Unsupervised Deep Subgraph Anomaly Detection

Zheng Zhang, Liang Zhao
Department of Computer Science, Emory University, USA
{zheng.zhang, liang.zhao}@emory.edu

2022 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-5099-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDM54844.2022.00086

***Abstract*—Effectively mining anomalous subgraphs in networks is crucial for many application scenarios, such as disease outbreak detection, financial fraud detection, and activity monitoring in social networks. Identifying anomalous subgraphs is extremely challenging due to their complex topological structures and high-dimensional attributes, various notions of anomalies, and the exponentially large subgraph space in a given graph. Existing classical shallow models typically rely on handcrafted anomaly measure functions, which cannot handle common situations when such prior knowledge is unavailable. Recently, deep learning-based methods provide an end-to-end way that learns the anomaly measure functions. However, although they have achieved great success in detecting node-level, edge-level, and graph-level anomalies, detecting anomalous at the subgraph level has been largely under-explored due to enormous difficulties in subgraph representation learning, supervision, and end-to-end anomaly quantification. To circumvent the above mentioned challenges, this paper proposes a novel deep framework named Anomalous Subgraph Autoencoder (AS-GAE) to extract the anomalous subgraphs in an unsupervised and weakly supervised manner. Specifically, we first develop a location-aware graph auto-encoder to uncover the anomalous areas in the given graph according to the mismatch during the reconstruction. Then a supermodular graph scoring function module is proposed to assign reasonable anomaly scores to the subgraphs in the extracted anomalous areas. The superiority of our proposed method was demonstrated through extensive experiments on two synthetic datasets and nine real-world datasets.**

***Index Terms*—Anomaly detection, Deep graph autoencoder**

I. Introduction

Network data is a popular type of data that describes the properties of discrete objects and their pairwise relationship. Given a network, one of the major tasks in the field of network data mining is the detection of anomalous subgraphs. A subgraph can be defined as an anomaly when its connectivity structure or attributive properties can be described as an outlier in the graph. For example, the significant difference in subgraph topological structure is one type of anomaly subgraph. As shown in Figure 1(a), in a regular lattice material network, one might expect the inserted impurity as an outlier because the topological structure is different from other areas in the graph. Similarly, infectious disease researchers may have an interest in discovering a new unknown infectious disease at the early stage of a disease outbreak (e.g. COVID-19) from the health surveillance network. In normal cases, the count of cases for different disease symptoms should follow a statistical distribution such as the Poisson distribution. As shown in Figure 1(b), a group of connected nodes with significant abnormal symptom attributes may indicate a potential disease outbreak is taking place.

Existing graph anomaly detection algorithms [1] can be categorized into traditional shallow methods [2], [3] and deep learning-based method [4], [5]. Previous shallow approaches to detecting anomaly subgraphs have mainly focused on manually defining anomaly quantification metrics for subgraphs and developing methods to extract anomalous patterns based on the designed measures. Although the ideas behind these shallow methods are simple and intuitive, the shallow mechanisms are suffered from the limited capability of capturing non-linear properties to discriminate complex anomalies from graphs with high-dimensional features and irregular topological structures [6]. More importantly, these methods require prior knowledge to determine the measurement function for detecting anomalies, which are usually unavailable due to the unknown nature of anomaly subgraph patterns in many practical applications [7], [8], and cannot make the detection process end-to-end. On the other hand, deep learning-based methods, which have received growing attention in recent years [9], can extract expressive representations of objects, such as nodes or graphs, to effectively distinguish abnormal and normal objects. Previous works have shown impressive progress in many graph anomaly detection tasks [9].

Despite the success in generalizing deep learning techniques to graph anomaly detection problems, most previous works only focus on detecting node- or edge-level anomalies. The task of anomaly subgraph mining has been largely under-explored and has just started to attract attention. However, there is no trivial way to simply apply these methods to accomplish the task of detecting anomalous subgraphs due to several unique challenges: (1) **Difficulty in obtaining sufficient training labels in an end-to-end manner without intensive supervision**. Deep learning approaches heavily rely on the training objectives to optimize model parameters. For anomalous subgraph detection, this necessitates sufficient training labels and appropriate loss function such that models can effectively discriminate the anomalous patterns. Unfortunately, the training labels are hard to sufficiently obtain due to the exponential possible subgraphs in a given graph, and designing proper objectives for detecting anomalous subgraphs is challenging because there is usually no prior knowledge about the anomalies. (2) **Difficulty in preserving both geodesic distance and topological similarity among nodes for representing subgraph anomaly**. Existing end-to-end works that consider node- or edge-level cannot be directly used for subgraph-level anomaly detection. For example, within the scope of a given anomaly subgraph, individual nodes or edges might be normal. It only turns out to be anomalous

978-1-6654-5099-7/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDM54844.2022.00086

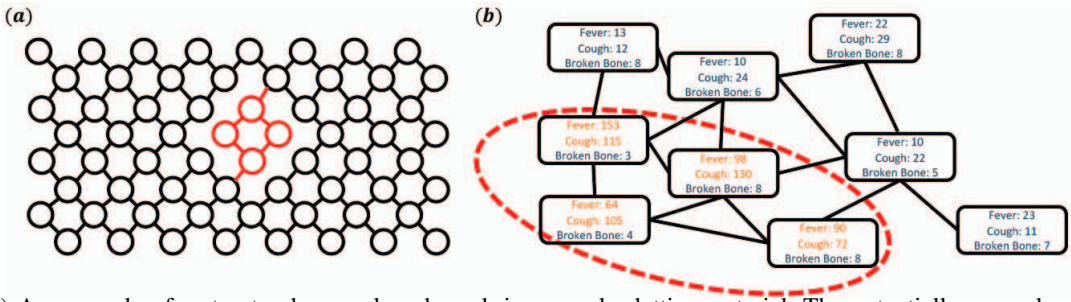

Fig. 1: (a) An example of a structural anomaly subgraph in a rugular lattice material. The potentially anomalous subgraph in red has a significantly different topology from other regions in the given network, which may be skeptical of a potential imperfection area. (b) An example of an attributive anomaly subgraph in a simulated disease outbreak network with three attributes of disease symptoms (fever, cough and broken bones). The potentially anomalous subgraph in the red dashed circle has an anomalous subset of attributes (cough and fever). The count number of these two attributes within the anomaly subgraph area are significantly higher than other nodes.

when considered as a group compared to other areas. Merely aggregating node- or edge-level anomaly scores can not reflect the subgraph abnormality. But how to jointly consider intra-subgraph structure, as well as the subgraph's position and relation to the whole graph, is indispensable yet very challenging. (3) **Difficulty in quantifying the degree of being abnormal under arbitrary and unknown anomaly types**. It is extremely hard to quantify the degree of abnormality for arbitrary patterns of attribute and topological structure of subgraph when a ground truth anomaly type is unknown. Previous shallow methods typically utilize a handcrafted measure function, which is limited by its low expressive power and generalizability to unseen anomalies. On the other hand, existing deep learning methods learn the scoring function from scratch, which is usually subject to the exponentially growing search space and can easily lead to overfitting issues.

