IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 July 2022, accepted 21 September 2022, date of publication 3 October 2022, date of current version 27 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3211306

SURVEY

Graph Anomaly Detection With Graph Neural Networks: Current Status and Challenges

HWAN KIM[1], BYUNG SUK LEE[2], WON-YONG SHIN[3,4], (Senior Member, IEEE), AND SUNGSU LIM[1], (Member, IEEE)

[1]Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, Republic of Korea
[2]Department of Computer Science, University of Vermont, Burlington, VT 05405, USA
[3]Department of Computational Science and Engineering, School of Mathematics and Computing, Yonsei University, Seoul 03722, Republic of Korea
[4]Department of Artificial Intelligence, Pohang University of Science and Technology (POSTECH), Pohang 37673, Republic of Korea

Corresponding author: Sungsu Lim (sungsu@cnu.ac.kr)

This work was supported in part by NRF funded by the Korean Government (MSIT) under Grant 2021R1A2C3004345; in part by IITP funded by the Korean Government (MSIT) through the Artificial Intelligence Convergence Innovation Human Resources Development (Chungnam National University) under Grant RS-2022-00155857; and in part by BK21 FOUR Program by Chungnam National University Research Grant, in 2022.

ABSTRACT Graphs are used widely to model complex systems, and detecting anomalies in a graph is an important task in the analysis of complex systems. Graph anomalies are patterns in a graph that do not conform to normal patterns expected of the attributes and/or structures of the graph. In recent years, graph neural networks (GNNs) have been studied extensively and have successfully performed difficult machine learning tasks in node classification, link prediction, and graph classification thanks to the highly expressive capability via message passing in effectively learning graph representations. To solve the graph anomaly detection problem, GNN-based methods leverage information about the graph attributes (or features) and/or structures to learn to score anomalies appropriately. In this survey, we review the recent advances made in detecting graph anomalies using GNN models. Specifically, we summarize GNN-based methods according to the graph type (i.e., static and dynamic), the anomaly type (i.e., node, edge, subgraph, and whole graph), and the network architecture (e.g., graph autoencoder, graph convolutional network). To the best of our knowledge, this survey is the first comprehensive review of graph anomaly detection methods based on GNNs.

INDEX TERMS Dynamic graph, graph anomaly detection, graph neural network, node anomaly, static graph.

I. INTRODUCTION

A graph is an effective data structure for efficiently representing and extracting complex patterns of data and is used widely in numerous areas like social media, e-commerce, biology, academia, communication, and so forth. Data objects represented in a graph are interrelated, and the objects are typically represented as nodes and their relationships as edges between nodes. The structure of a graph refers to how the nodes are related via individual edges, and can effectively represent even far-reaching relationships between nodes. Moreover, graphs can be enriched semantically by augmenting their structural representations with attributes of nodes and/or edges.

Anomaly detection is the process to identify abnormal patterns that significantly deviate from patterns that are typically observed. This is an important task with increasing needs and applications in various domains. There have been significant research efforts on anomaly detection since Grubbs et al. [1] first introduced the notion of anomaly (or outlier). Since then, with the advancement of graph mining over the past years, graph anomaly detection has been drawing much attention [2], [3].

Early work on graph anomaly detection has been largely dependent on domain knowledge and statistical methods, where features for detecting anomalies have been mostly

The associate editor coordinating the review of this manuscript and approving it for publication was Zhipeng Cai.

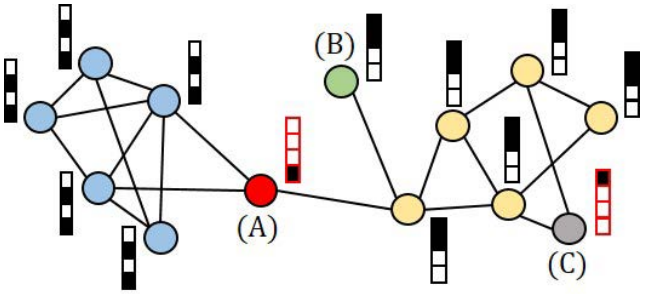

FIGURE 1. Example of graph anomaly detection. Nodes A and C are detected anomalous attribute-wise. Nodes A and B are detected anomalous structure-wise (as they do not belong to any community). Using GNN detects node A to be anomalous both attribute-wise and structure-wise.

handcrafted. This handcrafted detection task is naturally very time-consuming and labor-intensive. Furthermore, real-world graphs often contain a very large number of nodes and edges labeled with a large number of attributes, and are thus large-scale and high-dimensional. To overcome the limitations of the early work, considerable attention has been paid to deep learning approaches recently when detecting anomalies from graphs [4]. Deep learning's multi-layer structure with non-linearity can examine large-scale high-dimensional data and extract patterns from the data, thereby achieving satisfactory performance without the burden of handcrafting features [5], [6].

More recently, graph neural networks (GNNs) have been adopted to efficiently and intuitively detect anomalies from graphs due to the highly expressive capability via the message passing mechanism in learning graph representations (e.g., [7] and [8]). With GNNs, learning and extracting anomalous patterns from graphs, even those with highly complex structures or attributes, are relatively straightforward as GNN itself handles a graph with attributes as the input data [9]. The state-of-the-art graph anomaly detection approaches [7], [10] combine GNN with existing deep learning approaches, in which GNN captures the characteristics of a graph and deep learning captures other types of information (e.g., time). Fig. 1 illustrates an example of graph anomaly detection with GNN. Suppose that nodes (A) and (C) are detected anomalous in terms of the node attributes, and nodes (A) and (B) are detected anomalous in terms of the graph topology. Then, only node (A) should be detected anomalous if both node attributes and graph topology are taken into account together as anomalous factors. GNN models enable us to detect such anomalies by examining both graph topology and node attributes simultaneously.

In this survey, we provide an overview of GNN-based approaches for graph anomaly detection and review them primarily by the types of graphs, namely static graphs and dynamic graphs. Compared with other surveys on related topics — on graph anomaly detection (in general) [2], [3], graph anomaly detection specifically using deep learning [11], [12], and general anomaly detection using deep learning [13], [14] — this survey aims to touch on the unique angle of graph anomaly detection *using GNN models*. Given the significance of GNN and the active ongoing research efforts for its use in graph anomaly detection, it is our conviction that a comprehensive survey on this particular topic is timely and beneficial to the research community.

