Motif-Level Anomaly Detection in Dynamic Graphs

Zirui Yuan, Minglai Shao, and Qiben Yan, *Senior Member, IEEE*

***Abstract*— As many real-world networks evolve over time, such as social networks, user-item networks, and IP-IP networks, anomaly detection for dynamic graphs has attracted growing attention. Most existing studies focus on detecting anomalous nodes or edges but fail to detect anomalous motif instances. In this paper, we propose MADG, a general Motif-level Anomaly Detection framework for dynamic Graphs, which can identify the anomaly in different motifs. Motifs are specific subgraph structures that frequently occur in a network and have been widely used in network analysis. In order to learn discriminative motif-level representations and leverage the temporal information from the dynamic graph, we design motif-augmented GCN and temporal self-attention. We first use motif-augmented GCN to model the topological structure among nodes and motif instances to learn their representations at each snapshot. Then, we feed the representations of multiple snapshots into the self-attention layer with relative temporal encoding in order to capture the evolutionary patterns. Extensive experiments on real-world dynamic graph datasets demonstrate the effectiveness of our proposed MADG framework.**

***Index Terms*— Anomaly detection, dynamic graph, motif, graph convolutional network, self-attention, autoencoder.**

I. Introduction

In RECENT years, anomaly detection in graphs has attracted increasing attention from both industry and academia with the prevalent use of graph data in social networks, e-commerce, financial transaction, and cybersecurity. Detecting anomalies in graphs can be essential in maintaining social stability and business interests. Examples include identifying crowd activities using social media data or detecting network intrusion [1], [2], [3], [4]. Compared with static graphs, dynamic graphs contain affluent temporal information and can better model the evolving objects and relationships in real-world networks. As a result, anomaly detection for dynamic graphs gains increasing popularity within the graph anomaly detection research community.

The existing graph anomaly detection methods can be classified as node-level, edge-level or subgraph-level, depending on the targeted objects of detection. In the present study, we concentrate on the anomaly detection of a specific type of subgraph in dynamic graphs. We define this task as motif-level anomaly detection, which aims to detect anomalous motif instances.

Motif is a specific subgraph pattern that occurs more frequently than would be expected by chance in a network. Some common types of network motifs include triangles, quadrilaterals, and bipartite motifs. These motifs can be found in a wide range of biological, social, and technological networks. A large body of research within the network science literature focuses on using motifs to analyze the functional properties of networks [5], [6], [7].

Due to the prevalence of motifs in networks, certain anomalies may attempt to evade detection by disguising their activities as a typical motif pattern within the network. If the detection methods lack robustness in such situations, the anomalies have the potential to inflict significant societal and commercial losses. Consequently, there is a pressing need for an efficacious technique that can accurately identify anomalous motif instances.

Many efforts to detect anomalies in dynamic graphs have been made in the recent decade, which can be categorized into traditional non-deep learning methods and deep learning-based methods. The former techniques typically involve modeling the patterns of normal data, such as those described in [8], where the authors assume a stable evolutionary mechanism governing the generation and removal of edges within a dynamic graph. They introduce a novel link prediction method that fits the evolutionary patterns of the graph, and anomalies can be identified based on deviations between observations and prediction outcomes. Nonetheless, as the dimensionality of data increases, these methods suffer from limited performance and large computational costs.

Recently, deep learning-based methods have shown superior performance in dynamic graph learning. They are able to capture implicit rules from the given data and generate desired low-dimensional representations of networks which are very useful for anomaly detection. NetWalk [9] applies a deep graph embedding technique to dynamic graphs and leverages a clustering-based detector to detect anomalies. More recently, AddGraph [10], StrGNN [11], H-VGRAE [12] and TADDY [13] further propose end-to-end deep neural network models for anomaly detection in dynamic graphs.

Despite the success, the existing deep learning-based methods only focus on anomaly detection at the level of node or edge. However, motif-level anomalies may not be evident at the level of individual nodes or edges but rather emerge from the complex evolving pattern of the motif. Thus, it is necessary to consider the motif as a whole when detecting anomalies rather than focusing exclusively on individual nodes

Manuscript received 2 June 2022; revised 29 March 2023; accepted 26 April 2023. Date of publication 3 May 2023; date of current version 11 May 2023. This work was supported in part by the NSFC Program under Grant 62272338 and in part by the China Postdoctoral Science Foundation under Grant 2022T150470 and Grant 2021M702448. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Valeria Loscri. *(Corresponding author: Minglai Shao.)*

Zirui Yuan and Minglai Shao are with the School of New Media and Communication, Tianjin University, Tianjin 300072, China (e-mail: yzr@tju.edu.cn; shaoml@tju.edu.cn).

Qiben Yan is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: qyan@msu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2023.3272731

or edges. This requires a more sophisticated approach than node or edge-level anomaly detection methods can provide.

Detecting anomalous motif instances in a dynamic graph is not a trivial task. The main challenges are as follows: 1) Given that motifs can vary significantly in size and shape, the anomaly detection approach must be sufficiently general to adapt to various motifs. 2) Structural information is a significant factor when considering the anomalous degree of a graph object. However, it is important to note that motifs represent higher-order structures and encoding their structural information directly from the original network is not feasible. 3) Temporal information also plays a significant role in anomaly detection. Given that an individual motif instance typically appears within a short time span, it is difficult to learn its temporal information in a dynamic graph.

In order to overcome these challenges, in this paper, we propose a novel **M**otif-level **A**nomaly **D**etection framework for dynamic **G**raph (**MADG** for abbreviation). For challenge 1, we examine a universal approach to obtain motif-level representations from nodes. Specifically, we develop an attention-based motif feature aggregator that can aggregate information from nodes within each instance of a motif. This aggregator is capable of adapting to diverse shapes of motifs. For challenge 2, to incorporate the structural information of motifs, we have enhanced the original Graph Convolutional Network (GCN) by establishing connections among motifs and nodes, resulting in a motif-augmented GCN. As for challenge 3, the motif instance may appear for a short time, but its nodes usually exist for a long time. In order to obtain the motif representations with temporal information, we develop a temporal self-attention module to learn evolving patterns of nodes. Following this, we utilize our motif feature aggregator to aggregate information from nodes of each motif instance. After obtaining the latent representation of motif instances and nodes, we reconstruct the topological structures of the motif-augmented network and use the reconstruction errors as the anomaly score of motif instances. To summarize, the main contributions of this paper are as follows:

- We propose MADG, a novel unsupervised learning framework for motif-level anomaly detection in dynamic graphs, which learns representations for each motif instance and detects anomalous motif instances based on reconstruction errors.
- We design an effective method for dynamic graph learning, which integrates local topological and temporal dependency.
- The proposed MADG framework is scalable and capable of adapting to various types of motifs, including triangle and quadrilateral motifs. Our scalability study further revealed that MADG maintains robust performance even as the motif size increases while the computation time does not increase significantly.
- We conduct extensive experiments on four real-world datasets. The experimental results demonstrate the effectiveness of MADG in detecting motif-level anomalies in dynamic graphs.

The remainder of this paper is organized as follows. Section II reviews the related work briefly. Section III formally defines the problem of motif-level anomaly detection in dynamic networks and introduces some preliminaries. Section IV presents the details of the proposed MADG framework, and the experimental results are demonstrated in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

A. Dynamic Graph Anomaly Detection

Since the dynamic graph can model the evolving objects and relationships in real-world networks, anomaly detection in dynamic graphs gains increasing attention. The existing works can be categorized into non-deep learning and deep learning-based methods. We first introduce non-deep learning methods. GOutlier [14] employs a structural connectivity model to detect outliers in graph streams and designs a reservoir sampling method to maintain structural summaries of the underlying network. CAD [15] detects anomalous edges by measuring information regarding changes in the graph structure and edge weights. CM-Sketch [16] is a sketch-based method that considers both the local structural information and historical behavior to identify irregular edges. StreamSpot [17] is a clustering-based approach that introduces a similarity function for heterogeneous graphs property comparison based on the relative frequency of local substructures. It leverages a centroid-based clustering method to model the behaviors of graph streams. SpotLight [18] is a randomized sketching-based method that guarantees a large mapped distance between anomalous and normal instances in the sketch space.

Since the non-deep learning methods cannot capture the non-linear properties [19], recent studies have adopted deep learning techniques and achieved great success. NetWalk [9] applied the network embedding techniques to dynamic graphs. They leverage a random walk-based encoder with an autoencoder to learn node representations on the initial graph and incrementally update as the edges change. Finally, NetWalk adopts a dynamic clustering-based anomaly detector to score the abnormality of each edge. AddGraph [10] combined temporal, structural and attribute information to measure the abnormality of edges in dynamic graphs. They adopt GCN as a structural features extractor and use attention-based Gated Recurrent Units (GRU) to combine short-term and long-term dynamic evolution. StrGNN [11] first extracts the h-hop enclosing subgraph of edges and proposes a node labeling function to identify the role of each node in the subgraph. Then, stacked GCN and GRU are used to capture the spatial and temporal information. Finally, they propose a "context-dependent" negative sampling strategy to train the model. HVGRAE [12] builds a hierarchical model by combining variational graph autoencoder and recurrent neural network. It captures normal patterns in training set by maximizing the likelihood of the adjacency matrix and node attributes via variational inference. The edge reconstruction probability is used to measure the abnormality. TADDY [13] uses a dynamic graph transformer model with spatial-temporal node encoding to aggregate spatial and temporal knowledge simultaneously.

