4-2023

Subgraph centralization: A necessary step for graph anomaly detection

Zhong ZHUANG

Kai Ming TING

Guansong PANG
Singapore Management University, gspang@smu.edu.sg

Shuaibin SONG

Subgraph Centralization: A Necessary Step for Graph Anomaly Detection

Zhong Zhuang* Kai Ming Ting* Guansong Pang† Shuaibin Song*

Abstract

Graph anomaly detection has attracted a lot of interest recently. Despite their successes, existing detectors have at least two of the three weaknesses: (a) high computational cost which limits them to small-scale networks only; (b) existing treatment of subgraphs produces suboptimal detection accuracy; and (c) unable to provide an explanation as to why a node is anomalous, once it is identified. We identify that the root cause of these weaknesses is a lack of a proper treatment for subgraphs. A treatment called Subgraph Centralization for graph anomaly detection is proposed to address all the above weaknesses. Its importance is shown in two ways. First, we present a simple yet effective new framework called Graph-Centric Anomaly Detection (GCAD). The key advantages of GCAD over existing detectors including deep-learning detectors are: (i) better anomaly detection accuracy; (ii) linear time complexity with respect to the number of nodes; and (iii) it is a generic framework that admits an existing point anomaly detector to be used to detect node anomalies in a network. Second, we show that Subgraph Centralization can be incorporated into two existing detectors to overcome the above-mentioned weaknesses.

1 Introduction

Attributed networks are omnipresent in various applications because they are a powerful means to represent node information as well as the relationship between nodes well. Examples are: (a) In a social network [16] (e.g., Facebook, Blog, LinkedIn), users with attribute information following or being followed by other users can be seen as nodes and connections in an attributed network. (b) In a citation network [8] of scientific publications, authors denote nodes, and the connections between nodes denote joint publications between authors.

Anomalous node detection in an attributed network has many applications, e.g., fraud detection, social spam detection and academic misconduct detection.

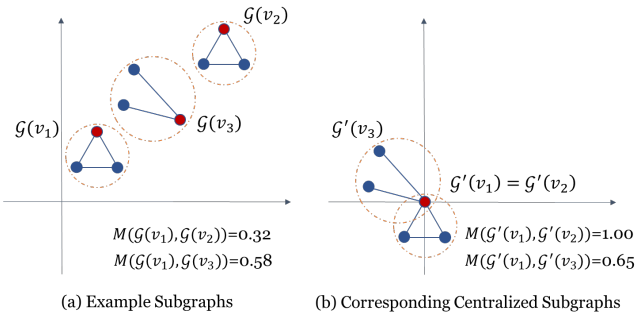


Figure 1: (a) Three example subgraphs of a (not-shown) network in the space of given node vectors. The red node represents the source node v of a subgraph $\mathcal{G}(v)$. M denotes a similarity measure between subgraphs. (b) In the translated space, each centralized subgraph $\mathcal{G}'(v)$ has all its nodes translated to the same extent as source node v is translated to the origin. $\mathcal{G}'(v_1) = \mathcal{G}'(v_2)$ because they have the same structure.

Existing anomalous node detectors can be categorized into two approaches, i.e., full-graph-based detectors and subgraph-based detectors. The former has high computational cost which limits them to small-scale networks only. We discover that the latter does not use subgraphs to their full potential resulting in suboptimal detection accuracy. In addition, both types of detectors are unable to provide an explanation as to why a node is anomalous, once it is identified.

The importance of subgraphs can be understood from the fact that each node v is influenced by all the nodes connected to it directly and transitively in a subgraph centered at v. The nodes connected via many hops outside the subgraph have negligible or no influence. Therefore, the similarity between subgraphs determine whether a subgraph is normal or anomalous.

Though the second existing approach has the same understanding, the methodology has a serious weakness, i.e., no centralization is performed before computing the similarity between subgraphs. As a result, the similarity between subgraphs can be misleading. An example is shown in Figure 1(a): subgraph $\mathcal{G}(v_1)$ is deemed to be more similar to $\mathcal{G}(v_3)$ than $\mathcal{G}(v_2)$ simply because the

*National Key Laboratory for Novel Software Technology, Nanjing University, {zhuangz,tingkm,songsb}@lamda.nju.edu.cn
†Singapore Management University, gspang@smu.edu.sg

node positions of the first two graphs are closer, i.e., the node positions play a more important role than the graph structures in the similarity calculation.

Intuitively, the similarity between subgraphs rely primarily on the graph structure, and the absolute positions in the space of the given node vectors shall play no role in the calculation. This can be achieved by translating all nodes of a subgraph $\mathcal{G}(v)$ to the same extent as the source node v is translated to the origin. Figure 1(b) shows the translated subgraphs (in the translated space) of those shown in Figure 1(a) (in the original space). Indeed, the similarity between the translated subgraphs $\mathcal{G}'(v_1)$ and $\mathcal{G}'(v_2)$ is unity because they have the same structure; and each of the them is less similar to $\mathcal{G}'(v_3)$ due to the difference in structures.

We call this technique *Subgraph Centralization*. We show that it is a necessary step in a proposed new graph anomaly detection framework and in addressing all the above-mentioned weaknesses of existing detectors.

