Uxue Garcia Rodriguez

Graph Neural Networks Explainability in Social Networks: Evaluation of Methods and Feature Relevance

FINAL MASTER'S PROJECT

Directed by Dr. Jordi Duch Gavaldà

Master's degree in Computer Security Engineering and Artificial Intelligence

UNIVERSITAT ROVIRA i VIRGILI

Tarragona

2025

Abstract

This work analyses the predictive capabilities of a GNN on a social network and its later explanatory abilities. Network nodes were labeled based on the augmented features of their first-level neighbors. GraphSAGE and GCN were trained using only the augmented features and on invented and augmented features. Then, explainability was assessed using GNNExplainer, PGExplainer, and GraphMask, and compared against ground-truth logic using fidelity, unfaithfulness, and characterization metrics. GNNExplainer proved most effective, closely aligning with the labeling rules. The results highlight both the strengths and limitations of these explainers in capturing meaningful structural patterns in the graph data.

Resumen

Este trabajo analiza las capacidades predictivas de una GNN en una red social y su posterior capacidad explicativa. Los nodos fueron etiquetados según la información estructural de sus vecinos más cercanos. Se entrenaron los modelos GraphSAGE y GCN usando únicamente estas características, así como otras inventadas. La explicabilidad se evaluó con GNNExplainer, PGExplainer y GraphMask, y se comparó con la lógica real mediante métricas de fidelity, unfaithfulness y characterization. GNNExplainer fue el más efectivo, alineándose con las reglas de etiquetado. Los resultados evidencian fortalezas y limitaciones de estos métodos para capturar patrones estructurales relevantes en grafos.

Resum

Aquest treball analitza les capacitats predictives d'una GNN en una xarxa social i la seva posterior capacitat explicativa. Els nodes van ser etiquetats segons la informació estructural dels seus veïns més pròxims. Es van entrenar els models GraphSAGE i GCN usant únicament aquestes característiques, així com altres inventades. La explicabilitat es va avaluar amb GNNExplainer, PGExplainer i GraphMask, i es va comparar amb la lògica real mitjançant mètriques de fidelity, unfaithfulness i characterization. GNNExplainer va ser el més efectiu, alineant-se amb les regles d'etiquetatge. Els resultats evidencien fortaleses i limitacions d'aquests mètodes per a capturar patrons estructurals rellevants en grafs.

Keywords: Graph Neural Networks, Explainability, GNNExplainer

Table of Contents

List of Figures

List of Tables

1 Introduction

In today's digital era, real-world data is highly represented as a network of interconnected objects that evolved into **complex ecosystems** that play a crucial role in modern theories of information creation and distribution within organizations, where modeling relationships between social entities as complex graphs that reveal patterns of interaction beyond the boundaries of individual objects. (Tabassum et al., 2018). Graph Neural Networks (GNNs) have emerged as a powerful tool for modeling these structures (Kakkad et al., 2023), enabling the analysis of both feature information and structural data from graphs (Fan et al., 2019). However, as these models gain popularity (Platonov et al., 2023), their **interpretability becomes increasingly important** (Pope et al., 2019), especially in these fields, given that they are often considered 'black boxes' (Yuan et al., 2020a), (Fang et al., 2023).

In the context of Graph Neural Networks, explainability refers to the capacity to render a model's predictions **transparent and understandable** (Zhang et al., 2024). It is important to understand why the system fails, highlight features that correlate specific patterns or classes, and thus reveal any inherent weaknesses or limitations in the learning process (Bugueño et al., 2024).

Without a clear understanding of the inner workings that drive a model's predictions, the trustworthiness of deep models is undermined, restricting their deployment in areas where fairness, privacy, and safety are paramount. To ensure that these models can be used safely and reliably, it is essential not only to achieve high predictive accuracy but also to offer explanations that are understandable to users from diverse fields. This requirement for transparency has spurred extensive research into explanation techniques, ultimately aiming to provide insight into the decision-making processes of deep neural networks (Yuan et al., 2020b).

