Anomaly Subgraph Detection Through High-Order Sampling Contrastive Learning

Ying Sun[1,2], **Wenjun Wang**[1,3*], **Nannan Wu**[1*] and **Chunlong Bao**[2]
[1]College of Intelligence and Computing, Tianjin University, China
[2]School of Digital and Intelligent Industry, Inner Mongolia University of Science and Technology, China
[3]Yazhou Bay Innovation Institute, Hainan Tropical Ocean University, China
{yingsun, wjwang, nannan.wu}@tju.edu.cn, baochunlong0@gmail.com

Abstract

Anomaly subgraph detection is a crucial task in various real-world applications, including identifying high-risk areas, detecting river pollution, and monitoring disease outbreaks. Early traditional graph-based methods can obtain high-precision detection results in scenes with small-scale graphs and obvious anomaly features. Most existing anomaly detection methods based on deep learning primarily concentrate on identifying anomalies at the node level, while neglecting to detect anomaly groups in the internal structure. In this paper, we propose a novel end-to-end Graph Neural Network (GNN) based anomaly subgraph detection approach(ASD-HC) in graph-structured data. 1)We propose a high-order neighborhood sampling strategy to construct our node & k-order neighbor-subgraph instance pairs. 2)Anomaly features of nodes are captured through a self-supervised contrastive learning model. 3) Detecting the maximum connected anomaly subgraph is performed by integrating the Non-parameter Graph Scan statistics and a Random Walk module. We evaluate ASD-HC against five state-of-the-art baselines using five benchmark datasets. ASD-HC outperforms the baselines by over 13.01% in AUC score. Various experiments demonstrate that our approach effectively detects anomaly subgraphs within large-scale graphs.

1 Introduction

Anomaly Detection (AD) is a vital application in real-world datasets. AD typically involves two primary tasks: Anomalous Node Detection (AND) and Anomaly Subgraph Detection (ASD). Existing AD approaches are usually categorized into two main groups: graph-based and deep learning-based approaches. Early traditional graph-based ASD approaches can obtain high-precision detection results in scenes with small-scale graphs and obvious anomaly features[Akoglu *et al.*, 2015]. As the demand for large-scale data applications grows and AI technology advances, numerous deep learning-based AD approaches continue to be proposed. These approaches learn node representations through deep and shallow machine-learning models. Some of them project the node embeddings into Euclidean or hyperbolic space and identify outliers based on their distances. Others define anomaly-measuring functions to calculate anomaly scores for all nodes, with the top-K anomalous nodes identified as the detection result[Pang *et al.*, 2021]. In summary, most recent AD approaches based on deep learning techniques primarily focus on identifying anomalies at the node level rather than detecting anomalous subgraphs with their internal structures.

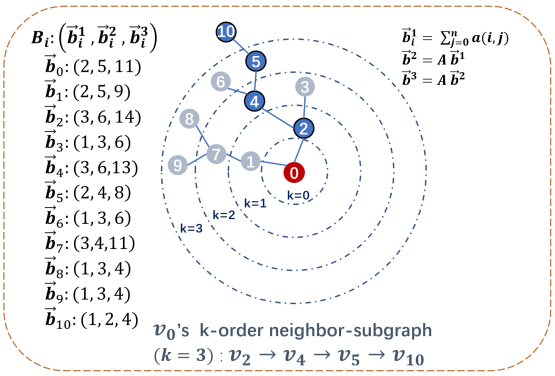

Figure 1: A toy example illustrates our sampling strategy for the k-order neighbor-subgraph. $A \in \mathbb{R}^{n \times n}$ is the adjacency matrix and B_i denotes the numbers of the high-order neighbors of node v_i. With $k = 3$, our k-order neighbor-subgraph sampling procedure selects node v_0's neighbor v_2 based on $max(\vec{\mathbf{b}}_1^3, \vec{\mathbf{b}}_2^3)$. Subsequently, node v_4 is chosen based on $max(\vec{\mathbf{b}}_3^3, \vec{\mathbf{b}}_4^3)$.

Graph anomalies often appear as structural anomalies and contextual anomalies. A structural anomaly refers to an anomalous pattern or behavior in the overall structure of a system or dataset. It involves deviations from the expected relationships, connections, or configurations within the data. A contextual anomaly refers to an anomalous behavior or pattern that is considered unusual within a specific context or environment, such as a specific time period. Anomaly subgraphs are clusters of nodes exhibiting anomaly features along with associated structures. Typically, an anomaly subgraph consists mostly of anomalous nodes, with only a minority being normal in it. However, detecting anomaly subgraphs faces multiple challenges: (1) The rarity of anoma-

*Corresponding authors: Nannan Wu, Wenjun Wang

lous nodes in large graphs may result in ineffective representations of anomaly features for subgraphs. (2) The challenge of deriving anomaly features for subgraphs leads existing deep learning-based AD approaches to neglect the overall anomalous attributes of interconnected nodes (i.e., subgraphs) within graphs. (3) The limited capacity to represent subgraphs within existing deep-learning models hampers the development of an end-to-end ASD algorithm. (4) Detecting subgraph is an optimization problem with exponential growth, posing a significant challenge in terms of time cost. With the expansion of the graph's size, the number of all possible subgraphs increases exponentially, leading to a significant rise in computational costs.

Our paper aims to detect the maximum connected anomaly subgraph in a graph and proposes a novel end-to-end ASD approach. Initially, we investigate a high-order sampling strategy to build node-subgraph instance pairs for contrastive learning, as shown in Fig.1. Next, we utilize a GNN module to capture the representation embeddings of instance pairs within the graph. Additionally, we apply a self-supervised contrastive learning method to reveal the anomalous characteristics of nodes and subsequently rank them based on their anomaly scores. Furthermore, we establish an evaluation function utilizing Non-Parametric Scan Statistics (NPSS) to gauge the level of anomalous significance in a subgraph. Finally, we employ a suboptimal procedure to detect the maximum anomaly subgraph within the graph.