In order to address all the above mentioned challenges, this paper proposes a novel deep **A**nomalous **S**ub**g**raph **A**uto**e**ncoder (**AS-GAE**) framework for detecting anomaly subgraphs. Specifically, to overcome the issue of lacking sufficient ground truth anomalies, we train the model in an unsupervised or weakly supervised manner by optimizing self-supervised learning objectives. Therefore, our framework does not require any prior knowledge about the anomalies. To simultaneously maintain geodesic distance and topological similarity information among nodes to represent subgraph anomalies, a novel *location-aware graph autoencoder* module is proposed to extract expressive representation vectors. Therefore, a residual graph can be constructed from the original graph by uncovering the mismatched areas of the graph during the reconstruction. Finally, to assign a reasonable anomaly score to any subgraph with arbitrary attributes and topologies, a *supermodular graph anomalies quantification* module is applied to introduce an inductive bias in a collective way.

II. Related Work

Anomaly subgraph detection in networks. There has been a long time of research efforts on the subjects of anomaly subgraph detection in network data [1]. One mainstream of works is to select a subset of nodes or edges from graphs with respect to a given objective function and constraints. Typically, SODA [2] identifies the anomaly subgraph by giving an input subgraph query template, where the measure metrics are based on the deviations in linkage compared to its neighborhood [10]. Chen *et at.* [3] proposed a method to assign a manually designed anomaly score function f of nodes and maximize the total anomaly score $\sum_{i=1}^{k} f(v_i)$ of a subset of nodes according to a constraint on the structure of found nodes set (e.g. connected graph). However, the key to their methods is the manually designed anomaly score function which requires prior knowledge and can not be extended when the potential anomaly type is unknown. Another stream of works focuses on mining small subgraphs with anomalous connectivity structure in the graph by analyzing the spectral information from the given topology property of the graph. Specifically, Miller *et at.* [11] applies eigenvalue decomposition to the modularity matrix of the graph, and shows that the existence of a small anomalous connected subgraph would result in the appearance of high variance in the subspace spanned by combinations of eigenvectors. Sharpnack *et at.* [12] developed a method by utilizing a generalized likelihood ratio test and a Lovász extended scan statistic for detecting a well-connected region of vertices. However, these works can only consider the topological structure of graphs and omit the rich information that is carried by attributes, which means they cannot handle attribute anomalies or mixed anomalies.

Recently, several anomaly detection studies in an unsupervised manner by measuring the attribute and topology difference between the internal and external structure of a given subgraph are proposed. AMEN [13] is a typical work that was proposed to model the attribute distribution and measure the normality of subgraphs by considering the consistency between internal and external information. Similarly, ANOMALOUS [14] leverages a joint framework to fuse attribute and network structure information and performs attribute selection and anomaly detection based on CUR decomposition and residual analysis. However, due to the low representative power of manually designed shallow similarity measures, these methods can not effectively distinguish anomalies from graphs.

Deep learning in graph anomaly detection. The deep

approaches are capable of learning the expressive representations of nonlinear relationships within the complex structure of objects [15]–[17]. The recent success studies of graph neural networks (GNNs), further enrich the capability of deep learning models for graph data mining tasks, including graph anomaly detection [9], [18]. However, many existing models such as DCI [19] and CARE-GNN [20] are supervised models that require ground truth labels, which are usually not available. Under the unsupervised learning manner, one stream of works is applying deep one-class classification [4] loss to the learned representation vectors of nodes in the graph, then the anomaly scores are calculated according to the relative distance from nodes to a sphere in the latent space. Another stream of works [21], [22] typically employs an autoencoder framework to reconstruct the graph, where the anomaly scores of nodes/edges are computed as their reconstruction errors. Although extensive efforts have been devoted to detecting anomalous nodes/edges in graphs, little work has been done on detecting anomalous subgraphs. DEEPFD [23] is a recent attempt to address this problem. It first uses a graph autoencoder framework to learn the latent representation of nodes, then applies a density-based clustering algorithm DBSCAN in the latent representation space to detect a dense cluster, where the extracted dense cluster is considered as the anomaly subgraph in the given graph. However, the effectiveness of their proposed model is highly dependent on the assumption that the representation vectors of nodes in anomalous subgraphs are close to each other, while other normal nodes are uniformly distributed in the remaining latent space. This assumption may be invalid for networks when the normal nodes are shared similar topology or attributes (e.g. a biological network). Furthermore, this method cannot guarantee that the extracted nodes are distributed close together in the graph, which may contradict the idea of extracting an anomalous subgraph.

III. Problem Formulation

We consider an attributed network as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is a set of nodes that $N = |\mathcal{V}|$ denotes the number of nodes in the graph and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. We also let $\mathbf{X} \in \mathbb{R}^{N \times p}$ denotes the node attribute matrix and $\mathbf{A} \in \mathbb{R}^{N \times N}$ represents the adjacency matrix. Specifically, the attribute of node v_i can be expressed as a p dimensional vector $x_i \in \mathbb{R}^p$. $A_{ij} = 1$ denotes there is an edge connecting nodes v_i and $v_j \in G$, otherwise $A_{ij} = 0$. A subgraph h of the given G is represented as $H = (\mathcal{V}_H, \mathcal{E}_H)$ where $\mathcal{V}_H \subseteq \mathcal{V}$ is a subset of nodes and $\mathcal{E}_H$ is the corresponding set of edges. $\mathbf{X}_H, \mathbf{A}_H$ are the corresponding node attribute and adjacency matrices. With the preliminary notion of the attributed network, we formalize the anomalous subgraph detection problem as follows:

Problem 1. **Anomalous subgraph detection.** *Given a graph G, the task of anomalous subgraph detection is to search for a subgraph $H \subseteq G$ that is most different from the majority of graph, where the degree of being abnormal is quantified by a score function f.*

The main goal of this paper is to effectively extract anomalous subgraphs from a given graph, which is extremely difficult to address due to several unique challenges: 1) Difficulty in obtaining sufficient training labels in an end-to-end manner without intensive supervision. 2) Difficulty in maintaining both geodesic distance and topological similarity information among nodes for representing subgraph anomalies. 3) Difficulty in quantifying the degree of being abnormal under arbitrary and unknown anomaly types.