Our main contributions are summarized as follows:

1. Formally define normal and anomalous nodes in a network. As far as we know, we are the first to provide such definitions on graph anomaly detection.
2. Propose a simple but effective new framework called Graph-Centric Anomaly Detection or GCAD which employs Subgraph Centralization as a necessary step. GCAD is capable of detecting anomaly on large-scale networks and providing an explanation why a node is anomalous/normal. The framework also admits an existing point anomaly detector to be used to perform graph anomaly detection.
3. Demonstrate that Subgraph Centralization can be applied to two existing detectors to improve their detection accuracy, and empower their ability to explain their predictions.

GCAD is distinguished from the two existing approaches with two unique features, i.e., the use of centralized subgraphs as the basis for similarity measurement and explainability. Subgraphs are the key to the definition of node anomalies and the design of the GCAD framework. The centralization is simple yet crucial to the success of GCAD. The explainability of GCAD depends on the centralized subgraphs, but independent of the point anomaly detector used in GCAD.

2 Related Work

To detect anomalies, it is vital to model some characteristics of a network that represent the normality of the network. We categorize existing works into two approaches, along the line whether the entire network or subgraphs are used in the modeling:

The full-graph-based approach is to model some form of normal characteristics from an entire network. ANOMALOUS [26] selects attributes on the space of features based on the structure of a network and applies residual analysis to detect anomalies. DOMINANT [5] combines GCN [12] with an autoencoder to reconstruct the network (via both the attribute matrix and adjacency matrix) such that the normal nodes are reconstructed with small errors. A few other works [7, 13] have followed the same methodology with some improvements. Oddball [1] models the normal characteristics of a network via a power-law model. OCGNN [22] applies GCN and hypersphere learning [18] to perform representation learning on a network.

The subgraph-based approach utilizes various ways to extract subgraphs from a network. CoLA [15] generates a large set of positive and negative subgraphs, where a normal subgraph depicts the normal relationship between each node and its neighbouring structure in the network; and a negative subgraph does not. This set is used to train a classifier. ANEMONE [9] and SL-GAD [27] employ self-supervised learning by generating subgraphs surrounding target nodes and performing patch-level and contextual-level contrastive learning.

However, none of them utilize Subgraph Centralization, leading to the problem we mentioned in Section 1.

3 Node Anomaly: Definitions

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{X}\}$ be an undirected attributed network, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ denotes the set of nodes, $\mathcal{E}$ denotes the set of edges, $\mathcal{X} \in \mathbb{R}^{n \times d}$ denotes the attribute matrix of nodes, and each node vector has d attributes.

Node anomaly detection is defined as:

Definition 1. (Node Anomaly Detection) *The task of node anomaly detection in a network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{X}\}$ is to identify the few node anomalies which have characteristics different from the majority of the nodes in the network, and to rank all the nodes on the basis of their anomaly scores such that the node anomalies are ranked higher than the normal ones.*

To operationalize the above generic definition, a method must be devised to provide a score for each node. We propose to use h-subgraphs as the basic means to do this in the next subsection.

3.1 Node Anomalies Based on h-subgraphs

Definition 2. (h-subgraph) *An h-subgraph $\mathcal{G}^h(v) = (\mathbf{V}, \mathbf{E}, \mathbf{X})$ is a subgraph rooted at source node v such that the shortest path between any node in $\mathcal{G}^h(v)$ and v has length $\leq h$, where $h \in \mathbb{N}$ is the maximum depth of the subgraph.*

Given a network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{X}\}$ having n nodes, there are a total of n h-subgraphs. Among the n h-subgraphs, the ones which are few and different in terms of some measure from the majority of the h-subgraphs are regarded as anomalous; and the source nodes of these h-subgraphs are regarded as node anomalies in $\mathcal{G}$. We propose to use a similarity measure to characterize the differences between h-subgraphs.

The node anomaly based on h-subgraphs is defined as follows. Let the set of nodes $\mathcal{V}$ consists of two subsets $\mathcal{V}^A$ and $\mathcal{V}^N$ which denote the sets of anomalous nodes and normal nodes, respectively, where $|\mathcal{V}^N| \gg |\mathcal{V}^A|$, and $M(\cdot, \cdot)$ be a similarity measure between two h-subgraphs.

DEFINITION 3. *Node u in $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{X}\}$ is*

- *A normal node if $\mathcal{G}^h(u)$ is similar to most other h-subgraphs, i.e., $\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} M(\mathcal{G}^h(u), \mathcal{G}^h(v))$ is large.*
- *An anomalous node if $\mathcal{G}^h(u)$ is dissimilar to most other h-subgraphs, i.e., $\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} M(\mathcal{G}^h(u), \mathcal{G}^h(v))$ is small.*

In practice, the difference can be parameterized as: u is an anomalous node if $\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} M(\mathcal{G}^h(u), \mathcal{G}^h(v)) < \tau$; otherwise u is a normal node. Or simply select the top `m` anomalous nodes which have the lowest similarities.

4 Proposed Framework: GCAD

We propose a new framework called GCAD (Graph-Centric node Anomaly Detection) to detect node anomalies in a network. GCAD scores every node by examining the similarity between h-subgraphs extracted from a network. A node v has a high score if the h-subgraph with v as the source node is dissimilar to most other h-subgraphs in the network.

The proposed framework consists of four main components: subgraph extraction and centralization, subgraph embedding, anomaly detection and subgraph depth-based weighted anomaly scoring, as shown in Algorithm 1.