This work addresses a fundamental question: Are current explainability techniques capable of correctly discerning **important structural relationships** in the classification of nodes in social networks?

To address this problem, a series of controlled experimental approaches are performed, where node labels are assigned based solely on the **attributes of their first level neighboring nodes**, disregarding the node's own characteristics, creating a hierarchy of relationships that should be captured by explainability models. This design allows to evaluate whether the GNN models used (SAGE and GCN) can understand these **neighborhood relationships** and correctly classify nodes. By knowing in advance how the clusters have been created, the accuracy with which different explainability techniques (GNNExplainer, PGExplainer and GraphMask) can explain the predictions made by the models is evaluated.

This research not only contributes to the **validation of explainability** methods in GNNs but also provides insights into the relative importance of structural attributes versus node attributes in social network classification tasks. The results obtained have the potential to improve both the design of predictive models and the understanding of the social dynamics captured in these networks.

In the following chapters, the methodology followed throughout the process will be described in detail, starting with the steps used to create the labels, followed by the design of the applied GNN architecture, and finally, the design of the experiments to evaluate and analyze the explainability methods.

1.1 State of art

Graph theory, an essential branch of discrete mathematics (Newman, 2003b), has its origins in the famous Königsberg bridges problem, posed in the 18th century (Fortunato, 2010). The city of Königsberg, divided by the Pregel River and connected by seven bridges, intrigued its inhabitants with a puzzle: is it possible to cross all bridges exactly once and return to the starting point? Leonhard Euler solved this problem in 1736 by demonstrating its impossibility through an innovative approach that represented regions as nodes V(G) and bridges as edges E(G) (Biggs et al., 1986). This work not only solved the puzzle but also laid the groundwork for graph theory, which has since grown beyond its recreational origins into a **well-established field**, essential for understanding the mathematical principles that define discrete structures (Chung & Lu, 2006).

Formally, a graph G is defined as an ordered pair (V(G), E(G)), where V(G) is a non-empty set of vertices (nodes) and E(G) is a set of edges (links) (Diestel, 2000), (Bondy & Murty, 1976). Each edge in E(G) connects exactly two vertices from V(G). If an edge *e* connects vertices *u* and *v*, it can be denoted as *e* = {u,v}, and *u* and *v* are called the endpoints of e (Bollobás, 1998), (Bondy & Murty, 1976). Parameters such as the order v(G) (number of vertices) and size e(G) (number of edges) quantify structural properties, while concepts like paths, cycles, and node degree (number of edges connected to a node) are essential for analyzing connectivity and flow in networks (Gross et al., 2018), (Newman, 2010). (Newman & Park, 2003)

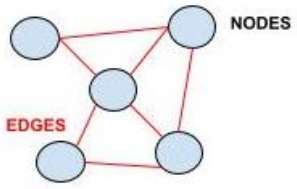

Figure 1: Basic graph network representation

These mathematical foundations of graph theory have enabled the development of analytical tools and algorithms that are now widely applied across disciplines, from traditional transportation and logistics problems to the modern analysis of social networks and complex systems. (Newman, 2010), (Easley & Kleinberg, 2010)

The **evolution** of graph theory has been tied to its ability to model complex systems. In the 20th century, Reinhard Diestel's work in "Graph Theory" expanded the theoretical framework by introducing concepts such as infinite graphs and tree decompositions, crucial for distributed algorithms and large-scale network analysis (Diestel, 2000). With the advent of the digital revolution, it was observed that many real systems, especially social networks, follow patterns similar to the Barabási-Albert model (BarabáSi & Albert, 1999). This model shows that, as the network grows, new nodes tend to connect with those that already have many connections, forming "hubs" or central nodes. Studies on networks such as Facebook and Twitter have shown this behavior, supporting the idea of preferential attachment (Kwak et al., 2010), (Mislove et al., 2007).