IV. Methodology

In order to design an effective method for detecting anomalous subgraph in attributed networks by addressing the above mentioned challenges, we propose a novel framework named deep **A**nomalous **S**ub**g**raph **A**uto**e**ncoder (**AS-GAE**), which is shown in the Figure 2. In the rest of this section, we first introduce the details of the two core components of our whole framework AS-GAE: *location-aware graph autoencoder* in Section IV-A and *supermodular graph anomalies quantification* in Section IV-B. Then in Section IV-C, in order to optimize the parameters in these two modules without intense supervision, we present the objective function of AS-GAE under an unsupervised learning setting and a weakly supervised learning manner. Specifically, a novel *location-aware graph autoencoder*, which is shown in Figure 2(a), is first performed to map the input graph G into low-dimensional embeddings $\mathbf{Z}$ by an encoder ϕ, and then reconstruct the graph structure and features from the low-dimensional embeddings through a decoder φ. Then the disparity between the original graph and the reconstructed graph is used to generate the residual graph $\mathcal{R}$ as our core to identify the candidate anomaly subgraphs. Especially, we propose a novel location-aware encoder to enhance relative dependency information among nodes in a collective way for representing subgraph anomalies. As shown in Figure 2(b), among the candidate anomaly subgraphs in the built residual graph $\mathcal{R}$, we jointly detect the most anomaly subgraph and learn the underlying anomaly scoring function by a novel *supermodular graph anomalies quantification* module. Finally, to optimize all the parameters involved in the AS-GAE framework without intense supervision, we propose two learning strategies where one is unsupervised and the other is weakly supervised. Therefore, our framework is not limited to any specific anomaly types, nor does it require prior knowledge about the anomaly patterns.

A. Candidate anomaly subgraphs extraction by location-aware graph autoencoder

To detect the most anomalous subgraph, we first extract a few candidate anomalous subgraphs from the given input graph. Most existing methods that require intense supervision cannot handle this problem due to the lack of prior knowledge of potential anomalous patterns. One-class classification [4] and reconstruction-based method [24] are two commonly used ways to find the anomalies in an unsupervised manner. However, since the number of possible subgraphs in a given graph grows exponentially, one-class classification-based methods can easily lead to overfitting issues due to

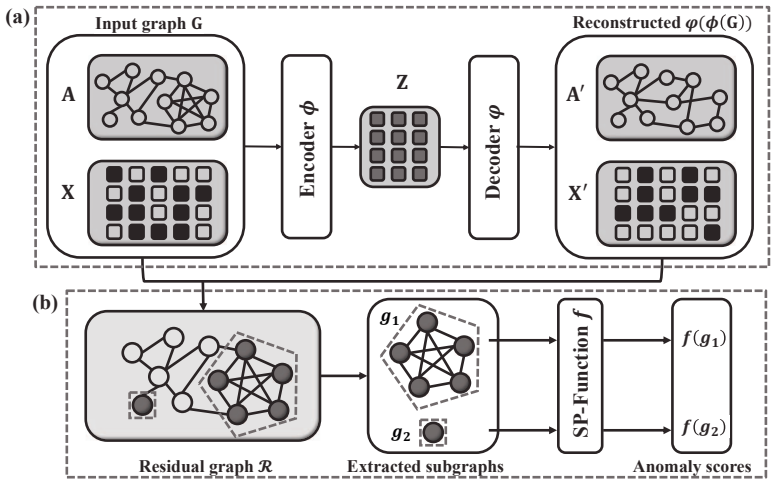

Fig. 2: The overall framework of our proposed model, which is composed of two components: (a) *location-aware graph autoencoder* and (b) *supermodular graph anomalies quantification*. The *location-aware graph autoencoder* module first maps the information of both local neighborhoods and global locations into low-dimensional latent embeddings Z by a location-aware encoder ϕ, then reconstructs both the adjacency matrix $\mathbf{A}$ and node features $\mathbf{X}$ through the decoder φ. Then the residual graph $\mathcal{R}$ is built according to the mismatch between the reconstructed graph $\varphi(\phi(G))$ and original graph G to uncover the potential anomalous area. For each subgraph in the constructed graph, the *supermodular graph anomalies quantification* module applies a graph supermodular function (SP-Function f) to assign an anomaly score to it.

the huge search space. On the other hand, the reconstruction-based method is able to avoid graph scanning by constructing the residuals between true data and estimated data. Specifically, reconstruction-based anomaly detection methods are established by finding low-dimensional representations of the original data, where anomalous and normal data are expected to be separated from each other. Traditional methods, such as Principal Component Analysis (PCA), suffer from the limited expressive capability of their model, which may lead to false alarms [25]. Graph autoencoder, which serves as an unsupervised deep neural network technique, can capture the nonlinear correlations between data due to its strong approximation power. After obtaining these low-dimensional embeddings, they are restored to the original data space, which is called the reconstruction of the original graph data. By reconstructing data from low-dimensional embeddings, the nature of the majority pattern in the data is captured while outliers and noise are filtered out. Reconstruction error, which is defined as the degree of mismatch between the original data and the reconstructed data from low-dimensional embeddings, can be considered a strong indicator for describing anomalies in the dataset [5], [21], [26]–[28]. Specifically, larger reconstruction errors indicate a higher probability of anomalies, as they deviate significantly from the majority patterns.

Although existing graph autoencoder models can handle individual nodes, they can not accurately handle subgraphs because autoencoder models can not accurately reconstruct subgraph context. Specifically, as shown in Figure 3, they can not tell whether the neighbors of a node are in the region of the subgraph or in a far away region, which leads to the loss of subgraph information in the learned representation. The key reason behind it is the insufficient expressive power

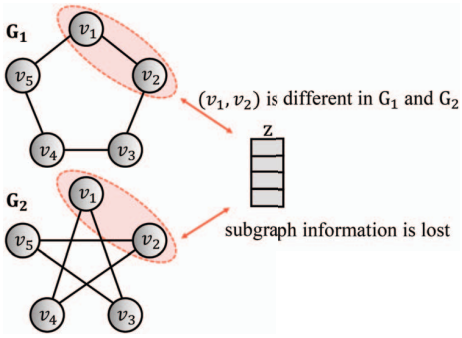

Fig. 3: Example of plain graph autoencoders failing to detect potential anomalous subgraphs. Since the global positions of nodes in the learned latent representation are lost, nodes can not be aware of which node is directly connected to them. Although (v_1, v_2) is different in G_1 and G_2, the learned representations will be identical. Therefore, (v_1, v_2) in G_1 and G_2 will be encoded into the same embedding, resulting in a large reconstruction error and causing the model to tend to treat normal subgraphs as abnormal.

of ordinary graph neural network encoders in representing subgraphs because they can not retain the global location of nodes in the graph, which results in the inability to capture the dependence between nodes in a given subgraph and hence the failure to detect the real anomalies.

In order to correctly reveal the potential contextual anomaly areas, we design a location-aware graph autoencoder to incorporate the locational information of nodes in the network when performing message aggregation. To be specific, we first sample a set of C random anchor nodes $\mathcal{V}_C = \{v_1, v_2, \ldots, v_C\}$ in the given graph G. Then the shortest distance from each node to anchors, which is denoted as $\mathbf{S} \in \mathbb{R}^{N \times C}$, is calculated

and treated as additional node labels. Thus the global location of each node can be inferred from the shortest distances to all anchor nodes (relabeled nodes attributes). The modified location-aware graph-autoencoder is able to capture the dependence properties among nodes when they are located in a close area within the global context of the graph. This module plays a critical role in correctly reconstructing the normal areas and uncovering the real anomalous areas in the graph.

