(i)}}^{-\frac{1}{2}} \mathbf{X}_p^{(i)} \mathbf{\Theta}\right), \end{aligned} \tag{1}$$

where θ is the parameter set of GNN. For simplicity, here we directly adopt a one-layer GCN [3] , where $\widetilde{\mathbf{A}_p^{(i)}} = \mathbf{A}_p^{(i)} + \mathbf{I}$ is the adjacency matrix added self-loop, $\widetilde{\mathbf{D}_p^{(i)}}$ is the degree matrix of the ego-net $\mathcal{G}_p^{(i)}$, $\mathbf{\Theta} \in \mathbb{R}^{f \times d}$ is the weight matrix of the GCN layer, d is the dimension of embedding, and $\sigma(\cdot)$ is the ReLU activation function. Here GCN can be replaced by other types of GNN alternatively. For patch-level contrastive learning, we pick the embedding of masked target node by letting $\mathbf{h}_p^{(i)} = \mathbf{H}_p^{(i)}[1,:]$. It is worth noting that, although the corresponding input $\mathbf{X}_p^{(i)}[1,:]$ is a zero vector, the embedding $\mathbf{H}_p^{(i)}[1,:]$ becomes informative by aggregating the attributes of other nodes in the ego-net via GNN.

Then, ANEMONE computes the embedding of target node v_i by a MLP module. We denote the attribute vector of v_i as $\mathbf{x}^{(i)} = \mathbf{X}[i,:]$, and the target node embedding $\mathbf{z}_p^{(i)}$ is given as follows:

$$\mathbf{z}_p^{(i)} = MLP_\theta\left(\mathbf{x}^{(i)}\right) = \sigma\left(\mathbf{x}^{(i)} \mathbf{\Theta}\right). \tag{2}$$

Here the weight is shared with the GNN in Eq. (1), which ensures that $\mathbf{h}_p^{(i)}$ and $\mathbf{z}_p^{(i)}$ are projected into the same embedding space.

After that, a contrastive learning module is built to learn the agreement between $\mathbf{h}_p^{(i)}$ and $\mathbf{z}_p^{(i)}$. Specifically, we utilize a bilinear layer to calculate their similarity score:

$$s_p^{(i)} = Bilinear\left(\mathbf{h}_p^{(i)}, \mathbf{z}_p^{(i)}\right) = \sigma\left(\mathbf{h}_p^{(i)} \mathbf{W}_p \mathbf{z}_p^{(i)\top}\right), \tag{3}$$

where $\mathbf{W}_p$ is a trainable matrix, and $\sigma(\cdot)$ is Sigmoid function.

To learn a discriminative contrastive network, we introduce a *negative sampling* strategy for model training. That is, for a given score $s_p^{(i)}$ (to distinguish, we denote it as "positive score"), we calculate the negative score $\tilde{s}_p^{(i)}$ by:

$$\tilde{s}_p^{(i)} = Bilinear\left(\mathbf{h}_p^{(j)}, \mathbf{z}_p^{(i)}\right) = \sigma\left(\mathbf{h}_p^{(j)} \mathbf{W}_p \mathbf{z}_p^{(i)\top}\right), \tag{4}$$

where $\mathbf{h}_p^{(j)}$ is acquired from the ego-net centered at another node v_j ensuring that $i \neq j$. In practice, our contrastive learning model is trained in a mini-batch manner. Therefore, $\mathbf{h}_p^{(j)}$ can be easily obtained from other target nodes in the same batch. With $s_p^{(i)}$ and $\tilde{s}_p^{(i)}$, the patch-level contrastive network is trained with a Jensen-Shannon divergence [13] objective function:

$$\mathcal{L}_p = -\frac{1}{2n} \sum_{i=1}^{n} \left(log\left(s_p^{(i)}\right) + log\left(1 - \tilde{s}_p^{(i)}\right)\right). \tag{5}$$