TABLE I
NOTATION DESCRIPTION

Notation	Description
$\mathbb{G} = \{\mathcal{G}^t\}_{t=1}^{T}$	The dynamic graph from timestamp 1 to T.
$\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$	The snapshot at timestamp t.
$\mathcal{V}^t$	The node set at timestamp t.
$\mathcal{E}^t$	The edge set at timestamp t.
n^t	The number of nodes, $\lvert\mathcal{V}^t\rvert$.
$v_i^t \in \mathcal{V}^t$	The i-th node in the node set at timestamp t.
$e_{i,j}^t \in \mathcal{E}^t$	An edge between v_i^t and v_j^t at timestamp t.
$\mathbf{A}^t \in \mathbb{R}^{n^t \times n^t}$	The adjacency matrix at timestamp t.
$\mathbb{M}^t = \{\mathcal{M}_i^t\}_{i=1}^{\overline{n}^t}$	The motif set at timestamp t.
$\mathcal{M}_i^t = (\mathcal{V}_{\mathcal{M}_i}^t, \mathcal{E}_{\mathcal{M}_i}^t)$	The i-th motif instance at timestamp t.
$\widehat{\mathcal{V}}^t$	The node set of motif-augmented network at timestamp t.
$\overline{\mathcal{V}}^t$	The virtual node set of motif instances at timestamp t.
$\overline{n}^t$	The number of virtual nodes, $\lvert\overline{\mathcal{V}}^t\rvert$.
$\hat{\mathcal{E}}^t$	The edge set of motif-augmented network at timestamp t.
$\overline{\mathcal{E}}^t$	The new edges related to motif.
$\mathbf{H}^{(l)}$	The output representation of l-layer motif-augmented GCN.
d	The size of the encoding dimension.
$\mathbf{h}_i$	The representation of the i-th node.
$\mathbf{Z}^t$	The output representation of temporal self-attention.
$\hat{\mathbf{Z}}^t$	The output representation of motif feature aggregator.
$\hat{\mathbf{A}}^{Rec}$	The reconstructed adjacency matrix of motif-augmented network.

In order to learn an anomaly score for each edge, they adopt a negative sampling strategy to generate pseudo anomalous edges.

Most of the above approaches are designed to detect node-level or edge-level anomalies, while our proposed MADG framework focuses on motif-level anomalies.

B. Subgraph-Level Anomaly Detection

The most pertinent reference to our motif-level anomaly detection task is subgraph-level anomaly detection. In motif-level anomaly detection, our objective is to assess the degree of abnormality of motifs, which are distinct subgraph structures that frequently appear in networks. This implies that only certain types of predetermined structures require evaluation. Conversely, in subgraph-level anomaly detection, anomalous subgraphs may take any form. Therefore, motif-level anomaly detection can be viewed as a distinct subvariant of subgraph-level anomaly detection.

Subgraph-level anomaly detection is more challenging than node-level and edge-level anomaly detection. Since the nodes and edges in an anomalous subgraph might be normal, we need to consider the abnormality from the view of the entire subgraph. Although many non-deep learning-based methods for this task have been proposed in the past, there is still a lack of studies using deep learning-based methods. Recently, some work have detected anomalous subgraphs via graph representation techniques.

SADE [20] is a two-stage subgraph anomaly detection framework. This framework combines role discovery and random walk to learn subgraph representations. Then, it adopts isolation forest, a conventional anomaly detection method, to identify anomalous subgraphs. HO-GAT [21] can detect abnormal nodes and motif instances in static graphs simultaneously. In order to learn the dependencies among nodes and motifs, HO-GAT creates a motif-augmented network and uses Graph Attention Network (GAT) to learn the latent representations of nodes and motif instances.

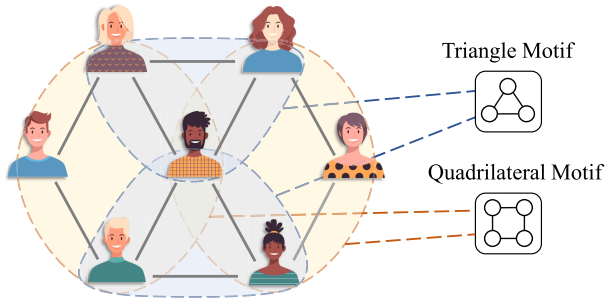


Fig. 1. An illustration of the triangle motif and the quadrilateral motif in social networks.

Unlike the aforementioned works, this paper aims to detect anomalous motif instances in dynamic graphs, which is a more complex scenario. In addition to considering the structural information in each static snapshot, it is necessary to leverage the temporal information from the dynamic graph. Due to the adequate performance of HO-GAT in static graphs, we also adopt the motif-augmented network to build adjacency among nodes and motif instances at each timestamp. Differently, we further design a novel temporal self-attention module to aggregate information from multiple timestamps.

III. PROBLEM DEFINITION

In this section, we first introduce some concepts and notations and then formulate our problem. For convenience, the notations used in this paper are listed in Table I.

A. Dynamic Graph

Given a dynamic graph with a maximum timestamp of T, it can be defined as $\mathbb{G} = \{\mathcal{G}^t\}_{t=1}^{T}$, where each $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$ represents a snapshot of the dynamic graph at timestamp t, $\mathcal{V}^t$ and $\mathcal{E}^t$ represent the node set and the edge set of snapshot $\mathcal{G}^t$ respectively. An edge $e_{i,j}^t = \left(v_i^t, v_j^t, w_{i,j}^t\right) \in \mathcal{E}^t$ means that there is an interactions with weight $w_{i,j}^t$ between node v_i^t and node v_j^t at the timestamp t, where $v_i^t, v_j^t \in \mathcal{V}^t$. For convenience, we let $n^t = \left|\mathcal{V}^t\right|$ denote the number of nodes at timestamp t. We use an adjacency matrix $\mathbf{A}^t \in \mathbb{R}^{n^t \times n^t}$ to represent the edges in $\mathcal{E}^t$, where $\mathbf{A}_{i,j}^t = w_{i,j}^t$ if $e_{i,j}^t = \left(v_i^t, v_j^t, w_{i,j}^t\right)$ in $\mathcal{E}^t$, otherwise $\mathbf{A}_{i,j}^t = 0$.

B. Motif

In this paper, we aim to identify the anomalous motif instances in each snapshot $\mathcal{G}^t$. Motif, as called as subgraph of interest or frequent subgraph, refers to a specific subgraph structure (e.g., common substructures in sets of biomolecular

compounds) that frequently occurs in a network such as edges, triangles and quadrilaterals. It is an essential building block in the complex network and has been widely studied [5], [6], [7].

Detecting anomalous motif instances is vital for both network research and applications. In the previous studies [7], [22], [23], Z-score has been widely used to discover motifs. They usually identify the densely connected subgraph with the largest Z-score as the motif. Among various motif structures, the triangle motif is the most widely studied motif, which contains three nodes and three edges. For example, some studies leverage the triangle motif in social networks to learn better representations [24], [25]. The triangle motif represents a ternary closure relationship, which is a common phenomenon in the social network. Following the previous works [21], [26], [27], [28], we focus on detecting anomalous triangle motif instances, but our framework can be easily extended to other types of motifs as well.

For each snapshot $\mathcal{G}^t$, we can obtain a motif set $\mathbb{M}^t = \left\{\mathcal{M}_i^t\right\}_{i=1}^{\overline{n}^t}$, where each $\mathcal{M}_i^t = \left(\mathcal{V}_{\mathcal{M}_i}^t, \mathcal{E}_{\mathcal{M}_i}^t\right)$. $\overline{n}^t$ is number of motif instances. For triangle motif, $\mathcal{V}_{\mathcal{M}_i}^t = \left\{v_{i_1}, v_{i_2}, v_{i_3}\right\}$ and $\mathcal{E}_{\mathcal{M}_i}^t = \left\{\left(v_{i_1}, v_{i_2}\right), \left(v_{i_2}, v_{i_3}\right), \left(v_{i_1}, v_{i_3}\right)\right\}$.

C. Motif-Level Anomaly Detection

In order to identify the anomalous motif instances in the dynamic graph, we need to measure the abnormality of all motif instances in any snapshot $\mathcal{G}^t$. Here, we use $f\left(\mathcal{M}_i^t\right)$ for each $\mathcal{M}_i^t \in \mathbb{M}^t$, where $f(\cdot)$ is a learnable anomaly score function. A motif instance with a large score indicates that this motif instance has a high anomalous probability. Since labeled data for graph anomaly detection is hard to collect, we aim to design an unsupervised way to learn $f(\cdot)$. In particular, we follow the commonly-used autoencoder-based anomaly detection paradigm to design our model.

D. Autoencoder-Based Anomaly Detection Paradigm

Rumelhart et al. [29] first introduced the autoencoder in 1986, which is a type of algorithm used to learn a low-dimensional informative representation for the input data in an unsupervised way. Generally speaking, an autoencoder contains two main components: an encoder and a decoder. The encoder compresses the input data into a low-dimensional latent representation, and the decoder reconstructs an approximation of the original input data. This process can be formulated as follows:

$$\tilde{\mathbf{X}} = D(E(\mathbf{X})), \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{N\times n}$ is the original input data, E is the encoding function to compress $\mathbf{X}$ into a low-dimensional representation $E(\mathbf{X}) \in \mathbb{R}^{N\times d}$, D is the decoding function to recover the input data from low-dimensional representation, and $\tilde{\mathbf{X}} \in \mathbb{R}^{N\times n}$ is the recovered results output from D. The autoencoder is trained to reconstruct the $\mathbf{X}$ well enough, giving rise to the following optimization problem:

$$\min_{D,E} = \left\|\mathbf{X} - D(E(\mathbf{X}))\right\|. \tag{2}$$

Theoretically, given that the latent representations' dimensionality d is much lower than the input data's dimensionality n, the autoencoder model is forced to capture the essence of the input data. The essence depends on the distribution of the majority (i.e., normal data), and anomalies are a negligible part of the input data which do not respect the essence. In order to minimize the overall reconstruction error, the autoencoder model is influenced more by the normal data. In this way, the anomalies will not be reconstructed well by the autoencoder. Therefore, the data instances which have high reconstruction error are more anomalous.

