Improving Anomaly detection in Heterogeneous Graphs using GAN-Enhanced Explainability

Premanand Ghadekar   
Dept. of Information Technology  
VIT, Pune  
[premanand.ghadekar@vit.edu](mailto:premanand.ghadekar@vit.edu)

Vaishali More   
Dept. of Information Technology  
VIT, Pune  
[vaishali.more21@vit.edu](mailto:vaishali.more21@vit.edu)

Harshita Bhagat  
Dept. of Information Technology  
VIT, Pune  
[harshita.bhagat21@vit.edu](mailto:harshita.bhagat21@vit.edu)

Sarthak Khare  
Dept. of Information Technology  
VIT, Pune  
[khare.sarthak211@vit.edu](mailto:khare.sarthak211@vit.edu)

Parth More  
Dept. of Information Technology  
VIT, Pune  
[parth.more21@vit.edu](mailto:parth.more21@vit.edu)

Chinmay Saraf  
Dept. of Information Technology  
VIT, Pune  
[chinmay.saraf211@vit.edu](mailto:chinmay.saraf211@vit.edu)

***Abstract— Amid the era of rapid integration of Graph Neural Networks (GNNs) into diverse applications, such as anomaly detection in heterogeneous graphs, the pursuit of interpretable and dependable decision-making models has gained paramount significance. The comprehension of the rationale behind GNN predictions proves vital for instilling trust in their outcomes. While existing GNN explanation methods have made progress in this domain, their ability to provide precise and authentic explanations remains limited. To address these deficiencies, a novel approach named "GAN-Enhanced Explainability" (GEE) is introduced, built upon the foundation of Generative Adversarial Networks (GANs). GEE comprises a dual-component architecture: a generator responsible for producing explanations and a discriminator facilitating the refinement of the explanation generation process. Additionally, this research extends this foundation by incorporating innovative visualization techniques to augment the interpretability of GNN-based anomaly detection, thereby forging a new path in the realm of explainable AI for intricate graph structures.***

***Keywords— Graph anomaly detection, Graph Attention Network, Generative Adversarial Network***

# **Introduction**

Anomalies, often referred to as outliers, represent infrequent and significant deviations from the typical patterns or behaviors observed within a given dataset or system. Detecting and understanding anomalies is of paramount importance across various domains, including data analysis, security, quality control, and many more. The ability to identify anomalies is a fundamental aspect of anomaly analysis, and it involves a variety of methods, ranging from statistical techniques to machine learning algorithms and domain-specific rules. The method selection depends on the features of data and specific problem under consideration. However, detecting anomalies is not always a straightforward task, as distinguishing genuine anomalies from noise or minor data fluctuations can be challenging. Moreover, the definition of what constitutes an anomaly can be subjective, varying from one context to another.

Graphs have gained widespread popularity in numerous application domains. Graphs are applied in social activities for modeling friendships, in biology for representing genetic relationships, in communication for network analysis, and in e-commerce for visualizing product recommendations and user interactions. In these graphs, entities are represented as nodes or vertices, and their relationships are denoted by edges. The utilization of structural information within graphs has introduced a more intricate challenge—Graph Anomaly Detection (GAD) in non-Euclidean spaces. GAD entails identifying anomalies within graph elements, such as sub-graphs, edges, or nodes, either within a single graph or within a graph’s database.

Understanding the reasoning behind a model's predictions is not only a matter of intellectual curiosity but a critical necessity, particularly in applications with significant consequences. While considerable progress has been made in the development of Graph Neural Network (GNN) explainers, these techniques often face limitations in providing precise and authentic explanations. These limitations arise from the inherent complexity of GNNs, which function as highly non-linear and black-box models, thereby making it challenging to extract insights. To address these shortcomings, this research introduces a novel approach called "GAN-Enhanced Explainability" (GEE). This method builds upon the foundation of Generative Adversarial Networks (GANs) to enhance the interpretability of GNN predictions.

In this paper, we will delve further into the GEE framework, providing a comprehensive overview of its components and training process. We will also present experimental results demonstrating the effectiveness of GEE in enhancing the interpretability and reliability of GNN-based anomaly detection. By combining GANs with advanced visualization methods, we aim to bridge the gap between the capabilities of GNNs and the demand for transparent, interpretable decision-making processes in the realm of complex graph structures.