Fig. 2 shows the timeline of the surveyed methods. The survey in Section III is organized according to the classification framework used by other surveys on graph anomaly detection [2], [3], [11], [12]. This area is still new, and yet the published methods cover a broad range of graphs (static versus dynamic, plain versus attributed) and anomaly types (structure, node, edge, subgraph) although the distribution of the study topics seems skewed toward node anomalies in static graphs. Additionally, in Section IV, we share our opinions on several promising opportunities and challenges pertaining to graph anomaly detection using GNN.

II. GNN FRAMEWORKS

GNNs have been widely used as an effective method to extract useful features, especially from attributed networks, while performing graph representation learning (also known as network embedding) through the message passing mechanism. Model architectures of GNNs have been actively developed, and include, but not limited to, graph convolutional network (GCN) [15], GraphSAGE [16], and graph attention network (GAT) [17]. The main idea of such GNN models is message passing, which aggregates individual features from h-hop neighbors. The message passing mechanism of GNNs can be expressed as follows [18]:

$$\mathbf{m}_u^{(k-1)} \leftarrow \text{AGGREGATE}^{(k-1)}(\{\mathbf{h}_v^{(k-1)}, \forall v \in \mathcal{N}(u) \cup \{u\}\}),$$
$$\mathbf{h}_u^{(k)} \leftarrow \text{UPDATE}^{(k-1)}\left(\mathbf{h}_u^{(k-1)}, \mathbf{m}_u^{(k-1)}\right),$$

where AGGREGATE and UPDATE are arbitrary differentiable functions and $\mathbf{m}_u$ is the message that is aggregated from the neighborhood $\mathcal{N}(u)$ of node u as well as the node itself. At the k-th layer of a GNN, the previous embedding $\mathbf{h}_u^{(k-1)}$ of u is updated to $\mathbf{h}_u^{(k)}$ by aggregating neighborhood information via message passing. The initial embeddings are set to the features of nodes, and the output at the final layer corresponds to the embeddings of the underlying GNN model.

GCN originally presented an effective network representation model that naturally combines the network structure and node attributes in the learning process. Subsequently, Hamilton et al. [16] introduced GraphSAGE, which samples nodes for message aggregation in a neighbor to handle large graphs. GAT learns the weights of each node in a neighbor during message passing.

There have been follow-up studies that use alternative model architectures. For example, Wang et al. [19] introduced a graph stochastic neural network (GSNN) to alleviate the inefficiency of a deterministic way used in existing GNN models. Most existing GNN approaches embed networks into either the Euclidean or hyperbolic space. These approaches have the issue of inflexibility in modeling a complex graph topology. To address this issue, Zhu et al. [20] proposed a

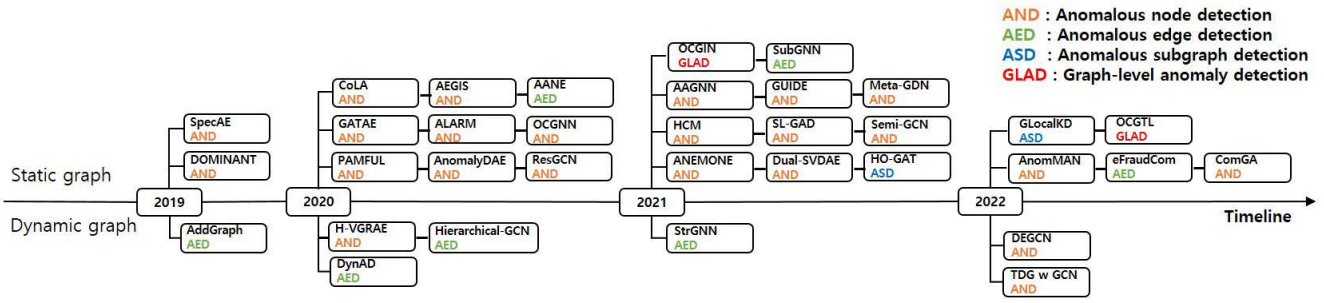

FIGURE 2. Timeline of graph anomaly detection methods using GNN models. The methods above the timeline are for static graphs, and those below are for dynamic graphs.

graph geometry interaction learning (GIL) algorithm, which employs *both* Euclidean and hyperbolic spaces.

III. GNN-BASED GRAPH ANOMALY DETECTION

The surveyed methods are categorized by the graph types (static versus dynamic), anomaly types (nodes, edges, subgraphs, and whole graphs), and network architectures. Table 1 outlines the organization of our survey.

A. GNN-BASED STATIC GRAPH ANOMALY DETECTION

A majority of research efforts on static graph anomaly detection addressed node anomalies, and only a few addressed edge anomalies and subgraph anomalies.

1) ANOMALOUS NODE DETECTION

Detecting anomalous *nodes* using GNNs was carried out mostly in attributed graphs. That is, each of GNN-based methods extracts node attribute information as well as structural information from a static attributed graph and evaluates the anomaly score of nodes using a certain scoring algorithm. Various GNN-based approaches have been proposed to effectively extract the necessary features from attributed graphs. We categorize these methods according to their network architectures and then briefly describe the key ideas behind each method.

On the basis of the structural information, anomalous nodes can be further divided into the following three types: global anomalies, structural anomalies, and community anomalies [12]. Global anomalies are referred to as deviated node attributes in the graph; structural anomalies are referred to as deviated structural information in the graph; and community anomalies are referred to as both deviated node attributes and structural information in the same community.

Note that a fair number of GNN-based approaches are built upon the graph autoencoder (GAE) framework instantiated with either GCN or GAT. For anomalous node detection in static graphs, we review GNN-based GAE methods as well as standalone GNN methods.

a: GCN-BASED GAE FRAMEWORK

GAE has been most widely used for detecting graph anomalies. For anomalous node detection in static graphs, while existing GAEs mostly use GCN in the encoder, they adopt their own decoders depending on the perspective on which each method focuses. Additionally, an anomaly scoring function, following the decoder, identifies the abnormality by scoring each node on the basis of reconstruction errors from the decoder.

It is likely that graphs are sparse in real-world situations. Moreover, complex interactions between individual nodes are difficult to capture due to the non-linear characteristics. To address these issues, *deep anomaly detection on attributed networks (DOMINANT)* [7] detected the global and structural anomalies using GCN as the encoder while discovering node representations from a given attributed graph. Decoders in DOMINANT were designed in the sense of reconstructing the original graph structure and nodal attributes.

