Semi-supervised Anomaly Detection with Extremely Limited Labels in Dynamic Graphs

Jiazhen Chen[1], Sichao Fu[2]✉, Zheng Ma[3], Mingbin Feng[1]✉, Tony S. Wirjanto[1], and Qinmu Peng[2]

[1] Department of Statistic and Actuarial Science, University of Waterloo
{j385chen, ben.feng, twirjanto}@uwaterloo.ca
[2] School of Electronic Information and Communications, Huazhong University of Science and Technology
fusichao_upc@163.com, pengqinmu@hust.edu.cn
[3] Cheriton School of Computer Science, University of Waterloo
z43ma@uwaterloo.ca

Abstract. Semi-supervised graph anomaly detection (GAD) has recently received increasing attention, which aims to distinguish anomalous patterns from graphs under the guidance of a moderate amount of labeled data and a large volume of unlabeled data. Although these proposed semi-supervised GAD methods have achieved great success, their superior performance will be seriously degraded when the provided labels are extremely limited due to some unpredictable factors. Besides, the existing methods primarily focus on anomaly detection in static graphs, and little effort was paid to consider the continuous evolution characteristic of graphs over time (dynamic graphs). To address these challenges, we propose a novel GAD framework (EL^2-DGAD) to tackle anomaly detection problem in dynamic graphs with extremely limited labels. Specifically, a transformer-based graph encoder model is designed to more effectively preserve evolving graph structures beyond the local neighborhood. Then, we incorporate an ego-context hypersphere classification loss to classify temporal interactions according to their structure and temporal neighborhoods while ensuring the normal samples are mapped compactly against anomalous data. Finally, the above loss is further augmented with an ego-context contrasting module which utilizes unlabeled data to enhance model generalization. Extensive experiments on four datasets and three label rates demonstrate the effectiveness of the proposed method in comparison to the existing GAD methods.

Keywords: Graph anomaly detection · Semi-supervised learning · Graph neural network · Graph contrastive learning.

1 Introduction

Anomaly detection identifies irregular patterns in data that deviate from expected behavior [2]. In today's interconnected world, many datasets naturally form complex network structures that are often contaminated with anomalies, such as fraudulent transactions in financial networks or malicious accounts in social networks [5, 21]. Given the inherent complexity of these graph-structured datasets and the severe consequences of ignoring anomalies, graph anomaly detection (GAD) has garnered significant attention in recent years.

Owning to the rarity of anomalies, existing GAD methods mainly rely on unsupervised approaches, including reconstruction-based [7, 14, 24] and contrastive learning-based techniques [12, 17, 26]. These methods detect anomalies as deviations from learned normal patterns but mostly focus on static graphs, overlooking the dynamic evolution of nodes, edges, and attributes. Recently, dynamic GAD has received increasing research interest, which requires models capable of capturing both structural dependencies and temporal patterns [25]. While some methods model dynamic graphs as discrete snapshots using techniques such as GCNs, GRUs, or transformers [1, 13, 25], they are primarily unsupervised and rely heavily on the quality of the pseudo labels, which may risk identifying noisy samples as anomalies due to the lack of prior anomaly knowledge.

In practice, it is often possible to acquire a limited set of labeled examples. This has spurred increased attention towards semi-supervised GAD, which combines labeled data with unlabeled data to improve detection performance [4, 8, 10, 15, 19]. For instance, SemiADC [15] uses generative adversarial networks to learn the feature distribution of normal nodes and trains a classifier to distinguish them from labeled anomalies. Similarly, SAD [19] uses a memory-enhanced deviation network to enforce significant deviations between abnormal and normal nodes, and incorporates a contrastive learning module to harness the potential of unlabeled data. Despite their promising results, these methods rely heavily on labeled anomalies, which are scarce and difficult to obtain, thus limiting their practical effectiveness in real-world applications.