**Objectives:**

1. To increase the interpretability of graph anomaly detection.
2. To detect anomalies in heterogeneous graphs, where nodes and relationships belong to different types.
3. To handle class imbalance, often prevalent in real-world anomaly detection scenarios

# **Literature Survey**

Hwan Kim et al reviewed the recent progress in the domain of graph anomaly detection employing GNN models. They categorized GNN-based techniques based on several criteria, including graph type, anomaly type, and network architecture. Nevertheless, their study did not investigate the utilization of explainable GNNs in the context of graph anomaly detection, nor did it delve into the advancements in class balancing techniques [1]. Xiaoxiao Ma et al. offer an in-depth survey that follows a task-oriented approach, classifying previous research based on their capability to detect anomalous graph elements. They have curated a collection of neural network techniques, cross-referencing them with the research papers in which they were discussed, provided datasets with relevant links, and covered work done on various graph types [2]. The paper authored by Shiwen Wu et al. presents a study of the research in the context of recommender systems that employ Graph Neural Networks (GNNs). In particular, it offers a categorization of GNN-based recommendation models that are based on the types of information they utilize and the tasks they handle. It has also elaborated on F1@K (Precision@K and Recall@K combination) [3]. In the work authored by Abdullah Hamid et al., a comprehensive study is presented for the fake news detection challenge. The study is divided into two core components: (i) the detection of text-based misinformation (TMD), and (ii) the identification of structure-based misinformation (SMD). To tackle the TMD task, the authors employed two distinct approaches, incorporating both Bag-of-Words (BoW) and BERT embeddings within various fusion strategies. In contrast, for the SMD task, Graph Neural Networks (GNNs) were harnessed for different conspiracy theories related to COVID-19 [4]. This paper by Yanling Wang et al. presented a novel Self Supervised Learning (SSL) framework for node-based representation learning. This helped to capture graph properties by clustering the complete graph into different parts. The decoupling approach helped to address learning difficulty and is less biased towards hard instances. Framework was developed using GNN models: GAT, GCN, GraphSage and GIN [5]. The paper by Anshika Chaudhary et al. introduces a DL-based approach, using GNN, for detecting anomalies in social networks like email and Twitter. It combines statistical measures to understand the structure of anomalous nodes, emphasizing the role of hidden layers in impact assessment. GNN utilizes network structure for improved anomaly detection and the approach can adapt to evolving network patterns, ensuring robust detection [6]. The paper by Jaekoo Lee et al. presents a DL-based approach to predict dynamic graph evolution by learning spatio-temporal features. It focuses on detecting dynamic anomalies through node affinity scores, expanding deep learning to non-grid domains. It enables deep learning on dynamic graphs, capturing evolving relationships and effectively handles sparse graphs by leveraging scale-free properties [7]. This article by Xuanguang Chen explores urban computing's role in enhancing urban life quality and focuses on data processing challenges. It investigates various applications of GNNs or urban computing data processing and concludes that GNNs offer superior results. The paper highlights GNNs' superiority in various urban computing aspects, driving progress and suggests multidisciplinary approaches and potential combinations of GNNs with other deep learning models [8]. Saddam Aziz et al. presented a methodology in their research for identifying malicious attacks in Internet of Vehicular Networks, employing an xNN in conjunction with LSTM, and DNN architectures. Feature selection challenges were addressed using K-Means clustering for scoring and ranking.They were able to achieve high accuracy for all models, with xNN achieving 99.7%, with LSTM achieving 90%, and with DNN achieving 92%. Additionally, developed model successfully detected CAN message attacks [9]. Jongmo Kim et al introduced an innovative approach, an evolving graph framework, which is made to address both the intricacies and the non-static characteristics of attribute networks in conjunction with GNNs. This framework comprises two primary elements: feature selection to streamline the snapshots and the utilization of subgraph embedding techniques to derive method-based temporal patterns from these snapshots [10]. Xuhong Wang et al. proposed OCGNN, an innovative one-class classification framework for the detection of anomalies in graphs. OCGNN leverages the representation capabilities of GNNs in conjunction with the traditional one-class objective. In the future, OCGNN is envisioned to have graph-level and dynamic node anomaly detection as well. [11]. Lingqiang Xie et al. introduced an anomaly detection technique that combines Graph Neural Networks (GNN) with a dynamic threshold mechanism, aimed at ensuring consistent, prolonged operational performance and the assessment of satellite health. The accuracy of the proposed algorithm exceeds 98% when applied to telemetry data from satellite power systems, and it significantly enhances the detection performance for anomalies in telemetry time series data [12]. Xiaoling Lin et al. proposed a novel DL methodology employed to identify anomalies in air quality data by integrating both spatial and temporal correlations. The approach utilizes a Context-augmented Graph Autoencoder to effectively address the intricate spatiotemporal characteristics of the data associated with air quality [13]. Qiang Huang et al. introduce GraphLIME which is a local interpretable explainable framework for GNN. GraphLIME generates finite features as predictions and explanations by utilizing feature information from N-hop network neighbors and the HSIC Lasso for nonlinear interpretability. Future work includes explaining structural graph patterns and giving nonlinear explanations for node sets [14]. Yiqiao Li et al. present a survey of the frameworks that are developed for the GNN explanations and provide future directions for research. The researchers delved into a comprehensive examination of XGNNs, both with and without incorporating existing Explainable AI (XAI) approaches, aiming to enhance the interpretability of GNN models. This research suggested developing standard rules for XAI methods [15].