To later successfully approach the evaluation of different explainability models to test how it identifies the important features in the task of classifying nodes in social networks, it is necessary to first design a model that successfully captures all the

features and produces adequate results to be able to perform the evaluation. First, it is necessary to design a model that successfully captures all the features and produces adequate results to be able to perform the evaluation. For all this, it is essential to understand how GNNs, especially Graph Convolutional Neural Networks (GCN) and GraphSAGE, apply in this context.

For instance, in Ying et al. (2018) the authors introduce a **scalable GCN framework** that efficiently aggregates information from a node's local neighborhood to generate meaningful embeddings. Although originally designed for the recommender system in Pinterest, this approach is highly adaptable to social networks, where users are connected in complex ways, as it allows the model to capture both direct and indirect relationships (Ying et al., 2018). The graph convolution operations presented can be modified to provide a sound basis for effective node classification. As demonstrated by Kipf & Welling (2016) the proposed GCNs can simultaneously encode both graph structure and node features efficiently, significantly outperforming other methods in semi-supervised classification tasks. This capability is crucial for correctly interpreting social interactions and assigning accurate labels to different network actors, while maintaining a reasonable computational cost even in complex graphs (Kipf & Welling, 2016).

Similarly, in their work on inductive representation learning on large graphs, Hamilton et al. (2017) present **GraphSAGE**, an innovative framework that effectively generates node embeddings by sampling and aggregating information from local node neighborhoods. This approach demonstrates significant advantages for social network analysis, as it enables the model to generalize to previously unseen nodes and for graphs without node features. The authors' experiments on real-world datasets, including social platforms like Reddit, reveal that GraphSAGE consistently outperforms some methods for node classification tasks. By learning aggregator functions rather than individual embedding, GraphSAGE captures both topological structure and node features, making it particularly effective for identifying user types and communities within social networks. This inductive learning capability represents a substantial advancement over transductive approaches for analyzing large-scale, evolving social media platforms. (Hamilton et al., 2017)

But these models are not only used for node classification, but they have also been tested in various tasks related to social networks:

- **Fake news detection:** Shu et al. (2019) applied GNNs for fake news detection on social media, introducing a deep hierarchical co-attention model.
- **Recommendations:** Wang et al. (2019) introduced Knowledge Graph Attention Network for recommendation which represents an initial attempt to leverage structural knowledge through an information propagation mechanism.
- **Link prediction:** Zhang et al. (2018) proposed SEAL, a new framework that leverages GNNs to simultaneously learn from local subgraphs data, node embeddings, and attributes for effective link prediction.

Although the previous examples analyze the different model's performances across tasks such as node classification, fake news detection, recommendations, and link prediction, their practical adoption relies on a crucial additional factor,

explainability. To address this challenge, some explainability methods have emerged:

1. **GNNExplainer**

   In the paper GNNExplainer: Generating Explanations for Graph Neural Networks (Ying et al., 2019), the authors introduce GNNExplainer, a method that explains any GNN's predictions without requiring changes to the underlying model or re-training. It leverages the recursive neighborhood aggregation mechanism to identify key subgraphs and the most relevant node features influencing the prediction. This method has been used in a number of investigations.

2. **PGExplainer**

   In the paper Parameterized Explainer for Graph Neural Network (Luo et al., 2020) the authors propose PGExplainer, a method designed to offer a general understanding of GNN models by explaining several instances at once. PGExplainer leverages the latent representations learned by GNNs to identify the key subgraph structures that are pivotal to their predictions. Moreover, its efficiency in inductive scenarios makes it particularly suitable for real-world applications.

3. **GraphLime**

   In the paper GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks (Huang et al., 2020), GraphLIME method is introduced that works with any model to explain GNN predictions. It uses information from nearby neighbors and their predicted labels with a tool called HSIC Lasso to find complex relationships, remove extra noise, and keep only the important features so users can trust the results and compare models.