To our knowledge, no prior work has explored dynamic GAD under severely limited supervision, where only a handful of labeled anomalies are available during training. To bridge this gap, we present the first investigation into this problem and propose a novel framework, EL^2-DGAD, which enhances the robustness of existing GAD methods in dynamic settings with minimal supervision. First, a transformer-based dynamic graph encoder is introduced to effectively capture both temporal and structural properties of dynamic graphs. Unlike existing methods that primarily focus on local dependencies using GNN-based approaches [19, 22], our encoder leverages both local and global attention mechanisms. The former extracts fine-grained patterns from a node's neighborhood, while the latter aggregates these patterns across the graph, thereby enhancing the detection of subtle and extensive anomalies across the graph. Additionally, continuous-time embeddings are integrated into the transformer's attention mechanism, enabling precise, time-sensitive representations. This contrasts with discrete snapshot-based methods [1, 12, 25], which lose precision by grouping events into fixed intervals, making it harder to detect time-sensitive anomalies. To effectively utilize the limited labels, we propose an ego-context hypersphere classification loss, which learns a context-aware hypersphere boundary by considering both the ego (target instance) and its historical context (graph dynamics). Rather than modeling the diverse and irregular nature of anomalies, it establishes a robust boundary for normal data, reducing reliance on a large number of labeled anomalies. Lastly, to further improve model generalization, we integrate an ego-context contrastive loss, which ensures the consistency of neighboring information with target instances by utilizing the vast amount of unlabeled data. Extensive experiments on four datasets and three extremely low label rates demonstrate the effectiveness of the proposed method in comparison to existing GAD methods.

2 Methodology

2.1 Problem Formulation

In this paper, we define a dynamic graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of nodes and $\mathcal{E}$ consists of temporal edges. Each edge, denoted by $\delta^t = (v_i, v_j, t, x_{ij})$, captures an interaction from node v_i to node v_j at time t, with x_{ij} serving as the edge feature. Additionally, We define $\mathcal{G}^t$ as the subgraph containing all events up to time t, and $\mathcal{G}^{t^-}$ as the same, excluding the event at t. Our proposed EL^2-DGAD framework aims to detect anomalous edges in this dynamic graph setting, where each edge δ^t is associated with a latent label y^t, indicating whether the edge is normal ($y^t = 0$) or abnormal ($y^t = 1$). In this study, we have access to only a limited number of labeled edges, denoted as Y^L, while the majority of edges remain unlabeled (Y^U), with $|Y^L| \ll |Y^U|$. The goal is to develop an algorithm that assigns an anomaly score s^t to each edge, represented as $f(\delta^t) = s^t$, to effectively distinguish between normal and anomalous edges.

2.2 Overview

The overview of EL^2-DGAD is shown in Figure 1. Specifically, for each input sample, an ego and a context graph are simultaneously defined. Each of the graphs is encoded through a transformer-based dynamic graph encoder, which builts on top of a series of local and global attentions (Section 2.3). During model training, we introduce an ego-context hypersphere classification loss in Section 2.4 that classifies edges by quantifying the similarity between their ego representation and the corresponding dynamic neighboring context. Finally, an ego-context contrastive loss in Section 2.4 is incorporated that is designed to mine and learn from the patterns of normality prevalent within the graph based on unlabeled data.

2.3 Graph Encoder

In scenarios with extremely limited labeled data, a robust model must capture both temporal and structural dependencies in dynamic graphs. Inspired by