1. **Methodology**

GNNs work by propagating information through the graph. The information can be represented as features associated with the edges and nodes of the graph. The GNN learns to update the features of each node related to the features of its neighbors. This process is repeated for multiple steps, until the GNN learns to represent each node with a feature vector that can be used to make predictions about the graph.

There are many different GNN architectures, each with its own strengths and weaknesses. Some of the well known GNN architectures include:

* Graph Convolutional Networks (GCNs): Inspired by CNNs, GCNs are a subtype of GNNs that utilize a convolution operation to gather the features of a node's neighbors.
* Graph Attention Networks (GATs): GNN which uses an attention mechanism to focus on the most important neighbors of a node when updating its features are GATs.
* Graph Transformers: Graph transformers are a type of GNN that is inspired by transformers,which are one of neural networks categories employed for tasks related to NLP. Graph transformers use a self-attention mechanism to learn graph dependencies which are long-range.

Anomaly detection is a critical task. In GCN, every neighborhood node has the same importance. In GAT, the main idea behind GAT is to compute that coefficient implicitly rather than explicitly as GCNs do.

The coefficient, which is now represented by , was suggested by the authors of GAT to be calculated using node attributes and then fed into an attention function. Lastly, the attention weights are subjected to the softmax function, which produces a probability distribution. In terms of math, we have:

Some nodes might be more important than others which is not handled in the case of GCNs since every node has the same weightage. Also GATs are more efficient than Graph Transformers in some cases. Graph Transformers can be computationally expensive to train and use, while GATs are more efficient. That is why we will be implementing anomaly detection on GATs.

**Algorithm**: Heterogeneous Graph anomaly detection.

**Input:** Input graphs, labeled anomalies.

**Output:** loss, AUC values.

**Required:** hidden channels,output channels

**Steps:**

1: Initialize HGT model with specified parameters

2: Transfer the model and data to the desired device (e.g., CPU)

3: Set up Adam optimizer, initializing it with learning rate of 0.005 and weight decay of 0.00001.

4: Create a list of views for movies and actors based on random flags

5: Create a list of HeteroData objects for each view

6: Initialize adjacency matrices (A), node type matrices (T), and feature matrices (X) for each view

7: Initialize parameters for the HGT model (weight and weight\_node\_type)

8: Define the HGT model architecture, including linear layers and HGTConv layers

9: Define the loss function, including terms for edge prediction, node type prediction, and feature reconstruction

10: for epoch in range(1, 101) do

Set optimizer gradients to zero

Forward pass through the HGT model

Compute the loss

Backpropagate and update model parameters

Calculate anomaly scores on the test set

Compute AUC for performance measurement

Update and best AUC if the current is higher

11: Output the trained HGT model and best AUC

**Training GAN for explanations:**

Anomaly detection becomes very complex for graph anomalies. It is very important to explain these graph anomalies, we proposed GAN architecture for this purpose. This model creates a weighted matrix of input graph and anomaly prediction by GNN which gives the subgraph of

the explanation. In this research, we trained the GAN model on dataset**.** It has two parts: generator and discriminator.

**Generator:** Generator is trained on input graph and anomaly prediction by GNN to generate weight matrix. The generator network learns to create synthetic data points that resemble the normal data distribution. This matrix is given as input for the discriminator and its feedback is taken. Generator architecture and weights are updated based on feedback given by the discriminator.