Mathematically, given the input graph G, our proposed location-aware autoencoder first maps the adjacency matrix $\mathbf{A}$, node attributes matrix $\mathbf{X}$, and distances to anchors matrix $\mathbf{S}$ into the latent representation vectors $z \in \mathbb{R}^{N \times D}$ through location-aware encoder ϕ, where D is the dimension of latent representation and $z_i \in \mathbb{R}^D$ is the corresponding latent vector for the node v_i. Formally, one convolutional operation of a location-aware encoder can be represented as:

$$\mathbf{X}^{(\ell+1)} = g^{(\ell)}\left(\mathbf{X}^{(\ell)}, \mathbf{S}, \mathbf{A} | \mathbf{W}^{(\ell)}\right), \tag{1}$$

where $g^{(\ell)}$, $\mathbf{X}^{(\ell)}$ and $W^{(\ell)}$ is the graph convolution function, the latent embeddings of nodes, and a trainable weight matrix at layer ℓ, respectively. We take the attribute matrix $\mathbf{X}$ as the input of the first layer, which is equivalent to $\mathbf{X}^{(0)}$. Then, we reconstruct both node attributes matrix $\mathbf{X}'$ and adjacency matrix $\mathbf{A}'$ from extracted latent embeddings z by decoder φ. Specifically, the reconstructed attribute x'_i of node v_i can be represented as:

$$x'_i = \rho\left(z_i | \mathbf{W}_{\mathrm{X}}\right), \tag{2}$$

and the reconstructed link A'_{ij} between node v_i and node v_j can be represented as:

$$A'_{ij} = \varrho\left(\{z_i, z_j\} | \mathbf{W}_{\mathrm{A}}\right), \tag{3}$$

where ρ and ϱ are two multilayer perceptron (MLP) modules and $\mathbf{W}_{\mathrm{X}}$ and $\mathbf{W}_{\mathrm{A}}$ are two trainable weight matrices. Then the learning objective is composed of two self-supervised tasks, which we can write as

$$L(G, \varphi(\phi(G))) = \lambda \left\|\mathbf{X} - \mathbf{X}'\right\| + (1-\lambda)\left\|\mathbf{A} - \mathbf{A}'\right\|, \tag{4}$$

where ϕ is the location-aware encoder and φ is the nonlinear decoder. λ is a hyperparameter to tune the trade-off between the importance of structure information and attribute information and $\|\cdot\|$ is commonly chosen to be the ℓ_2-norm.

The residual graph $\mathcal{R}$ can then be built upon the mismatch between the original graph G and the reconstructed graph $\varphi(\phi(G))$. We first define the reconstruction error r_i of one node v_i as:

$$r_i = \lambda \left\|x_i - x'_i\right\| + (1-\lambda) \sum_j \left\|A_{ij} - A'_{ij}\right\|, \tag{5}$$

then the residual graph $\mathcal{R}$ is constructed as:

$$\mathcal{R} = \left(\mathcal{V}_{\mathcal{R}}, \mathcal{E}_{\mathcal{R}}\right), \mathcal{V}_{\mathcal{R}} = \{v_i | v_i \in \mathcal{V} \ \ \& \ \ r_i < \tau\}, \tag{6}$$

where $\mathcal{E}_{\mathcal{R}}$ is the corresponding set of edges to the set of nodes $\mathcal{V}_{\mathcal{R}}$ and τ is a threshold to filter out the nodes.

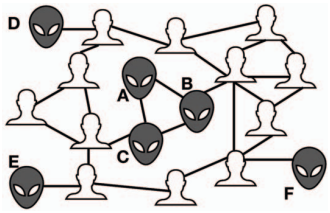

Fig. 4: An example of the synergy of anomalies in a social network. Here users A, B, and C are potential fraudulent users, which are equally abnormal in their individual behavior as users D, E, and F, respectively. But it is natural to consider users A, B, and C are more anomalous because of their connections to potentially fraudulent users, which might reinforce the implies that they are anomalous.

B. Quantifying anomalous scores of subgraphs by graph supermodular function

Given the extracted residual graph $\mathcal{R}$ from the original graph G by the reconstruction results of the location-aware graph autoencoder, the nodes $\mathcal{V}_{\mathcal{R}}$ in the residual graph can exhibit as a set of connected components $\{g | g \subseteq \mathcal{R}\}$ where each component g is an induced subgraph of the original graph G. Here a key question is how to evaluate the degree of abnormality of each extracted component in the residual graph. It is worth noting that the subgraph-level anomaly detection problem is different from the node-level anomaly detection problem, which only considers individual nodes without considering their overall dependencies within close-range regions. Instead, when individual nodes are connected in the same component, the synergy of anomalies is crucial for detecting subgraph-level anomalies. For example in Figure 4, in an online social network, it is more desirable to treat a subgraph of three connected potentially fraudulent users as more anomalous rather than three separate isolated users because they have abnormal neighbors. Therefore, more concretely, when quantifying the anomaly of a given subgraph, its value needs to be no less than the sum of the anomalies of all its individual partitions. This indicates that the anomaly quantization function should be supermodular.

Given a graph G with a collection of N labeled nodes $\mathcal{V} = \{v_1, \ldots, v_N\}$, the set of all subgraphs can be represented as $\mathcal{H} = \{H | H \subseteq G\}$. A graph scoring function $f : \mathcal{H} \to \mathbb{R}$ assigns a real value to any graph $H \in \mathcal{H}$. Here suppose H is the subgraph that is induced by set of nodes $\mathcal{V}_H$. We also use $H_{+\{u\}}$ to denote the subgraph which is induced by the set of nodes $\mathcal{V}_H \cup \{u\}$ when $\{u\} \notin \mathcal{V}_H$. We give the definition of a graph supermodular function as following:

Definition 1. *For all $Q, H \in \mathcal{H}$ that $\mathcal{V}_Q \subseteq \mathcal{V}_H \subseteq \mathcal{V} \backslash \{u\}$, a graph function $f(\cdot)$ is said to be supermodular if and only if*

$$f(Q_{+\{u\}}) - f(Q) \leq f(H_{+\{u\}}) - f(H). \tag{7}$$

A simplest example of supermodular function on graph can be given as $f(G) = |\mathcal{V}| + |\mathcal{E}|$, where $|\mathcal{E}|$ denotes the number of edges in graph G.