To capture complex interactions of GAE-based approaches in high-dimensional graphs, Zhang et al. [21] proposed a one-class classification-based framework, called *dual support vector data description based autoencoder (Dual-SVDAE)*, which aims to learn the hypersphere boundary on the normal nodes' latent space based on the graph structure and attributes in order to detect global and structural anomalies. Specifically, its AE has a dual encoder–decoder architecture — 1) a GCN-based encoder and an inner product-based decoder for structural features and 2) a multilayer perceptron (MLP)-based encoder–decoder for attributes. The nodes outside of the hypersphere space are regarded as anomalous nodes.

Subsequently, Yuan et al. [22] presented a dual GAE framework, *higher-order structure based anomaly detection (GUIDE)*, which leverages the higher-order structures in modeling the complex interaction to detect global and structural anomalies. Specifically, it consists of the attribute AE for attribute embeddings and the structure AE for structural embeddings based on the attention mechanism.

Since GCNs learn node representations via message passing from the neighbors, they could over-smooth the representations, making anomalous nodes less distinguishable from the normal nodes. To alleviate the over-smoothing issue, *spectral AE (SpecAE)* [8] was presented an approach for detecting global and community anomalies by employing the Laplacian to calculate the inter-node distances and to

TABLE 1. Taxonomy and summary of graph anomaly detection methods using GNN models.

Graph type	Anomaly type	Network architecture	Method	Summary (key issue addressed → solution)
Static graph	Node anomaly	GCN-based GAE	DOMINANT [7]	Complex interactions, sparsity, non-linearity → GCN-based encoder
			Dual-SVDAE [21]	Overfitting for normal & abnormal → hypersphere embedding space
			GUIDE [22]	Complex interactions → higher-order structure decoder
			SpecAE [8]	Over-smoothing issue → tailored embedding space
			ComGA [23]	Over-smoothing issue → community-specific representation
			ALARM [24]	Heterogeneous attributes → multiple GCN-based encoders
			AnomMAN [25]	Heterogeneous attributes → multiple GCN-based encoders
			SL-GAD [26]	Contextual information → subgraph sampling & contrastive learning
		GCN alone	Semi-GCN [27]	Label information → semi-supervised learning by GCN
			HCM [28]	Label & contextual information → hop-count prediction model
			ResGCN [29]	Over-smoothing issue → GCN with residual-based attention
			CoLA [30]	Targeting issue of GAE → contrastive self-supervised learning
			ANEMONE [31]	Contextual information → multi-scale contrastive learning
			PAMFUL [32]	Contextual information → pattern mining algorithm with GCN
		GAT-based GAE	AnomalyDAE [33]	Complex interactions → GAT-based encoder
			GATAE [34]	Over-smoothing issue → GAT-based encoder
			AEGIS [35]	Handling unseen nodes → generative adversarial learning with GAE
		Other GNN-based model	OCGNN [36]	Targeting issue of GAE → GNN with hypersphere embedding space
			AAGNN [37]	Targeting issue of GAE → GNN with hypersphere embedding space
			Meta-GDN [38]	Hard work to label anomalies → meta-learning with auxiliary graphs
	Edge anomaly	GCN-based GAE	AANE [39]	Noise or adversarial links → GAE with a loss for anomalous links
			eFraudCom [40]	Fraud detection → heterogeneous graph and representative data sampling
		GCN alone	SubGNN [41]	Fraud detection → GIN and extracting and relabeling subgraphs
	Subgraph anomaly	GCN alone	GLocalKD [42]	Graph-level anomalies → joint learning global & local normality
		GAT-based GAE	HO-GAT [43]	Abnormal subgraphs → hybrid-order attention with motif instances
	Graph-level anomaly	GCN alone	OCGIN [44]	Graph-level anomaly detection → graph classification with GIN
			OCGTL [45]	Hypersphere collapse → set of GNNs for embedding
Dynamic graph	Edge anomaly	GCN and GRU	AddGraph [10]	Long-term patterns → temporal GCN with attention-based GRU
			DynAD [46]	Long-term patterns → temporal GCN with attention-based GRU
			Hierarchical-GCN [47]	Dynamic data evaluation → temporal & hierarchical GCN
			StrGNN [48]	Structural change → mining unusual temporal subgraph structures
	Node anomaly	GCN & DRNN-based GAE	H-VGRAE [49]	Anomalous nodes → modeling stochasticity and multi-scale ST dependency
		GCN and GRU	DEGCN [50]	To capture node- and global-level patterns → DGCN and GGRU
		GCN alone	TDG with GCN [51]	Malicious connections on traffic → extracting TDGs

simultaneously embed attributes and relations in the underlying graph. It used the GCN encoder and the deconvolutional decoder. Similarly, Luo et al. [23] proposed *community-aware attributed graph anomaly detection framework (ComGA)* to detect global and community anomalies by propagating community-specific representations of each node with a gateway. In a community detection module, communities were encoded and decoded for community representations. In a tailored deep GCN (tGCN) module, the representations, structural information, and nodal attributes were used as input of GCN layers. The gateway concatenates the community feature, structure, and attribute. Finally, in an anomaly detection module, the well-designed structure decoder and attribute decoder reconstruct the output from the tGCN module.

Most existing studies do not carefully take multi-view properties into consideration. The complementary information from multi-view attributes can help detect anomalies more efficiently. In a multi-view attributed network, each node is affiliated with a set of multi-dimensional features (attributes), which can be represented by k distinct feature spaces along with k views. By making use of the multi-view attributes, Peng et al. [24] proposed a *deep multi-view framework for anomaly detection (ALARM)* for detecting global and structural anomalies. Specifically, it employs multiple GCNs to embed a multi-view attributed graph as an encoder and uses two decoders, each of which reconstructs the graph structure and attributes.

Chen et al. [25] proposed a framework, called *anomaly on multi-view attributed networks on attributed networks (AnomMAN)*, which aims to detect community anomalies. It first decomposes multi-view attributed graphs into k-subgraphs and encodes the subgraphs with GCNs. Then, the representations from different subgraphs are concatenated using the attention mechanism and reconstructed by a decoder with respect to both structures and attributes of the multi-view graphs.