**Discriminator:** The discriminator architecture classifies graphs for that it learns to distinguish between real and generated graphs by generator. The discriminator tries to fool the generator and this information is given as feedback for the generator and update.



Fig 1. Architecture of the proposed system using GAN based GNN

# 

# **Results and Discussion**

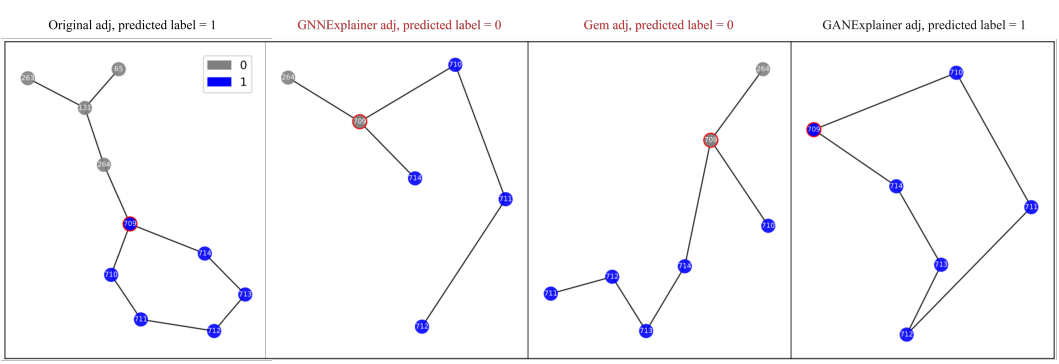


Fig 2 Graph Structure:

Decoding Graph Neural Network (GNN) Predictions

The introduced "GAN-Enhanced Explainability" (GEE) approach demonstrated its effectiveness in enhancing the interpretability of GNNs for detection of anomaly in intricate graph structures. The study aimed to address the limitations of existing GNN explainers and provide accurate explanations for GNN predictions.

The GAN model, trained on a real-world dataset, harnessed the potential of Generative Adversarial Networks to augment the interpretability of GNN-based anomaly detection.

Figure 2 illustrates Node 709 labeled as 1 in Tree-Cycles when K is set to 6. Within this depiction, nodes marked in blue signify that the target GNN predicts a label of 1 for them, while gray nodes indicate a prediction of 0 by the implemented GNN. The red circular node highlights the need to clarify the reason for assigning label of 1 by implementing GNN. The initial subimage showcases the original structure of the graph, where the implemented GNN accurately predicts a label of 1. The following subfigures (from the second to the fourth) depict the explanations produced by Gem, GNNExplainer, and GANExplainer, respectively, for Node 709.

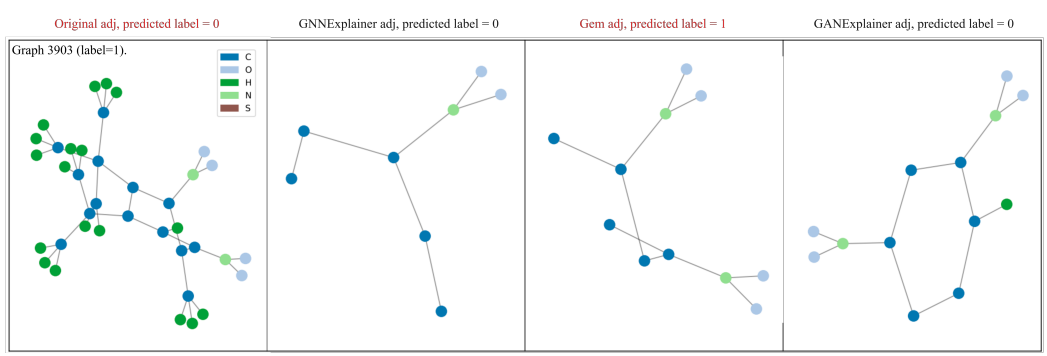


Fig 3 Different Explainers (GNNExplainer, Gem, GEE) comparisons

Fig 3. Illustrates the explanation visualization for Mutagenicity with K set to 15.Label of Graph 3903 is 1, whereas the label of Graph 3904 is 0. The initial column in this representation showcases the original structure of the graph along with the label predicted by the target GNN. The subsequent three features elucidate the graphs with explanations generated by GNNExplainer, Gem, and GANExplainer, respectively. Fig 4. is the other example that we have visualized.