To fully utilize the reconstruction results by our location-aware graph autoencoder and make the whole framework

in an end-to-end manner, we hence propose a novel deep graph supermodular neural network by extending the previous submodular deep learning model [29]. Specifically, given an input graph G, one updating function of node v_i by deep graph supermodular neural network at layer ℓ can be expressed as:

$$x_i^{(\ell+1)} = \sigma^{(\ell)}(\mathbf{w}_1^{(\ell)\mathsf{T}} x_i^{(\ell)} + \sum_{j \in \mathcal{N}(i)} \mathbf{w}_2^{(\ell)\mathsf{T}} x_j^{(\ell)}), \tag{8}$$

where $\sigma^{(\ell)}$ is a non-negative non-decreasing convex function and $\mathbf{w}^{(\ell)}$ is a non-negative weight matrix. The whole deep graph supermodular neural network can be achieved by stacking L layers of updating operation in Equation 8 and a summation operation over all nodes, which is expressed as:

$$f(G) = \sum_i x_i^{(L)}, \tag{9}$$

where $x_i^{(L)}$ is the scalar value of node v_i after L stacked graph convolution operations.

Theorem 1. *For any graph G with non-negative node attributes, the graph scoring function $f : \mathcal{H} \to \mathbb{R}$ that is defined in Equation 9 is supermodular.*

Proof. We first prove that one layer of graph operation function on all nodes in Equation 8 is supermodular. For simplicity and without loss of generality, we suppose the input node attributes $x^{(\ell)}$ are one-dimensional non-negative scalar values and let $\mathbf{w}_1^{(\ell)} = \mathbf{w}_2^{(\ell)} = \mathbf{w}^{(\ell)}$, hence we can rewrite Equation 8 as

$$x_i^{(\ell+1)} = \sigma^{(\ell)}(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \mathbf{w}^{(\ell)} x_j^{(\ell)}).$$

Since the linear combinations of node scalar values $\sum_{j \in \mathcal{N}(i) \cup \{i\}} \mathbf{w}^{(\ell)} x_j^{(\ell)}$ is a modular function and $\sigma^{(\ell)}$ is a non-negative non-decreasing convex function, the composition convolution operation is supermodular [29]. Then, it is straightforward to know that in Equation 9 the summation over all supermodular functions is still supermodular. □

C. Unsupervised and Weakly supervised learning objectives of AS-GAE

To optimize the parameters involved in the previously mentioned components *location-aware graph autoencoder* and *supermodular graph scoring function* without intense supervision, we first present the objective function of the proposed framework under an unsupervised learning manner and then a weakly supervised strategy when partial labels are available.

1) Unsupervised model: After the training process, the anomalous scores can be given by the function $f(g), \forall g \subseteq \mathcal{R}$. Subgraphs with higher anomaly scores are considered to have a higher probability of being anomalies. The loss function of AS-GAE under an unsupervised setting can be described as

$$\min_{\phi,\varphi,r} L\left(G, \phi(\varphi(G))\right) + \beta F(\mathcal{R}),$$
$$s.t. F(\mathcal{R}) = \sum_{g \subseteq \mathcal{R}} \max\{0, r - f(g)\} - \gamma \left\| r \right\|,$$

where $\mathcal{R}$ is the residual graph and $f(\cdot)$ is the supermodular scoring function. β and γ are two hyperparameters, and $\|\cdot\|$ is commonly chosen to be the ℓ_2-norm.

The core idea of designing the loss function for unsupervised anomaly subgraph mining is that the majority of the graph should be normal. The first term in the loss function is the reconstruction error of the location-aware graph autoencoder, which is given in Equation 4. The second term is used to further penalize components in the residual graph with relatively small anomalous scores. For example, when there are multiple subgraphs in the residual graph, due to the rarity of the anomaly, it is reasonable to assume that most of the subgraphs found should still be normal. The learnable parameter r serves as a threshold for filtering the subgraphs in the residual graph. When the anomaly score of a subgraph given by f is larger than the threshold r, then no further penalization is given because it has a high probability to be a true anomaly subgraph.

2) Weakly Supervised Model: We generalize the unsupervised learning model to a slightly looser constraint setting. In some real-world situations, an imprecise ground true anomaly area can be provided, but not the exact type of anomaly. For example, when a protest event occurs, it is difficult to locate an accurate location range (e.g. a street) in a short period of time, but it may be easy to obtain a wider range (e.g. a city). When given a large range of unknown anomalies, it is worth designing an algorithm to reduce the extent of the anomaly region. We name this setting weakly supervised anomaly detection. The difference point is that there exists an anomaly subgraph in the given area of nodes $\mathcal{Q}$ and there are no anomalies outside this scale. Thus we modify the second term to a formulation, which first further penalizes the subgraphs in the residual graph that lie outside the scale $\mathcal{Q}$. On the other hand, it encourages the subgraph with the largest anomaly score that is within the scale. In this setting, we denote a given scale $\mathcal{Q} \subseteq \mathcal{V}$ as a collection set of nodes from the graph G, where all nodes in the ground truth anomaly subgraph are belonged to $\mathcal{Q}$. The loss function of our proposed model for weakly supervised learning can be described as

$$\min_{\phi,\varphi,f} L\left(G, \phi(\varphi(G))\right) + \beta F(\mathcal{R}),$$
$$s.t. F(\mathcal{R}) = -\max_{g \subseteq \mathcal{R}} \{(1 - t_g) \log f(g)\} + \sum_{g \subseteq \mathcal{R}} t_g \log f(g),$$

where $\mathcal{R}$ is the residual graph and t_g represents whether the component g is outside the contextural scale of given nodes collection $\mathcal{Q}$, where $t_g = 0$ when $g \subseteq \mathcal{Q}$ otherwise $t_g = 1$.

V. Experimental Results

In this section, we first introduce the experimental settings, then the effectiveness of our proposed framework in both unsupervised and weakly supervised manners is presented through a set of comprehensive experiments. All the experiments are conducted on a 64-bit machine with four NVIDIA TITAN Xp GPUs. The proposed method is implemented with Pytorch deep learning framework [30]. The link to our code is at the github repository https://github.com/rollingstonezz/subgraph_anomaly_detection_icdm22.

A. Experimental Settings

1) Synthetic datasets: In order to examine the effectiveness of our proposed method in detecting different types

of anomalous subgraphs in a given graph, we first generate two sets of synthetic datasets, including a structural anomaly synthetic dataset and an attributive anomaly synthetic dataset. We provide a brief description of these two datasets as follows:

Structural anomaly synthetic dataset. We first apply several existing commonly used graph generator models to generate the *background graph*. In this paper, we investigate the following classic graph generators as *background graph*: (1) Erdös-Rényi (ER) model that can sample graphs with given node and edge number uniformly at random; (2) Watts-Strogatz (WS) model that can generate small-world properties graph; (3) Barabási-Albert (BA) model that can generate scale-free graphs; and (4) complete graphs. Then a small subgraph with a different structural property is inserted into the background graph as the ground truth anomaly subgraph. We extend previous works [2], [3] to insert two types of anomaly subgraphs: chain and dense graphs, to demonstrate the effectiveness of our proposed framework in detecting both sparse and dense anomaly subgraphs and the failure of previous benchmark models in this task.