IV. Methodology

In this section, we introduce a motif-level anomaly detection framework for dynamic graphs. An overview of the proposed framework MADG is shown in Figure 2. It consists of four essential components: namely *motif-augmented GCN*, *temporal self-attention*, *motif feature aggregator*, and *anomaly detector*. The details of each component are described in this section.

A. Motif-Augmented GCN

GCN [30] extends the idea of convolution from image to graph. It can extract features from the local neighborhood. Stacking multiple layers of GCN and non-linear activation functions can learn discriminative node representations. Most previous works [10], [31], [32] use graph neural networks like GCN to learn node representations for anomaly detection. However, motifs are higher-order structures, and it is essential to treat motifs holistically and learn motif-level representations to accurately detect motif-level anomalies. Therefore, we propose motif-augmented GCN to learn the complex mutual influence of motif-to-motif and motif-to-node. At first, we create a motif-augmented network by regarding both the original nodes and the motif instances as the motif-augmented nodes, their interconnection as edges. The details of the motif-augmented network are explained as follows.

By searching in the original network at timestamp t, we can obtain a motif instance set $\mathbb{M}^t = \left\{\mathcal{M}_i^t\right\}_{i=1}^{\overline{n}^t}$. The motif-augmented network $\hat{\mathcal{G}}^t = \left\{\hat{\mathcal{V}}^t, \hat{\mathcal{E}}^t\right\}$ is then constructed by creating $\overline{n}^t$ virtual nodes to represent $\overline{n}^t$ motif instances and build edges among motif instances and original nodes.

The motif-augmented node set $\hat{\mathcal{V}}^t$ consists of $n^t + \overline{n}^t$ nodes, where the first n^t nodes are the nodes in the original network and the rest $\overline{n}^t$ nodes are the virtual nodes representing the $\overline{n}^t$ motif instances in $\mathbb{M}^t$. $\overline{\mathcal{V}}^t$ is defined as the node set of $\overline{n}^t$ virtual nodes.

The motif-augmented edge set $\hat{\mathcal{E}}^t$ is a union of $\mathcal{E}^t$ and $\overline{\mathcal{E}}^t$, where $\mathcal{E}^t$ is the original edge set in timestamp t and $\overline{\mathcal{E}}^t$ is a new edge set including the new edges related to $\overline{\mathcal{V}}^t$. The details of the rules to construct $\overline{\mathcal{E}}^t$ are explained as follows:

(1) For the i-th virtual node in $\overline{\mathcal{V}}^t$ representing motif instances $\mathcal{M}_i^t$, we add a new edge between the i-th virtual node and each node in motif instances $\mathcal{M}_i^t$.
(2) For each $e = \left(v_i, v_j, w_{i,j}\right) \in \mathcal{E}^t$, if node v_i or node v_j is included in a motif instance $\mathcal{M}_k^t$, then we add a new edge

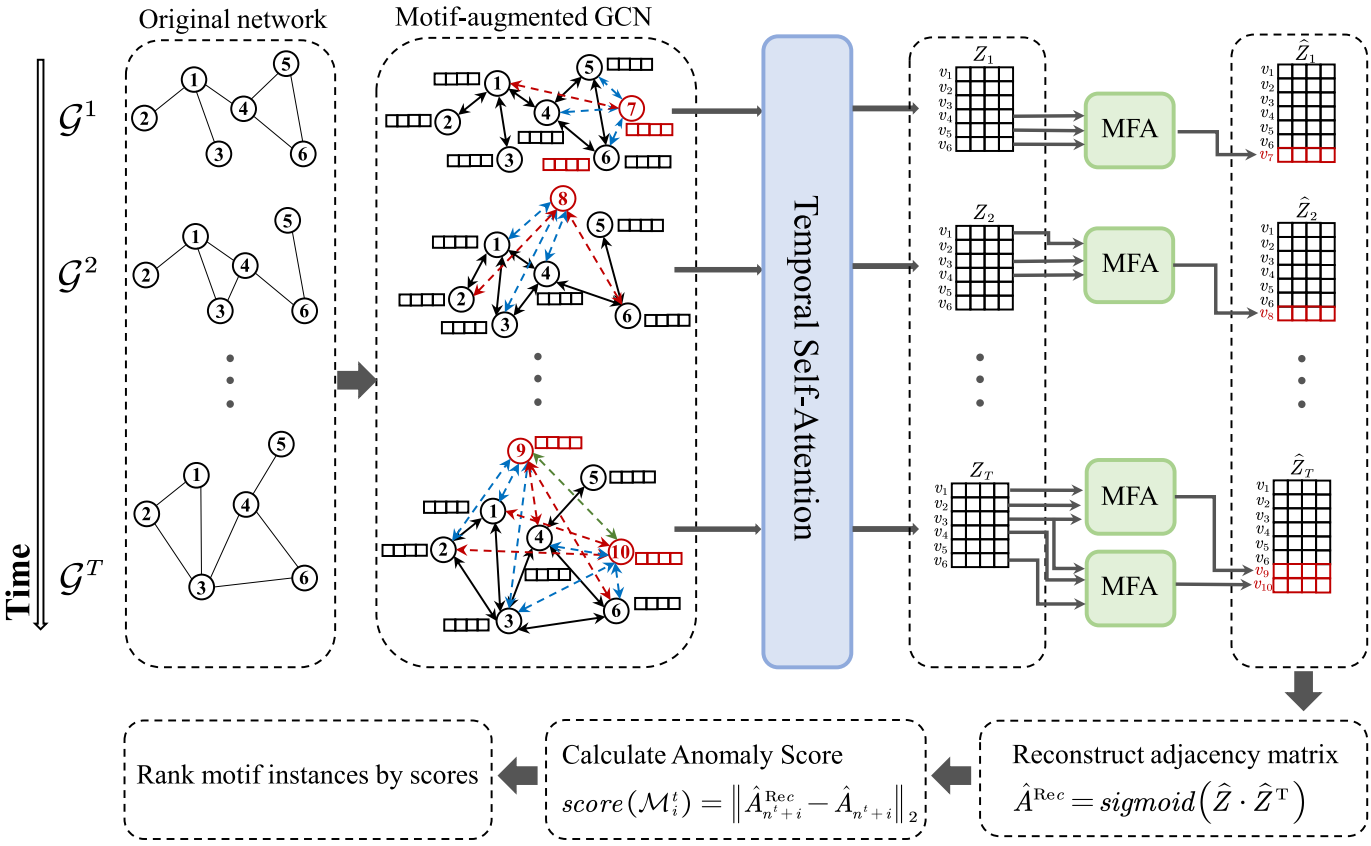


Fig. 2. The overview of the proposed MADG framework. As the first step, motif-augmented GCN creates new interactions among the original nodes and the virtual nodes representing motif instances and performs the message passing over these nodes to encode structural information into node representations. The colors of the new edges correspond to the edge generation strategies we describe in section IV-A. Then we feed the node representations of multiple snapshots into temporal self-attention to capture the evolutionary patterns of the dynamic graph and use the motif feature aggregator(MFA for short) to aggregate information for each motif instance. Finally, we reconstruct the adjacency matrix of motif-augmented networks. The reconstruction errors are minimized to train the framework, and the motif instance with high reconstruction error will be identified as an anomaly.

between node v_i or node v_j and the k-th virtual node that represents the motif instance $\mathcal{M}_k^t$ with weight $w_{i,j}$.

(3) For each motif instance pair $\mathcal{M}_i^t$ and $\mathcal{M}_j^t$, if $\mathcal{V}^t_{\mathcal{M}_i} \cap \mathcal{V}^t_{\mathcal{M}_j} \neq \emptyset$, then we add a new edge between the i-th virtual node and the j-th virtual node in $\overline{\mathcal{V}}^t$, which represent motif instances $\mathcal{M}_i^t$ and motif instances $\mathcal{M}_j^t$ respectively.

An example of the motif-augmented network is shown in Figure 3.

After creating a motif-augmented network for each snapshot, we stack L GCN layers to encode the structural dependency in motif-augmented networks. Each layer of the encoder can be expressed as follows:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{3}$$

where $l \in [0, L-1]$. $\tilde{\mathbf{A}} = \hat{\mathbf{A}} + \mathbf{I}$ is the adjacency matrix of motif augmented network with self-loops. $\tilde{\mathbf{D}}$ is the degree matrix of this snapshot, and $\tilde{\mathbf{D}}_{i,i} = \sum_j \tilde{\mathbf{A}}_{i,j}$ denotes the degree of the i-th node. $\mathbf{W}^{(l)} \in \mathbb{R}^{d \times d}$ is a feature map matrix. $\mathbf{H}^{(0)} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{n^t}, \mathbf{h}_{n^t+1}, \ldots, \mathbf{h}_{n^t+\overline{n}^t}]$ is the initial node representations of motif-augmented network.

After applying the L-layer motif-augmented GCN encoder in each snapshot, the output node representations $\mathbf{H}_t^L \in \mathbb{R}^{(n^t+\overline{n}^t)\times d}$ contains the structural information in the motif-augmented network at the timestamp t.

