پانزدهمین سمینار احتمال
و فرآیندهای تصادفی
۸ و ۹ شهریور ۱۴۰۴
دانشگاه کردستان

15th Seminar On Probability and Stochastic Processes
August 30-31, 2025
University of Kurdistan

انجمن آمار ایران

Anomaly Detection in Dynamic Graphs: From Representation Learning to Contrastive Approaches

Hadi Zare*[1], Saeed Rakhshani [1], Ghazaleh Niknam [2]

[1]Department of Data Science and Technology, University of Tehran, Iran.

[2] Ingham Institute for Applied Medical Research, UNSW, Australia.

Abstract

Graphs effectively model complex interactions in real-world data, especially when enriched with node/edge features. Dynamic graphs, whose structures and attributes evolve over time, pose additional challenges in analysis and modeling. Anomaly detection in such graphs is critical in domains like fraud detection and cybersecurity. This paper reviews recent advances in dynamic graph representation learning, focusing on self-supervised contrastive methods and graph augmentations that improve anomaly detection. We also survey key evaluation metrics and discuss open challenges and future research directions.

Keywords: Dynamic Networks, Graph Neural Networks, Anomaly Detection, Contrastive Learning.

Mathematics Subject Classification (2020): xxAxx, xxBxx, xxCxx.

1 Introduction

Many real-world systems—such as social networks, financial infrastructures, biological systems, and cybersecurity environments—exhibit complex relational structures that are best modeled using graphs. In a graph, nodes represent entities, and edges denote interactions or relationships between them. Moreover, in most practical settings, graphs include additional information such as node or edge attributes, making them even richer and more expressive Hamilton, W.L. (2017). The graph structure enables a wide range of machine learning tasks, including node classification, link prediction, community detection, and recommendation systems Niknam (2023). Among these, anomaly detection has emerged as a particularly critical task, especially in domains where detecting irregular behavior can have significant real-world implications, such as fraud prevention, intrusion detection, or abnormal activity monitoring Akoglu (2015). However, in many real scenarios, graph structures are not static. New nodes and edges appear or disappear over time, attributes evolve, and interaction patterns change. These temporal dynamics give rise to dynamic graphs, which introduce additional layers of

*Corresponding author, h.zare@ut.ac.ir

complexity in modeling and analysis. Capturing both the structural and temporal patterns in such graphs is essential for reliable anomaly detection Kipf (2016). To address this challenge, recent advances in graph representation learning aim to encode nodes, edges, or entire graphs into low-dimensional vectors that preserve structural, attribute-based, and temporal information. These representations serve as the foundation for downstream tasks, including anomaly detection. While early approaches focused on supervised or reconstruction-based learning, more recent models leverage self-supervised and contrastive learning techniques, which exploit graph augmentations to learn robust embeddings without requiring labeled data You (2020). This paper provides an analytical overview of this landscape, beginning with the foundations of graph data, continuing through the evolution of representation learning techniques, and culminating in the application of these techniques to anomaly detection in dynamic graphs. We further discuss the role of graph augmentation, evaluation metrics, and open research challenges in this fast-growing area.

2 Graph Data: Structure, Types, and Applications

Graphs offer a powerful and flexible framework for modeling complex relationships among entities. Formally, a graph can be represented as $G = (V, E, X)$, where V is the set of nodes, E is the set of edges, and X denotes the matrix of node features. In some cases, features are also defined for edges, especially in weighted or attributed graphs. Unlike tabular or sequential data, graph-structured data is inherently non-Euclidean, which introduces unique challenges for learning and inference.

2.1 Types of Graphs by Structure

Graphs can be categorized based on their structural properties:

- **Static Graphs**: The topology remains fixed over time. These graphs are typically used to model systems in a snapshot or steady state.
- **Dynamic Graphs**: The node set, edge set, and associated features evolve over time. These graphs better represent systems with continuously changing relationships, such as user interactions or transaction networks.
- **Attributed Graphs**: Each node and/or edge carries feature information (e.g., user profile, product category, edge weights).
- **Directed and Weighted Graphs**: Some graphs capture directional relationships (e.g., information flow) or quantify interaction strength using edge weights.