The existing GAE-based approaches do not effectively extract the contextual information (e.g., neighboring nodes and subgraphs). In addition, they focused primarily on node-level representation learning. To address these issues, Zheng et al. [26] proposed *self-supervised learning for graph anomaly detection (SL-GAD)*, where the GCN-based encoder and decoder are used to capture node- and subgraph-level features along with a self-supervised learning strategy for community anomalies. Specifically, a target node and subgraphs centered on each target node (i.e., contextual information) were sampled and fed into the encoder. Afterward, generative attribute reconstruction and contrastive learning modules were leveraged to extract the useful features in a self-supervised fashion. The GCN-based decoder reconstructs the node attributes, and then the anomaly score is calculated on the basis of both modules.

b: GCN FRAMEWORK

Since labeling anomalies is labor-intensive and costly, many efforts have focused on unsupervised learning. Nonetheless,

labeling information can help enhance the model performance. For this reason, Kumagai et al. [27] proposed a simple yet effective GCN-based method, called *semi-GCN*, which is capable of embedding nodes into a hypersphere space while taking advantage of the structure and attribute features of a graph by stacking graph convolutional layers in order to detect global anomalies.

To address the issue above in unsupervised learning and leverage global context information, Huang et al. [28] proposed the *hop-count based model (HCM)* based on self-supervised learning with the *PairwiseDistance* algorithm [52] in which hop counts of node pairs are leveraged as labels [52]. Specifically, it consists of a preprocessing module for labeling, GCNs for learning the representation of graphs, and the MLP module for calculating the anomaly scores.

Pei et al. [29] proposed *residual GCN (ResGCN)* to alleviate the sparsity and over-smoothing issues in modeling attributed graphs. It uses the attention mechanism with residual information, modeled by MLP layers, in a graph.

Basically, GAE can bring sub-optimal performance on the anomaly detection task. To tackle this problem raised by GAE-based methods, Liu et al. [30] proposed *contrastive self-supervised learning (CoLA)*, which concentrates on modeling the relations between nodes and their neighboring subgraphs to detect the community anomalies. More precisely, the well-organized node and its subgraph pairs were carefully sampled, and then the GCN-based contrastive learning model embedded the pairs. Subsequently, a readout module computed discriminative scores, while showing the abnormality of a target node of the pairs.

Jin et al. [31] presented *graph anomaly detection framework with multi-scale contrastive learning (ANEMONE)*, which detects community anomalies, to overcome the sub-optimality based on a multi-scale contrastive learning model. In the model, two GCN-based contrastive networks are trained at a patch-level (i.e., node–node) and a context-level (i.e., node–egonet) to catch the information in multiple graph scales. Then, a statistical anomaly estimator computes the anomaly scores of target nodes on the basis of the patch- and context-level scores.

There were a few studies that jointly take into account pattern mining and GNNs for graph anomaly detection. For example, Zhao et al. [32] introduced a GCN-based framework, *pattern mining and feature learning (PAMFUL)*, which employs pattern mining to guide the GCN in aggregating local information in order to catch global patterns in the sense of detecting the global and structural anomalies.

c: GAT-BASED GAE FRAMEWORK

While the GCN-based framework exhibits good performance in anomalous node detection, its simple aggregation operation, which can cause the over-smoothing issue, limits the GCN's ability to extract useful features. To overcome this, GAT-based approaches have been developed as an alternative.

Earlier studies did not take into account the complex interactions between nodes and attributes due to the shallow learning architecture. To solve this issue, Fan et al. [33] proposed a *deep joint representation learning framework for anomaly detection through a dual AE (AnomalyDAE)*, which captures the complex interactions between the structure and attributes to detect global and structural anomalies. Specifically, it consists of a GAT-based structure encoder and an attribute encoder, each employing an inner product decoder.

Moreover, to alleviate the over-smoothing issue in GCNs, the *graph attention-based AE (GATAE)* [34] embedded the input attributed graph by using multiple graph attention layers as an encoder in order to detect global and structural anomalies. An inner product decoder was used for reconstruction of the graph structure, and a decoder with the same architecture as its encoder was used to reconstruct node attributes.

Since most efforts have not yet handled the unseen nodes, the underlying model may be unnecessarily retrained for newly discovered nodes in graphs. To address the issue, *adversarial graph differentiation network (AEGIS)* [35] detected the community anomalies using an attention-based graph differentiation network (GDN). Specifically, an encoder based on GDNs first embeds the graphs at node- and neighborhood-levels, and then a GDN-based decoder reconstructs the input graphs along with the anomaly ranking list.

d: OTHER GNN FRAMEWORKS

Wang et al. [36] proposed *one-class graph neural network (OCGNN)*, which aims at detecting the global anomalies. OCGNN combines the representation capability of GNNs with a hypersphere learning objective function, which enables us to learn a hypersphere boundary and detect points that are outside the boundary as anomalous nodes.

Despite significant efforts, it remains an open challenge to learn suitable communities for effective anomaly detection. Additionally, defining anomalies is nontrivial since the deviation patterns of anomalies are not easily disclosed. To address these issues, Zhou et al. [37] introduced *abnormality-aware GNN (AAGNN)*, which employs a subtractive aggregation method to decide the pattern deviation between a target node and its nearby nodes along with hypersphere learning for community anomaly detection. Specifically, by subtracting the feature of a target node from the aggregated feature in the k-hop neighborhood, abnormality of the node is determined for further embedding into the hypersphere space.

Due to the expensive labeling cost, there have been only a few methods designed in a supervised manner. Ding et al. [38] introduced a supervised GNN-based framework, *few-shot network anomaly detection via cross-network meta-learning (Meta-GDN)*, which aims to detect the global anomalies on the target graph by using very limited knowledge of ground-truth anomalies from auxiliary graphs with the GCN-based encoder. Meta-GDN was built upon graph deviation networks (GDNs), which first score each node by using a GCN-based anomaly score learner and then use the deviation loss to enforce the model to give high anomaly scores to nodes whose features deviate from normal nodes significantly.

2) ANOMALOUS EDGE DETECTION

Anomalous edges generally represent different or atypical interactions between nodes in a graph. Research on such anomalous edge detection in a static graph has been relatively limited.