The main contributions of this paper are as follows.

- We propose a novel end-to-end algorithm ASD-HC designed for detecting anomaly subgraphs. ASD-HC employs a novel high-order neighborhood sampling and contrastive learning method to extract the anomaly features of nodes within the graph comprehensively.
- We present a suboptimal detecting anomaly subgraph method with non-parameter scan statistics assessing function, effectively reducing the run-time cost.
- Our approach is evaluated against five state-of-the-art deep learning-based anomaly detection algorithms on five real-world benchmark datasets, confirming its effectiveness and accuracy.

2 Related Works

Traditional graph-based approaches achieve detection capabilities by extracting features from graph data[Wu *et al.*, 2019; Sun *et al.*, 2020; Sun *et al.*, 2022]. However, they encounter limitations in scalability due to the constraints imposed by the size of the graphs on their algorithms. Most, if not all, recent anomaly detection approaches are based on deep learning techniques, with a predominant focus on anomalous node detection(AND). There are also a few approaches that tackle anomaly subgraph detection(ASD).

Contrastive learning-based AND: Contrastive learning, a self-supervised approach drawing insights from unlabeled data, has become popular in anomaly detection. It eliminates the need for manual labeling, reducing associated costs. [Jin *et al.*, 2021] estimates node anomalous scores through the contrast of node & node and node& ego-net multi-scale instance pairs. In [Liu *et al.*, 2022], contrastive learning is applied to capture anomaly features and calculate the anomaly scores of nodes. Its instance pairs consist of the node and its K-hop neighborhood which is constructed through a random walk procedure. Another approach is presented by [Hu *et al.*, 2023], which detects anomalous nodes via the subgraph-aligned contrastive learning across multiple views of the graph. [Duan *et al.*, 2023] calculates the anomaly scores of nodes via contrastive learning among node-node,node-subgraph, and subgraph-subgraph multi-scale instance pairs between the original view and the augmentation view.

Deep learning-based ASD: Recently, several deep learning-based ASD approaches have been proposed for different scenarios. For instance, [Wu *et al.*, 2022] focuses on detecting anomaly subgraphs in multiple attributed networks within the federated learning framework. [Zhang *et al.*, 2023] employs the k-hop Breadth-First Search strategy and Triadic closure to identify anomaly subgraphs in dynamic graphs. [Huang *et al.*, 2023] proposes a Hybird-Order GAT model to detect anomalous nodes and anomalous motif instances within an attributed network. [Zhang and Zhao, 2022] employs a location-aware graph autoencoder to reconstruct the given graph and identify anomalous areas through a residual graph, which reveals disparities between the original and the reconstructed graphs. Hence, [Zhang and Zhao, 2022] is the only one among the above approaches, which detects the anomaly subgraph in scenarios most similar to our approach.

3 Preliminaries

In this paper, we first define an attributed graph as $G = (V, E, \mathbf{A}, \mathbf{X})$, where V and E are the node and edge sets, $\mathbf{A} \in \mathbb{R}^{n\times n}$ denotes the adjacency matrix of G, $\mathbf{X} \in \mathbb{R}^{n\times m}$ is the observed m dimensional attribute sets of nodes in G, $E \subseteq V \times V$, $n = |V|$ denotes the number of nodes ($n \in \mathbb{R}$), $S = (V_S, E_S, \mathbf{A_S}, \mathbf{X_S})$ is a connected subgraph of G, where $V_s \subseteq V$ and $E_s \subseteq E$.

Definition 3.1 (**Anomaly subgraph detection).** Given $G = (V, E, \mathbf{A}, \mathbf{X})$ and $S = (V_S, E_S, \mathbf{A_S}, \mathbf{X_S})$, $S \subseteq G$ is a connected subgraph of an attributed graph G, the function $f(S) \in \mathbb{R}$ measures the significantly anomalous lever of a subgraph S. If $f(S) \geq K$ (K is a constant), then S is defined as "Anomaly Subgraph".

To evaluate the significantly anomalous lever of a subgraph within a graph, we introduce the Higher Criticism(HC) statistic to define our objective function $f(S)$. Notably, HC is one of the Non-Parametric Graph Scan statistics(NPGS) that have been utilized in road congestion detection, event detection, and other real-world applications. Our approach can also be generalized to other NPGSs.

Definition 3.2 (**Maximum anomaly subgraph).** Given the significant level of anomaly feature α (e.g.,0.25), the maximum anomaly subgraph of the graph G, denoted as $\hat{S}$, is defined as follows:

$$\hat{S} = \underset{S\subseteq G}{\arg\max}\, f\big(\alpha, N_\alpha(S), N(S)\big) = \frac{N_\alpha(S) - N(S)\alpha}{\sqrt{N(S)\alpha(1-\alpha)}}. \quad (1)$$