Attributive anomaly synthetic dataset. We also generalize previous simulation studies on detecting disease outbreak [31]. Specifically, a random geometric graph model is first utilized to place N nodes uniformly at random in a rectangular domain. An edge between nodes is formed if a pair of nodes is within a spatial distance threshold. A small circle area that is set at random is considered the ground truth outbreak area, where the sub-network within this area is considered the anomaly subgraph. To simulate disease outbreaks, for all nodes in the normal region, their node attributes are sampled from a given Poisson distribution $P_n(x) = \frac{\lambda_n^x e^{-\lambda_n}}{x!}$. For nodes in the outbreak area, another Poisson distribution $P_a(x) = \frac{\lambda_a^x e^{-\lambda_a}}{x!}$ with a different expectation value λ_a is applied to a certain percentage of attributes.

2) Real-world datasets: To further evaluate the performance of our proposed method and comparison methods in real-world scenarios, nine public real-world attributed network datasets, including four citation network datasets, three social network datasets, one communication dataset, and one materials dataset, are utilized as benchmark datasets in our experiments. We provide a brief description of these real-world datasets as follows:

Citation networks. ACM [21], Cora, CiteSeer, and Pubmed [32] are four commonly used public citation network datasets. Each dataset contains bag-of-words representation of documents and citation links between the documents.

Social networks. Blog and Flickr [21] are two social networks formed by online users. All users are with a list of tags, and their following relationships form connections. Wiki [19] is another social network that describes co-editing relationships between editors on Wikipedia pages.

Communication network. Email [33] is an e-mail communication network of a large European research institution. Nodes indicate members of the institution. An edge between a pair of members indicates that they exchanged at least one email.

Background graph	Complete		ER random		WS small-world		BA scale-free	
Anomaly type	Dense	Chain	Dense	Chain	Dense	Chain	Dense	Chain
L1SUB	0.557	0.533	0.861	0.511	0.521	0.486	0.963	0.503
AMEN	0.554	0.976	0.601	0.632	0.654	0.522	0.711	0.501
DEEPFD	0.505	0.495	0.974	0.947	0.497	0.488	0.973	0.947
DOMINANT	0.409	0.486	0.598	0.611	0.589	0.701	0.598	0.456
DEEPSVDD	0.953	**1.000**	0.489	**1.000**	**1.000**	**1.000**	0.475	0.981
AS-GAE	**0.976**	**1.000**	**0.984**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

TABLE I: AUC scores of the structure anomaly synthetic datasets. The best performance on each dataset is in bold.

Material network. OMDB [34] is an open-access electronic structure database for three-dimensional organic crystals. Nodes indicate atoms and the edges are formed according to the distance between nodes.

The ground-truth anomalous labels are available in the datasets ACM, Blog, Flickr, and Wiki by previous studies [19], [21]. For the rest datasets, we further investigate the effectiveness of mining subgraph-level anomalies of AS-GAE by following the generation strategy used in previous works [21], [22], [35] to inject synthetic anomalous subgraphs into the original data manually. Specifically, for datasets excluding OMDB, we first randomly sample a subgraph in the given graph, then replace the attributes of the nodes in the sampled subgraph with random distant nodes' features for injected attributive anomalies or make the sampled subgraph fully connected for injected structural anomalies. For the OMDB dataset, we manually inject one impurity cell into the regular periodic lattice and consider the injected impurity cell as the ground truth anomaly subgraph. The basic statistics and the number of anomalies in datasets are summarized in Table II.

3) Benchmark models: We compare our proposed method with several existing unsupervised anomaly detection algorithms, including two non-deep learning methods and three state-of-the-art deep learning methods:

(1) **L1SUB** [11] is a method to detect small, anomalous graphs embedded into background networks. It computes the $L1$-norm of the eigenvectors of the graph modularity matrix and finds the subspace of the eigenvector with the abnormal variance.

(2) **AMEN** [13] is a method that proposed a measure to leverage both structural and attributive information to quantify internal consistency and external separability of nodes.

(3) **DEEPFD** [23] is a deep graph neural network method to detect group anomalies in a given network. It first generates the latent embeddings of each node by a graph autoencoder model. Then the anomalous subgraphs are extracted by a dense block detection algorithm DBSCAN.

(4) **DOMINANT** [21] is a deep graph autoencoder method to detect anomalies in an attributed network. It utilizes a graph neural network-based architecture for processing both graph structure and features.

(5) **DEEPSVDD** [4] is a deep one-class classification based anomaly detection method named deep SVDD. We generalize this method to do the task of subgraph anomaly detection in a most intuitive way. For each node in the graph, we select its k-hop subgraph and map it to a high dimensional sphere by a graph neural network. Then the most distant subgraph from the sphere will be considered the most anomaly subgraph.

4) Evaluation metrics: We utilize the AUC score, a widely used evaluation metric for anomaly detection tasks that mea-

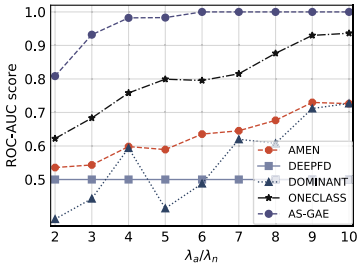

Fig. 5: The experimental results for the attribute anomaly synthetic dataset with varying ratio λ_a/λ_n of the expectation values of the normal and abnormal distribution.

sures the area under the plot of true positive rate against false positive rate, to quantify the performance of AS-GAE and comparison methods. Specifically, we report the AUC score of detecting the set of nodes in the anomalous subgraph, where a larger value indicates a higher detection performance.

B. Unsupervised learning results

1) Effectiveness results: We compare our proposed method AS-GAE with benchmark methods on both synthetic datasets and real-world datasets in an unsupervised learning manner. The comparison of AUC scores for the structural anomaly synthetic dataset is provided in Table I, by extensive combinations of different background graphs and inserted anomalous subgraphs. In Figure 5, we illustrate the results of experiments on attributive anomaly synthetic dataset, where we vary the division ratio λ_a/λ_n of the expectation values of normal and abnormal distributions for a robustness test. Finally, we provide the results of real-world datasets in Table II. We summarize our observations on the effectiveness of AS-GAE and comparison models below:

(1) The results demonstrate the strength of our proposed method by consistently achieving the best results in all eight structure anomaly synthetic tasks, all nine attributive anomaly synthetic tasks, and eight out of nine real-world datasets. Specifically, our results outperformed the benchmark models by over 31.7% on average for structure anomaly synthetic datasets, 41.6% on average for attributive anomaly synthetic datasets, and 16.1% on average for real-world datasets.

(2) It is also worth noting that the deep learning-based benchmark methods (DEEPFD, DOMINANT, DEEPSVDD, and AS-GAE) show a more competitive performance than the non-deep learning-based benchmark methods (L1SUB and AMEN), by over 15.6% on average for structure anomaly synthetic datasets, 13.8% on average for attributive anomaly synthetic datasets, and 12.9% on average for real-world datasets, which arguably indicates that non-deep learning methods have limited capability to effectively discriminate anomalies from graphs with complex structures or rich attributes.

(3) Our proposed method shows a stronger detection performance compared to other deep learning methods. A possible reason is that our method takes advantage of the dependence relationships information within the context of the graph to acquire a more competitive performance.