Duan et al. [39] proposed *anomaly aware network embedding (AANE)*, which was designed for plain graphs and implemented using the GCN-based GAE framework. This method adjusts the fitting loss and the ''anomaly-aware'' loss, which consists of both the deviation loss and the removal loss. The probability from the loss functions is the score of an edge. An edge with a lower probability is more likely to be an anomalous edge.

Song et al. [41] proposed *subgraph-based framework (SubGNN)* for fraud detection. Sub-graphs near target edges are extracted and relabeled for entity independence. The proposed relational graph isomorphism network (R-GIN) learns the features for precise fraud detection.

Zhang et al. [40] proposed a *competitive graph neural network (CGNN)*-based fraud detection system (eFraudCom) to detect fraudulent behaviors on an e-commerce platform. The CGNN is a GCN-based GAE system. The eFraudCom system consists of a data processor and a fraud detector. Specifically, in the data processor, representative normal data were sampled, and a heterogeneous graph with the sampled normal data and the rest was generated; and in the fraud detector, neighbors in the graph were sampled and anomalous edges were detected by the CGNN.

3) ANOMALOUS SUBGRAPH DETECTION

Anomalous subgraph detection is far more challenging than anomalous node or edge detection. For one noticeable thing, nodes and edges in an anomalous subgraph may be considered normal in themselves. Moreover, a subgraph can be very diverse in its structure and size. Presumably due to this challenge, there exist only a limited number of studies in the literature. We have found one GAT-based GAE method.

Ma et al. [42] proposed a GCN method, *global and local knowledge distillation (GLocalKD)*, which extracts graph- and node-level representations from a set of graphs by using two different stacked GCN modules to detect locally or globally anomalous subgraphs. Specifically, it consists of two GCNs, namely, a random target network and a predictor network, that have the same GCN architecture and incur the same distillation losses. GLocalKD learns the graph normality at a fine-grained level by using the predictor network to predict the graph- and node-level representations produced by the random target network.

Huang et al. [43] proposed a GAT-based GAE framework, called *hybrid-order graph attention network (HO-GAT)*, to detect anomalous nodes and subgraphs simultaneously by using the structure and attributes of abnormal nodes and subgraphs. To define the most representative subgraph, the motif — the widely studied higher-order structure characterized as a densely connected subgraph — was leveraged. To model the relations between nodes and motifs, HO-GAT effectively measured the significance of four relations: a node to another node, a node to a motif, a motif to a node, and a motif to another motif. Specifically, the graph attention layer was used in the encoder in order to capture information of the four relations mentioned above. Additionally, two individual decoders were employed to reconstruct the structure and attributes.

4) GRAPH-LEVEL ANOMALY DETECTION

Graph-level anomaly detection (GLAD) finds graphs that are notably different from most of the graphs in a set of graphs. There have been limited studies on addressing this task. As mentioned above, GNN-based methods have proven its capabilities in graph-structured data, and several methods such as OCGNN [36] and DOMINANT [7] have successfully applied GNN to graph anomaly detection tasks. Recently, there have been attempts to adapt GNN-based graph classification methods to GLAD tasks.

Zhao and Akoglu [44] proposed *one-class graph-level anomaly detector (OCGIN)* based on Graph Isomorphism Network (GIN) [53], which has drawn attention to barely studied GLAD tasks while improving the detection performance. GIN [53] generalizes the Weisfeiler-Lehman (WL) test [54], which can efficiently distinguish a broad class of graphs [55], and achieves powerful classification capability among GNNs. Specifically, it optimizes the one-class Deep Support Vector Data Description (deep SVDD) objective function [56] at the output layer of a GIN model.

To further improve the performance and overcome the hypersphere collapse where the deep one-class objective encourages all graph embeddings in the training data to concentrate within a hypersphere, Qiu et al. [45] presented a new GNN-based approach, *one-class graph transformation learning (OCGTL)*, which integrates advantages of deep one-class classification (deep OCC) [44], [56] and self-supervised anomaly detection with learnable transformations [57]. Specifically, OCGTL consists of a set of GNNs to embed its input graphs into a latent space. Afterward, the various embeddings were used for training to be close to the reference embedding.

B. GNN-BASED DYNAMIC GRAPH ANOMALY DETECTION

Unlike a static graph, *temporality* is an important factor in a dynamic graph whose structure or attributes change over time. Recently, various methods for detecting anomalies in graphs changing or evolving over time have been proposed based on graph communities, compression, decomposition, distance metrics, and probabilistic modeling of graph features [2]. There are several approaches proposed for dynamic graphs where GCN is combined with deep learning methods that are suitable for temporal processing, such as recurrent neural network (RNN), gated recurrent unit (GRU), and transformer. A few studies addressed detecting anomalies in edges or in nodes of a dynamic graph. There is no study that addresses detecting anomalous subgraphs yet.

1) ANOMALOUS EDGE DETECTION

The basic concept of using GCN and GRU together is that GCN extracts features from a graph and GRU captures historical information useful for anomaly detection. Alongside the fact that GCNs do not consider the temporal factors in dynamic graphs, *anomaly detection in dynamic graph (AddGraph)* [10] combined GCN and GRU, which enables us to integrate long-term and short-term patterns in a window in order to describe the normal edges by using structural, attribute, and temporal information. Specifically, it leveraged GCN to process the previous node state with edges in the current graph by harnessing the structural and attribute features of nodes. The nodes' states in a short window were used as the short-term information. The GCN output and the shot-term information were then combined in a contextual attention-based GRU to extract the hidden state of nodes in order to calculate the anomaly probability of an edge.

Note that Zheng et al.'s work [10] requires the entire set of nodes as the input and, thus, is not able to effectively manage newly added nodes. To address this issue, Zhu et al. [46] proposed *anomaly detection in dynamic network (DynAD)*, which uses evolving GCNs, consisting of multiple layers at each time, and attention-based GRU for adaptive parameter learning.

There are multi-level data structure (in addition to temporal factors) inherent in graphs. Thus, a hierarchical convolutional network model may be conducive to simultaneously capture global and local features from those graphs. In this regard, Wang et al. [47] employed a *hierarchical GCN* with a Laplacian-based graph coarsening algorithm to leverage the multi-level structure. In addition, they captured temporal features by using GRU as well.