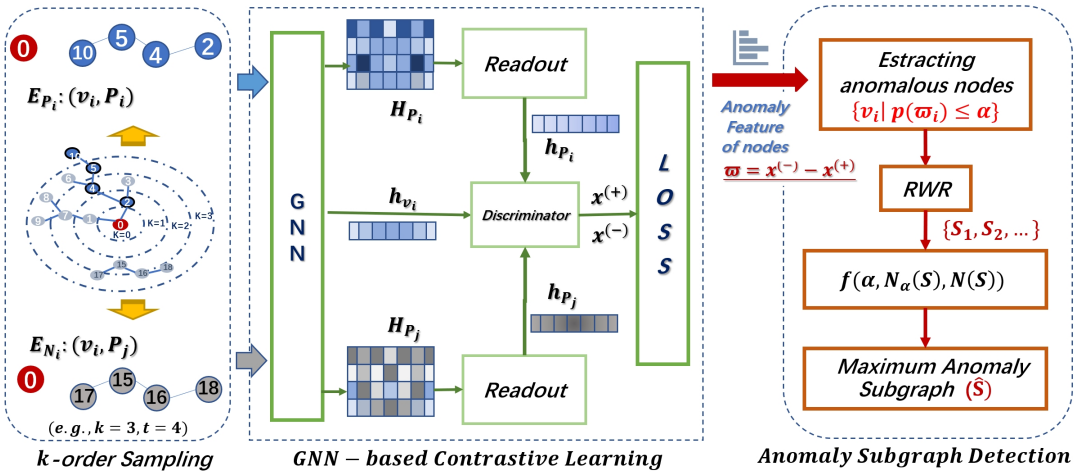

Figure 2: Illustration of our ASD-HC approach for maximum anomaly subgraph detection. ASD-HC comprises three components: (1) Preprocessing through k-order neighborhood sampling; (2) GNN-based contrastive learning; (3) Identification of the maximum anomaly subgraph in the provided graph, relying on the anomalous nodes with the significant anomaly features. $p(\varpi_i)$ is the p-value that the null hypothesis is invalid for the gap between positive and negative samples following a uniform distribution.

where $N_\alpha(S) = \Sigma_{v_i \in V_S} \delta(p(\varpi_i) \leq \alpha)$ denotes the number of anomalous nodes of S, ϖ_i is the anomaly score of node v_i, and $p(\varpi_i)$ denotes the p-value associated with ϖ_i. As for the function $\delta(\cdot) = 1$ if its input is True, otherwise $\delta(\cdot) = 0$. $N(S) = \Sigma_{v_i \in V_S} \delta(1)$ is the total number of nodes in S.

4 Methodology

4.1 High-Order Neighborhood Sampling Strategy

This paper proposes a high-order neighborhood sampling method to find the k-order neighbor subgraph of each node. Firstly, the high-order neighborhood weight of the graph is represented by matrix $\mathbf{B} = (\vec{\mathbf{b}}^{(1)}, \vec{\mathbf{b}}^{(2)}, .., \vec{\mathbf{b}}^{(k)}) \in \mathbb{R}^{n \times k}$, where vector $\vec{\mathbf{b}}^{(k)} \in \mathbb{R}^n$ denotes the number of k-order neighbors for each node. Let $\vec{\mathbf{b}}_i^{(1)} = \sum_{j=1}^{n} a(i,j), 1 \leq i \leq n$, where $a(i,j)$ denote the elements of the adjacency matrix $\mathbf{A}$. Secondly, the k-order neighborhoods of nodes are calculated as the following equation:

$$\vec{\mathbf{b}}^{(k)} = \mathbf{A}\vec{\mathbf{b}}^{(k-1)}, \tag{2}$$

where k is a hyperparameter. $\vec{\mathbf{b}}^{(k)}$ is also represented as a column vector, where each element quantifies the influence of the k-order neighborhood of each node. Therefore, the k-order maximum influential neighbor $\hat{v_i}$ of the current node v_i is defined in Eq.(3).

$$\hat{v_i} = \underset{j \in N_i}{\arg\max}\ \vec{\mathbf{b}}_j^{(k)}, \tag{3}$$

where N_i denotes the neighbors of the node v_i. Our high-order neighborhood sampling strategy sets $\hat{v_i}$ as the next node of v_i in the k-order neighbor subgraph.

4.2 Contrastive Learning of Anomaly Features

To enhance the efficiency of our algorithm, we initially preprocess the data and subsequently generate the candidate sets of positive and negative samples, i.e., positive and negative neighbor subgraphs, respectively. In our paper, we extract k-order **maximal/minimal neighbor-subgraphs** from the high-order neighborhood to construct contrastive instance pairs. E_{P_i} denotes the positive instance pair comprising a target node v_i and its corresponding *k-order maximal neighbor-subgraph*, denoted as P_i (initiated with v_i). On the other hand, the negative instance pair E_{N_i} consists of v_i and the *k-order minimal neighbor-subgraph* derived from another node. They are defined as follows:

$$E_{P_i} = (v_i, P_i), \qquad E_{N_i} = (v_i, P_j), \quad s.t. \quad i \neq j. \tag{4}$$

To facilitate the comparison of nodes and subgraphs in positive and negative instance pairs, it is essential to execute the training process for nodes and subgraphs within the same parameter space.

1). Node Embedding: Our approach trains the node embeddings using the weight matrix W and the activate function $\sigma(\cdot)$, consistent with the GCN module, the input is the attribute of nodes without the structural data. Therefore, the module is similar to the MLP:

$$h_{v_i}^{(l)} = \sigma(h_{v_i}^{(l-1)} W^{(l-1)}) \tag{5}$$

where $h_{v_i}^{(0)} = \mathbf{X}(i)$ is the attribute vector of node v_i, i.e., the i-th row of the matrix $\mathbf{X}$ associated with the graph G.