As the base unit of operation is h-subgraphs, the algorithm begins to extract n h-subgraphs from a given network consisting of n nodes.

To enable similarity measurements among h-subgraphs, one key step is to centralize all these h-subgraphs to nullify the unwanted interference of node positions in the node vector space, since we are not interested in their absolute difference in the node vector space. Only then it is meaningful to examine the (dis)similarity between h-subgraphs, purely based on the structure of individual h-subgraphs. This process is

Algorithm 1 GCAD

Require: Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$; subgraph depth: h; the parameter of depth-based weighted score: λ
Ensure: Anomaly Scores $\hat{Y}$ for all $v \in \mathcal{V}$
1: $\mathcal{S} \leftarrow$SEC($\mathcal{G}$, h); (Extract & Centralize Subgraphs)
2: **for** each $\mathcal{G}^h(v) \in \mathcal{S}$ **do**
3: $\quad \mathbf{e}_v = \phi(\mathcal{G}^h(v))$; (Embed each $\mathcal{G}^h$ to a vector)
4: **end for**
5: Let $\mathbf{E}$ be the matrix of embedded vectors $\mathbf{e}_v$ for all $v \in \mathcal{V}$
6: $Y \leftarrow$ Detector($\mathbf{E}$); (Score $v \in \mathcal{V}$ with a point detector)
7: Compute the reweighted scores $\hat{Y}$ via Eq. 4.4;
8: **return** $\hat{Y}$

subgraph centralization. Both the extraction and centralization are conducted in line#1 in Algorithm 1).

Then, an embedding method is required to generate an embedded representation for each h-subgraph (line#3 in Algorithm 1). An existing point anomaly detector can then be applied on the set of embedded vectors to detect node anomalies which are different from the majority in the set, as shown in line#6 in Algorithm 1. Since each h-subgraph is derived from a source node, the anomaly score of each h-subgraph denotes the anomaly score of its source node.

Finally, to derive the final score of a node u (line#6 in Algorithm 1), a weighted anomaly scoring method is used to aggregate the anomaly scores of source nodes of all h-subgraphs which contain node u.

We provide the details of the four steps in the following four subsections.

4.1 Subgraph Extraction and Centralization (SEC). Figure 2a provides an illustration of the subgraph extraction process. Given an h setting, extracting h-subgraphs from a given network is straightforward: each node in the network is treated as a source node u; and the h-subgraph with source node u is extracted via connected nodes up to depth h. The two 1-subgraphs are in the top-right corner in Figure 2b.

The aim, according to Definition 3, is to compare the structures of individual h-subgraphs which are determined from the relative positions of nodes in an h-subgraph, but independent of the absolute positions of the nodes in the node vector space. To achieve this aim, the subgraph centralization component maps all source nodes of h-subgraphs to the origin of the node vector space. To ensure that the structure of every h-subgraph remains unchanged, every node in an h-subgraph is translated to the same extent in which its source node is translated to the origin. This centralization nullifies the unwanted interference of the original positions in the node vector space, and enables only the structures of the h-subgraphs to be compared. The pro-

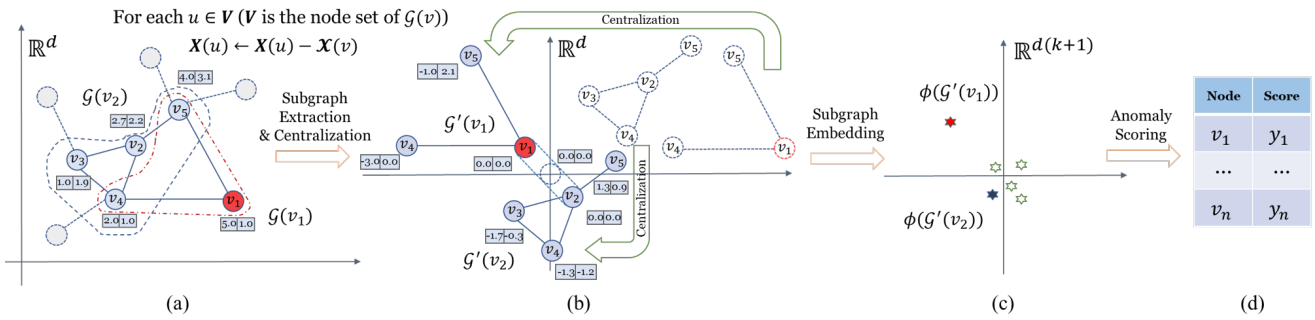


Figure 2: Overview of the GCAD framework. Only the first three components are shown here, i.e., h-subgraph extraction and centralization, subgraph embedding, and anomaly scoring. (a) The red node v_1 in the node vector space $\mathbb{R}^d$ is an anomalous node because it has connections with two distant nodes v_4 and v_5 while the other nodes are connected to neighbouring nodes only. Each dotted loop indicates an h-subgraph, extracted from the given network, where $h = 1$. (b) The two example 1-subgraphs are centralized such that their source nodes are translated to the origin. For clarity, the source nodes v_1 and v_2 are drawn besides the origin without overlapping (two parallel doted lines extended from the origin). Subfigure (c) shows the embeddings of the centralized subgraphs in the embedded $\mathbb{R}^{d(k+1)}$ space. $\phi(\mathcal{G}(v_1))$ is an outlier which is different from other points in this space.

cess is illustrated in Figure 2b. The detailed procedure of the above two processes is shown in Algorithm 2.