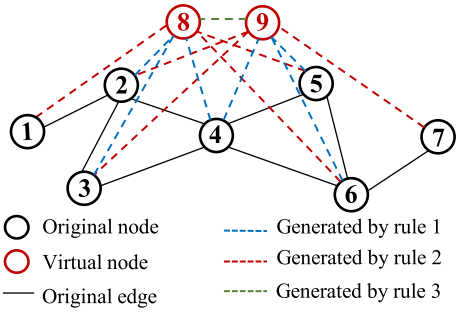


Fig. 3. An illustration of the motif-augmented network. There are two triangle motif instances in this network, and we represent them with two virtual nodes indexed by 8 and 9. The new edges are created according to the rules explained in section IV-A.

B. Temporal Self-Attention

As the snapshots evolve, the output node representations of L-layer motif-augmented GCN vary across timestamps. When we identify anomalies in a snapshot, a straightforward idea is to observe the surrounding multiple continuous snapshots simultaneously. Based on this intuition, the self-attention

mechanism are very suitable for capturing temporal variations of the dynamic graph. In the attention operation, each position can aggregate the information from any other timestamp. Some previous work [13], [33], [34] has introduced the attention mechanism in dynamic graph representation learning. Inspired by the relative position encoding in [35], we design a temporal self-attention layer with relative temporal encoding.

Specifically, after motif-augmented GCN, we obtain node representations $\mathbf{H}_t^L \in \mathbb{R}^{(n^t+\bar{n}^t)\times d}$ of each timestamp t. We let $\mathbf{H}_v = \left[\mathbf{h}_v^1, \mathbf{h}_v^2, \ldots, \mathbf{h}_v^T\right]$ represent a representation matrix of node v, where $\mathbf{h}_v^t \in \mathbb{R}^d$ is the representation of node v from $\mathbf{H}_t^L$ and T is the maximum number of timestamps. Then, representation matrices of a batch of nodes are fed into temporal self-attention, and we let $\mathbf{H}$ represent the input. Note that node v may not appear in every timestamp. For that case, we use an empty vector to fill the position where v does not appear. The main process is the same as the basic self-attention [36], which can be expressed as follows:

$$\mathbf{Q} = \mathbf{H}\mathbf{W}_Q, \mathbf{K} = \mathbf{H}\mathbf{W}_K, \mathbf{V} = \mathbf{H}\mathbf{W}_V, \tag{4}$$

$$\mathbf{A} = \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_K}}, \tag{5}$$

$$\mathbf{H}' = softmax(\mathbf{A})\mathbf{V}. \tag{6}$$

The input $\mathbf{H}$ is projected by three matrices $\mathbf{W}_Q \in \mathbb{R}^{d\times d_K}$, $\mathbf{W}_K \in \mathbb{R}^{d\times d_K}$ and $\mathbf{W}_V \in \mathbb{R}^{d\times d_V}$ to the corresponding representations $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ representing queries, keys, and values, respectively. $\mathbf{A}$ is an attention weight matrix capturing the similarity between queries and keys. For simplicity, we use single-head self-attention and set $d_K = d_V = d$.

Self-attention can aggregate features from any other timestamps by a weighted summation operation. Attention weight $\mathbf{A}$ is the weight of this operation. Considering that different relative positions should result in different influences, some attention works add position encoding into input $\mathbf{H}$ or attention weight $\mathbf{A}$. In this work, we design a relative temporal encoding and add it to the attention weight $\mathbf{A}$ as bias, which can be expressed as follows:

$$\mathbf{A} = \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_K}} + \mathbf{b}_{\phi(t_1,t_2)}, \tag{7}$$

where $\mathbf{b}_{\phi(t_1,t_2)}$ is a learnable scalar indexed by $\phi(t_1, t_2) = t_1 - t_2$, and shared across all layers. By using $\mathbf{b}_{\phi(t_1,t_2)}$, the influence of the difference of timestamp is considered in our attention module. Each timestamp can adaptively attend to all other timestamps.

C. Motif Feature Aggregator

Since most motif instances only appear in one timestamp, temporal self-attention cannot capture their temporal information directly. The output of temporal self-attention is representations of all nodes in multiple snapshots. In order to get discriminative motif instance representation with temporal information, we need an effective method to aggregate features from nodes in motif instances.

As stated in the previous subsection, self-attention can adaptively aggregate features, and it has been widely used to generate the representation of the entire sequence. For example, in the BERT model [37], [38], they add a special token at the beginning of each sequence. After computing, the special token's output representations are fed into downstream tasks as sequence-level representations. For the graph analysis task, some work [35], [39] have already adopted this method to generate graph-level representations with effective results. Inspired by these studies, we use self-attention with supernode to generate motif-level representations. The details of this method are shown as follows.

First, we initial a supernode representation $\mathbf{h}_{i_0}$. For a given triangle motif instance $\mathcal{M}_i^t = \left(\mathcal{V}_{\mathcal{M}_i}^t, \mathcal{E}_{\mathcal{M}_i}^t\right)$, $\mathbf{h}_{i_1}$,$\mathbf{h}_{i_2}$,$\mathbf{h}_{i_3}$ are the representations of the three nodes in $\mathcal{V}_{\mathcal{M}_i}^t$. Then, we feed the feature matrix $\mathbf{H} = \left[\mathbf{h}_{i_0}, \mathbf{h}_{i_1}, \mathbf{h}_{i_2}, \mathbf{h}_{i_3}\right] \in \mathbb{R}^{4\times d}$ as input into self-attention. The self-attention operation is the same as the basic self-attention we introduced in section IV-B.

After self-attention, we obtain the $\mathbf{H}' = [\mathbf{h}'_{j_0}, \mathbf{h}'_{j_1}, \mathbf{h}'_{j_2}, \mathbf{h}'_{j_3}]$. Among them, $\mathbf{h}'_{j_0}$ can aggregate the information of the whole motif instance, and we use it as the representation of this motif instance. The parameters of the supernode are shared among all motif instances.

D. Loss Function and Anomaly Detection

After temporal self-attention operation, we obtain a latent nodes representation matrix $\mathbf{Z}^t \in \mathbb{R}^{n^t\times d}$ for each timestamp t. Then we use the motif feature aggregator module to generate representations for each motif instance and obtain $\hat{\mathbf{Z}}^t \in \mathbb{R}^{(n^t+\bar{n}^t)\times d}$. $\hat{\mathbf{Z}}^t$ is the desired low-dimensional representation of the motif-augmented network, which captures structural and temporal information.

Then, we need to learn an anomaly score for each motif instance from their representations. Here, we consider an end-to-end framework. We use $\hat{\mathbf{Z}}^t$ to reconstruct the adjacency matrix of the motif-augmented network at timestamp t. For $\hat{\mathbf{Z}}^t = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{n^t}, \mathbf{h}_{n^t+1}, \ldots, \mathbf{h}_{n^t+\bar{n}^t}]$, we reconstruct the adjacency matrix $\hat{\mathbf{A}}^{Rec}$. One widely adopted strategy of the topological structure reconstruction lies in calculating the inner product between the latent representations of the two nodes and applying a sigmoid activation function [21], [31], [32], [40]. It can be expressed as follows:

$$\hat{\mathbf{A}}^{Rec} = sigmoid(\hat{\mathbf{Z}}^\top\hat{\mathbf{Z}}). \tag{8}$$

The loss function is designed as a mean square error:

$$\mathcal{L} = \left\|\hat{\mathbf{A}}^{Rec} - \hat{\mathbf{A}}\right\|_2, \tag{9}$$

where $\hat{\mathbf{A}}$ is a normalized adjacency matrix of motif-augmented network.

Since the original adjacency of anomalies is difficult to reconstruct, we use each motif instance's reconstruction error as its anomaly score. For example, the anomaly score of motif instance $\mathcal{M}_j^t$ is:

$$score(\mathcal{M}_j^t) = \left\|\hat{\mathbf{A}}_{n^t+j,:}^{Rec} - \hat{\mathbf{A}}_{n^t+j,:}\right\|_2, \tag{10}$$

where $\hat{\mathbf{A}}_{n^t+j,:}^{Rec}$ is the reconstruct adjacency of motif instance $\mathbf{M}_j^t$, and $\hat{\mathbf{A}}_{n^t+j,:}$ is the $(n^t + j)$-th row in the adjacency matrix $\hat{\mathbf{A}}$. Since the first n^t nodes in $\hat{\mathbf{A}}$ are the nodes of original

networks, the motif instance $\mathcal{M}_j^t$ is represented by the (n^t+j)-th node.

In the end, we summarize our MADG framework in Algorithm 1.

Algorithm 1 MADG

Input: Graph stream: $\mathbb{G} = \{\mathcal{G}^t\}_{t=1}^T$, motif set of each timestamp: $\{\mathbb{M}^t\}_{t=1}^T$.
Output: Anomaly scores of each motif instance.

```
Randomly initialize node embeddings.
repeat
    for t = 1 to T do
        Encode node representations H_t^L via
         motif-augmented GCN in section IV-A;
    end
    Aggregate temporal information via temporal
     self-attention in section IV-B;
    Generate motif representations via motif feature
     aggregator in section IV-C;
    for t = 1 to T do
        Calculate reconstruction error L^t via loss
         function in section IV-D;
    end
    Minimize L = sum_{t=1}^T L^t;
until Convergence;
Calculate anomaly scores via Equation (10).
```

E. Complexity Analysis

In this subsection, we analyze the complexity of each component in MADG. In motif-augmented GCN, we stack L layers GCN. Each GCN layer has a computation of $\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{XW}$ whose complexity is $\mathcal{O}(|\widehat{\mathcal{V}}^t|cdh)$, where $|\widehat{\mathcal{V}}^t|$ is the number of nodes in motif-augmented network at timestamp t, c is the average degree, d represents the feature dimension of input data $\mathbf{X}$ and h is the number of feature map of $\mathbf{W}$. In our framework, we set $h = d$. The complexity of motif-augmented GCN is $\mathcal{O}(L\sum_{t=1}^T|\widehat{\mathcal{V}}^t|cdh)$, where T is the number of timestamp. As for temporal self-attention, the time complexity is $\mathcal{O}(nT^2d)$, where n is the number of nodes in the whole dynamic network. For motif feature aggregator, it costs $\mathcal{O}(md)$, where m is the number of motif instances in all snapshots. For anomaly detector, the time complexity is $\mathcal{O}(\sum_{t=1}^T|\widehat{\mathcal{V}}^t|^2)$. To sum up, the overall complexity of our proposed MADG framework is $\mathcal{O}((L\sum_{t=1}^T|\widehat{\mathcal{V}}^t|cd + nT^2 + m)d + \sum_{t=1}^T|\widehat{\mathcal{V}}^t|^2)$.