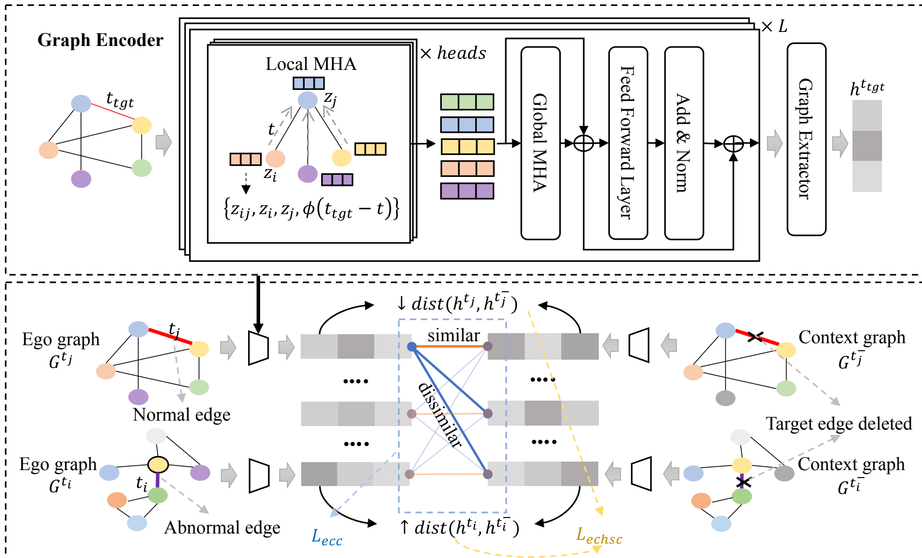


Fig. 1: An illustration for the architecture of EL^2-DGAD.

recent graph transformer advances [3, 23], We propose a transformer-based dynamic graph encoder that leverages local attention for fine-grained neighborhood extraction, global attention for graph-wide aggregation, and continuous-time embeddings to seamlessly capture temporal dynamics. Concretely, our graph encoder takes a subgraph at time t as the input (e.g., $\mathcal{G}^t$ or $\mathcal{G}^{t^-}$) and generates node representations that reflect their dynamic state. We start by aggregating 1-hop neighborhood information through masked multi-head self-attention (local MHA). To capture the evolving nature of dynamic graphs, we define the entries of Q, K, and V at layer l as follows:

$$Q_{ij}^l = \mathrm{MLP}(\mathrm{Concat}(x_{ij}, z_i^{l-1}, z_j^{l-1}, \phi(\Delta_t)))W^{Q,l} \tag{1}$$

$$K_{ij}^l = \mathrm{MLP}(\mathrm{Concat}(x_{ij}, z_i^{l-1}, z_j^{l-1}, \phi(\Delta_t)))W^{K,l} \tag{2}$$

$$V_{ij}^l = \mathrm{MLP}(\mathrm{Concat}(x_{ij}, z_i^{l-1}, z_j^{l-1}, \phi(\Delta_t)))W^{V,l} \tag{3}$$

where z_i^{l-1} is the output for node i from layer $l-1$, and z_i^0 is a learnable embedding vectors specified by the degree of v_i, as detailed in [23]. MLP denotes the multi-layer perceptrons. $\Delta_t = t - t_{ij}$, where t_{ij} is the occurring time of edge (v_i, v_j). $\phi(\cdot)$ is the sinusoidal embedding function [20], which generates a time representation by applying sine and cosine functions to the input time difference Δ_t, scaled by a range of frequencies.

Then, the attention output of node i for a single head can be formulated as:

$$a_i^l = \sum_j \mathrm{softmax}_j \left(\frac{Q_{ij}^l {K^l}_{ij}^T}{\sqrt{d}} M_{ij} \right) V_{ij}^l \tag{4}$$

where M is the adjacency matrix of the input graph. Next, we fed the outputs obtained from the local MHA, as the input to the regular multi-head self-attention in [20] (global MHA). The layer l's output is then defined as

$$z^l = \mathrm{LN}(\mathrm{FFN}(z^{l-1,loc} + z^{l-1,glo})) + z^{l-1,loc} + z^{l-1,glo} \tag{5}$$

where $z^{l-1,loc}$ and $z^{l-1,glo}$ are the output from the local MHA and global MHA, LN and FFN denote the layer normalization and the feed-forward blocks.

2.4 Model Training

Ego-context HSC loss To train the model, we incorporate the hypersphere classification (HSC) loss [18], which is well-suited for anomaly detection tasks when labeled anomalies are scarce. Traditional classifiers assume similar data points cluster naturally, but anomalies rarely form distinct clusters [2]. The HSC loss overcomes this by constructing a spherical decision boundary that pulls normal samples toward a fixed center (the zero vector) and pushes anomalies away. Specifically, the HSC loss of a given sample at t is defined as:

$$\mathcal{L}_t^{hsc} = -y^t \log l(x^t) - (1 - y^t) \log(1 - l(x^t)) \tag{6}$$

where $l(z) = \exp(-\|z\|^2)$, x^t and y^t are the latent representation and label of an interaction event at time t. The final classification output is given by $1 - l(x^t)$, which represents the probability of an interaction being classified as normal.