Algorithm 2 SEC

Require: Network: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$; subgraph depth: h
Ensure: Set of centralized h-subgraphs $\mathcal{S}$

```
Initialize S = {};
for each v ∈ V do
    Extract G^h(v) = (V, E, X);(subgraph extraction)
    for each u ∈ V do (V includes the source node v)
        X(u) ← X(u) − X(v); (Centralization)
    end for
    S ← S ∪ {G^h(v)};
end for
return S
```

Once the above process is completed, the majority of centralized h-subgraphs which are similar to each other constitute the normal h-subgraphs in a network. The few centralized h-subgraphs which are dissimilar to the normal ones are anomalies.

To facilitate such a comparison, we need to embed each h-subgraph into a vector, which is the topic of the next subsection. For brevity, we use $\mathcal{G}^h(v)$ to denote a centralized h-subgraph with source node v hereafter.

4.2 Subgraph Embedding. A subgraph embedding method ϕ to map each centralized h-subgraph $\mathcal{G}^h(v)$ into a vector $\phi(\mathcal{G}^h(v))$. We use the widely-used subgraph embedding method, Weisfeiler-Lehman (WL) [20, 25]. More specifically, for every h-subgraph $\mathcal{G}^h(v) = \{\mathbf{V}, \mathbf{E}, \mathbf{X}\}$, we utilize the WL scheme in the node vector space to get the embedding for every node of an h-subgraph. The key idea is to create an explicit propagation scheme that leverages and iteratively updates the current node vector by averaging over the node vectors in the neighbourhoods.

Let the node vector $\mathbf{X}^0(u) = \mathbf{X}(u) \in \mathbb{R}^d$ for each node $u \in \mathbf{V}$. The node vector of the kth iteration $\mathbf{X}^k(u)$, for $k \geq 1$, is computed via WL as follows:

(4.1)
$$\mathbf{X}^k(u) = \frac{1}{2}\left(\mathbf{X}^{k-1}(u) + \frac{1}{deg(u)} \sum_{w \in \mathbf{N}(u)} \mathbf{X}^{k-1}(w)\right),$$

where $deg(u)$ denotes the degree of node u, and $\mathbf{N}(u)$ is set of the one-hop neighbours of node u in the h-subgraph $\mathcal{G}^h(v)$.

Definition 4. *The WL-embedded vector of a node u in an h-subgraph $\mathcal{G}^h(v) = \{\mathbf{V}, \mathbf{E}, \mathbf{X}\}$ is defined as $\phi(u) \in \mathbb{R}^{d(k+1)}$:*

(4.2)
$$\phi(u) = [\mathbf{X}^0(u), \ldots, \mathbf{X}^k(u)]^\top.$$

Note that the concatenation of embedding from each iteration is different from other GNN-based methods in that GNN-based embedding does not concatenate outputs of different layers.

Definition 5. *The WL-embedded vector $\mathbf{e}_v$ of the entire h-subgraph $\mathcal{G}^h(v)$ is defined as the mean embedding of all nodes in the h-subgraph:*

(4.3)
$$\mathbf{e}_v = \phi(\mathcal{G}^h(v)) = \frac{1}{|\mathbf{V}|} \sum_{u \in \mathbf{V}} \phi(u).$$

Because the largest k for an h-subgraph is h. Therefore, we have set $k = h$.

4.3 Point Anomaly Detector. Given a set of embedded vectors, an existing unsupervised anomaly detector can then be used to detect anomalies that are different from the majority in the set.

Let $\mathbf{E}$ be the matrix of the embedded vectors $\mathbf{e}_v$. We assume that $Detector(\mathbf{E})$ is trained from $\mathbf{E}$ and produces output Y which is a one-dimensional matrix of anomaly scores y_v for all nodes v in the network. Any existing point anomaly detector can be used here.

4.4 Depth-based Weighted Score. The point anomaly detector in the last subsection produces a score y_v for node v which is the source node of h-subgraph $\mathcal{G}^h(v)$ for every node in a given network. The score can be used directly as the anomaly score of each node. But, we introduce a depth-based weighted score here for further improvement.

Let u be the node of interest; and $\mathcal{V}_u$ be the set of all source nodes of h-subgraphs which contain u. The depth-based weighted score aggregates scores from all h-subgraphs in $\mathcal{V}_u$. The idea is to place a larger weight (less than 1) to the score y_v of $\mathcal{G}^h(v)$ if u is closer to v.

The weight is formulated using a parameter λ as $\lambda^{\ell(v,u)}$, where $0 \leq \lambda < 1$; and $\ell(v,u)$ is the number of hops from v to u in $\mathcal{G}^h(v)$.

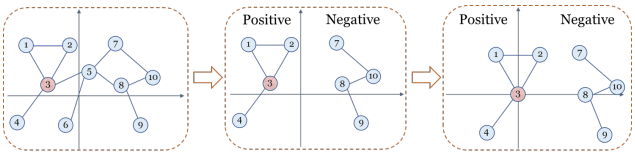
The final anomaly score $\hat{y}_u$ for node u is defined as:

$$\hat{y}_u = \frac{\sum_{v \in \mathcal{V}_u} \lambda^{\ell(v,u)} y_v}{\sum_{v \in \mathcal{V}_u} \lambda^{\ell(v,u)}}. \tag{4.4}$$

5 Applying Centralization to Two Existing Detectors

Here we show that Subgraph Centralization is applicable to an existing subgraph-based detector CoLA [15], and an existing full-graph-based detector OCGNN [22]. They are described in the next two subsections.