4. **Captum**

   In the paper Captum: A Unified and Generic Model Interpretability Library for PyTorch (Kokhlikyan et al., 2020), the authors present Captum, an open-source library for PyTorch that implements a variety of gradient- and perturbation-based attribution methods. It efficiently scales to handle large inputs and works with many types of models, helping users understand which features drive model predictions.

In the ongoing exploration of methods to enhance the explainability of Graph Neural Networks (GNNs), Yuan et al. (2021) introduces **SubgraphX**, a novel approach that focuses on identifying significant subgraphs with the objective of understanding GNN predictions. This method employs Monte Carlo tree search to efficiently explore various subgraphs within a given graph. Wang et al. (2024) applied this approach to analyze GNN vulnerabilities by generating explanatory subgraphs to perform evasion and backdoor attacks, demonstrating its utility not only in interpretability but also in security assessments of GNN models.

In addition to these approaches, explainability in graph regression tasks is also being explored. For example, Zhang et al. (2023) introduce **RegExplainer**, which tackles the challenges of interpreting GNN predictions in regression settings. RegExplainer generates post-hoc explanation subgraphs that highlight the key factors

influencing the regression outcome, all without requiring retraining or modifying the original model. Similarly, Royat et al. (2024) present the **GINTRIP framework**, which focuses on spatio-temporal GNNs by not only accurately predicting future signals but also providing interpretable, time-aware subgraphs.

To conclude the review of the state of the art, various GNN architectures developed to address a range of challenges have been examined, along with several explainers designed to elucidate the reasoning behind their solutions. However, despite these advances, the explainability of these models remains relatively underexplored (Yuan et al., 2020b), even though understanding the reasoning behind these models is essential. (Patel & Sahni, 2022).

This work tries to explore that specific aspect by evaluating whether **explainers can highlight the meaningful connections** that establish predictions, particularly in settings where node labels depend on the attributes of their neighbors, rather than being misled by irrelevant features.

1.2 Regarding the problem

In a social network environment, where almost all users are part of some network (Escobar et al., 2021), understanding how Graph Neural Networks make decisions is of critical importance. This research addresses the fundamental question of whether GNNs primarily focus on node attributes or on the overall graph structure when making predictions.

A key aspect of this study is the assignment of node labels based on neighboring node metrics, effectively implementing the principle of "I am classified this way because my neighbors have these characteristics." This approach deliberately emphasizes the relational nature of social networks, where a nodes identity is partially determined by its connections. The central question becomes: can GNN explainers correctly identify this relationship-based classification mechanism?

The expectation is that, if the model is truly capturing the structural patterns embedded in the graph, **the explainer should highlight the specific relationships** as the primary drivers of each prediction. This would indicate that the **explainer is correctly tracing the rationale behind the node labels**.

When invented features are introduced as noise into the system, ideally, both the model's predictions and its explanations should remain largely unchanged if the GNN is truly learning from structural patterns. If the model's behavior or the explainer's interpretations significantly shift in response to these arbitrary features, it indicates that the GNN is not properly capturing the intended neighborhood-based classification mechanism.

This research examines GNN explainers' ability to accurately attribute the models' decisions to the actual determining factors, the metrics of neighboring nodes, rather than being misled by noise features. By testing the explainer's resilience to noise, it can be evaluated if they correctly identify that a nodes classification is because of its **network position** and the **characteristics of its neighbors**, not from arbitrary individual attributes.

The results of this investigation will enhance the understanding of both GNN behavior and the reliability of GNN explainability methods in social network contexts, ultimately contributing to more transparent and trustworthy graph-based machine

learning systems.