V. Experiments

In this section, we evaluate the proposed MADG framework on four real-world dynamic graphs.

A. Experimental Setup

1) Dataset: In order to confirm the superiority of our proposed MADG framework, we adopted four real-word dynamic graph datasets that have been studied in previous works [9], [10], [11], [35]. The statistics of these datasets are shown in Table IV.

The UCI Messages network is collected from an online social platform of students at the University of California, Irvine. Each node represents a user in the platform, and each edge indicates that there is a message interaction. The Email-DNC dataset is collected from the 2016 Democratic National Committee email leak. Each node corresponds to a person, and the edge indicates an email communication. Bitcoin-Alpha and Bitcoin-OTC are two rating networks collected from two Bitcoin platforms named Alpha and OTC. Each node represents a user from the platform in these two datasets, and each edge denotes that a user has rated another. Since ground-truth anomalies are difficult to obtain, plenty of works inject anomalies into real-world datasets to evaluate their proposed methods [2]. For example, in the edge anomaly detection task, previous studies inject anomalous edges by randomly linking two unrelated nodes in the graph, while the original edges in the graph are normal [9], [10], [13]. Following their approach, we inject a certain percentage of anomalous motif instances to investigate our model. To be concrete, we randomly choose three nodes in the graph to generate anomalous triangle motif instances, which significantly deviate from the occurrence patterns of original triangle motif instances.

2) Baselines: We compared MADG against three popular deep learning-based dynamic graph anomaly detection methods.

- **NetWalk** [9]: The method utilizes random walk and auto-encoder to learn node embeddings and detects anomalies via k-means clustering. The node embeddings and cluster centers will be updated along with the timestamps.
- **AddGraph** [10]: AddGraph is a semi-supervised dynamic graph anomaly detection approach, which uses GCN and an attention-based Gated Recurrent Units (GRU) to capture structural information and temporal information, respectively.
- **TADDY** [13]: TADDY is a transformer-based anomaly detection framework for dynamic graphs. It leverages a dynamic graph transformer model with spatial-temporal node encoding to learn discriminative knowledge from dynamic graphs.

3) Experimental Design: In our experiments, each dataset is divided into ten snapshots according to the timestamps. The baseline models mentioned above are designed for edge anomaly detection and cannot directly discover anomalous motif instances. In order to obtain the anomalous motif instance using these methods, for triangle motif instances, we use the average anomaly score of three contained edges as the anomaly score of the motif instance. In the experiments on different datasets, the MADG framework is trained by Adam optimizer with a learning rate of 0.005 for 20 epochs. The dimension of representations d is 64. The input node representations are randomly initialized, and the motif representation is initialized by the mean of the representations of contained nodes. In addition, the motif-augmented GCN module stack

TABLE II
THE PERFORMANCE COMPARISON OF THE TRIANGLE MOTIF ANOMALY DETECTION IN AUC MEASURES

	UCI Messages	Email-DNC	Bitcoin-Alpha	Bitcoin-OTC
1% anomalies				
NetWalk	0.6519	0.7793	0.5887	0.6377
AddGraph	0.5285	0.7801	0.4770	0.4994
TADDY	0.8325	0.6997	0.9417	0.8924
MADG	**0.9568**	**0.9623**	**0.9758**	**0.9725**
5% anomalies				
NetWalk	0.5717	0.7699	0.6307	0.6819
AddGraph	0.6154	0.7337	0.4752	0.4671
TADDY	0.7414	0.7050	0.9438	0.8202
MADG	**0.9580**	**0.9564**	**0.9713**	**0.9718**
10% anomalies				
NetWalk	0.5771	0.7679	0.6233	0.6624
AddGraph	0.6278	0.7232	0.47602	0.5182
TADDY	0.7250	0.7011	0.9355	0.8226
MADG	**0.9527**	**0.9621**	**0.9721**	**0.9679**

TABLE III
PRECISION@K AND RECALL@K ON DIFFERENT DATASETS WITH 10% ANOMALIES

Precision@K												
	UCI Messages			Email-DNC			Bitcoin-Alpha			Bitcoin-OTC		
K	50	100	200	50	100	200	50	100	200	50	100	200
NetWalk	0.2000	0.1700	0.2250	0.9800	0.9600	0.8450	0.4200	0.4600	0.5550	0.3400	0.4100	0.5050
AddGraph	0.4600	0.3900	0.2900	0.7800	0.8400	0.8100	0.5400	0.4500	0.4100	0.4000	0.3500	0.3850
TADDY	0.0600	0.1000	0.1300	0.8000	0.7300	0.6500	0.5000	0.6000	0.6050	0.7400	0.6800	0.6350
MADG	**0.9400**	**0.9300**	**0.8750**	**1.0000**	**0.9800**	**0.9850**	**0.9200**	**0.8500**	**0.8800**	**0.8000**	**0.8500**	**0.8800**
Recall@K												
	UCI Messages			Email-DNC			Bitcoin-Alpha			Bitcoin-OTC		
K	50	100	200	50	100	200	50	100	200	50	100	200
NetWalk	0.0108	0.0183	0.0485	0.0145	0.0284	0.0500	0.0104	0.0227	0.0548	0.0049	0.0118	0.0290
AddGraph	0.0248	0.0421	0.0626	0.0115	0.0249	0.0481	0.0133	0.0222	0.0405	0.0057	0.0101	0.0221
TADDY	0.0032	0.0108	0.0280	0.0118	0.0216	0.0385	0.0123	0.0296	0.0597	0.0106	0.0196	0.0365
MADG	**0.0507**	**0.1003**	**0.1888**	**0.0148**	**0.0291**	**0.0585**	**0.0227**	**0.0419**	**0.0868**	**0.0115**	**0.0244**	**0.0506**

TABLE IV
THE STATISTICS OF THE DATASETS

Dataset	#nodes	#edges	#Avg.Degree
UCI Messages	1899	13838	14.57
Email-DNC	1866	39264	42.08
Bitcoin-Alpha	3777	24173	12.80
Bitcoin-OTC	5881	35588	12.10

two GCN layers. For the other baselines, we retain the settings described in the corresponding papers.

We use AUC, precision@k, and recall@k as the performance metrics. AUC is the area under the ROC curve, which ranges from 0 to 1. Precision@k is the proportion of true anomalies discovered in the top K instances. Recall@k refers to the percentage of true anomalies in the top K instances among all the ground truth anomalies. We first test the anomalous motif detection in datasets with 1%, 5%, and 10% anomalies, respectively. To verify the scalability of MADG, we extend the anomalous triangle motif detection to anomalous quadrilateral motif detection. Finally, we study the impacts of different parameters on MADG.

B. Anomaly Detection for Triangle Motif

In this subsection, we evaluate our proposed MADG framework and baseline methods in anomalous triangle motif detection. The anomaly detection performance comparison of average AUC on all timestamps is demonstrated in Table II. Compared to the baseline method, MADG reaches a large performance gain on the four datasets with varying anomaly proportions. From the precision@k and recall@k results in Table III, we can observe that MADG is able to rank anomalies higher. We attribute the excellent performance of MADG to the effective learning of motif-level structural information. Figure 4, Figure 5, and Figure 6 illustrate the results of anomaly detection on each snapshot. Since the baseline models need the first 50% data as clean data to pre-train, we only report the results of the latter five snapshots. The results indicate that MADG beats baselines on almost all snapshots. Even though TADDY achieves higher AUC scores in some snapshots, the performance of our proposed method is more robust under all snapshots. A possible reason is that we use temporal self-attention to aggregate information, which ensures that the representations of any snapshot can learn enough discriminative knowledge.