2). k-Order Neighbor-subgraph Embedding: As the GNN[Scarselli *et al.*, 2009] method is widely used for processing graph data, our paper leverages this model to learn embeddings of nodes in the k-order neighbor-subgraph from the graph's structure and attribute information. In this paper, we conduct experiments using the GCN [Kipf and Welling, 2017] model. Additionally, the GCN module can be substituted with other GNN models, such as GAT[Veličković *et al.*, 2018], GraphSAGE[Hamilton *et al.*, 2018], etc. For an attributed graph $G = (V, E, A, X)$, our GCN module calculates the following iteration equation:

$$H^{(l)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l-1)} W^{(l-1)}) \tag{6}$$

where $\tilde{A} = A + I$, I is an identity matrix, $\tilde{D} = D + I$, D is the degree matrix of graph G, $\sigma(\cdot)$ is an activation function(e.g. ReLU), $W^{(l)}$ is the l-layer trainable parameter matrix, $H^{(0)} = X$ is the initial input matrix. Therefore, the k-order neighbor-subgraph P_i, being a set of connected nodes, undergoes its representation learning through the GCN module, utilizing the adjacency matrix A_{P_i} and the attribute matrix X_{P_i}, then Eq.(6) is rewritten as the following equation:

$$H_{P_i}^{(l)} = \sigma(\tilde{D}_{P_i}^{-\frac{1}{2}} \tilde{A}_{P_i} \tilde{D}_{P_i}^{-\frac{1}{2}} H_{P_i}^{(l-1)} W^{(l-1)}) \tag{7}$$

For computational convenience, a readout layer follows the output of Eq.(7) to transform the matrix into a vector.

$$h_{P_i} = Readout(H_{P_i}) = \frac{1}{t} \sum_{j=1}^{t} H_{P_i}(j) \tag{8}$$

where $t = |P_i|$ is the number of nodes in the k-order neighbor-subgraph P_i, $H_{P_i}(j)$ is the j-th row of matrix H_{P_i}.

3). Loss of Contrastive Learning Module We employ the Bilinear function to evaluate the relationships x between node embeddings h_v and k-order neighbor-subgraph embeddings h_P, and the relationships of the positive and negative instance pairs $x^{(+)}$, $x^{(-)}$ are denoted by Eq.(9), respectively.

$$\begin{aligned} x_i^{(+)} &= Bilinear(h_{v_i}, h_{P_i}); \\ x_i^{(-)} &= Bilinear(h_{v_i}, h_{P_j}), \quad s.t. \quad i \neq j. \end{aligned} \tag{9}$$

The loss function in our contrastive learning module utilizes the Binary Cross-Entropy Loss function in conjunction with a Sigmoid layer, shown in Eq.(10).

$$\mathfrak{L} = -\sum_{i=1}^{n_B} (y_i \log(\sigma(x_i)) + (1 - y_i) \log(1 - \sigma(x_i))) \tag{10}$$

where n_B is the batch size, $\sigma()$ denotes the Sigmoid function, y represents the labels of positive(E_P) and negative(E_N) instance pairs, defined as follows.

$$y_i = \begin{cases} 1, & v_i \in E_{P_i} \\ 0, & v_i \in E_{N_i} \end{cases} \tag{11}$$

4). Anomaly Score Function: The intuitive concept of contrastive learning is to train the correlation value of a node's positive instance pair, $x^{(+)}$, to approach 1, and the node's negative instance pair, $x^{(-)}$, to approach 0. The majority of nodes in the graph are expected to be normal and adhere to the rule stated above. However, for the anomalous node, the values of $x^{(+)}$ and $x^{(-)}$ deviate from this expected pattern. Like the other contrastive learning model, we introduce the anomaly score function (shown in Eq.(12)) to derive the anomaly features for nodes in the graph.

$$\varpi_i = x_i^{(-)} - x_i^{(+)}, \tag{12}$$

where i denotes the node v_i. When v_i is a normal node, it can be inferred from the above equation that ϖ_i is a negative number, ideally set to -1. Conversely, v_i is an anomalous node, ϖ_i will be greater than -1. The greater the anomaly of the node, the farther the value of ϖ_i deviates from -1. The vector ϖ represents the anomaly features of each node learned through our approach.

Algorithm 1 Anomaly Subgraph Detection With High-Order Neighborhood Sampling Contrastive Learning (ASD-HC)

Input: $G = (V, E, A, X)$, $\mathbf{k}$ (*k-order*), $\mathbf{t}$ (*the k-order neighbor-subgraph size*), E (*number of epoch*), n_B (*the batch size*), the anomalous threshold α
Output: Maximal connected anomaly subgraph $\hat{S}$

```
Randomly initializing the parameters of model W;
Initialization: b^(1) = Σ a(i,j),  i,j ≤ |V|;   e ← 0;
Calculate the k-order vector b^(k) = A^(k-1) b^(1);
for all node i ∈ V do
    Generating positive subgraphs {P_i} with max(b_i^(k));
    Generating negative subgraphs {N_i} with min(b_i^(k));
end for
while e ≤ E do
    Selecting the batch set {v_0, v_1, ..., v_{n_B}} ∈ V
    for v_i ∈ {v_0, v_1, ..., v_{n_B}} training do
        E_P ← (v_i, P_i);  E_N ← (v_i, P_j),  s.t. i ≠ j;
        Embedding by GCN : h_{v_i}, h_{P_i} as Eq.(5), (8);
        Quantifying E_{P_i} and E_{N_i}: x_i^(+), x_i^(-) as Eq.(9);
        Defining labels of E_{P_i} and E_{N_i}: y_i^(+) ← 1,   y_i^(-) ← 0;
        Training W as loss: 𝔏 ← BCEWithLogitsLoss(x, y);
    end for
end while
for all nodes testing do
    Calculating anomaly score ϖ_i ← x_i^(-) − x_i^(+), v_i ∈ V;
end for
Filtering out anomalous nodes: Ω ← {v_i | p(ϖ_i) ≤ α}
RWR start with Ω in G: → {S_0, S_1, S_2, ...}
while S ∈ {S_0, S_1, S_2, ...} do
    Ŝ ← arg max f(S)
end while
return Ŝ;
```

4.3 Anomaly Subgraph Detection Algorithm

To create an efficient algorithm for detecting connected anomaly subgraphs, we adopt the following steps:

1 Assuming that the ϖ values follow a uniform distribution, then calculate the probability of ϖ, i.e. $p(\varpi)$.

2 Filtering out the anomalous nodes with $p(\varpi)$ values less than a specified threshold α.

3 Constructing connected anomaly subgraphs within a larger graph based on the filtered anomalous nodes via the Random Walk with Restart (RWR) algorithm.