To capture the dynamic nature of graphs, we modify the HSC loss to incorporate temporal and structural context. Specifically, we represent the interaction at time t using two graph perspectives: the ego graph $\mathcal{G}^t$, which captures the evolving interactions including the event at t, and the context graph $\mathcal{G}^{t-}$, which reflects the recent neighborhood structure up to but not including t.

The latent representation for the interaction at time t is then defined as the difference between these two graph perspectives, i.e., $x^t = h^t - h^{t^-}$, where h^t and h^{t^-} are the graph representations for ego and context graphs, respectively. Here, h^t is computed as $h^t = f_o(\mathrm{concat}(z_i^t, z_j^t))$, where f_o is a two-layer MLP and z_i^t is the outputs of node i for $\mathcal{G}^t$ from the graph encoder. A separate graph encoder with different weights is used to compute h^{t-} for the context graph.

By replacing the original latent representation in the HSC loss with $h^t - h^{t-}$, we formulate the ego-context HSC loss as

$$\mathcal{L}_t^{echsc} = y^t \|h^t - h^{t^-}\|^2 - (1 - y^t) \log(1 - \exp(-\|h^t - h^{t^-}\|^2)) \tag{7}$$

This modified loss directly measures how an interaction alters the graph representation relative to its historical context, thereby classifying interactions based on their alignment with established patterns in the evolving graph.

Ego-context Contrastive Loss To further prevent overfitting with limited labeled data, we add a contrastive learning component that leverages abundant unlabeled samples. This component maximizes similarity between ego-context graph pairs from the same edge and minimizes similarity between pairs from different edges, based on the assumption that most edges naturally align with their context.

Specifically, the loss for an edge occurring at time t is defined as:

$$\mathcal{L}_t^{ecc} = -\log(S(f_p(h^t), f_p(h^{t^-}))) - \log(1 - S(f_p(h^t), f_p(h^{t^-_{neg}}))) \tag{8}$$

where $h^{t^-_{neg}}$ is the context graph representation of a different edge (selected randomly within a batch), and $S(\cdot,\cdot)$ computes the cosine similarity between two inputs and applies a linear mapping to convert the value within 0 and 1. The use of $f_p(\cdot)$, an MLP projector function, aligns with standard practices in contrastive learning which ensures that the unsupervised learning framework does not overshadow the primary task.

Finally, the overall training loss is defined as:

$$\mathcal{L} = \mathcal{L}^{echsc} + \lambda \mathcal{L}^{ecc} \tag{9}$$

where λ is a hyper-parameter controlling the contribution of the contrastive loss.

3 Experiments

3.1 Experimental Settings

Datasets and Model Implementation We evaluate our approach on four dynamic network datasets: UCI, Digg, Reddit, and Wikipedia [6, 11, 16]. Since UCI and Digg lack labeled anomalies, we inject 3% synthetic anomalies similar to [13]. All datasets are chronologically split into training (50%), validation (20%), and testing (30%) to reflect real-world scenarios where future data is unknown during training. We evaluate performance under extremely limited labels, providing only 1, 2, or 3 labeled anomaly examples, with labeled normal samples retained proportionally.

To construct the ego-graph $\mathcal{G}^t$, we retain all edges up to and including time t. The context graph $\mathcal{G}^{t^-}$ similarly includes edges up to t but excludes those occurring exactly at t, ensuring it represents only historical information. To manage the potentially large sizes of $\mathcal{G}^t$ and $\mathcal{G}^{t^-}$, we sample 25, 10, and 5 neighboring nodes at the first, second, and third hops, respectively, while ensuring sampled events occur before the target edge's timestamp. In addition, our model uses a two-layer transformer with 4 attention heads, and the hidden size is set to 128 across all modules. The contrastive loss weight λ is set to 0.01 for Digg, UCI, Reddit and 10 for Wikipedia. The models are trained for 20 epochs using the Adam optimizer [9], with a learning rate of 0.0001 for the Digg and UCI datasets, and 0.001 for the Wikipedia and Reddit dataset.