C. Anomaly Detection for Quadrilateral Motif

To verify the scalability of MADG, we extend the model to detect anomalous quadrilateral motif instances. The quadrilateral motif is another frequently occurring subgraph structure. We only need to change the triangle motif in the motif-augmented network to the quadrilateral motif. In

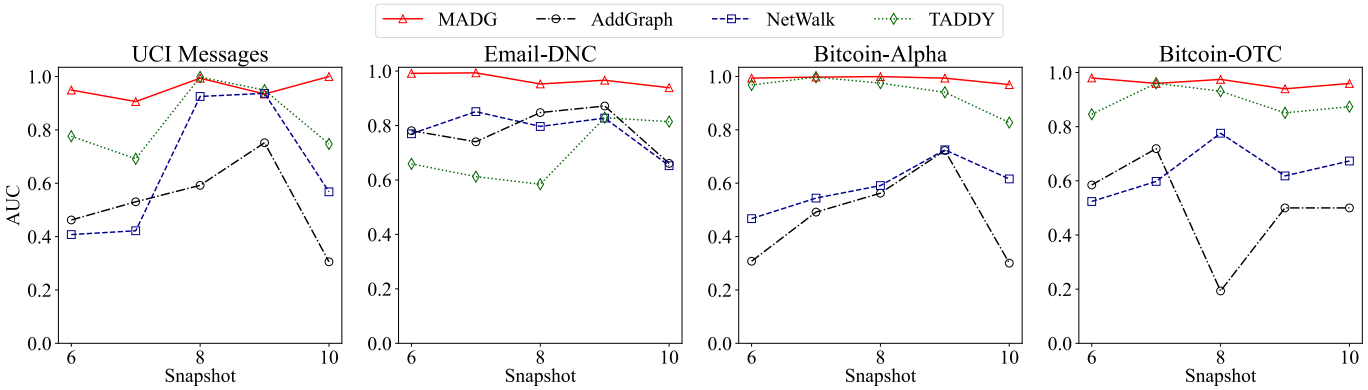


Fig. 4. The AUC values of triangle motif anomaly detection on the latter five snapshots. The anomaly ratio is 1%.

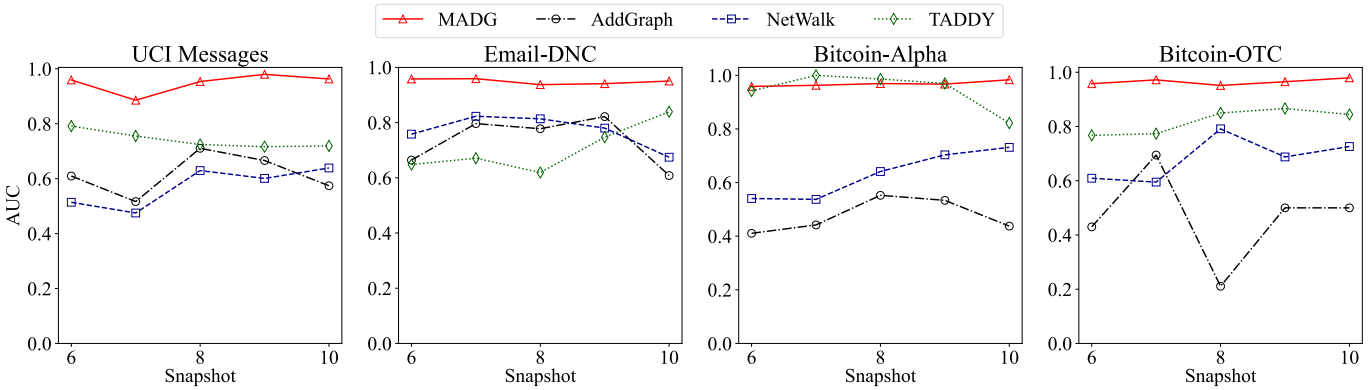


Fig. 5. The AUC values of triangle motif anomaly detection on the latter five snapshots. The anomaly ratio is 5%.

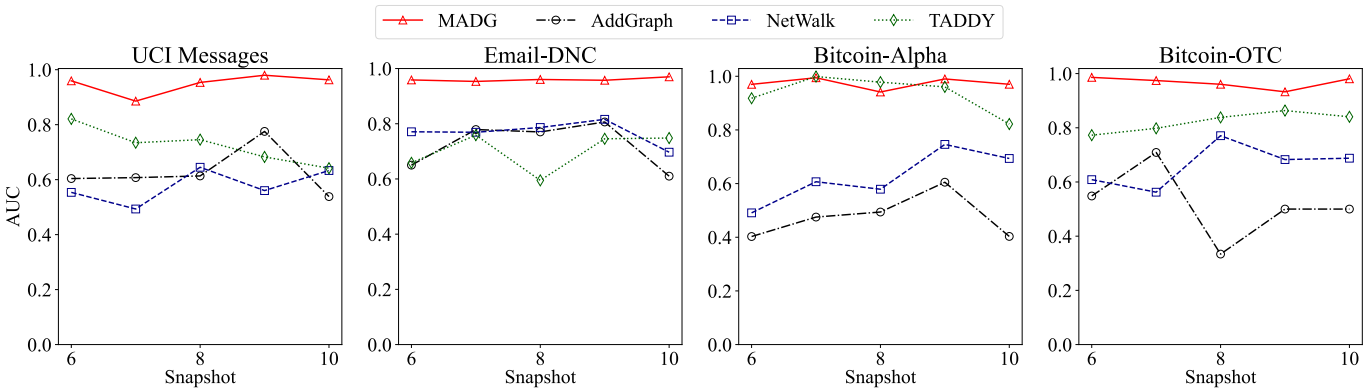


Fig. 6. The AUC values of triangle motif anomaly detection on the latter five snapshots. The anomaly ratio is 10%.

particular, we use the approach described in section IV-A to create the virtual node and its adjacency for every quadrilateral motif instance. The anomaly detection performance comparison of average AUC on all timestamps is demonstrated in Table V. The results show that the proposed MADG framework outperforms other methods in this task with a high performance, which further demonstrates that our model can be extended to detect other motif structures.

D. Parameter Sensitivity

1) Dimension d and GCN Layers L: In this subsection, we investigate how the size of the encoding dimension d and the number of GCN layers L affect the performance of our proposed MADG framework. Specifically, we set the range of d to $\{4, 8, 16, 32, 64\}$ and the range of L to $\{1, 2, 3, 4, 5\}$. Other parameters are set as optimum. We test the performance on different datasets with 5% anomalous motif instances and report results in Figure 7 and Figure 8.

According to the results of different datasets, we can observe that when d is small, increasing the d will improve the performance. However, after reaching its peak, the performance will degrade as the d increases. For UCI Messages, Email-DNC, and Bitcoin-OTC, the boundary point is $d = 8$. For Bitcoin-Alpha, the boundary point is $d = 32$. A possible

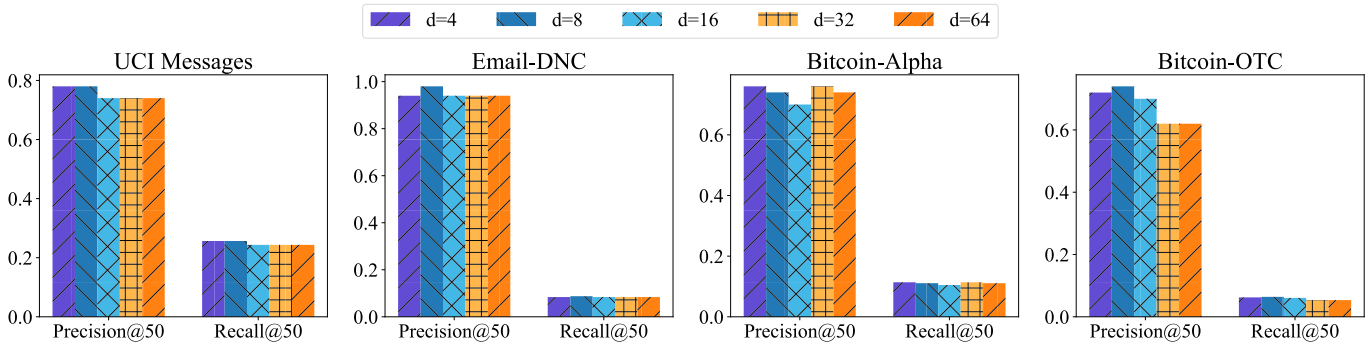


Fig. 7. Sensitivity w.r.t. encoding dimension d.

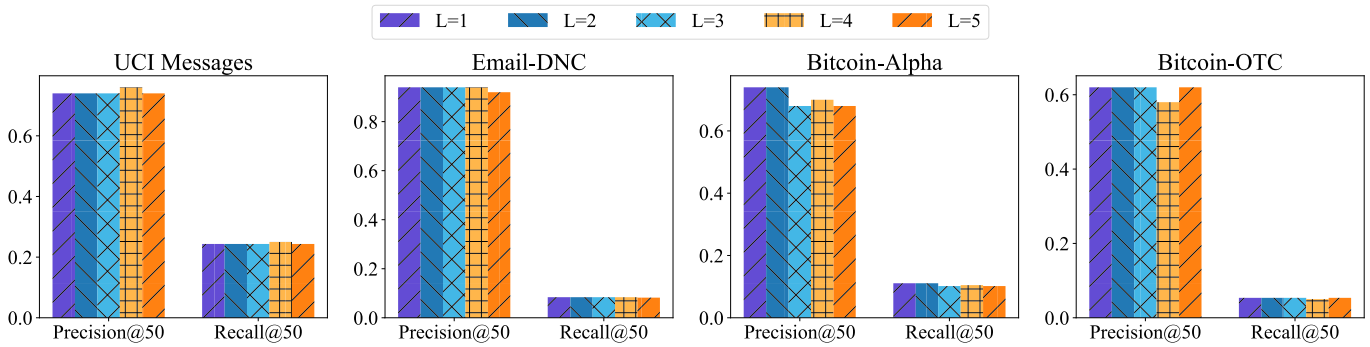


Fig. 8. Sensitivity w.r.t. the number of GCN layers L.