To take structural changes around the target edge in a window into consideration, Cai et al. [48] proposed *structural graph neural network (StrGNN)*, which produces h-hop subgraphs from each temporal graph and uses GCN and GRU to extract features and temporal dependencies. Specifically, h-hop subgraphs centered on the target edge at each timeline were extracted firstly. Then, GCNs and pooling technique were used to exploit features from the subgraphs. Subsequently, GRUs captured the temporal dependencies.

2) ANOMALOUS NODE DETECTION

Dynamic graphs often show complex variation patterns, which can be interpreted by the stochasticity and spatiotemporal relationships between nodes and edges. However, previous deterministic methods could not handle such a problem. To solve this, Yang et al. [49] proposed *hierarchical variational graph recurrent autoencoder (H-VGRAE)* by employing GCN-based GAE and RNN—specifically, by stacking multiple GCN layers and combining the GCN layers with dilated RNN (DRNN) layers in the encoder. The same methods in the encoder were used in the decoder except the Bernoulli and Gaussian MLP for reconstruction in order to detect anomalous nodes. H-VGRAE jointly learns the anomaly features of the nodes and edges, and detects node anomalies using automatic threshold selection.

Zhang et al. [50] proposed a novel *dynamic evolving graph convolutional network (DEGCN)* model to capture evolving patterns of both local node-level and global graph-level software behaviors. It consists of three stages. First, the multi-scale graphs are generated by sliding windows. Second, on a directed GCN (DGCN) model, in-degree node features and out-degree node features are summarized simultaneously, and on a graph encoding-based GRU (GGRU) model, the evolving patterns of graphs are learned for its time steps. Third, the features from DGCN and GGRU are combined, and MLP layer calculates the anomality score.

Zola et al. [51] proposed a graph-based approach to detect malicious connections on traffic networks. Specifically, in the first stage, a temporal dissection operation was used to split the entire network information into time intervals and to extract Traffic Dispersion Graphs (TDGs). In the second stage, new synthetic samples for each TDG were created to balance the number of attackers and normal nodes. Finally, the TDGs were trained by GCNs and anomalous nodes were detected.

IV. OPPORTUNITIES AND CHALLENGES

GNN-based graph anomaly detection is a quite challenging problem with lots of potentials to build sophisticated methods based upon the existing approaches presented in this survey, particularly for detecting edge anomalies and subgraph anomalies. Further, there are research opportunities and open challenges that are important and not addressed yet. This section summarizes some of them.

A. EXPLAINABLE GNNs FOR DETECTING GRAPH ANOMALIES

The output of anomaly detection methods is either an anomaly score or a top-k ranking list. To intuitively understand the meaning of the score and list, additional explanations need to be provided. Since GNN-based methods are inherently less interpretable than traditional machine learning approaches, it is important to resolve the issue along with explainable models for anomaly detection. Currently, there is relatively little work on designing explainable GNN models for graph anomaly detection (e.g., Deng et al. [58]).

B. IDENTIFICATION OF GRAPH ANOMALIES WITH GNNs

Anomalies are mostly recognized on the basis of a reconstruction loss and a distance-based stochastic function. Although this loss function is capable of capturing deviating patterns in training, the reconstruction loss is vulnerable to noise and the distance-based function does not work properly when anomalous and normal data are distant in the embedding space. These challenges will be no less, or rather worse, for the problem of identifying graph anomalies. While many GNN-based approaches embed graphs into either Euclidean or hyperbolic spaces, they do not fully utilize the information available in graphs or lack the flexibility to model intrinsic

complex graph geometry [20]. Therefore, it is a promising future research direction to design a suitable embedding space and loss functions to detect graph anomalies.

C. CLASS IMBALANCE IN GRAPH ANOMALY DETECTION WITH GNNs

Imbalance between normal and anomalous data is inevitable since the anomalies tend to occur rarely. As the model performance is heavily dependent on training data, the class imbalance is a challenge that must be overcome. Rather than naïvely using negative sampling, a better strategy may be augmenting the training data, which is also a challenging issue in graph domains. There is little research carried out to resolve the issue of class imbalance in graph anomaly detection with GNNs in the literature (e.g., Zhao et al. [59]).

D. ANOMALY DETECTION WITH GNNs FROM HETEROGENEOUS GRAPHS

A heterogeneous graph is a graph that contains multiple types of nodes and edges and often occurs in practice. Studies on heterogeneous graph anomaly detection with GNNs have been largely under-explored in the literature. Handling the heterogeneity in solving the anomaly detection problem along with GNNs would be challenging, particularly if compounded by the temporal factors in a dynamic graph, due to the modeling difficulty including extraction of multiple relations of nodes/edges. Feasible strategies may include (1) sampling target edges or subgraphs via meta-paths and (2) extracting useful information via embedding the types of heterogeneous graphs.

E. FEW-SHOT GRAPH ANOMALY DETECTION WITH GNNs

Although GNN approaches have rapidly advanced over the past a few years, few-shot graph anomaly detection with GNNs has nor been studied much yet. While, in real-world scenarios, it is easy to obtain a few labeled anomalies from graphs similar to the target graph, there have been only a few recent attempts to leverage such a potential to develop GNN models that can leverage a small number of labeled anomalies (e.g., Ding et al. [38]). There are still open challenges on designing diverse meta-learning algorithms to conduct few-shot graph anomaly detection aided by GNNs.

V. CONCLUSION

In this paper, we provided a comprehensive survey of state-of-the-art graph anomaly detection methods that were built upon GNN models. Additionally, we presented several opportunities and challenges for further research in this area. A majority of research efforts have been concentrated on detecting node anomalies from static graphs, while types of graphs and anomaly types are rapidly expanding to include dynamic graphs and edge/subgraph/graph-level anomalies. This trend is expected to continue in the coming years. Some specific application domains, such as anomaly detection in IoT, in program analysis (e.g., Android malware and OS verification), remain for future research. We hope that this survey will trigger and stimulate more active research on graph anomaly detection using GNNs and be a stepping stone for subsequent surveys as the research progresses.

REFERENCES

[1] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, Feb. 1969.

[2] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, "Anomaly detection in dynamic networks: A survey," *Wiley Interdiscipl. Rev., Comput. Stat.*, vol. 7, no. 3, pp. 223–247, Mar. 2015.

[3] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Mining Knowl. Discovery*, vol. 29, no. 3, pp. 626–688, May 2015.