5.1 Centralize a Subgraph-based Detector - CoLA. Contrastive self-supervised Learning framework for Anomaly detection on attributed networks (CoLA) [15] is a recent state-of-the-art graph anomaly detection that uses contrastive learning of subgraphs to learn a classifier so that it can be used to produce an anomaly score for each node. We show below that our subgraph centralization can be easily plugged into CoLA to enhance its detection accuracy by performing Subgraph Centralization to CoLA's subgraphs before learning a classifier. Particularly, as shown in Figure 3, the new version called CoLA_SC takes three steps to obtain the centralized subgraphs. The first step is to traverse each node in random order as the target node within every epoch. Then, for a given target node, a positive subgraph and a negative subgraph are sampled via random walk with restart (RWR) [21]. The positive subgraph is generated from the target node; and the negative subgraph is from a node different from the target node. The first two steps are the same as the way in [15]. In the third step, both the positive and negative subgraphs are centralized as the way in Algorithm 2. After obtaining the centralized subgraphs, exactly the same training steps are used to learn a CoLA detector. The details can be found in [15].



(a) Full Graph (b) Sampled Subgraphs (c) Centralized Subgraphs

Figure 3: CoLA_SC performs Subgraph Centralization after the subgraphs have been sampled from a network. The red node denotes the target node in this example.

5.2 Centralize a Full-Graph-based Detector - OCGNN. One-class graph neural network (OCGNN) [22] integrates GNNs and one-class hypersphere learning to provide an end-to-end detector. The enhanced version with Subgraph Centralization is called OCGNN_SC. The additional functionalities are given as follows. Before training the end-to-end detector, random walk [21] is used to generate a subgraph for each node, and then each subgraph is centralized to the origin of the node vector space, as described before. An average pooling function is employed as the readout function for GNN, followed by the same hypersphere learning as in OCGNN.

6 Experimental Design and Results

The experiments are designed to answer the following questions:

i. Is Subgraph Centralization a necessary step in the problem of graph anomaly detection?

ii. How does Subgraph Centralization derive its explainability?

To answer the first question, we use the following baseline methods:

- **Oddball** [1] is a detector specially designed to detect the types of node anomalies which have clique and star. It learns a power-law model of a network to represent the normal neighbourhoods, and computes the deviation of each node from the expected value of the model as the anomaly score.

- **DOMINANT** [5] is a GCN-Autoencoder-based method, denoted as DOM. It applies two decoders to reconstruct the attribute and adjacency matrix. The anomaly score is computed by combining two reconstruction errors with a trade-off coefficient.
- **CoLA** [15] is a contrastive learning method that trains a GCN-based model to discriminate node anomalies from normal nodes in a network
- **OCGNN** [22] is a method utilizing GCN [12] and hypersphere learning [18] to detect anomalies.

In GCAD [1], we use the recent Isolation Distributional Kernel [19] (IDK for short) as the point anomaly detector in line#6 in Algorithm 1. We also examine alternative existing unsupervised anomaly detectors in an ablation study later. And other experimental settings are given in the Appendix [11].

	# Nodes	# Edges	# Attributes	# Anomalies
Watts-Strogatz	500	1500	2	24
SBM_{stru}	1000	5839	10	25
RGG_s	500	2195	2	20
RGG_l	500	60639	2	20
$Lattice_l$	1200	2340	2	20
$Lattice_s$	1220	2410	2	99
ACM	16484	82175	8337	597
Cora	2708	5803	1433	150
Citeseer	3327	5139	3703	150
Pubmed	19717	46424	500	600

Table 1: Data characteristics of datasets used

The investigation for the first question is reported in Section 6.1. The answer to the second question is presented in Section 6.2. Two additional experiments, that examine the effect of h in h-subgraphs and a scaleup test, are reported in the following two subsections.

6.1 Main Evaluation. This evaluation employs two sets of datasets, i.e., synthetic datasets and existing datasets. Their data characteristics are given in Table 2. They are briefly described as follows.

Synthetic datasets. We create six synthetic datasets using four classical graph generation models [10, 23, 4, 3]. For RGG and Lattice, we create two types of datasets, whose normal node in one is the anomalous node in the other. Such setting can help us evaluate the ability of the methods in detecting different types of anomalies apart from the type found in the real datasets. (See the Appendix [11] for details).

Existing datasets. Like previous work, the original citation network datasets, i.e., Cora, Citeseer, Pubmed and ACM, are assumed to have no anomalies and two types of anomalies are injected. They are the same as used in [15].

[1]Code is available at `https://github.com/IsolationKernel/Codes/tree/main/IDK/GraphAnomalyDetection`.