2.2 Temporal Modeling in Dynamic Graphs

Dynamic graphs are typically modeled in two ways:

- **Discrete-time graphs (snapshots)**: The graph is observed at fixed intervals as a sequence of static snapshots:
  $G=\{G_1,G_2,\ldots,G_t\}$
  This approach allows the use of traditional graph learning techniques on each time slice.
- **Continuous-time graphs (event-based)**: The graph evolves via timestamped interactions represented as events:
  $E=(u_i,v_i,t_i)$.
  This representation captures fine-grained temporal dynamics and is useful for modeling real-time systems such as messaging platforms or financial transactions.

2.3 Real-World Applications of Graph Data

The expressive nature of graph structures has led to their adoption in a wide range of domains:

- **Social networks**: Modeling user interactions, community detection Zare (2020), link prediction, and malicious behavior detection.
- **Biological and chemical networks**: Understanding protein-protein interactions, gene regulatory networks, or molecular structures.
- **Cybersecurity**: Identifying abnormal access patterns, lateral movements, or suspicious communication behaviors in IT infrastructure.
- **Financial systems**: Detecting fraudulent transactions, analyzing payment networks, and modeling customer behavior.
- **Recommendation systems**: Capturing user-item interactions and learning personalized suggestions based on graph proximity.

In many of these settings, graph structures are inherently dynamic, with continuous evolution in connectivity and attributes. Thus, modeling dynamic graphs becomes not only relevant but necessary for accurate and timely analysis Skarding (2021).

3 Graph Representation: Objectives, Methods, and Role in Learning

Graph data, especially within dynamic structures, present significant challenges due to their high dimensionality, dependencies, and temporal complexity. To effectively utilize such data in machine learning frameworks, it is essential to first map them into a compact and meaningful vector space that preserves structural patterns, node features, and temporal relationships. This process is known as Graph Representation Learning.

3.1 Objectives of Graph Representation

The primary goal in this field is to learn mappings that convert nodes, edges, or entire graphs into feature vectors such that:

- Nodes with similar roles or neighborhoods have close embeddings.
- Both local and global graph features are preserved.
- The dynamic structure and behavior over time are taken into account.
- The representations are suitable for downstream tasks such as classification, prediction, and anomaly detection Kipf (2016)

3.2 Categories of Representation Learning Methods

a) Proximity-based Methods

These methods aim to preserve statistical proximity between nodes. For example, the DeepWalk algorithm learns node embeddings by leveraging random walks and techniques similar to word2vec. Node2vec extends this approach by allowing control over the extent of local versus global exploration. While simple and scalable, these methods are limited to static graphs without node features Hamilton, W.L. (2017).

b) Graph Neural Networks (GNNs)

Models such as Graph Convolutional Networks (GCN) and GraphSAGE aggregate the features of each node's neighbors to build more interactive representations. The Graph Attention Network (GAT) improves this process by introducing an attention mechanism. These models

are well-suited for attributed graphs but, in their initial forms, were primarily designed for static graphs Veli (2017).

c) Temporal Models for Dynamic Graphs

To more accurately model structural evolution over time, advanced temporal models have been developed:

- EvolveGCN learns time-dependent parameters for GCNs to analyze graph snapshots over time.
- Temporal GAT (TGAT) and Temporal Graph Network (TGN) utilize attention and temporal memory to capture interaction histories.
- DySAT combines self-attention along structural and temporal dimensions, achieving notable results on dynamic data Rossi, E. (2020).

d) Generative and Graph Autoencoder Models

These approaches aim to learn embeddings by reconstructing the graph structure.

- Graph Autoencoder (GAE) and Variational Graph Autoencoder (VGAE) train based on adjacency structures.
- Adversarially Regularized Graph Autoencoder (ARGA) regularizes the embedding distribution via adversarial training.
- GMM-VGAE integrates probabilistic learning and clustering, enhancing the detection of hidden anomalies in the representation space Kipf (2017).

e) Self-Supervised and Contrastive Learning

Models such as DGI, GRACE, BGRL, and especially GraphCL generate different views of the graph (e.g., by altering structure, features, or subgraphs) to learn noise- and transformation-robust representations. The model is trained to bring embeddings of two correlated versions of a node or graph closer while pushing apart unrelated ones. This category has laid the foundation for advanced anomaly detection techniques, as anomalies typically exhibit distinct behavior under such transformations You (2020).