Table 1: Performance comparison for all methods in terms of AUC.

Dataset	Wikipedia			UCI			Digg			Reddit		
Labels	1	2	3	1	2	3	1	2	3	1	2	3
AddGraph	0.631	0.641	0.663	0.732	0.732	0.754	0.739	0.740	0.742	0.578	0.580	0.582
TGAT	0.660	0.674	0.694	0.764	0.790	0.808	0.800	0.804	0.812	0.527	0.541	0.555
GDN	0.516	0.529	0.533	0.510	0.546	0.547	0.511	0.510	0.513	0.532	0.531	0.533
SL-GAD	0.467	0.524	0.526	0.503	0.497	0.502	0.494	0.497	0.504	0.503	0.505	0.500
TADDY	0.616	0.617	0.618	0.769	0.765	0.772	0.811	0.841	0.842	0.516	0.517	0.518
SAD	0.661	0.674	0.710	0.774	0.798	0.816	0.809	0.820	0.841	0.561	0.566	0.592
PREM	0.549	0.555	0.555	0.531	0.532	0.536	0.512	0.513	0.515	0.516	0.535	0.536
EL^2-DGAD	**0.722**	**0.725**	**0.732**	**0.839**	**0.840**	**0.842**	**0.839**	**0.844**	**0.846**	**0.611**	**0.615**	**0.616**

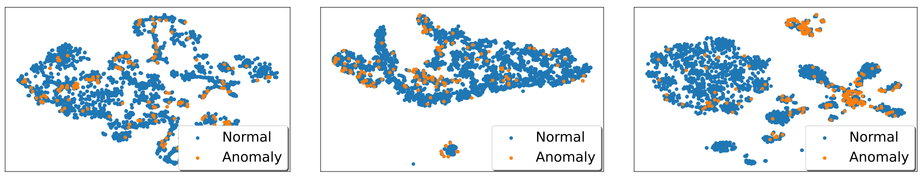
Baselines and Performance Evaluation We benchmark our proposed method against several recent GAD models, namely GDN [8], SL-GAD [26], PREM [17], AddGraph [25], TADDY [13], TGAT [22], and SAD [19]. Among these, GDN, SL-GAD, and PREM are designed for static graph anomaly detection, while AddGraph, TADDY, TGAT, and SAD are tailored for dynamic graph settings. Note that only GDN, TADDY, AddGraph, and SAD can utilize label information by design. For a fair, label-inclusive comparison for other models, we perform a similar setting as in [19]: applying a cross-entropy loss to the edge representations, which are derived from the concatenated node embeddings of the edge endpoints. Thus, the performance of all models can be evaluated based on the classification results.

3.2 Comparison with State-of-the-art GAD

Extensive results in Table 1 show that EL^2-DGAD consistently outperforms all baselines across four datasets, especially under extremely limited labeled anomalies. The decline in performance for GDN, PREM, and SL-GAD validates our hypothesis regarding the critical role of dynamic information. The underperformance of AddGraph and TADDY is likely due to their use of discrete graph snapshots and the lack of a robust semi-supervised loss, which is essential when labels are scarce. Despite SAD being a recent advanced semi-supervised AD approach for dynamic graphs, it also shows a decline in performance under severe label scarcity. Specifically, SAD relies on a deviation network to generate pseudo-labels, which weakens the contrastive learning component when only a few labeled examples are available. In contrast, our method decouples supervised and unsupervised learning, allowing effective generalization with minimal labels. Moreover, our approach offers two key advantages. First, our ego-context hypersphere loss delivers a more precise instance-wise comparison than the deviation network, which relies on global statistics. Second, our model employs a more robust transformer-based encoder in comparison to the GAT encoder in SAD, which additionally improves dynamic graph learning from a global perspective.

Table 2: Performance evaluation on ablation experiments in terms of AUC.