[4] Y. Li, N. Liu, J. Li, M. Du, and X. Hu, "Deep structured cross-modal anomaly detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[5] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019.

[6] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognit. Lett.*, vol. 42, pp. 11–24, Jun. 2014.

[7] K. Ding, J. Li, R. Bhanushali, and H. Liu, "Deep anomaly detection on attributed networks," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, May 2019, pp. 594–602.

[8] Y. Li, X. Huang, J. Li, M. Du, and N. Zou, "SpecAE: Spectral AutoEncoder for anomaly detection in attributed networks," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2233–2236.

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Mar. 2020.

[10] L. Zheng, Z. Li, J. Li, Z. Li, and J. Gao, "AddGraph: Anomaly detection in dynamic graph using attention-based temporal GCN," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4419–4425.

[11] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Comput.*, vol. 22, no. 1, pp. 949–961, Jan. 2019.

[12] 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 Trans. Knowl. Data Eng.*, early access, Oct. 8, 2021, doi: 10.1109/TKDE.2021.3118815.

[13] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2021.

[14] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Feb. 2017, pp. 1–14.

[16] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 1025–1035.

[17] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Feb. 2018, pp. 1–12.

[18] W. L. Hamilton, *Graph Representation Learning*. San Rafael, CA, USA: Morgan & Claypool, 2020.

[19] H. Wang, C. Zhou, X. Chen, J. Wu, S. Pan, and J. Wang, "Graph stochastic neural networks for semi-supervised learning," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, Oct. 2020, pp. 1–10.

[20] S. Zhu, S. Pan, C. Zhou, J. Wu, Y. Cao, and B. Wang, "Graph geometry interaction learning," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, Oct. 2020, pp. 7548–7558.

[21] F. Zhang, H. Fan, R. Wang, Z. Li, and T. Liang, "Deep dual support vector data description for anomaly detection on attributed networks," *Int. J. Intell. Syst.*, vol. 37, no. 2, pp. 1509–1528, Sep. 2021.

[22] X. Yuan, N. Zhou, S. Yu, H. Huang, Z. Chen, and F. Xia, "Higher-order structure based anomaly detection on attributed networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 2691–2700.

[23] X. Luo, J. Wu, A. Beheshti, J. Yang, X. Zhang, Y. Wang, and S. Xue, "ComGA: Community-aware attributed graph anomaly detection," in *Proc. 15th ACM Int. Conf. Web Search Data Mining*, Feb. 2022, pp. 657–665.

[24] Z. Peng, M. Luo, J. Li, L. Xue, and Q. Zheng, "A deep multi-view framework for anomaly detection on attributed networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 6, pp. 2539–2552, Jun. 2022.

[25] L.-H. Chen, H. Li, and W. Yang, "AnomMAN: Detect anomaly on multi-view attributed networks," 2022, *arXiv:2201.02822*.

[26] 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 Trans. Knowl. Data Eng.*, early access, Oct. 12, 2021, doi: 10.1109/TKDE.2021.3119326.

[27] A. Kumagai, T. Iwata, and Y. Fujiwara, "Semi-supervised anomaly detection on attributed graphs," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.

[28] T. Huang, Y. Pei, V. Menkovski, and M. Pechenizkiy, "Hop-count based self-supervised anomaly detection on attributed networks," in *Proc. Eur. Conf. Mach. Learn. Princ. Knowl. Discovery Databases (ECML PKDD)*, Jul. 2022.

[29] Y. Pei, T. Huang, W. van Ipenburg, and M. Pechenizkiy, "ResGCN: Attention-based deep residual modeling for anomaly detection on attributed networks," *Mach. Learn.*, vol. 2021, pp. 1–23, Sep. 2021.

[30] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, and G. Karypis, "Anomaly detection on attributed networks via contrastive self-supervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2378–2392, Jun. 2022.

[31] M. Jin, Y. Liu, Y. Zheng, L. Chi, Y.-F. Li, and S. Pan, "ANEMONE: Graph anomaly detection with multi-scale contrastive learning," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 3122–3126.

[32] T. Zhao, T. Jiang, N. Shah, and M. Jiang, "A synergistic approach for graph anomaly detection with pattern mining and feature learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2393–2405, Jun. 2022.

[33] H. Fan, F. Zhang, and Z. Li, "Anomalydae: Dual autoencoder for anomaly detection on attributed networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 5685–5689.

[34] Z. You, X. Gan, L. Fu, and Z. Wang, "GATAE: Graph attention-based anomaly detection on attributed networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2020, pp. 389–394.

[35] K. Ding, J. Li, N. Agarwal, and H. Liu, "Inductive anomaly detection on attributed networks," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1288–1294.

[36] X. Wang, B. Jin, Y. Du, P. Cui, Y. Tan, and Y. Yang, "One-class graph neural networks for anomaly detection in attributed networks," *Neural Comput. Appl.*, vol. 33, pp. 12073–12085, Mar. 2021.

[37] S. Zhou, Q. Tan, Z. Xu, X. Huang, and F.-L. Chung, "Subtractive aggregation for attributed network anomaly detection," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 3672–3676.

[38] K. Ding, Q. Zhou, H. Tong, and H. Liu, "Few-shot network anomaly detection via cross-network meta-learning," in *Proc. Web Conf.*, Apr. 2021, pp. 2448–2456.

[39] D. Duan, L. Tong, Y. Li, J. Lu, L. Shi, and C. Zhang, "AANE: Anomaly aware network embedding for anomalous link detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 1002–1007.

[40] G. Zhang, Z. Li, J. Huang, J. Wu, C. Zhou, J. Yang, and J. Gao, "eFraudCom: An E-commerce fraud detection system via competitive graph neural networks," *ACM Trans. Inf. Syst.*, vol. 40, no. 3, pp. 1–29, Jul. 2022.

[41] J. Song, X. Qu, Z. Hu, Z. Li, J. Gao, and J. Zhang, "A subgraph-based knowledge reasoning method for collective fraud detection in E-commerce," *Neurocomputing*, vol. 461, pp. 587–597, Oct. 2021.

[42] R. Ma, G. Pang, L. Chen, and A. V. D. Hengel, "Deep graph-level anomaly detection by glocal knowledge distillation," in *Proc. ACM Int. Conf. Web Search Data Mining (WSDM)*, Jan. 2022, pp. 704–714.