3.3 The Role of Representation in Downstream Tasks

Learned graph representations serve as a crucial interface between graphs and practical applications such as:

- Node or graph classification, e.g., predicting user roles or labeling molecules
- Link prediction, e.g., identifying potential relationships in social networks or recommendation systems
- Clustering, for uncovering hidden structures like communities or similar behaviors
- Anomaly detection, i.e., identifying nodes or edges whose behavior or structure deviates from the norm

In the following section, focusing specifically on anomaly detection, we will survey various approaches for detecting anomalies in dynamic graphs and explore how powerful representations facilitate the discovery of abnormal behaviors.

4 Anomaly Detection in Dynamic Graphs

In many real-world systems, some behaviors or structures deviate from the dominant patterns. These unusual behaviors, known as anomalies, may indicate unexpected, high-risk, or even malicious events Detecting anomalies in dynamic graphs—where relationships, features, and network structures evolve over time—is considerably more challenging because continuous, intermittent, and multi-dimensional changes must be accounted for.

4.1 Types of Anomalies in Dynamic Graphs

In dynamic graphs, anomalies can manifest in several forms:

- **Node Anomalies:** Abnormal behavior of a node relative to its history or structural context.
- **Edge Anomalies:** Formation of unusual or unprecedented connections between nodes.
- **Subgraph Anomalies:** Incongruent patterns within a specific network subregion.
- **Temporal Anomalies:** Sudden changes in interaction patterns, features, or structure over time.

The combination of these anomaly types introduces significant analytical and computational complexities in modeling.

4.2 Anomaly Detection Approaches

a) Reconstruction-based Approaches

The core idea behind these methods is to reconstruct the graph or its features, with anomalies identified as instances showing high reconstruction error.

- Models like Graph Autoencoder (GAE) reconstruct graph structure via an encoder–decoder framework.
- Variational Graph Autoencoder (VGAE) extends this idea probabilistically by modeling uncertainty.
  In dynamic graphs, these models operate on sequences of snapshots or events, where temporal deviations in reconstruction can signal anomalies Khan (2022); Molaei (2024).

b) Embedding Distribution Modeling

Instead of direct reconstruction, some methods analyze the density distribution of embeddings.

- The GMM-VGAE model combines a probabilistic autoencoder and Gaussian Mixture Model (GMM) to cluster embeddings and detect outliers as anomalies.
  These methods are generally well-suited for snapshot-based dynamic graphs, as they allow distribution evaluation at each time step Niknam (2023).

c) Contrastive Learning-based Approaches

These approaches aim to learn representations robust to reasonable graph perturbations (e.g., partial node or feature removal).

- GraphCL creates two different views of a graph through augmentation and a contrastive loss to extract stable features.
- In anomaly detection, nodes whose embeddings are unstable under such augmentations may be considered anomalous.
- Methods like AnomalyCL refine positive and negative pair definitions to directly target anomaly detection Liu (2021); Rafiei (2024).

d) Temporal and Memory-based Models

For graphs with continuous or instantaneous interactions, models have been designed to capture temporal dependencies using dynamic memory modules.

- Temporal Graph Networks (TGN) maintain memories for each node, considering historical behavior to predict future interactions.
- In anomaly detection, Graph Deviation Network (GDN) learns normal behaviors over time to identify abrupt deviations.
  These models perform well on timestamped data such as security logs or messaging platforms.

e) Hybrid and Multi-task Models

Some approaches combine several of the above methods. For example, models that perform reconstruction, use contrastive loss, and incorporate latent distribution modeling simultaneously.
Such hybrids enable self-supervised anomaly detection from multiple perspectives.
Examples include advanced variants of ARGA or reinforcement-enhanced VGAE versions trained with augmentation-aware strategies Zheng (2021).

4.3 Fundamental Challenges

Despite notable progress, anomaly detection in dynamic graphs still faces key challenges:

- **Lack of Labeled Data:** Anomalies are often rare, hidden, and unlabeled in many domains, necessitating unsupervised or semi-supervised learning.
- **Contextual Definition:** Anomaly is a context-dependent concept—what is normal in one system might be anomalous in another.
- **Scalability:** Increasing graph size demands methods that are computationally efficient and memory-light.
- **Complex Dynamics:** The interplay of structural, feature, and temporal changes requires models capable of simultaneously handling multiple dimensions.