4 Assessing the candidate connected anomaly subgraphs with NPGS in Eq.(1).

5 Iterative optimization processing.

Based on hypothesis testing and statistical theory, if the null hypothesis is valid and the distribution of ϖ should theoretically be uniform, the p-values are expected to follow a normal distribution within the interval [0, 1]. The p-value of the observed data is relatively low under the null hypothesis, which means that the observed data represent an even more extreme scenario. This provides evidence for rejecting the null hypothesis. Consequently, in our paper, we establish a predefined threshold α (e.g., $\alpha = 0.25$), then when the probability of a node's anomalous score ϖ_i, represented as $p(\varpi_i)$, is less than or equal to α, it indicates that the node v_i has a

Dataset	#nodes	#edges	#attributes	#anomalies
Cora	2,708	5,803	1,433	150
ACM	16,484	90,557	8,337	597
Citeseer	3,327	5,198	3,703	150
BlogCatalog	5,196	172,783	8,189	300
Flickr	7,575	241,304	12,407	450

Table 1: Statistics of datasets involved in our experiments.

high likelihood of being an anomalous node. A collection of anomalous nodes Ω is filtered based on the condition of $p(\varpi_i) \leq \alpha$.

As we know, the computational complexity of graph detection increases exponentially with larger graph sizes. The computational cost is one of the challenges of the graph detection process. Our approach employs the RWR algorithm [Tong *et al.*, 2006] to construct candidate connected anomaly subgraphs. The induced subgraphs are constructed around a set of anomalous nodes $\Omega = \{v_i | p(v_i) \leq \alpha\}$ but are not confined solely to these anomalous nodes. Therefore, this strategy addresses the challenges associated with traditional anomaly subgraph detection algorithms and significantly mitigates the computational demands for subgraph identification in large graphs.

5 Experiments

Our approach and baselines are all evaluated on the five benchmark datasets, as shown in Table 1. We explore all experiments on the GPU: GeForce GTX 1080 Ti (11GB) $\times 2$, with a total memory of 128GB.

5.1 Dataset

Our paper leverages five widely recognized datasets to assess the performance of our algorithm and baselines. These datasets are commonly employed in other deep learning-based approaches[Zheng *et al.*, 2023; Liu *et al.*, 2022; Jin *et al.*, 2021], and their key information is outlined in Table 1. These datasets are categorized into two types:

Citation Networks: Cora[1][McCallum *et al.*, 2000], CiteSeer[2][Lawrence *et al.*, 1999], ACM[Sen *et al.*, 2008][Tang *et al.*, 2008]. In these graphs, nodes represent corresponding documents, edges denote links between those documents, and their node features are extracted from the text contents of each document.

Social Networks: BlogCatalog[3] and Flickr[Tang and Liu, 2009]. BlogCatalog data is extracted from the blog-sharing website BlogCatalog, while Flickr is sourced from the image-sharing platform Flickr. In these two graphs, nodes represent users of the respective websites, and edges signify the relationships between users. Additionally, the features of nodes are derived from the tag contents of the posted blogs and photos, respectively.

[1]www.cora.justresearch.com

[2]www.scienceindex.com

[3]http://www.blogcatalog.com

Algorithms	Cora	ACM	Citeseer	Flickr	BlogCatalog
DOMINANT	0.8124	0.7986	0.8267	0.7454	0.6465
CoLA	0.8942	0.8322	0.8786	0.7468	**0.7800**
ANEMONE	0.8975	0.8398	**0.9137**	0.6620	0.6417
GRADATE	0.8421	0.8231	0.8079	0.7181	0.6058
AS-GAE	0.6786	0.5019	0.6730	0.5021	0.4947
ASD-HC	**0.9217**	**0.8786**	0.9057	**0.7719**	**0.7800**

Table 2: AUC evaluation of anomalous node detection by ASD-HC and its competitive baselines across five benchmark datasets.

5.2 Experiment Setting

Metrics: We use the AUC metric to evaluate both anomalous node detection and anomaly subgraph detection, which is commonly used to evaluate the performance of a binary classification model. AUC quantifies the model's ability to distinguish between positive and negative instances. A higher AUC value, closer to 1, indicates better performance.

Baselines: Our experiments explore five state-of-the-art deep learning-based baselines for anomalous node detection and one method (AS-GAE) for anomaly subgraph detection.

1) **DOMINANT**[Ding *et al.*, 2019] employs a deep graph autoencoder method and utilizes graph structure and features for detecting anomalous nodes in a graph.
2) **ANEMONE**[Jin *et al.*, 2021] is an anomalous node detection method based on Graph Neural Networks (GNN), aiming to identify graph anomalies using multi-scale patch and context-level contrastive learning.
3) **CoLA**[Liu *et al.*, 2022] is an anomaly detection algorithm focusing on nodes, utilizing a GNN-based contrastive learning method at the node-subgraph level. It computes anomaly scores for nodes by evaluating representations derived from nodes and subgraphs in positive and negative instance pairs.
4) **GRADATE**[Duan *et al.*, 2023] is an anomalous node detection approach based on node-node, node-subgraph, and subgraph-subgraph multi-view contrastive learning.
5) **AS-GAE**[Zhang and Zhao, 2022] is an anomaly subgraph detection algorithm utilizing a graph autoencoder. It generates a residual graph by reconstructing the given graph and detects anomaly subgraphs by identifying different areas between the original graph and its reconstructed graph.