(4) As shown in Table I, in the structure anomaly synthetic datasets experiments, our proposed method consistently achieves superior anomaly detection performance with respect to different combinations of background graphs and anomaly subgraphs, which proves the robustness of our proposed method. In comparison, the DEEPFD method and DEEPSVDD method have significantly different performances on different background graphs. For example, the DEEPFD method shows strong detection performance when having a random graph or scale-free graph as a background graph, while the DEEPSVDD method has competitive performance when the background graph is a complete graph or small-world graph. In addition, the L1SUB method also achieves over 21.7% better performance on detecting dense subgraphs than detecting chain subgraphs.

2) Ablation study: Here we investigate the impact of the proposed components of AS-GAE. We first consider a variant **NO LA** that removes the *location-aware encoder* module, where it only uses a plain graph autoencoder to reconstruct the graph. To study the effectiveness of the proposed *supermodular graph scoring* module, we further construct a variant **NO SF** that substitutes the supermodular function with a simple modular function (where we use summation over nodes in this case). Due to the limitation of pages, we only present the results of six real-world datasets in Table III.

(1) Our full model has achieved the best performance on five out of six datasets. Specifically, the full model outperforms the variants **NO LA** and **NO SF** by 12.7% and 2.8% on average, respectively. The performance results validate that both *location-aware encoder* and *supermodular graph scoring* modules can benefit the task of subgraph anomaly detection.

(2) The detection performance drops significantly on the Wiki, Email, and OMDB datasets when we remove the location-aware module. We observe that these three datasets have significantly fewer dimensions of node attributes (average dimension $= 15.3$) than the other three networks (average dimension $= 1,878$), which may indicate that the *location-aware encoder* module plays a more critical role in detecting subgraph anomalies when the number of dimensions of node attributes is relatively small.

3) Sensitivity analysis: Here we investigate the impact of parameters on the effectiveness of AS-GAE.

(1) We first conduct the experiments to explore the impact of the size of the anomaly subgraph with respect to a given graph. Due to the limited space, we only show the result on the material dataset. We set the range of anomaly subgraph ratio from 2.5% to 25%. According to Table IV, the AUC scores show an upward and convergence trend with the decrease of anomaly subgraph ratio. The performance tends to drop significantly when the percentage of anomalies is larger than 10%. One possible reason is that when the rate of anomalies is high, it contradicts the rarity of anomalies.

(2) We then investigate the impact of the dimension D of latent embedding in our autoencoder. The results of varying the values of D from 8 to 256 on five real-world datasets are shown on the left of Figure 6. According to the results, we can observe that the performance is steady that the average standard deviation is only 1.4%. There is a peak value of

Dataset	Email	Cora	Citeseer	Pubmed	OMDB	Blog	Flickr	ACM	Wiki
Nodes	1,005	2,708	3,327	19,717	1,124	5,196	7,575	16,484	8,227
Edges	25,571	5,429	4,732	44,338	17,522	171,743	239,738	71,980	744,652
Features	42	1,433	3,703	500	4	8,189	12,047	8,337	0
Anomalies	57	106	197	340	84	300	450	600	217
L1SUB	0.523	0.486	0.563	0.575	0.591	0.485	0.502	0.530	0.463
AMEN	0.603	0.626	0.645	0.773	0.768	0.534	0.605	0.621	0.442
DEEPFD	0.572	0.658	0.704	0.776	0.846	0.497	0.500	0.501	0.499
DOMINANT	0.674	0.752	**0.832**	0.840	0.618	0.781	0.749	0.748	0.487
DEEPSVDD	0.730	0.703	0.693	0.622	0.938	0.635	0.642	0.734	0.453
AS-GAE	**0.753**	**0.829**	0.795	**0.925**	**0.980**	**0.784**	**0.764**	**0.751**	**0.561**

TABLE II: AUC scores on nine real-world datasets. The best performance on each dataset is in bold.

Dataset	Wiki	Email	Cora	CiteSeer	Pubmed	OMDB
No LA	0.485	0.564	0.806	0.791	**0.825**	0.504
No SF	0.557	0.729	0.799	0.697	0.817	0.974
Full	**0.561**	**0.753**	**0.829**	**0.795**	**0.825**	**0.980**

TABLE III: Ablation study results. Full refers to our complete model. NO LA refers to a variant that removes the location-aware module. NO SF refers to another variant that substitutes the supermodular function with a simple modular function. The best AUC score of each dataset is in bold.

Anomaly rate	1/4	1/5	1/10	1/15	1/20	1/40
AS-GAE	0.734	0.817	0.961	0.968	0.980	0.978

TABLE IV: Influence of the AUC scores versus anomalous rate, which varies from 1/4 to 1/40 on OMDB dataset.

performance when the latent dimension D is from 16 to 256. More interestingly, the OMDB and Email datasets peak around $D = 16$, while the Cora, Citeseer, and Pubmed datasets peak at $D \geq 64$. One possible reason is that the OMDB and Email datasets have significantly less node attribute dimensionality than the Cora, CiteSeer, and Pubmed datasets, which may require less latent representation dimensionality to correctly reconstruct normal regions.

(3) We further explore how the number C of sampled anchors in the location-aware autoencoder would impact the effectiveness of AS-GAE. We vary the values of C from 25 to 200 on five real-world datasets as shown on the right of Figure 6. According to the figure, we can observe that the size of datasets is correlated with the value of C for the peak performance. For the smaller size datasets such as Email $(1,005)$ and OMDB $(1,124)$ their performances peak at $C \leq 100$, while peak performances occur at $C \geq 150$ for larger datasets such as CiteSeer $(3,327)$ and Pubmed $(19,717)$.

C. Weakly supervised learning results

In this part, we conduct experiments to examine the effectiveness of our generalized method in a weakly supervised learning manner. Specifically, given the anomalous subgraph of each dataset as the ground truth, we sample a larger subnetwork by random walk as the scale of the provided node-set $\mathcal{Q}$. We use the ratio value $q = \frac{|\mathcal{Q}|}{|g|}$ to measure the relative size of the given scale $\mathcal{Q}$, where $|g|$ is the size of the given ground truth anomaly subgraph. For each dataset, we vary the value of q from 2 to 12. The experimental results on five real-world datasets are reported in Figure 7. We can observe that detection performance continues to grow on all datasets as the given range is shrink. Specifically, the performance grows by 4.1% on average when the scale ratio q changes from 12 to 2. In addition, it is worth noting that

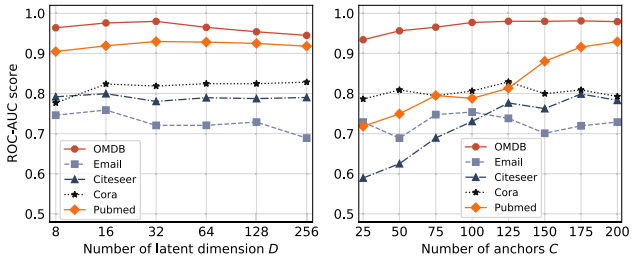

Fig. 6: Left: The sensitivity study for the number of latent dimensions D versus AUC scores on five real-world datasets. Right: The sensitivity study for the number of anchors C versus AUC scores on five real-world datasets.