Dataset	Wikipedia			UCI			Digg		
Labels	1	2	3	1	2	3	1	2	3
w/o Local MHA	0.563	0.569	0.582	0.771	0.775	0.772	0.808	0.796	0.802
w/o Global MHA	0.694	0.709	0.716	0.835	0.836	0.838	0.802	0.807	0.800
w/o $\phi(t)$ in MHA	0.465	0.473	0.474	0.789	0.791	0.800	0.719	0.730	0.756
w/o $\mathcal{L}^{ecc}$	0.653	0.683	0.683	0.838	0.837	0.840	0.815	0.821	0.821
w/o ego-context in $\mathcal{L}^{echsc}$	0.650	0.683	0.704	0.824	0.824	0.837	0.730	0.745	0.745
EL^2-DGAD	**0.722**	**0.725**	**0.732**	**0.839**	**0.840**	**0.842**	**0.839**	**0.844**	**0.846**



(a) AddGraph (b) SAD (c) EL^2-DGAD (ours)

Fig. 2: t-SNE Visualization on the Wikipedia dataset.

3.3 Ablation Experiments

Table 2 presents the results of the ablation study. Overall, EL^2-DGAD achieves optimal performance when all components are included. Specifically, removing the local MHA or global MHA significantly degrades performance, underscoring the importance of capturing both neighborhood structures and global interactions. Omitting fine-grained timestamp integration also leads to a considerable drop, as it hinders the model from capturing essential temporal dynamics. Additionally, the drop in performance without the contrastive loss component $\mathcal{L}^{ecc}$ indicates the value of leveraging unlabeled data to improve anomaly detection accuracy. Finally, eliminating the ego-context contrast from $\mathcal{L}^{echsc}$ forces all normal instances toward a single center, limiting the model's ability to capture diverse normal behaviors and effectively distinguish anomalies.

3.4 t-SNE Visualization

Figure 2 shows t-SNE visualizations of the output embeddings from the layer before anomaly scoring in the Wikipedia dataset, where the selected three models were trained with supervision from only one labeled anomaly. Our proposed EL^2-DGAD exhibits a relatively clearer separation, with anomalies forming distinct clusters that are well-separated from normal samples. This distinct clustering suggests that EL^2-DGAD more effectively learns discriminative embeddings, allowing it to capture the subtle distinctions between normal and anomalous interactions.

4 Conclusion

In this paper, we make the first attempt to tackle the anomaly detection problem in dynamic graphs under the challenging condition of extremely limited labels. Our proposed EL^2-DGAD framework introduces a transformer-based graph encoder to capture the evolving graph patterns from both local and global perspectives. This is further augmented with an ego-context hypersphere classification loss and contrastive loss, designed to effectively harness limited labels and abundant unlabeled data. Extensive experiments across various datasets and label rates demonstrate its superiority over existing GAD methods.

Acknowledgements

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant YCJJ20241203.

References

1. Cai, L., Chen, Z., Luo, C., Gui, J., Ni, J., Li, D., Chen, H.: Structural temporal graph neural networks for anomaly detection in dynamic graphs. In: Proceedings of the ACM International Conference on Information and Knowledge Management. pp. 3747–3756 (2021)
2. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys **41**(3), 1–58 (2009)
3. Chen, D., O'Bray, L., Borgwardt, K.: Structure-aware transformer for graph representation learning. In: Proceedings of the International Conference on Machine Learning. pp. 3469–3489 (2022)
4. Chen, J., Fu, S., Zhang, Z., Ma, Z., Feng, M., Wirjanto, T.S., Peng, Q.: Towards cross-domain few-shot graph anomaly detection. In: Proceedings of the IEEE International Conference on Data Mining (2024)
5. Chen, T., Tsourakakis, C.: Antibenford subgraphs: Unsupervised anomaly detection in financial networks. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 2762–2770 (2022)
6. De Choudhury, M., Sundaram, H., John, A., Seligmann, D.D.: Social synchrony: Predicting mimicry of user actions in online social media. In: Proceedings of the International Conference on Computational Science and Engineering. vol. 4, pp. 151–158 (2009)
7. Ding, K., Li, J., Bhanushali, R., Liu, H.: Deep anomaly detection on attributed networks. In: Proceedings of the SIAM International Conference on Data Mining. pp. 594–602 (2019)
8. Ding, K., Zhou, Q., Tong, H., Liu, H.: Few-shot network anomaly detection via cross-network meta-learning. In: Proceedings of the Web Conference. pp. 2448–2456 (2021)
9. Kinga, D., Adam, J.B., et al.: A method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations (2015)
10. Kumagai, A., Iwata, T., Fujiwara, Y.: Semi-supervised anomaly detection on attributed graphs. In: Proceedings of the International Joint Conference on Neural Networks. pp. 1–8 (2021)