Next, we will focus on one of the critical success factors in contrastive and self-supervised methods: **Graph Augmentation** — deliberate modifications in graph structure or features that enable learning more stable representations and thereby improve anomaly detection performance.

5 Graph Augmentation: Approaches and Its Role in Learning and Anomaly Detection

One of the key concepts in self-supervised and contrastive learning on graphs, especially in the context of representation learning and anomaly detection, is Graph Augmentation.
In its simplest definition, Graph Augmentation is the process of generating different, partial, or perturbed versions of a graph such that the overall semantic information is preserved.
These controlled modifications enable the model to learn representations that are robust against minor changes—a property that is particularly effective in identifying outlier or unstable behaviors, especially in dynamic graphs.

5.1 Objectives of Using Graph Augmentation

- **Enhancing Model Generalization:** Due to structural heterogeneity, graph data may have limited coverage. Augmentation helps the model be trained on a wider variety of scenarios.
- **Enabling Unsupervised Learning:** In the absence of labels, different augmentations of a graph can play the role of "positive" and "negative" examples for contrastive learning.
- **Highlighting Anomalies:** Nodes with unstable representations under augmentations are likely to exhibit anomalous behavior Zhao (2022).

5.2 Types of Graph Augmentation

Unlike image or text data, defining suitable augmentations in graphs is more challenging due to their nonlinear and non-Euclidean structures. Nonetheless, several popular methods have been developed, which can be applied independently or in combination:

a) Node Dropping
Randomly removing a subset of nodes creates an incomplete version of the graph. If the global structural information is preserved, this can help the model learn more fundamental features.

b) Edge Perturbation
Controlled addition or removal of edges modifies the graph's structure. This is particularly useful for testing the model's sensitivity to local relationships.

c) Attribute Masking/Corruption
Randomly deleting or altering features of some nodes or edges encourages the model to leverage structure for feature completion.

d) Subgraph Sampling
Extracting local subgraphs (e.g., via random walks or breadth-first search) helps the model learn more localized and scalable representations.

e) Graph Diffusion / Edge Dropping by Degree
Removing less important edges (e.g., those with low weight or connecting highly dense nodes) aims to focus on more significant structures.

f) Temporal Cropping (for Dynamic Graphs)
In temporal graphs, cropping part of the time history tests the model's robustness to temporal changes You (2020).

5.3 Role of Augmentation in Anomaly Detection

Augmentation not only improves representation quality but also plays a direct role in anomaly detection. Nodes whose embeddings become unstable under structural or feature perturbations are often on the fringes of normal behavior distributions.
In models like AnomalyCL, this instability is directly used as an anomaly signal. Additionally, some methods consider the drop in similarity between a node's representation before and after augmentation as an anomaly score Akoglu (2015).

5.4 Challenges in Designing Graph Augmentations

- **Preserving Structural Semantics:** Unlike images or text, removing a node can alter the entire graph's meaning. Designing augmentations that do not change the graph's semantics is critical.
- **Task Dependency:** An augmentation beneficial for classification might be ineffective or even harmful for anomaly detection.
- **Tuning Augmentation Strength:** The degree of perturbation must be chosen in accordance with the graph's stability and the learning objective.

6 Conclusion and Future Directions

Graphs, as powerful structures for modeling complex relationships in real-world data, play a prominent role across various scientific and industrial domains. With the growth of dynamic data, the need to analyze structures that evolve over time—particularly in tasks such as anomaly detection—has become increasingly important. In this article, we have followed an analytical approach to chart the conceptual path from graph definitions and types, through representation learning methods, to their application in anomaly detection in dynamic graphs.

Graph representation methods, from traditional random walk-based approaches to deep learning and memory-augmented models, all strive to capture graph structure, features, and dynamics in compact and meaningful vector forms. Among these, self-supervised learning methods—especially contrastive learning—utilizing diverse Graph Augmentation techniques have succeeded in producing robust and generalizable representations for downstream tasks like anomaly detection without requiring labeling.