	Oddball	DOM	CoLA		OCGNN		GCAD
			Orig	SC	Orig	SC	
Watts-Strogatz	.63	.63	.72	**.82**	.24	.77	**.82**
SBM_{stru}	**1.00**	**1.00**	**1.00**	**1.00**	.21	**1.00**	**1.00**
RGG_s	.56	.70	.70	.84	.43	.86	**.91**
RGG_l	.99	.93	**1.00**	**1.00**	.40	.40	**1.00**
$Lattice_l$	.94	.95	.80	.83	.30	**1.00**	**1.00**
$Lattice_s$	**1.00**	.92	.65	.73	.55	.92	.96
ACM	.75	.81	.82	**.88**	.37	.79	.85
Cora	.77	.90	.88	.90	.24	.88	**.92**
Citeseer	.73	.93	.90	.92	.26	.93	**.95**
Pubmed	.74	.91	**.95**	**.95**	.37	.89	.92
Rank	4.8	3.85	3.85	2.95	6.95	3.7	1.85
P-value	.0076	.0038	.0086	.1020	.0010	.0059	-

Table 2: Detection accuracy in terms of AUC of different methods. 'Orig' denotes the original method; and 'SC' denotes a method is modified with Subgraph Centralization. The last row shows the p-value obtained from a pair-wise significance test in comparison with GCAD. The second last row shows the ranking of each method, average all datasets.

Overall results. Observations from Table 2 are:

- GCAD is the best. It performs the best in seven out of the ten datasets, and the second best in the other three exceptions. It is also significantly better than each of the other six contenders at the $p = 0.01$ significance level. The only exception is CoLA_SC which employs the Centralization.
- Subgraph Centralization significantly improves the performance of CoLA and OCGNN close to the level of GCAD. The improvement over OCGNN is huge on almost all datasets[2]. Similarly, the improvement over CoLA is on all datasets that have room for improvement, albeit the degree of improvement is smaller.

6.2 Explainability. Table 3 shows examples of anomalous and normal nodes identified by GCAD via h-subgraphs. This explanation via visualization is possible because of the use of Subgraph Centralization before the anomaly detection. The top two normal h-subgraphs indicate the typical h-subgraphs that exist in a network (ranked at the bottom in the ranked list).

[2]There is no improvement for OCGNN_SC on RGG_l owing to the fundamental limitation of SVDD, i.e., SVDD always regards points far away from the data centroid as anomalies. This dataset has anomalies at the data centroid and the normal points are further away from the data centroid.

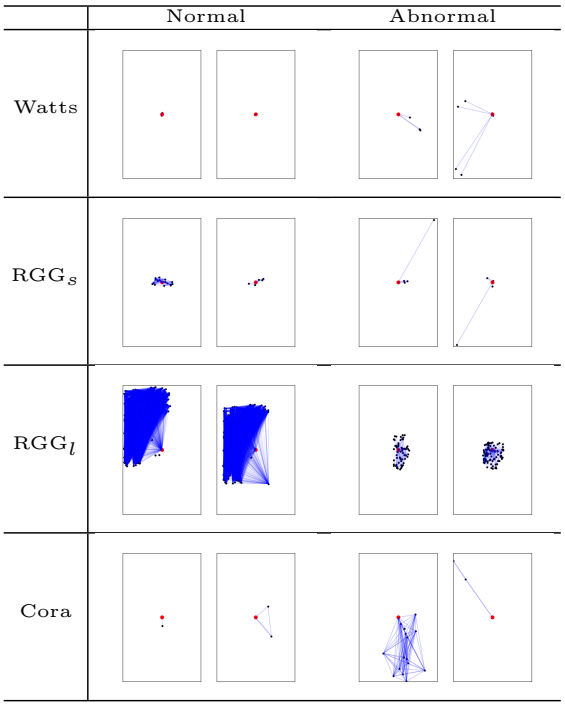


Table 3: Examples of anomalous and normal nodes identified by GCAD via h-subgraphs. Each red node denotes the source node of a centralized h-subgraph. These visualizations uses 1-subgraphs only. Note that many (normal) nodes of RGG$_l$ have many long connections and the few anomalous ones have short connections, unlike those in the other three datasets.

The top two anomalous h-subgraphs indicate the most unusual h-subgraphs in the network. The examples in Table 3 demonstrate that the h-subgraphs of the detected anomalies are significantly different from those normal h-subgraphs in each dataset.

Interestingly, *this explainability depends on the use of centralized subgraphs only*, and is independent of the point anomaly detector used in line#6 in Algorithm 1.

As none of the existing three detectors, i.e., CoLA, DOMINANT and OCGNN employ Subgraph Centralization for detection, they are unable to provide such a visualization to explain their detection outcomes. However, after utilizing Subgraph Centralization, CoLA_SC and OCGNN_SC have the same explainability.

6.3 The Effect of Parameter h on GCAD is examined here. GCAD performs the best when $h = 1$ on nine out of ten datasets and the only exception is RGG$_s$. This means that the simplest 1-subgraphs are sufficient to detect the anomalies on almost all the datasets. Figure 6 in the Appendix [11] shows the effect of h on different datasets.

6.4 Scaleup Test and Time Complexity. Figure 4 shows the result of a scaleup test for GCAD, DOMINANT and CoLA on Watts-Strogatz where the number of nodes is increased from 10^3 to 10^6. For a fair comparison, we measure the time each algorithm takes under the same environment.

The result shows that GCAD needs far fewer computations and can deal with large-scale million-node networks. DOMINANT and CoLA took a prohibitively long time when applied to the million-node network.

The time complexities of different components of GCAD are given in Table 4. GCAD has linear time complexity since $m \ll n$ and the other parameters are constant.