Anomaly injection: To evaluate the detecting ability of our algorithm, having anomaly ground truth in datasets is essential. However, all experimental datasets lack anomaly labels. We explore two types of anomaly injection: structural anomaly injection and attribute anomaly injection. We randomly select q nodes and induce q connected subgraphs using a random walk approach and then transform them into q fully connected subgraphs. Similarly, we randomly sample q connected subgraphs with the same number of nodes as Set T, and select k nodes as Set C. The attribute of each node in T is perturbed based on the Euclidean distance between it and the randomly selected node in C. The number of anomalies in each dataset can be found in Table 1.

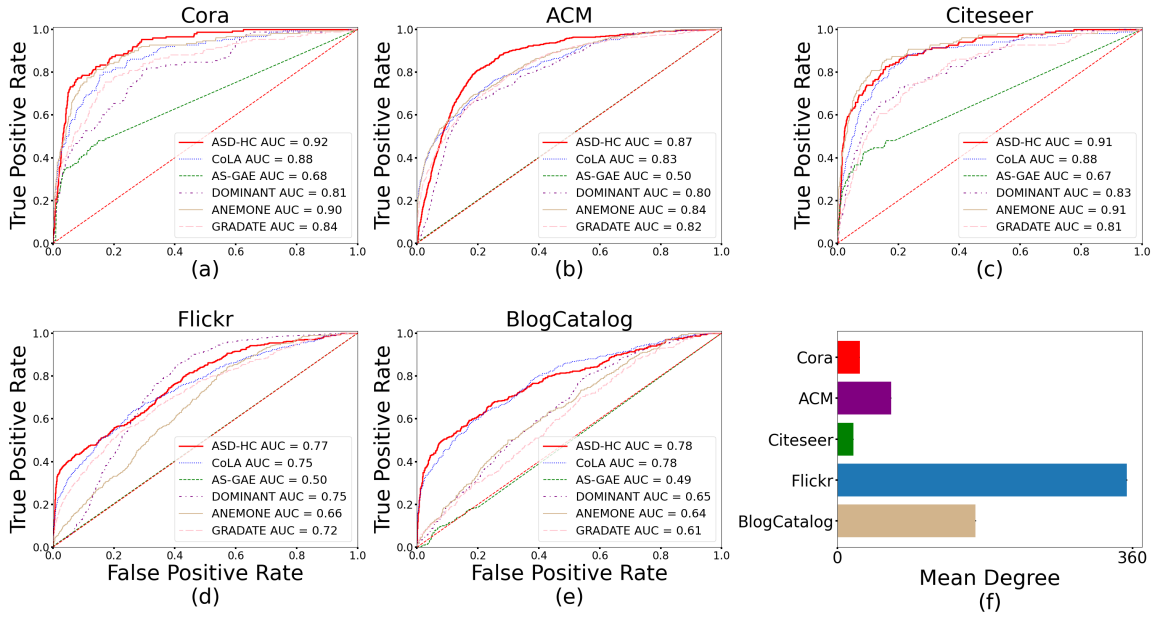

Figure 3: ROC curve evaluations for the detection of anomalous nodes on five distinct datasets, comparing ASD-HC to other baselines.

Methods	Cora				ACM				Citeseer				Flickr				BlogCatalog			
	$N_\alpha(S)$	$N(S)$	AUC	$f(\hat{S})$	$N_\alpha(S)$	$N(S)$	AUC	$f(\hat{S})$	$N_\alpha(S)$	$N(S)$	AUC	$f(\hat{S})$	$N_\alpha(S)$	$N(S)$	AUC	$f(\hat{S})$	$N_\alpha(S)$	$N(S)$	AUC	$f(\hat{S})$
AS-GAE	-	542	0.7590	-	-	22	0.5175	-	-	666	0.7951	-	-	4	0.25	-	-	1039	0.5458	-
ASD-HC(RWR)	166	389	0.9153	**15.29**	1240	2653	0.8139	**45.78**	184	385	0.9252	**18.02**	494	1259	0.7153	**24.08**	336	1060	0.6829	**15.23**
ASD-HC(BFS)	655	2502	0.9231	1.36	4117	16445	0.8784	0.10	667	2223	0.9310	5.45	1894	7575	0.7718	0.01	1300	5196	0.7783	0.03

Table 3: Evaluation of anomaly subgraph detection by two ASD-HCs and AS-GAE across five benchmark datasets. ("AUC" denotes the evaluation of anomalies within the detected anomaly subgraph, $f(\hat{S})$ denotes the significant level within the detected anomaly subgraph, where $N_\alpha(S) = \Sigma_{v_i \in V_S} \delta(p(\varpi_i) \leq \alpha)$ denotes the number of anomalous nodes in S, $N(S) = \Sigma_{v_i \in V_S} \delta(1)$ denotes the number of nodes in S, $\alpha = 0.25$.)

5.3 Experimental Result Analysis

Anomaly Detection Evaluation

In this section, we focus on evaluating the performance of our approach in detecting anomalous nodes within the graph and comparing it with other baseline methods. Due to the prevalent use of the ROC curve and the Area Under the Curve (AUC) metric for evaluating binary classification algorithms, our experiments utilize these two metrics to assess our algorithm's capability in distinguishing anomalous nodes from normal nodes within a graph. The figures from our experiments, as indicated in Table 2, illustrate that our approach outperforms the baselines across five benchmark datasets. In the Cora dataset, our approach, ASD-HC, achieves the highest AUC score of 0.9217, surpassing the competitive baseline ANEMONE by 2.42%. ASD-HC also attains the highest AUC score of 0.8786 in ACM, outperforming the competitive baseline ANEMONE by 3.88%. Similarly, in the Flickr dataset, ASD-HC achieves a score of 0.7719, consistently surpassing the baselines by at least 2.51%. Moreover, in the BlogCatalog dataset, ASD-HC and ColA both attain the highest score of 0.7800, exceeding other baselines by 13.35%. Although our approach does not maintain the highest score in the Citeseer dataset, it is only 0.8% less than ANEMONE.