Anomaly detection in dynamic graphs remains a challenging open problem due to the multidimensional nature of the data (structure, attributes, time), scarcity of labeled data, and conceptual variability in defining anomalies. Our review indicates that combining reconstruction-based, distributional, temporal, and contrastive methods can yield stronger and more reliable models. Additionally, precise evaluation metrics—especially for imbalanced data—play a key role in interpreting and validating results.

However, the research landscape still holds many opportunities ahead. Some promising future directions include:

- Designing unsupervised and semi-supervised models capable of accurately detecting anomalies with minimal labeled data.
- Developing application-aware augmentations that better preserve the semantic properties of graphs and more effectively highlight anomalies.
- Integrating temporal, structural, and attribute-based representations within multi-dimensional architectures.
- Creating and releasing large-scale real-world benchmark datasets for dynamic graphs with annotated anomalies.
- Advancing time- and behavior-aware evaluation metrics to better analyze performance in practical scenarios.

In summary, dynamic graphs provide a rich platform for deep analytical work, and anomaly detection in this space is not only a technical challenge but also a practical necessity across many critical domains. The fusion of principled representation learning with problem-driven insights will be the key to progress along this path.

References

Hamilton, W.L., R. Ying, and J. Leskovec, *Representation learning on graphs: Methods and applications.* arXiv preprint arXiv:1709.05584, 2017.

Ghazaleh Niknam and Soheila Molaei and Hadi Zare and Shirui Pan and Mahdi Jalili and Tingting Zhu and David Clifton, *DyVGRNN: DYnamic mixture variational graph recurrent neural networks.* Neural Networks, 2023. **165**: p. 596-610.

Akoglu, L., H. Tong, and D. Koutra, *Graph based anomaly detection and description: a survey.* Data mining and knowledge discovery, 2015. **29**(3): p. 626-688.

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

You, Y., et al., *Graph contrastive learning with augmentations.* Advances in neural information processing systems, 2020. **33**: p. 5812-5823.

Skarding, J., B. Gabrys, and K. Musial, *Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey.* iEEE Access, 2021. **9**: p. 79143-79168.

Veličković, P., et al., *Graph attention networks.* arXiv preprint arXiv:1710.10903, 2017.

Rossi, E., et al., *Temporal graph networks for deep learning on dynamic graphs.* arXiv preprint arXiv:2006.10637, 2020.

Kipf, T.N. and M. Welling, *Variational graph auto-encoders.* arXiv preprint arXiv:1611.07308, 2017.

Khan, W., et al., *Dvaegmm: dual variational autoencoder with gaussian mixture model for anomaly detection on attributed networks.* IEEE Access, 2022. **10**: p. 91160-91176.

Soheila Molaei and Ghazaleh Niknam and Ghadeer O. Ghosheh and Vinod Kumar Chauhan and Hadi Zare and Tingting Zhu and Shirui Pan and David A. Clifton, *Temporal dynamics unleashed: Elevating variational graph attention.* Knowledge-Based Systems, 2024. **299**: p. 112110.

Ghazaleh Niknam and Soheila Molaei and Hadi Zare and David Clifton and Shirui Pan, *Graph representation learning based on deep generative gaussian mixture models.* Neurocomputing, 2023. **523**: p. 157-169.

Liu, Y., et al., *Anomaly detection on attributed networks via contrastive self-supervised learning.* IEEE transactions on neural networks and learning systems, 2021. **33**(6): p. 2378-2392.

Zheng, Y., et al., *Generative and contrastive self-supervised learning for graph anomaly detection.* IEEE Transactions on Knowledge and Data Engineering, 2021. **35**(12): p. 12220-12233.

Zhao, T., et al., *Graph data augmentation for graph machine learning: A survey.* arXiv preprint arXiv:2202.08871, 2022.

Hossein Rafieizadeh, Hadi Zare, Mohsen Ghassemi Parsa, Hadi Davardoust, and Meshkat Shariat Bagheri, "DCOR: Anomaly detection in attributed networks via dual contrastive learning reconstruction",The Thirteenth International Conference on Complex Networks and Their Applications: COMPLEX NETWORKS 2024

H. Zare, M. Hajiabadi and M. Jalili, "Detection of Community Structures in Networks With Nodal Features based on Generative Probabilistic Approach," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 7, pp. 2863-2874, 1 July 2021, doi: 10.1109/TKDE.2019.2960222.