Note that the time complexities for CoLA and OCGNN do not include the deep learning time.

GCAD	Subgraph Extraction	$\mathcal{O}(nm)$
	Subgraph Centralization	$\mathcal{O}(nmf)$
	Subgraph Embedding (WL scheme [20])	$\mathcal{O}(feh)$
	Detector IDK [19]	$\mathcal{O}(nt\psi)$
	Total of GCAD	$\mathcal{O}(n(mf + t\psi) + feh)$
CoLA	Subgraph Generation	$\mathcal{O}(mnR(m + \delta))$
OCGNN	One GCN layer	$\mathcal{O}(pfu)$

Table 4: The time complexities of different components in GCAD. f and n denote the number of dimensions (of node vectors) and the number of nodes of a network. m and e denote the maximum numbers of nodes and edges, respectively, in a subgraph. R and δ denote the sampling rounds and the average degree in a network, respectively. p and u are the number of non-zero elements in adjacency matrix and the number of feature maps of the weight matrix, respectively.

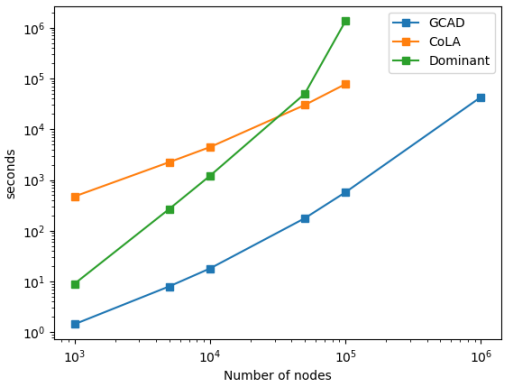


Figure 4: Scaleup test for GCAD, CoLA, and DOMINANT on the Watts-Strogatz dataset.

7 Ablation Studies on GCAD

In this section, we conduct two ablation studies. The first examines the effects of the two components in GCAD, i.e., Subgraph centralization and Depth-based

	IDK				iForest	OCSVM	LOF
Centralization	✗	✓	✗	✓	✓	✓	✓
Weighted score	✗	✗	✓	✓	✓	✓	✓
Watts-Strogatz	.57	**.82**	.66	**.82**	**.82**	**.82**	**.82**
SBM_{stru}	.58	**1.00**	.60	**1.00**	**1.00**	.99	**1.00**
RGG_s	.56	.90	.59	.91	.92	.93	**.95**
RGG_l	.88	**1.00**	.89	**1.00**	.51	.52	**1.00**
$Lattice_l$	.57	**1.00**	.60	**1.00**	**1.00**	**1.00**	.98
$Lattice_s$	.50	.96	.51	.96	**.98**	.96	.97
ACM	.73	.82	.76	**.85**	.71	**.85**	.83
Cora	.64	.90	.64	**.92**	.85	**.92**	**.92**
Citeseer	.59	.94	.59	**.95**	.81	**.95**	**.95**
Pubmed	.69	.89	.74	.92	.82	**.94**	.87
Rank	6.6	3.3	5.8	2.5	4.1	2.85	2.85
P-value	.0010	.0206	.0010	-	.0315	.5	.1238

Table 5: Results of two ablation studies on GCAD w.r.t. subgraph centralization, weighted score and point anomaly detectors in terms of AUC.

weighted score. The second investigates the generic nature of the GCAD framework by using different existing point anomaly detectors in line#6 in Algorithm 1. The unsupervised point anomaly detectors investigated are Isolation Forest (iForest) [14], OCSVM [17], and Local Outlier Factor (LOF) [2] and IDK [19] (we have used IDK as the default in the previous sections).

The results of the first study are shown in the second to the fifth columns in Table 5; and the results of the second study are shown in the fifth to the last columns.

We summarize the outcomes of these studies as:

- **Subgraph centralization** is a crucial component of GCAD. The accuracies of GCAD decline significantly on all the datasets without the component.

- **Weighted score**. The weighted score component provides a small AUC improvement on some datasets. It is useful especially when anomalous nodes are connected such as cliques. Through the depth-based weighted score, an anomalous node increases the anomaly scores of its anomalous neighbours. Therefore, all the anomalous nodes are ranked higher than without the weighted score.

- **Point anomaly detectors**. The results in the last three columns in Table 5 show that IDK performs better than iForest at $p = 0.05$ significance level. Overall, IDK has comparable accuracy as OCSVM and LOF. OCSVM performs poorly on RGG_l for the same reason stated in footnote 2 wrt OCGNN in Section 6.1 because they make the same assumption. The example runtimes on the Pubmed dataset of GCADs with IDK, iForest, OCSVM and LOF are 23, 40, 588 and 583 seconds, respectively. IDK is the recommended point anomaly detector at this point in time because it has the highest accuracy and run fastest.

8 Discussion

Anomalous Node Definitions. The importance of the definition of anomalous nodes can not be underestimated. Without a clear definition, how an anomaly relates to the normal ones is unclear. We cannot find such a definition in all the current works listed in the related work section. This has two implications. First, the analysis of the reason why (or the condition under which) a detector work or not becomes difficult, if not impossible. Second, anomalies are often implicitly/explicitly assumed to belong to a very restricted kind in existing works. For example, a node anomaly is assumed to be far away from an existing node in a network [5, 6]. This assumption is made on the basis that normal nodes are close to each other.