Based on the experimental data depicted in Figure 3, when the average degrees of nodes are small, such as in Cora, ACM, and Citeseer as shown in Figure 3(f), where the degrees are all less than 65.5, ANEMONE serves as the competitive baseline for our approach. When the degrees of graphs increase, CoLA becomes the primary competitive baseline for ASD-HC. In both scenarios, whether compared with ANEMONE or CoLA, ASD-HC either surpasses them or closely matches their performances. Therefore, our focus is directed toward analyzing the distinctions between ASD-HC and them. ANEMONE learns node embeddings through node&node and node&ego-net contrastive instance pairs within multi-views. It considers the influence from the node's direct neighbor to the target node. CoLA's result is derived from its node-subgraph contrastive instance pairs by employing the random walking method to sample 4 nodes for each subgraph. The choice of 4 is the default setting in its original algorithm and serves as the walking path length in RWR, which aligns with the ideal value described in its paper. We attribute the superior performance of ASD-HC over ANEMONE and CoLA to our high-order neighbor-subgraph sampling strategy, which enables a more comprehensive capture of the characteristics of node structure and relationships between key neighbors.

Compared to the state-of-the-art algorithms in deep learning-based anomaly detection within graph structural data, these results highlight the excellent ability of our approach to detect anomalous nodes in a graph. Figure 3

presents detailed data from our experiments on ROC curves. The ROC curve is generated by calculating True Positive Rate (TPR) and False Positive Rate (FPR) scores at various thresholds between predicted anomalous and normal instances. These curves are then plotted to illustrate the performance characteristics of the models. Figure 3 reveals that the AUC score corresponds to the values presented in Table 2. The ROC curves represented by the red lines in each dataset also demonstrate the effectiveness of our approach. However, in these experiments, we only assess the performance of detecting anomalous nodes for ASD-HC and all baselines. Anomalous node detection is not really our ultimate goal. Therefore, there is a need to further assess our algorithm and AS-GAE in the context of anomaly subgraph detection.

Anomaly Subgraph Detection Evaluation

After anomaly features have been extracted from each node through contrastive learning, we predict the distribution of the anomalies within the entire graph. Under the Uniform distribution, we calculate the p-values (probabilities) for all nodes' anomalous scores $p(\varpi)$. Anomalous nodes are identified by filtering out those with p-values lower than the specified significantly anomalous threshold (i.e., $p(\varpi_i) \leq \alpha$). In our experiments, the threshold α is set to 0.25. Additionally, our approach employing the RWR algorithm is designated as "ASD-HC(RWR)." To enable a thorough comparison, we replace RWR in our approach with the Breadth-First Search (BFS) algorithm, creating a variant denoted as "ASD-HC(BFS)." Additionally, AS-GAE serves as another baseline in the anomaly subgraph detection experiment. We evaluate the nodes in the maximum anomaly subgraph detected by our two variants ASD-HCs and AS-GAE, calculating AUC metrics, as shown in Table 3. The AUC metrics of both our ASD-HCs consistently outperform the AUC of AS-GAE, demonstrating an improvement of at least 13.01%. Furthermore, the detection result of AS-GAE is a set of nodes, not refined into anomalous and normal nodes. Hence, we represent its node count in the "$N(S)$" column. When comparing the two ASD-HCs, the subgraphs detected by ASD-HC(BFS) are notably larger and almost equal to the entire graph. Although ASD-HC(BFS)'s AUC slightly outperforms that of ASD-HC(RWR), this advantage is not deemed satisfactory. According to the values of $f(\hat{S})$ in Table 3, the subgraph detected by ASD-HC(RWR) demonstrates the most significantly anomalous levers on all benchmark datasets. In summary, ASD-HC(RWR) exhibits the most optimal capability in detecting anomaly subgraphs. Furthermore, ASD-HC(BFS)'s running time is longer compared to ASD-HC(RWR).

Parameter Studying

In our experiment, the batch size is set to 200, and the number of epochs is set to 100 for smaller graphs (e.g., Cora and Citeseer). For larger graphs with a substantial number of nodes or edges (e.g., ACM, Flickr, and BlogCatalog) reflecting more complex graph structures, additional training time is required. In such cases, we set the number of epochs to 400. The learning rate is set in the range of $[0.001, 0.0035]$. In addition to these above common parameters, our algorithm introduces three other crucial parameters: the order of neighbors (denoted as k) that we consider, the size of the neighbor-subgraph (denoted as t), and the significant level of the anomaly α. Typically, we set k equal to 3, t to 15, and assign a value to α within the range of $[0.15, 0.25]$.

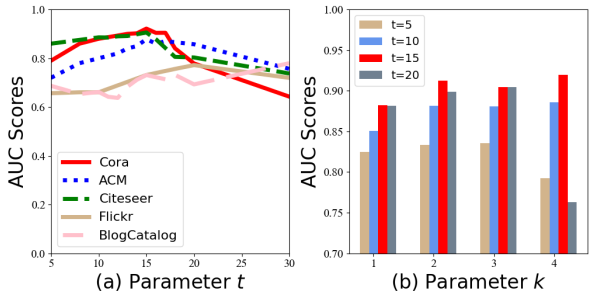

Figure 4: Evaluation of influences of k-order neighbors and the size of the neighbor-subgraph t. (a) As $k = 3$, evaluating the effect of t on the AUC, (b) When t is set to 5, 10, 15, and 20, respectively, AUC varies with k on the Cora dataset.