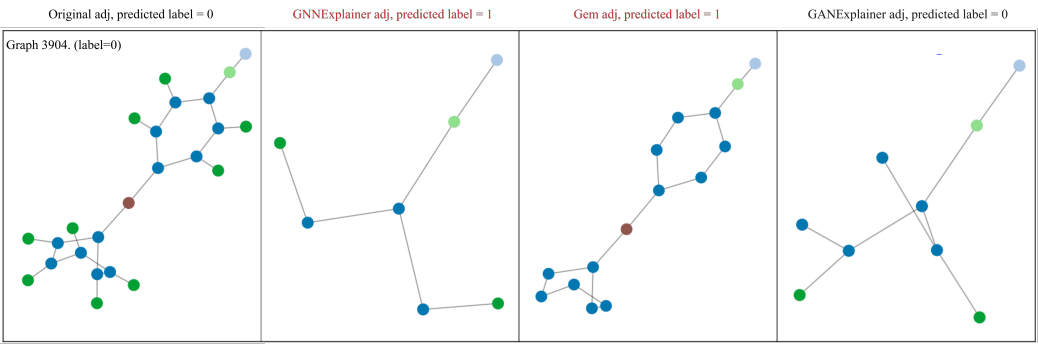


Fig 4 Different Explainers comparisons 2nd

Explanation accuracy was calculated by comparing the target GNN' predictions with the generated explanations. The formula for accuracy of explanation is as follows:

*…(iii)*

Here, 'f' represents the implemented GNN, 'g' is graph, 'Exp' stands for explanation, and 'Test' indicates the total count of items in the test set.

| Epoch | Loss | AUC |
| --- | --- | --- |
| 001 | 1.6282 | 0.7318 |
| 010 | 1.2603 | 0.7110 |
| 020 | 1.0907 | 0.8984 |
| 030 | 1.0731 | 0.9078 |
| 035 | 1.0670 | 0.9102 |
| 040 | 1.0645 | 0.9115 |
| 045 | 1.0631 | 0.9127 |
| 050 | 1.0623 | 0.9132 |
| 055 | 1.0618 | 0.9126 |
| 060 | 1.0614 | 0.9128 |

Table 1: Variation of loss and AUC values with epochs.

Table 2 displays the precision of explanations across synthetic datasets with varied K settings. The results demonstrate the consistent superiority of GEE in providing accurate explanations across all scenarios. In the context of the BA-Shapes dataset, GNNExplainer, Gem, and GEE exhibit commendable performance with synthetic datasets. However, GEE stands out by incorporating multiple enhancements, outperforming both GNNExplainer and Gem specifically on the BA-Shapes dataset. In the case of the Tree-Cycles dataset, GEE performs effectively, whereas neither GNNExplainer nor Gem achieves satisfactory results.

| K (edges) | BA-Shapes  5 6 7 8 | Tree-Cycles  6 7 8 9 |
| --- | --- | --- |
| GNN  Explainer | 0.7941 0.8824 0.9118 0.9118 | 0.2000 0.5429 0.7143 0.8571 |
| Gem | 0.94120.9412 0.9412 0.9412 | 0.7429 0.7429 0.7714 0.8857 |
| GEE | 0.7647 1.000 0.9706 0.9853 | 0.9143 1.0000 0.9714 1.0000 |

Table 2. Comparison of Performance Metrics on Synthetic Datasets Results on GNNExplainer, Gem, GEE models.

The outcomes from real-world datasets are showcased here, with the quantitative assessment outlined in Table 2. It unmistakably shows that the reported findings robustly validate the capability of our proposed GEE to produce explanations, consistently attaining high levels of accuracy across diverse datasets.

| K (edges) | Mutagenicity  15 20 25 30 | NCI1  15 20 25 30 |
| --- | --- | --- |
| GNN  Explainer | 0.6981 0.7188 0.7442 0.7834 | 0.6909 0.7031 0.7566 0.8004 |
| Gem | 0.6705 0.7027 0.7741 0.7949 | 0.6253 0.7055 0.7956 0.8126 |
| GEE | 0.6935 0.7442 0.7650 0.7857 | 0.6642 0.7494 0.7908 0.8273 |

Table 3. Comparison of Performance Metrics on Real-World Datasets Results on GNNExplainer, Gem, GEE models.