11. Kumar, S., Zhang, X., Leskovec, J.: Predicting dynamic embedding trajectory in temporal interaction networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1269–1278 (2019)
12. Liu, Y., Li, Z., Pan, S., Gong, C., Zhou, C., Karypis, G.: Anomaly detection on attributed networks via contrastive self-supervised learning. IEEE Transactions on Neural Networks and Learning Systems **33**(6), 2378–2392 (2022)
13. Liu, Y., Pan, S., Wang, Y.G., Xiong, F., Wang, L., Chen, Q., Lee, V.C.: Anomaly detection in dynamic graphs via transformer. IEEE Transactions on Knowledge and Data Engineering **35**(12), 12081–12094 (2021)
14. Luo, X., Wu, J., Beheshti, A., Yang, J., Zhang, X., Wang, Y., Xue, S.: Comga: Community-aware attributed graph anomaly detection. In: Proceedings of the ACM International Conference on Web Search and Data Mining. pp. 657–665 (2022)
15. Meng, X., Wang, S., Liang, Z., Yao, D., Zhou, J., Zhang, Y.: Semi-supervised anomaly detection in dynamic communication networks. Information Sciences **571**, 527–542 (2021)
16. Opsahl, T., Panzarasa, P.: Clustering in weighted networks. Social Networks **31**(2), 155–163 (2009)
17. Pan, J., Liu, Y., Zheng, Y., Pan, S.: Prem: A simple yet effective approach for node-level graph anomaly detection. In: Proceedings of the IEEE International Conference on Data Mining. pp. 1253–1258 (2023)
18. Ruff, L., Vandermeulen, R.A., Franks, B.J., Müller, K.R., Kloft, M.: Rethinking assumptions in deep anomaly detection. In: Proceedings of the International Conference on Machine Learning Workshop (2021)
19. Tian, S., Dong, J., Li, J., Zhao, W., Xu, X., Song, B., Meng, C., Zhang, T., Chen, L., et al.: Sad: Semi-supervised anomaly detection on dynamic graphs. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 2306–2314 (2023)
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
21. Wei, S., Wang, Y., Cheung, Y.m.: A branch elimination-based efficient algorithm for large-scale multiple longest common subsequence problem. IEEE Transactions on Knowledge and Data Engineering **35**(3), 2179–2192 (2023)
22. Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., Achan, K.: Inductive representation learning on temporal graphs. In: Proceedings of the International Conference on Learning Representations (2020)
23. Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., Liu, T.Y.: Do transformers really perform badly for graph representation? In: Advances in Neural Information Processing Systems. vol. 34, pp. 28877–28888 (2021)
24. Zhang, Z., Zhao, L.: Unsupervised deep subgraph anomaly detection. In: Proceedings of the IEEE International Conference on Data Mining. pp. 753–762 (2022)
25. Zheng, L., Li, Z., Li, J., Li, Z., Gao, J.: Addgraph: Anomaly detection in dynamic graph using attention-based temporal gcn. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 4419–4425 (2019)
26. Zheng, Y., Jin, M., Liu, Y., Chi, L., Phan, K.T., Chen, Y.P.P.: Generative and contrastive self-supervised learning for graph anomaly detection. IEEE Transactions on Knowledge and Data Engineering **35**(12), 12220–12233 (2023)