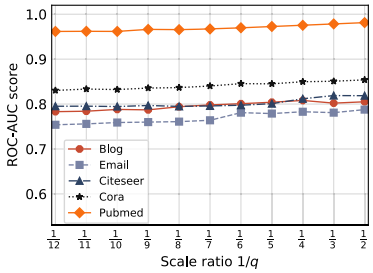

Fig. 7: The weakly supervised learning experiments on five real-world datasets with varying scale rates q.

the performance scores are almost converged to the values in the unsupervised learning setting when the given scale is comparable to the whole graph. The experimental results demonstrate that our proposed method effectively utilized the provided scale to further improve the anomaly detection power under our framework.

D. Training time analysis

The training time of AS-GAE under both unsupervised learning and weakly supervised learning settings are given in Table V with respect to a varying number of nodes and a fixed value of average node degree. The results indicate that the training time of AS-GAE increases almost linearly as the number of nodes in the graph increases. Furthermore, the training time of our framework in unsupervised learning and weakly supervised learning settings are comparable and grow at the same level.

N	1,000	2,000	3,000	4,000	5,000	6,000	7,000	8,000	9,000	10,000
Un	29.77	48.94	69.72	92.59	117.31	148.92	188.52	235.87	278.32	334.65
WS	31.46	53.89	77.51	101.22	130.21	163.54	206.54	256.31	300.45	362.42

TABLE V: The training time of AS-GAE respect to varying number of nodes N. Un and WS is short for unsupervised learning and weakly supervised learning. (Unit: second).

VI. Conclusion

This paper focuses on the crucial problem of detecting anomalous subgraphs from a given network under unsupervised learning or weakly supervised learning settings. The proposed framework deep **A**nomalous **S**ub**g**raph **A**uto**e**ncoder (**AS-GAE**) effectively addresses the unique challenges in anomaly subgraph detection by utilizing a *location-aware graph autoencoder* module to uncover the relative anomalous areas, and then a *supermodular graph anomalies quantification* module is applied to assign a reasonable anomaly score for the subgraphs in the built residual graph according to the reconstruction results of the autoencoder. Extensive experimental results on both synthetic and real-world datasets demonstrate the outstanding detection power of our model and the effectiveness of each module in our proposed framework.

Acknowledgment

This work was supported by the NSF Grant No. 1755850, No. 1841520, No. 2007716, No. 2007976, No. 1942594, No. 1907805, a Jeffress Memorial Trust Award, Amazon Research Award, NVIDIA GPU Grant, and Design Knowledge Company (subcontract number: 10827.002.120.04).

References

[1] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[2] M. Gupta, A. Mallya, S. Roy, J. H. Cho, and J. Han, "Local learning for mining outlier subgraphs from network datasets," in *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 2014.

[3] F. Chen, B. Zhou, A. Alim, and L. Zhao, "A generic framework for interesting subspace cluster detection in multi-attributed networks," in *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2017, pp. 41–50.

[4] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*. PMLR, 2018.

[5] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 665–674.

[6] Q. Wu, F. M. Wong, Y. Li, Z. Liu, and V. Kanade, "Adaptive reduced rank regression," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4103–4114, 2020.

[7] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[8] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[9] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, and L. Akoglu, "A comprehensive survey on graph anomaly detection with deep learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[10] A. Liu, Q. Wu, Z. Liu, and L. Xia, "Near-neighbor methods in random preference completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4336–4343.

[11] B. Miller, N. Bliss, and P. J. Wolfe, "Subgraph detection using eigenvector l1 norms," in *Advances in Neural Information Processing Systems*. Citeseer, 2010, pp. 1633–1641.

[12] J. Sharpnack, A. Krishnamurthy, and A. Singh, "Near-optimal anomaly detection in graphs using lovász extended scan statistic," in *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, 2013, pp. 1959–1967.

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

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

[15] Z. Zhang and L. Zhao, "Representation learning on spatial networks," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[16] S. Wang, X. Guo, and L. Zhao, "Deep generative model for periodic graphs," *arXiv preprint arXiv:2201.11932*, 2022.

[17] Q. Wu, W.-L. Hsu, T. Xu, Z. Liu, G. Ma, G. Jacobson, and S. Zhao, "Speaking with actions-learning customer journey behavior," in *2019 IEEE 13th International conference on semantic computing (ICSC)*. IEEE, 2019, pp. 279–286.

[18] Q. Wu, C. G. Brinton, Z. Zhang, A. Pizzoferrato, Z. Liu, and M. Cucuringu, "Equity2vec: End-to-end deep learning framework for cross-sectional asset pricing," in *Proceedings of the Second ACM International Conference on AI in Finance*, 2021, pp. 1–9.

[19] Y. Wang, J. Zhang, S. Guo, H. Yin, C. Li, and H. Chen, "Decoupling representation learning and classification for gnn-based anomaly detection," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[20] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 315–324.

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

[22] Y. Zheng, M. Jin, Y. Liu, L. Chi, K. T. Phan, and Y.-P. P. Chen, "Generative and contrastive self-supervised learning for graph anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[23] H. Wang, C. Zhou, J. Wu, W. Dang, X. Zhu, and J. Wang, "Deep structure learning for fraud detection," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 567–576.

[24] Y. She and A. B. Owen, "Outlier detection using nonconvex penalized regression," *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 626–639, 2011.

[25] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, 2006.

[26] H. Tong and C.-Y. Lin, "Non-negative residual matrix factorization with application to graph anomaly detection," in *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 2011, pp. 143–153.

[27] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[28] S. Wang, Y. Du, X. Guo, B. Pan, Z. Qin, and L. Zhao, "Controllable data generation by deep learning: A review," *arXiv preprint arXiv:2207.09542*, 2022.

[29] B. W. Dolhansky and J. A. Bilmes, "Deep submodular functions: Definitions and learning," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[31] M. Kulldorff, Z. Zhang, J. Hartman, R. Heffernan, L. Huang, and F. Mostashari, "Benchmark data and power calculations for evaluating disease outbreak detection methods," *Morbidity and Mortality Weekly Report*, pp. 144–151, 2004.

[32] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[33] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local higher-order graph clustering," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 555–564.

[34] S. S. Borysov, R. M. Geilhufe, and A. V. Balatsky, "Organic materials database: An open-access online database for data mining," *PloS one*, vol. 12, no. 2, p. e0171501, 2017.

[35] G. Pelosi, A. Cocchi, and S. Selleri, "Electromagnetic scattering from infinite periodic structures with a localized impurity," *IEEE Transactions on Antennas and Propagation*, vol. 49, no. 5, pp. 697–702, 2001.