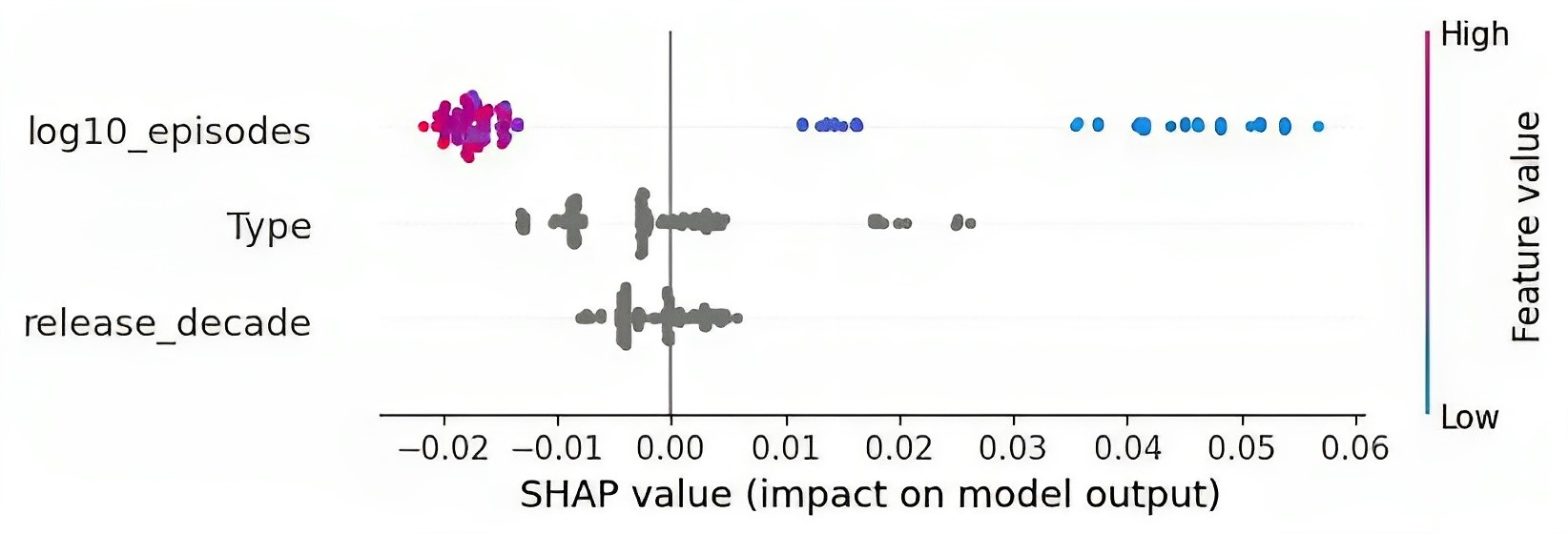


Fig.5 Results of SHAP, impact of features on output of model.

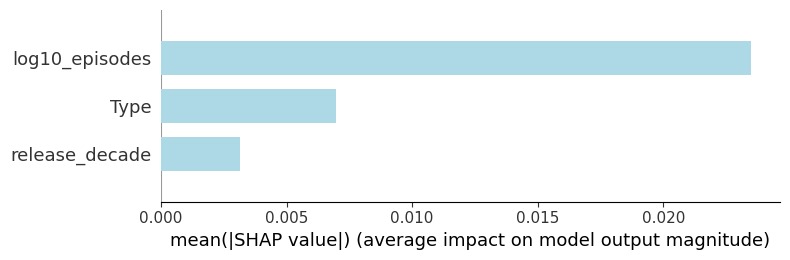


Fig.6 Visualization of Impact of SHAP Values on Model Performance

Figure 5 and 6 shows the output of the SHAP algorithm applied to the dataset. In this study, we have used this module for the purpose of anomaly explanation. values on the X axis show the impact of features on model output and Y axis highlights the features. SHAP helps to visualize the features and highlights the features that are more important for performance of the model, this helps the anomaly explanation.

# **Conclusion**

In summary, the "GAN-Enhanced Explainability" (GEE) approach was introduced to enhance the interpretability of GNNs for anomaly detection within intricate graph structures. Anomalies, representing significant deviations from typical patterns, hold paramount importance across numerous domains. The inherent complexity of GNNs often hinders the provision of accurate and genuine explanations for their predictions, necessitating the development of enhanced explainability methods.