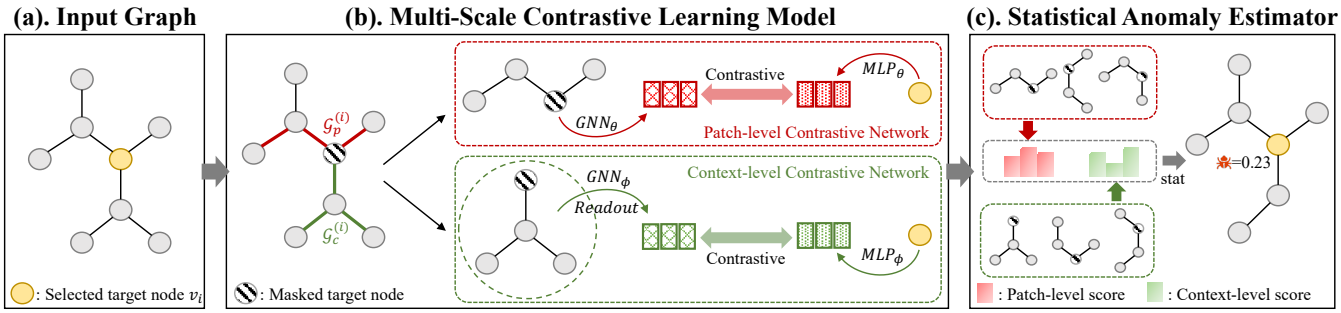


Figure 1: The overall pipeline of ANEMONE. The target is to predict the anomaly score of a selected target node v_i (the yellow node in (a)). In *multi-scale contrastive learning model* (b), the "node v.s. node" agreement and "node v.s. ego-net" agreement are learned by the patch- and context- level contrastive network respectively. In *statistical anomaly estimator*, the anomaly score is estimated with a statistical algorithm according to the multi-round predicted scores by contrastive learning model.

3.1.3 Context-level Contrastive Network. Symmetrically, the *context-level contrastive network* has a similar architecture with the patch-level one. Firstly, in analogy to Eq. (1), a siamese GNN module with parameter set ϕ generate the node embeddings $\mathbf{H}_c^{(i)}$ from the input ego-net $\mathcal{G}_c^{(i)}$, which is formulated by:

$$\mathbf{H}_c^{(i)} = GNN_\phi\left(\mathcal{G}_c^{(i)}\right) = \sigma\left(\widetilde{\mathbf{D}_c^{(i)}}^{-\frac{1}{2}}\widetilde{\mathbf{A}_c^{(i)}}\widetilde{\mathbf{D}_c^{(i)}}^{-\frac{1}{2}}\mathbf{X}_c^{(i)}\mathbf{\Phi}\right). \quad (6)$$

Note that *context-level contrastive network* has a different parameter set from the patch-level one, since the contrasts in two scales should be carried out in different embedding spaces.

The main distinction between patch- and context- level contrast is that, the latter tries to learn the agreement between the target node embedding and the ego-net embedding, which is obtained via a readout module:

$$\mathbf{h}_c^{(i)} = readout\left(\mathbf{H}_c^{(i)}\right) = \frac{1}{K}\sum_{j=1}^{K}\mathbf{H}_c^{(i)}[j,:]. \quad (7)$$

In this paper, we adopt average pooling as our readout function.

To project the target node's attribute to the same embedding space, a MLP module (similar to Eq. (2)) with parameter θ is leveraged to compute $\mathbf{z}_c^{(i)}$. Subsequently, the context-level score $s_c^{(i)}$ is estimated by a bilinear function with scoring matrix $\mathbf{W}_c$. Finally, the context-level network is trained by the objective function:

$$\mathcal{L}_c = -\frac{1}{2n}\sum_{i=1}^{n}\left(log\left(s_c^{(i)}\right) + log\left(1-\tilde{s}_c^{(i)}\right)\right). \quad (8)$$

3.1.4 Joint Training. In the training stage, we learn the two contrastive networks jointly. The overall objective function is:

$$\mathcal{L} = \alpha\mathcal{L}_c + (1-\alpha)\mathcal{L}_p, \quad (9)$$

where $\alpha \in [0,1]$ is a trade-off parameter to balance the importance between two components.

3.2 Statistical Anomaly Estimator

After the *multi-scale contrastive learning model* is well trained, ANEMONE utilizes a *statistical anomaly estimator* to calculate the anomaly scores for each node in the inference stage. First, for a given target node v_i, we generate R ego-nets for patch- and context-level contrastive networks respectively. Meanwhile, negative samples with an equal number are sampled. Feeding them into corresponding contrastive network, we obtain a total of $4R$ scores, which is: $[s_{p,1}^{(i)},\cdots,s_{p,R}^{(i)},s_{c,1}^{(i)},\cdots,s_{c,R}^{(i)},\tilde{s}_{p,1}^{(i)},\cdots,\tilde{s}_{p,R}^{(i)},\tilde{s}_{c,1}^{(i)},\cdots,\tilde{s}_{c,R}^{(i)}]$. We assume that an anomalous node has a smaller agreement with its adjacent structure and contexts. Therefore, we denote the base score as the difference between negative and positive scores:

$$b_{view,j}^{(i)} = \tilde{s}_{view,j}^{(i)} - s_{view,j}^{(i)}, \quad (10)$$

where the subscript "view" represents "p" or "c" and $j \in [1,\cdots,R]$.