[43] L. Huang, Y. Zhu, Y. Gao, T. Liu, C. Chang, C. Liu, Y. Tang, and C.-D. Wang, "Hybrid-order anomaly detection on attributed networks," *IEEE Trans. Knowl. Data Eng.*, early access, Oct. 6, 2021, doi: 10.1109/TKDE.2021.3117842.

[44] L. Zhao and L. Akoglu, "On using classification datasets to evaluate graph outlier detection: Peculiar observations and new insights," *Big Data J.*, vol. 2021, pp. 1–12, Dec. 2021.

[45] C. Qiu, M. Kloft, S. Mandt, and M. Rudolph, "Raising the bar in graph-level anomaly detection," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, May 2022, pp. 1–10.

[46] D. Zhu, Y. Ma, and Y. Liu, "A flexible attentive temporal graph networks for anomaly detection in dynamic networks," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Jan. 2021, pp. 870–875.

[47] B. Wang, T. Hayashi, and Y. Ohsawa, "Hierarchical graph convolutional network for data evaluation of dynamic graphs," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 4475–4481.

[48] L. Cai, Z. Chen, C. Luo, J. Gui, J. Ni, D. Li, and H. Chen, "Structural temporal graph neural networks for anomaly detection in dynamic graphs," in *Proc. ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, Oct. 2021, pp. 3747–3756.

[49] C. Yang, L. Zhou, H. Wen, Z. Zhou, and Y. Wu, "H-VGRAE: A hierarchical stochastic spatial–temporal embedding method for robust anomaly detection in dynamic networks," 2020, *arXiv:2007.06903*.

[50] Z. Zhang, Y. Li, W. Wang, H. Song, and H. Dong, "Malware detection with dynamic evolving graph convolutional networks," *Int. J. Intell. Syst.*, vol. 37, pp. 7261–7280, Mar. 2022.

[51] F. Zola, L. Segurola-Gil, J. L. Bruse, M. Galar, and R. Orduna-Urrutia, "Network traffic analysis through node behaviour classification: A graph-based approach with temporal dissection and data-level preprocessing," *Comput. Secur.*, vol. 115, Apr. 2022, Art. no. 102632.

[52] W. Jin, T. Derr, H. Liu, Y. Wang, S. Wang, Z. Liu, and J. Tang, "Self-supervised learning on graphs: Deep insights and new direction," 2020, *arXiv:2006.10141*.

[53] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *Proc. Int. Conf. Learn. Rep. (ICLR)*, Feb. 2019, pp. 1–17.

[54] B. Weisfeiler and A. Leman, "The reduction of a graph to canonical form and the algebra which appears therein," *NTI, Ser.*, vol. 2, no. 9, pp. 12–16, 1968.

[55] L. Babai and L. Kucera, "Canonical labelling of graphs in linear average time," in *Proc. Annu. Symp. Found. Comput. Sci. (SFCS)*, Oct. 1979, pp. 39–46.

[56] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2018, pp. 4393–4402.

[57] C. Qiu, T. Pfrommer, M. Kloft, S. Mandt, and M. Rudolph, "Neural transformation learning for deep anomaly detection beyond images," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Mar. 2021, pp. 8703–8714.

[58] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2021, pp. 4027–4035.

[59] T. Zhao, Y. Liu, L. Neves, O. J. Woodford, M. Jiang, and N. Shah, "Data augmentation for graph neural networks," in *Proc. 34th AAAI Conf. Artif. Intell.*, Jun. 2021, pp. 11015–11023.

HWAN KIM received the B.S. degree in law from Hannam University, Daejeon, South Korea, in 2013. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Chungnam National University, Daejeon. His research interests include data mining, machine learning with graphs, graph neural networks, and graph anomaly detection.

BYUNG SUK LEE received the B.S. degree in electronics engineering from Seoul National University, Seoul, South Korea, the M.S. degree in electrical engineering from KAIST, Daejeon, South Korea, and the Ph.D. degree in electrical engineering from Stanford University, Palo Alto, CA, USA. He is currently a Professor at the Department of Computer Science, University of Vermont, Burlington, VT, USA. His research interests include databases and database management systems, data mining and machine learning, data science, data stream processing, and information retrieval, with emphases on algorithms and performances. He has been a PI and co-PI of nearly 20 funded research projects, an author of more than 80 peer-reviewed publications, and has served as an organizer, the chair, and a member of program committees in almost 50 international conferences.

WON-YONG SHIN (Senior Member, IEEE) received the B.S. degree in electrical engineering from Yonsei University, Seoul, Republic of Korea, in 2002, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2004 and 2008, respectively.

In May 2009, he joined the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA, as a Postdoctoral Fellow, and was promoted to a Research Associate, in October 2011. From March 2012 to February 2019, he was a Faculty Member (tenure) of the Department of Computer Science and Engineering, Dankook University, Yongin, Republic of Korea. Since March 2019, he has been with the Department of Computational Science and Engineering, School of Mathematics and Computing, Yonsei University, where he is currently a Professor. He has been with the Department of Artificial Intelligence, Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea, as an Adjunct Professor, since September 2022. His research interests include information theory, communications, mobile computing, data mining, and machine learning.

Dr. Shin served as an Associate Editor for the *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, from 2014 to 2018. He served as an Organizing Committee Member for the 2015 IEEE Information Theory Workshop and the 2023 IEEE Consumer Communications and Networking Conference. He has also served as a technical program committee member or the chair for various conferences. He has received the Bronze Prize of the Samsung Humantech Paper Contest, in 2008, the KICS Haedong Young Scholar Award, in 2016, and the ICT Express Best Guest Editor Award, in 2021.

SUNGSU LIM (Member, IEEE) received the B.S. and M.S. degrees in mathematical sciences and the Ph.D. degree in knowledge service engineering from KAIST, Daejeon, South Korea, in 2009, 2011, and 2016, respectively. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Chungnam National University, Daejeon, South Korea. His research interests include mining and modeling large-scale networks and their theoretical aspects. He has received the Honorable Mention Prize of the Samsung Humantech Paper Contest, in 2012, the KAIST-Qualcomm Innovation Award, in 2016, and the NVIDIA Applied Research Accelerator Award, in 2022.