GEE, through its combination of generator and discriminator components, harnessed the potential of Generative Adversarial Networks to overcome these challenges. The generator produced explanations for GNN predictions, while the discriminator provided feedback to refine the explanation generation process. The research showcased the effectiveness of GEE in enhancing the interpretability and reliability of GNN-based anomaly detection.

By achieving the defined objectives, which encompass improved interpretability, the handling of class imbalance, and the identification of anomalous edges and sub-graphs in heterogeneous graphs, GEE emerges as a promising avenue for advancing explainable AI in the realm of complex graph structures.

This work contributes to the ongoing efforts aimed at improving the transparency and reliability of AI models, particularly in applications where the consequences of their decisions hold significant implications. It is anticipated that GEE and similar approaches will continue to narrow the gap between the capabilities of AI models and the increasing demand for transparent, interpretable decision-making processes in real-world applications.

##### **References**

1. Kim, Hwan, et al. "Graph anomaly detection with graph neural networks: Current status and challenges." IEEE Access(2022).
2. Ma, Xiaoxiao, et al. "A comprehensive survey on graph anomaly detection with deep learning." IEEE Transactions on Knowledge and Data Engineering (2021).
3. Wu, Shiwen, et al. "Graph neural networks in recommender systems: a survey." ACM Computing Surveys 55.5 (2022): 1-37.
4. Hamid, Abdullah, et al. "Fake news detection in social media using graph neural networks and NLP Techniques: A COVID-19 use-case." arXiv preprint arXiv:2012.07517 (2020).
5. Wang, Yanling, et al. "Decoupling representation learning and classification for gnn-based anomaly detection." Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. (2021).
6. A. Chaudhary, H. Mittal and A. Arora, "Anomaly Detection using Graph Neural Networks," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, pp. 346-350, doi: 10.1109/COMITCon.2019.8862186.
7. J. Lee, H. Bae and S. Yoon, "Anomaly Detection by Learning Dynamics From a Graph," in IEEE Access, vol. 8, pp. 64356-64365, 2020, doi: 10.1109/ACCESS.2020.2983987.
8. X. Chen, "Application of GNN in Urban Computing," 2020 International Conference on Communications, Information System and Computer Engineering (CISCE), Kuala Lumpur, Malaysia, 2020, pp. 14-17, doi: 10.1109/CISCE50729.2020.00010.
9. Aziz, Saddam, Muhammad Talib Faiz, Adegoke Muideen Adeniyi, Ka-Hong Loo, Kazi Nazmul Hasan, Linli Xu, and Muhammad Irshad. "Anomaly Detection in the Internet of Vehicular Networks Using Explainable Neural Networks (xNN)" Mathematics 10, no. 8: 1267,(2022).
10. Jongmo Kim , Kunyoung Kim, Gi-yoon Jeon, and Mye Sohn, "Temporal Patterns Discovery of Evolving Graphs for Graph Neural Network (GNN)-based Anomaly Detection in Heterogeneous Networks", Journal of Internet Services and Information Security (JISIS), volume: 12, number: 1 (February), pp. 72-82,(2022).
11. Wang, X., Jin, B., Du, Y. et al. One-class graph neural networks for anomaly detection in attributed networks. Neural Comput & Applic 33, 12073–12085 (2021).
12. Lingqiang Xie, Dechang Pi, Xiangyan Zhang, Junfu Chen, Yi Luo, Wen Yu, "Graph neural network approach for anomaly detection", Measurement, Volume 180,2021,109546.
13. X. Lin, H. Wang, J. Guo and G. Mei, "A Deep Learning Approach Using Graph Neural Networks for Anomaly Detection in Air Quality Data Considering Spatiotemporal Correlations," in IEEE Access, vol. 10, pp. 94074-94088, 2022.
14. Q. Huang, M. Yamada, Y. Tian, D. Singh and Y. Chang, "GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks," in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 7, pp. 6968-6972, 1 July 2023, doi: 10.1109/TKDE.2022.3187455.
15. Li, Yiqiao, Jianlong Zhou, Sunny Verma, and Fang Chen. "A survey of explainable graph neural networks: Taxonomy and evaluation metrics." arXiv preprint arXiv:2207.12599 (2022).