Then, we consider a statistical method for abnormality estimation. The behind intuition is that: 1) an anomalous node has relatively large base scores; 2) an anomalous node has unstable base scores under multiple ego-net sampling. Accordingly, we define the statistical anomaly scores $y_p^{(i)}$ and $y_c^{(i)}$ as the sum of mean and standard deviation for base scores:

$$\begin{aligned}\bar{b}_{view}^{(i)} &= \sum_{j=1}^{R} b_{view,j}^{(i)}/R,\\ y_{view}^{(i)} &= \bar{b}_{view}^{(i)} + \sqrt{\sum_{j=1}^{R}\left(b_{view,j}^{(i)} - \bar{b}_{view}^{(i)}\right)^2/R},\end{aligned} \quad (11)$$

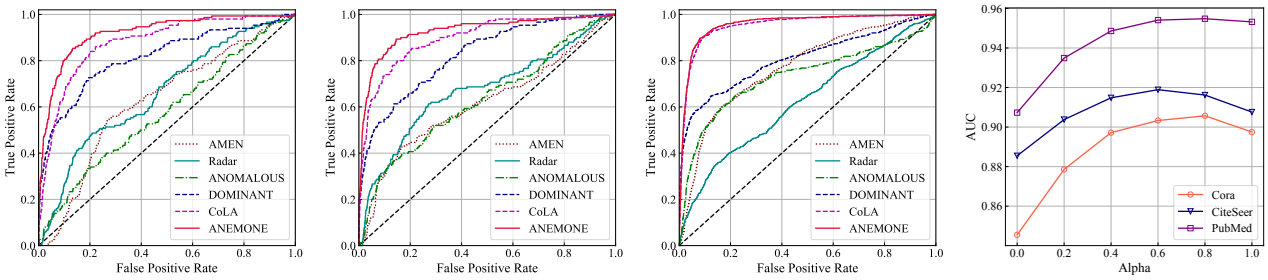
where the subscript "view" represents "p" or "c". Finally, we combine $y_p^{(i)}$ and $y_c^{(i)}$ into the final anomaly score $y^{(i)}$ for v_i, where the parameter α in Eq. (9) serves as a trade-off term:

$$y^{(i)} = \alpha y_c^{(i)} + (1-\alpha)y_p^{(i)}. \quad (12)$$

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Datasets. We conduct extensive experiments on three well-known citation network datasets, i.e. Cora, CiteSeer and PubMed. The statistics of the datasets are summarized in Table 1. Since these citation datasets have no anomalies by default and to evaluate our method in detecting different types of anomalies, we follow previous work [1, 6] to manually inject an equal number of attributive and structural anomalous nodes.



(a) ROC curve of Cora. (b) ROC curve of CiteSeer. (c) ROC curve of PubMed. (d) Trade-off parameter α w.r.t. AUC values.

Figure 2: ROC curves and the analysis for trade-off parameter α on three datasets.

Table 1: Basic statistics of the three datasets.

Datasets	# Nodes	# Edges	# Attributes	# Anomalies
Cora	2,708	5,429	1,433	150
CiteSeer	3,327	4,732	3,703	150
PubMed	19,717	44,338	500	600

Table 2: AUC of ANEMONE, its competitors and variants.

Methods	Cora	CiteSeer	PubMed
AMEN	0.6266	0.6154	0.7713
Radar	0.6587	0.6709	0.6233
ANOMALOUS	0.5770	0.6307	0.7316
DOMINANT	0.8155	0.8251	0.8081
CoLA	0.8779	0.8968	0.9512
CoLA$_{stat}$	0.8869	0.9047	0.9532
ANEMONE$_{mean}$	0.8963	0.9066	0.9524
ANEMONE$_{std}$	0.5402	0.7077	0.7440
ANEMONE	**0.9057**	**0.9189**	**0.9548**

4.1.2 Baselines. We compare ANEMONE with the following methods: AMEN [11], Radar [4], ANOMALOUS [10], DOMINANT [1] and CoLA [6]. We add a variant of CoLA, CoLA$_{stat}$, which integrates the proposed *statistical anomaly estimator* into CoLA. Our code is made available on GitHub [1], including the hyperparameter setting.