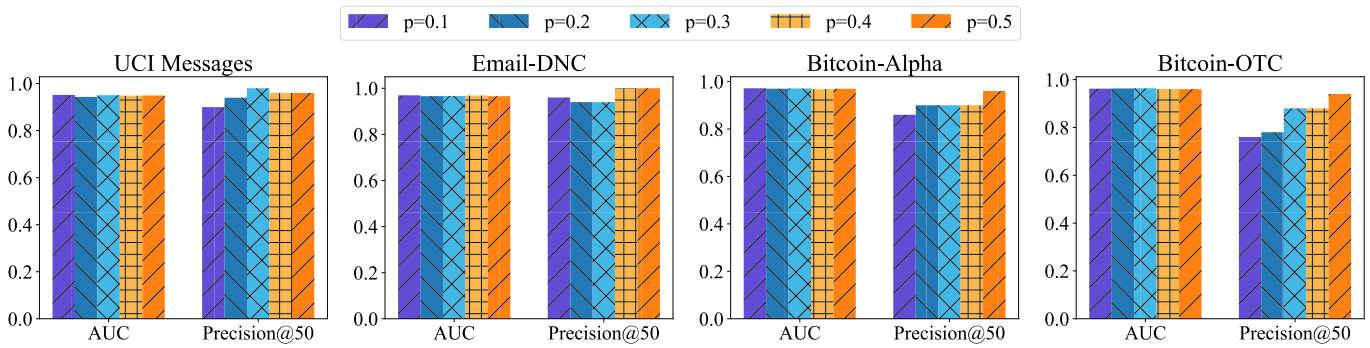


Fig. 9. Sensitivity w.r.t. anomaly ratio p.

reason is that when d is too small, the model cannot encode enough useful information. However, a large d will bring a more powerful fit ability, decreasing anomalies' reconstruction errors. Similar to d, the number of GCN layers L should not be too small or too large. The best choice is $L = 2$. Stacking more GCN layers will lead to the over-smoothing problem, which reduces the performance of MADG.

2) Anomaly Ratio: We further investigate the performance on different anomaly ratios p. The results are reported in Figure 9. We can observe that the AUC values maintain high performance as the anomaly ratio increases, and the precision@50 even grows. It demonstrates the stability of our model. Our explanation is that our model's optimization objective includes the reconstruction error of the motif instances and the reconstruction error of the original nodes. Although the number of anomalous motif instances increases, it is still a tiny number compared to the number of all nodes, which could hardly affect our model.

TABLE V
THE AUC RESULTS OF THE ANOMALY DETECTION FOR QUADRILATERAL MOTIFS WITH 10% ANOMALIES

Method	UCI Messages	Email-DNC	Bitcoin-Alpha	Bitcoin-OTC
NetWalk	0.5854	0.7645	0.6263	0.6564
AddGraph	0.5614	0.7237	0.4234	0.5140
TADDY	0.7110	0.4580	0.8903	0.7903
MADG	**0.9797**	**0.9788**	**0.9879**	**0.9825**

3) Window Size of Temporal Self-Attention: In the temporal self-attention module, our model aggregates information from multiple snapshots. We define the number of snapshots

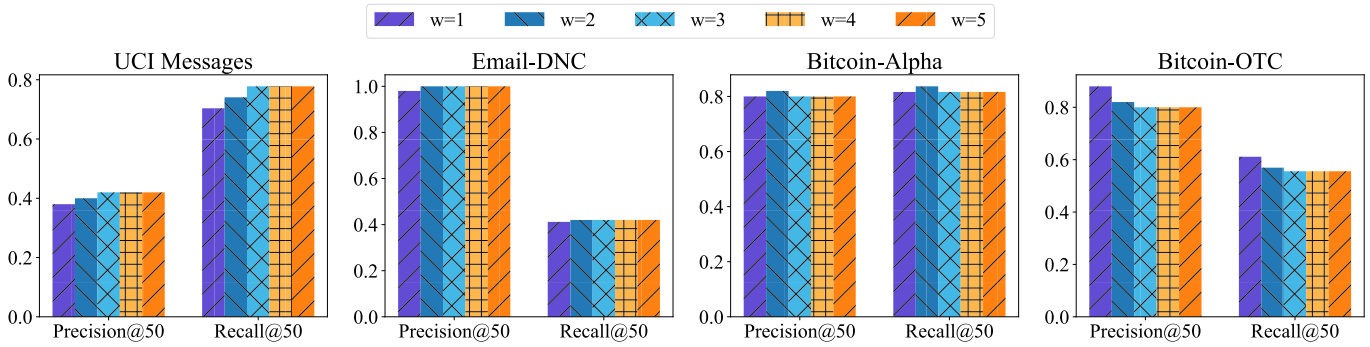


Fig. 10. Sensitivity w.r.t. window size w.

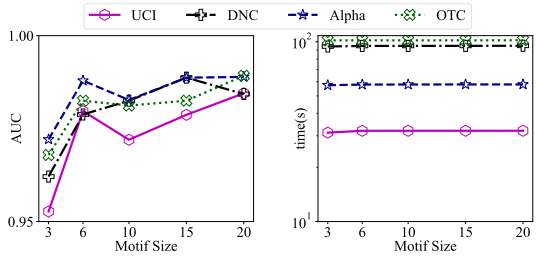


Fig. 11. Scalability w.r.t. the size of motif.

considered by temporal self-attention as the window size w. In this experiment, we investigate the effect of the window size. For convenience, we evaluate the performance on the fifth snapshot with 5% anomalies and report the results in Figure 10. The snapshots around the fifth snapshot will be considered by temporal self-attention. It can be observed that the performance on UCI Messages increases as the window size increases. However, the performance on the other two datasets is relatively insensitive, and the performance on the Bitcoin-OTC even decreases. The main reason is that nodes in these three datasets may only occur in several discontinuous snapshots. Thus a larger window size will not bring more information in this case. On the contrary, the nodes in UCI Messages occur more frequently. A larger window size can help temporal self-attention learn the evolutionary patterns better, improving anomaly detection performance.

E. Scalability Study

The scalability and efficiency of our MADG framework are demonstrated in Figures 11 (a) and (b), where we analyze its performance and runtime under varying motif sizes. Our experimental results reveal that MADG exhibits remarkable scalability, as an increasing motif size does not significantly increase computation costs while maintaining consistent performance. This is attributable to two key factors: firstly, our motif feature aggregator module effectively aggregates features from nodes in motif instances, regardless of the motif structure. Secondly, we use virtual nodes to represent motif instances, ensuring the computation is independent of the motif size.

TABLE VI
THE AUC RESULTS OF DIFFERENT VARIANTS

Method	UCI Messages	Email-DNC	Bitcoin-Alpha	Bitcoin-OTC
MADG	**0.9527**	**0.9621**	**0.9721**	**0.9679**
MADG-GCN	0.6492	0.7041	0.7031	0.7206
MADG-GAT	0.6481	0.7008	0.7005	0.7177
MADG-SAGE	0.6482	0.7010	0.7007	0.7179

F. Comparison With Different Encoders

In this experiment, we substitute the Motif-augmented GCN in MADG with three prominent graph neural networks: GCN [30], GAT [41], and GraphSage [42]. GCN is a popular convolutional neural network for graph learning that aggregates node features from neighbors. GAT is a self-attention based graph neural network that allows each node to selectively attend to its neighbors' features. GraphSage is a powerful inductive representation learning method that enables learning on large-scale graphs. We denote these MADG variants as MADG-GCN, MADG-GAT, and MADG-SAGE, respectively, in Table VI. Our findings reveal that the alternative graph neural network encoders do not perform optimally in the motif-level anomaly detection task. This is attributed to their node-level message-passing schemes, which fail to capture the anomalous behavior of motifs as a whole. In contrast, Motif-augmented GCN enhances the traditional node-level message-passing scheme with the motif-level message-passing scheme, thus facilitating the identification of this anomalous behavior. Consequently, it is essential to treat motifs holistically and learn motif-level representations to accurately detect motif-level anomalies.

G. Case Study

This subsection investigates a typical case of anomalous motif instances in a dynamic graph. Nodes 605, 642, and 1296 are three nodes that do not have any interactions in the UCI Messages dataset. However, we manually create their interaction with each other in the seventh snapshot to generate a typical anomalous triangle motif instance. The anomaly generation process is plotted in Figure 11. This motif instance has received the second high reconstruction error in our proposed model, which means a very high abnormal prob-

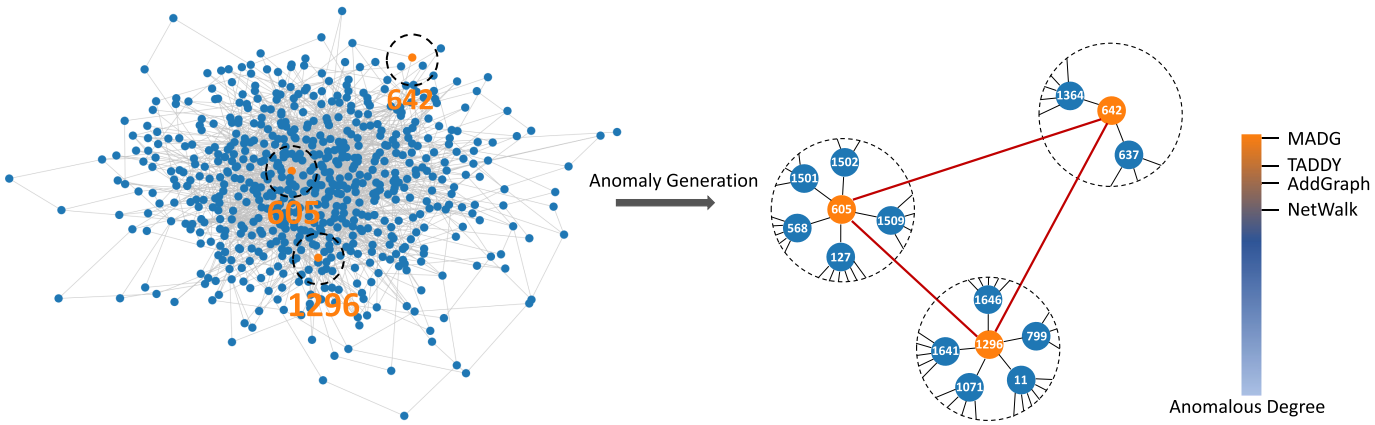


Fig. 12. Illustration of an anomaly case on the UCI Messages dataset. The left network is plotted according to the seventh snapshot of the UCI Messages dataset. Nodes 605, 642, and 1296 are three unrelated nodes over the dynamic graph, and they are far from each other. The anomaly appears when there are interactions among them. Anomaly detection methods will rank the anomaly score of every instance, and our method gives a higher anomalous degree to this anomaly case than other methods.