5.4 Ablation Studies

Effect of the Size of Neighbor-Subgraph

In Figure 4(a), with the order of neighbors k set to 3, the size of the neighbor-subgraph t increases accordingly. AUC shows a rising trend, reaching optimal values when t is in the range of $[15, 20]$, and then followed by a decline. This pattern is consistent across most datasets. However, in the BlogCatalog dataset, the AUC does not exhibit a downward trend until $t > 35$. After analysis, BlogCatalog has the highest edge-node ratio, indicating a tighter and more complex graph structure with stronger correlations between nodes. Enlarging the subgraph size t is beneficial for capturing node features with greater accuracy.

Effect of the High-Order of Neighbors

Taking the Cora dataset as an example, although specifying four different orders of neighbors ($k = 1, 2, 3, 4$), the AUC value achieves its optimal value when t is set to 15. However, when increasing the value of k, the AUC exhibits an upward trend, as shown in Figure 4(b).

6 Conclusion

Detecting anomaly subgraphs is vital for numerous real-world applications. Most existing anomaly detection methods overlook the identification of anomaly subgraphs in the graph data. In this paper, we present ASD-HC, a novel end-to-end Graph Neural Network (GNN)-based contrastive learning approach designed to detect the maximum connected anomaly subgraph within graph-structured data. ASD-HC employs a high-order neighborhood sampling strategy to construct node-subgraph instance pairs. It extracts anomaly features of nodes and defines an evaluation function based on the Non-Parametric Graph Scan statistics to detect the maximum connected anomaly subgraph. Our approach highlights its capabilities in detecting anomalous nodes and anomaly subgraphs, especially, outperforming baselines by over 13.01% in detecting the maximum anomaly subgraph. Experiment results show our approach effectively detects anomaly subgraphs in large-scale graphs.

Acknowledgements

This work is supported by the National Key Research and Development Program of China(31400) and the Natural Science Foundation of Inner Mongolia(2022LHMS06008). It is also partially sponsored by the Key R&D and Transformation Plan of Qinghai Province(2022-QY-218).

References

[Akoglu *et al.*, 2015] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data Min. Knowl. Discov.*, 29(3):626–688, 2015.

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

[Duan *et al.*, 2023] Jingcan Duan, Siwei Wang, Pei Zhang, En Zhu, Jingtao Hu, Hu Jin, Yue Liu, and Zhibin Dong. Graph anomaly detection via multi-scale contrastive learning networks with augmented view. In *AAAI*, pages 7459–7467. AAAI Press, 2023.

[Hamilton *et al.*, 2018] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2018.

[Hu *et al.*, 2023] Jingtao Hu, Bin Xiao, Hu Jin, Jingcan Duan, Siwei Wang, Zhao Lv, Siqi Wang, Xinwang Liu, and En Zhu. Samcl: Subgraph-aligned multiview contrastive learning for graph anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 2023.

[Huang *et al.*, 2023] Ling Huang, Ye Zhu, Yuefang Gao, Tuo Liu, Chao Chang, Caixing Liu, Yong Tang, and Chang-Dong Wang. Hybrid-order anomaly detection on attributed networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12249–12263, 2023.

[Jin *et al.*, 2021] 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*, pages 3122–3126. ACM, 2021.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR (Poster)*. OpenReview.net, 2017.

[Lawrence *et al.*, 1999] Steve Lawrence, C. Lee Giles, and Kurt D. Bollacker. Digital libraries and autonomous citation indexing. *Computer*, 32(6):67–71, 1999.

[Liu *et al.*, 2022] 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*, 33(6):2378–2392, 2022.

[McCallum *et al.*, 2000] Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Inf. Retr.*, 3(2):127–163, 2000.

[Pang *et al.*, 2021] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, 54(2):38:1–38:38, 2021.

[Scarselli *et al.*, 2009] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[Sen *et al.*, 2008] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Mag.*, 29(3):93–106, 2008.

[Sun *et al.*, 2020] Ying Sun, Wenjun Wang, Nannan Wu, Wei Yu, and Xue Chen. Anomaly subgraph detection with feature transfer. In *CIKM*, pages 1415–1424. ACM, 2020.

[Sun *et al.*, 2022] Ying Sun, Wenjun Wang, Nannan Wu, Chaochao Liu, Siddharth Bhatia, Yang Yu, and Wei Yu. AAAN: anomaly alignment in attributed networks. *Knowl. Based Syst.*, 249:108944, 2022.

[Tang and Liu, 2009] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *KDD*, pages 817–826. ACM, 2009.

[Tang *et al.*, 2008] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *KDD*, pages 990–998. ACM, 2008.

[Tong *et al.*, 2006] Hanghang Tong, Christos Faloutsos, and Jia-yu Pan. Fast random walk with restart and its applications. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 613–622, 2006.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[Wu *et al.*, 2019] Nannan Wu, Feng Chen, Jianxin Li, Jinpeng Huai, Baojian Zhou, Bo Li, and Naren Ramakrishnan. A nonparametric approach to uncovering connected anomalies by tree shaped priors. *IEEE Trans. Knowl. Data Eng.*, 31(10):1849–1862, 2019.

[Wu *et al.*, 2022] Nannan Wu, Ning Zhang, Wenjun Wang, Lixin Fan, and Qiang Yang. Fadman: Federated anomaly detection across multiple attributed networks, 2022.

[Zhang and Zhao, 2022] Zheng Zhang and Liang Zhao. Unsupervised deep subgraph anomaly detection. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 753–762, 2022.

[Zhang *et al.*, 2023] Chi Zhang, Wenkai Xiang, Xingzhi Guo, Baojian Zhou, and Deqing Yang. Subanom: Efficient subgraph anomaly detection framework over dynamic graphs, 2023.

[Zheng *et al.*, 2023] Yu Zheng, Ming Jin, Yixin Liu, Lianhua Chi, Khoa Tran Phan, and Yi-Ping Phoebe Chen. Generative and contrastive self-supervised learning for graph anomaly detection. *IEEE Trans. Knowl. Data Eng.*, 35(12):12220–12233, 2023.