4.1.3 Metric. A widely applied metric, ROC-AUC, is employed to evaluate the performance of anomaly detection. The ROC curve represents the plot of true positive rate against false positive rate, while AUC is the area under the ROC curve. The value of AUC is within $[0, 1]$, and a larger one indicates a better performance.

4.2 Effectiveness Evaluation

The ROC curves are demonstrated in Figure 2(a)-(c), while the comparison of AUC is given in Table 2. We make the following observations:

- In general, ANEMONE always outperforms all baseline methods on three benchmark datasets, which illustrates that the combination of the multi-scale contrastive learning technique and the statistical anomaly estimator significantly benefits the node-level anomaly detection.
- The deep learning-based approaches, i.e., DOMINANT, CoLA, and ANEMONE, outperform the shallow methods significantly, indicating that shallow mechanisms fail to capture anomalous patterns from high-dimensional attributes and complex underlying graph structure.
- CoLA$_{stat}$ shows a performance gain over CoLA, verifying the effectiveness of the proposed statistic anomaly estimator.

4.3 Ablation Study and Parameter Analysis

We further compare the results of ANEMONE and its variants, i.e., ANEMONE$_{mean}$ and ANEMONE$_{std}$, which only consider the mean value or standard deviation when estimating the anomaly score. As we can see in Table 2, both components in the anomaly estimator make a contribution to detecting anomalies, and the mean value of base scores has a greater correlation with node-level abnormality. Furthermore, the best performance is achieved by ANEMONE which combines both terms together.

The results of the effectiveness analysis for two contrastive scales are shown in Figure 2(d). We observe that the optimal performance is acquired when α is equal to 0.8 for Cora, 0.6 for CiteSeer and 0.8 for PubMed. Either larger or small values will lead to performance degradation. We conclude that both patch- and context-level contrastiveness can expose the exclusive anomalies in the corresponding scale. By jointly considering two perspectives, we can obtain the best results.

5 CONCLUSION

In this paper, we introduce a general framework named ANEMONE for graph anomaly detection. ANEMONE leverages a multi-scale contrastive learning technique to capture graph anomalies in multiple scales. A novel statistical estimation strategy is also included in ANEMONE for abnormality prediction. Extensive experiments on three benchmark datasets demonstrate the effectiveness of our proposed framework. For future work, we plan to investigate automatic learning approaches [15] to detect graph anomalies.

[1] https://github.com/GRAND-Lab/ANEMONE

REFERENCES

[1] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep anomaly detection on attributed networks. In *SDM*. SIAM, 594–602.

[2] Ming Jin, Yizhen Zheng, Yuan-Fang Li, Chen Gong, Chuan Zhou, and Shirui Pan. 2021. Multi-Scale Contrastive Siamese Networks for Self-Supervised Graph Representation Learning. In *IJCAI*.

[3] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*. 1–14.

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

[5] 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.

[6] 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).

[7] Yixin Liu, Shirui Pan, Ming Jin, Chuan Zhou, Feng Xia, and Philip S Yu. 2021. Graph Self-Supervised Learning: A Survey. *arXiv preprint arXiv:2103.00111* (2021).

[8] Yixin Liu, Shirui Pan, Yu Guang Wang, Fei Xiong, Liang Wang, and Vincent Lee. 2021. Anomaly Detection in Dynamic Graphs via Transformer. *arXiv preprint arXiv:2106.09876* (2021).

[9] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Quan Z Sheng, and Hui Xiong. 2021. A Comprehensive Survey on Graph Anomaly Detection with Deep Learning. *arXiv preprint arXiv:2106.07178* (2021).

[10] Zhen Peng, 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.

[11] Bryan Perozzi and Leman Akoglu. 2016. Scalable anomaly ranking of attributed neighborhoods. In *SDM*. SIAM, 207–215.

[12] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *ICDM*. IEEE, 613–622.

[13] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. In *ICLR*.

[14] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE TNNLS* 32, 1 (2020), 4–24.

[15] Miao Zhang, Steven Su, Shirui Pan, Xiaojun Chang, Ehsan Abbasnejad, and Reza Haffari. 2021. iDARTS: Differentiable Architecture Search with Stochastic Implicit Gradients. *arXiv preprint arXiv:2106.10784* (2021).

[16] Yizhen Zheng, V. Lee, Zonghan Wu, and Shirui Pan. 2021. Heterogeneous Graph Attention Network for Small and Medium-Sized Enterprises Bankruptcy Prediction. In *PAKDD*.