ability. The anomaly score output from different methods has different sizes and can not compare directly. For convenience to compare, we define the anomalous degree according to the ranking of the instance's anomaly score, which can be expressed as follows:

$$AnomalousDegree_i = \frac{Nmotifs - Ranking_i}{Nmotifs}, \quad (11)$$

where $AnomalousDegree_i$ is the anomalous degree of motif instances i, $Nmotifs$ is the total number of motif instances in this snapshot, and $Ranking_i$ is the anomaly score ranking of motif instance i. In this case, $Nmotifs = 178$. The ranking results and anomalous degree of this typical anomaly case are the following:

- MADG, ranking: 2, $AnomalousDegree$: 0.9888.
- TADDY, ranking: 21, $AnomalousDegree$: 0.8820.
- AddGraph, ranking: 32, $AnomalousDegree$: 0.8202.
- NetWalk, ranking: 50, $AnomalousDegree$: 0.7191.

Among all these methods, MADG gives this anomaly case the highest $AnomalousDegree$, which means our method can better identify anomalous motif instances in the dynamic graph.

VI. Conclusion and Future Work

In this paper, we introduced a novel problem of motif-level anomaly detection in the dynamic graph. It aims to detect anomalous motif instances which deviate from the underlying evolutionary pattern. To this end, we proposed the MADG framework, which can detect motif-level anomalies in the dynamic graph without explicitly labeled anomaly data. The MADG framework comprises four major components: motif-augmented GCN, temporal self-attention, motif feature aggregator, and anomaly detector. First, motif-augmented networks are created to model the structural dependencies among nodes and motif instances at each timestamp. Then, stacked GCN layers encode these networks into node representations. The node representations of multiple snapshots are fed as input to a novel temporal self-attention module, aiming to generate node representations with temporal information. After that, an attention-based motif feature aggregator generates representation for each motif instance. Then, the adjacency matrix of motif-augmented networks is reconstructed by the representations of nodes and motif instances. The reconstruction errors are minimized during the training of networks. Finally, the reconstruction errors of motif instances are adopted as anomaly scores. Our extensive experiments on real-world dynamic graphs corroborated the effectiveness of MADG in motif-level anomaly detection.

Our work can be extended in several interesting directions in future: (1) developing an algorithm for online motif-level anomaly detection, (2) integrating motif-level anomaly detection framework with motif discovery, and (3) incorporating motif-level anomaly detection with node and edge-level techniques to create a unified anomaly detection framework.

References

[1] 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.

[2] X. Ma et al., "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.

[3] S. Eltanbouly, M. Bashendy, N. AlNaimi, Z. Chkirbene, and A. Erbad, "Machine learning techniques for network anomaly detection: A survey," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Feb. 2020, pp. 156–162.

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

[5] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[6] H. Peng, J. Li, Q. Gong, Y. Ning, S. Wang, and L. He, "Motif-matching based subgraph-level attentional convolutional network for graph classification," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 5387–5394.

[7] R. Milo et al., "Superfamilies of evolved and designed networks," *Science*, vol. 303, no. 5663, pp. 1538–1542, 2004.

[8] H. Wang, J. Wu, W. Hu, and X. Wu, "Detecting and assessing anomalous evolutionary behaviors of nodes in evolving social networks," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 1, pp. 1–24, Feb. 2019.

[9] W. Yu, W. Cheng, C. C. Aggarwal, K. Zhang, H. Chen, and W. Wang, "NetWalk: A flexible deep embedding approach for anomaly detection in dynamic networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2672–2681.

[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. IJCAI*, Aug. 2019, pp. 4419–4425.

[11] L. Cai et al., "Structural temporal graph neural networks for anomaly detection in dynamic graphs," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 3747–3756.

[12] 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*.

[13] Y. Liu et al., "Anomaly detection in dynamic graphs via transformer," *IEEE Trans. Knowl. Data Eng.*, early access, Nov. 2, 2021, doi: 10.1109/TKDE.2021.3124061.

[14] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, May 2001, pp. 37–46.

[15] K. Sricharan and K. Das, "Localizing anomalous changes in time-evolving graphs," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2014, pp. 1347–1358.

[16] S. Ranshous, S. Harenberg, K. Sharma, and N. F. Samatova, "A scalable approach for outlier detection in edge streams using sketch-based approximations," in *Proc. SIAM Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, 2016, pp. 189–197.

[17] E. Manzoor, S. M. Milajerdi, and L. Akoglu, "Fast memory-efficient anomaly detection in streaming heterogeneous graphs," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1035–1044.

[18] D. Eswaran, C. Faloutsos, S. Guha, and N. Mishra, "SpotLight: Detecting anomalies in streaming graphs," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1378–1386.

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

[20] Y. Pei, F. Lyu, W. van Ipenburg, and M. Pechenizkiy, "Subgraph anomaly detection in financial transaction networks," in *Proc. 1st ACM Int. Conf. AI Finance*, Oct. 2020, pp. 1–8.

[21] L. Huang et al., "Hybrid-order anomaly detection on attributed networks," *IEEE Trans. Knowl. Data Eng.*, early access, Oct. 6, 2021, doi: 10.1109/TKDE.2021.3117842.

[22] L. Huang, C.-D. Wang, and H.-Y. Chao, "HM-modularity: A harmonic motif modularity approach for multi-layer network community detection," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2520–2533, Jun. 2021.

[23] W. Lin, X. Xiao, X. Xie, and X. Li, "Network motif discovery: A GPU approach," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 3, pp. 513–528, Mar. 2017.

[24] J. Xu et al., "Multivariate relations aggregation learning in social networks," in *Proc. ACM/IEEE Joint Conf. Digit. Libraries*, Aug. 2020, pp. 77–86.

[25] P. Shao, Y. Yang, S. Xu, and C. Wang, "Network embedding via motifs," *ACM Trans. Knowl. Discovery Data*, vol. 16, no. 3, pp. 1–20, Jun. 2022.

[26] C. E. Tsourakakis, J. Pachocki, and M. Mitzenmacher, "Scalable motif-aware graph clustering," in *Proc. 26th Int. Conf. World Wide Web*, Apr. 2017, pp. 1451–1460.

[27] P.-Z. Li, L. Huang, C.-D. Wang, and J.-H. Lai, "EdMot: An edge enhancement approach for motif-aware community detection," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 479–487.

[28] L. Huang, H. Chao, and G. Xie, "MuMod: A micro-unit connection approach for hybrid-order community detection," in *Proc. AAAI*, 2020, pp. 107–114.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," Inst. Cogn. Sci., California Univ. San Diego, San Diego, CA, USA, Tech. Rep., 1985.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[31] D. Zhu, Y. Ma, and Y. Liu, "DeepAD: A joint embedding approach for anomaly detection on attributed networks," in *Proc. Int. Conf. Comput. Sci.* Springer, 2020, pp. 294–307.

[32] K. Ding, J. Li, R. Bhanushali, and H. Liu, "Deep anomaly detection on attributed networks," in *Proc. SIAM Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, 2019, pp. 594–602.

[33] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "Dynamic graph representation learning via self-attention networks," 2018, *arXiv:1812.09430*.

[34] L. Yang, Z. Xiao, W. Jiang, Y. Wei, Y. Hu, and H. Wang, "Dynamic heterogeneous graph embedding using hierarchical attentions," in *Proc. Eur. Conf. Inf. Retr.* Springer, 2020, pp. 425–432.

[35] C. Ying et al., "Do transformers really perform bad for graph representation?" *arXiv preprint arXiv:2106.05234*, 2021.

[36] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[38] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[39] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.

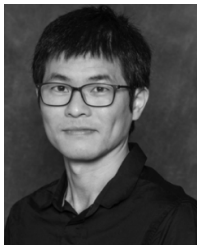
[40] H. Fan, F. Zhang, and Z. Li, "Anomalydae: Dual autoencoder for anomaly detection on attributed networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 5685–5689.

[41] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[42] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

Zirui Yuan received the B.E. degree in electronic information science and technology from Dalian Maritime University, Dalian, China, in 2021. He is currently pursuing the M.S. degree in computer technology with Tianjin University, Tianjin, China. His current research interests include anomaly detection, information propagation, and data mining.

Minglai Shao received the Ph.D. degree from the School of Computer Science and Engineering, Beihang University, Beijing, China. He was a Visiting Scholar with the State University of New York. He is currently an Associate Professor with the School of New Media and Communication, Tianjin University. His current interests include deep learning, machine learning, data mining, anomaly detection, and intelligent communication.



Qiben Yan (Senior Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Fudan University, Shanghai, China, and the Ph.D. degree from the Computer Science Department, Virginia Tech. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Michigan State University. His current research interests include wireless communications, wireless network security and privacy, mobile and the IoT security, and big data privacy. He was a recipient of the NSF CRII Award in 2016.