One crucial ingredient is missing in the above assumption, i.e., anomalies are rare and normal ones are plentiful in a network. It is possible that a network may contain two (or more) types of normal nodes in different parts. For example, one part has many subgraphs with close neighbouring nodes, and the other has many subgraphs with far away neighbouring nodes. As a result, anomalies in one part becomes normal ones in the other. Using the assumption misses all the anomalies in one part of the network, as we have identified with OCGNN in Section 6.1.

Note that the GCAD framework is generic, not coupled with any particular detector (as reported in Section 7). It is also not tied to any specific similarity measure, and it even admits no measure.

Anomaly Types in a Graph. We have identified that the commonly used datasets are likely to belong to one type of anomaly, where only relative positions of node vectors matter. That is why the proposed method works better than existing methods which do not employ subgraph centralization. It is possible that there are other types of graphs in which the proposed method does not work. Two possible example types are that anomalous nodes are due to (a) node vectors only but not graph structure, and (b) the absolute positions of node vectors play an important role in addition to graph structure. The former is not a graph problem and the node anomalies can be identified with a point anomaly detector. The latter is an open problem.

9 Conclusion

We show that Subgraph Centralization is a necessary technique in graph anomaly detection that has a significant influence on detection accuracy, and it empowers

any detectors to gain explainability that would otherwise be impossible. As this technique has time complexity linear to the number of nodes, it enables a detector to deal with large scale datasets. These advantages are shown in our proposed framework GCAD that incorporates Subgraph Centralization. Our empirical evaluation verifies that (i) GCAD is superior to four existing detectors in terms of its detecting accuracy, run time and scalability; and (ii) two existing detectors CoLA and OCGNN can reap at least two out of the three above-mentioned benefits of Subgraph Centralization if it is integrated into their algorithms.

Acknowledgement

Kai Ming Ting is supported by National Natural Science Foundation of China (62076120). Guansong Pang is supported in part by the Singapore Ministry of Education (MoE) Academic Research Fund (AcRF) Tier 1 grant (21SISSMU031).

References

[1] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. OddBall: Spotting anomalies in weighted graphs. In *KDD*, 410–421, 2010.

[2] Markus Breunig, Hans-Peter Kriegel, Raymond Ng, and Joerg Sander. LOF: Identifying density-based local outliers. In *ACM Sigmod Record*, 93–104, 2000.

[3] J. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*. Grundlehren der mathematischen Wissenschaften. Springer New York, 1998.

[4] Jesper Dall and Michael Christensen. Random geometric graphs. *Physical Review E*, 66(1), Jul 2002.

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

[6] Kay Liu, Yingtong Dou, Yue Zhao, Xueying Ding, Xiyang Hu, Ruitong Zhang, Kaize Ding, Canyu Chen, Hao Peng, Kai Shu, Lichao Sun, Jundong Li, George H. Chen, Zhihao Jia, and Philip S. Yu. BOND: Benchmarking Unsupervised Outlier Node Detection on Static Attributed Graphs In *NeurIPS Datasets and Benchmarks Track*, 2022.

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

[8] Xiao Huang, Qingquan Song, Jundong Li, and Xia Hu. Exploring expert cognition for attributed network embedding. In *WSDM*, 270–278, 2018.

[9] Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan. Anemone: Graph anomaly detection with multi-scale contrastive learning. In *CIKM*, 3122–3126, 2021.

[10] Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1), Jan 2011.

[11] Zhong Zhuang, Kai Ming Ting, Guansong Pang and Shuaibin Song. Subgraph Centralization: A Necessary Step for Graph Anomaly Detection. 2023. [Online]. Available: https://arxiv.org/abs/2301.06794.

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

[13] Yuening Li, Xiao Huang, Jundong Li, Mengnan Du, and Na Zou. SpecAE: Spectral autoencoder for anomaly detection in attributed networks. In *CIKM*, 2233–2236, 2019.

[14] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *IEEE ICDM*, 413–422, 2008.

[15] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15, 2021.

[16] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. In *Annual Review of Sociology*, 415–444, 2001.

[17] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. In *NeurIPS*, 582–588, 1999.

[18] David Tax and Robert Duin. Support vector data description. *Machine Learning*, 54:45–66, 01 2004.

[19] Kai Ming Ting, Bi-Cun Xu, Takashi Washio, and Zhi-Hua Zhou. Isolation distributional kernel: A new tool for kernel based anomaly detection. In *KDD*, 198–206, 2020.

[20] Matteo Togninalli, Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten Borgwardt. Wasserstein Weisfeiler–Lehman graph kernels. In *NeurIPS*, 6436–6446, 2019.

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

[22] Xuhong Wang, Baihong Jin, Ying Du, Ping Cui, Yingshui Tan, and Yupu Yang. One-class graph neural networks for anomaly detection in attributed networks. *Neural Computing and Applications*, 33(18):12073–12085, 2021.

[23] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[24] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[25] Bi-Cun Xu, Kai Ming Ting, and Yuan Jiang. Isolation graph kernel. *AAAI*, 35(12):10487–10495, May 2021.

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

[27] Yu Zheng and Ming Jin and Yixin Liu and Lianhua Chi and Khoa T. Phan and Yi-Ping Phoebe Chen. Generative and contrastive self-supervised learning for graph anomaly detection. *IEEE TKDE*, 2021.