2 Motivation, objectives and methodology

2.1 Motivation and objetives

In order to solve this problem and analyze the performance of the different explainer techniques, three different objectives are proposed:

1. **Generating features and labels for the social network.**

The first objective is to build a set of features for the nodes of the network dividing them into two categories:

- Invented features: Simulated attributes such as number of weekly posts, hours of activity, account age, user's category of interest, among others
- Augmented features: Attributes derived from the network structure itself, such as centrality[1], clustering coefficient[2] or PageRank[3].

Node labels will be assigned based solely on the augmented characteristics of their first-level neighbor nodes, ignoring the node's own features. This design is intended to embed a clear structural rule into the classification logic, which will later serve as the **ground truth for evaluating** the behavior of the GNN explainers.

2. **Evaluation of GNN models for understanding neighbor relationships.**

The second objective is to evaluate the ability of different GNN models, GraphSAGE and GCN to learn and capture the relationship between node labels and the features of their neighbors, identifying which model achieves a better understanding of these structural relationships.

3. **Analysis and validation of explainability techniques in the identification of relevant relationships.**

With prior knowledge of how the labels have been constructed, the third objective is to analyze and quantitatively evaluate the effectiveness of the three explainability techniques to correctly identify the neighboring nodes and connections that were relevant in the classification.

2.2 Methodology

The work is planned in several phases, as shown in the table below. In the **Definition** phase, a literature review is conducted to study explainability techniques in GNNs. This phase includes analyzing libraries such as *GNNExplainer*, *GPExpaliner*, and *GraphMask*. Next, during the **Design** phase, two main objectives are addressed. First, features and labels are defined for the social network by selecting a suitable graph dataset and creating specific rules based on the information of each node's neighbors. Second, the most suitable GNN model architecture is selected by comparing different models and optimizing their hyperparameters.

In the **Implementation** phase, GNN models are trained where node labels

[1] Centrality measures a node's importance in a complex network, shaping its dynamic behavior. (Rodrigues, 2019)
[2] It can be defined as the ratio of the number of triangles it forms to the maximum possible, reflecting how interconnected its neighbors are. (Saramäki et al., 2007)
[3] In graph theory, PageRank assigns each node a global importance score. (Chung, 2014)

depend on the augmented features of their neighbors. Explainability techniques are then applied to identify which neighbors influence the classification.

In the **Analysis** phase, the neighbor relationships known to be relevant are compared with those identified by the explainers, using specific evaluation metrics.

Finally, in the **Conclusion** phase, the ability of the explainer techniques to correctly distinguish the relevant structural relationships in the classification of nodes in social networks is evaluated.

Table 1: Proposed methodology

PHASES	OBJETIVES	TASKS
Definition	Review explainability techniques in GNNs	Literature review on GNNs
		Literature review on explainers
Design	1.Creation of features and labels for the social network	Selection of the graph dataset
		Generation of invented features
		Generation of augmented features
		Generation of the node labels
	2.Evaluation of the GNN models	Selecting the most suitable GNN model architecture
		Optimizing model hyperparameters
Implementation		Training the GNN with all generated features
	3.Analysis and validation of explainability techniques	Applying *explainability* techniques to identify influential edge connections
Analysis		Comparing ground true labeling rules with the ones found by the explainers
		Evaluation of the explainers based on the define metrics
Conclusion	Evaluation of the impact and limitations of *explainability* in GNNs	Analyzing the effectiveness of the techniques

3 Feature design and label assignment

3.1 Data source and analysis

For this research, data was obtained from the web SNAP[4], a resource developed by Jure Leskovec. Among the various collections available, the "Stanford Large Network Dataset Collection" offers different datasets relevant to machine learning applications. In this study, the "**ego-Facebook" dataset**, which has social circles from Facebook, proposed by Leskovec and Mcauley (2012), was selected as the primary data source. The dataset's structure was used to assign both synthetic *"invented" features* and attributes derived from the network structure to each node.

The network used comprised 4 039 nodes and 88 234 edges. Its structure was characterized by high connection density and well-defined community formation. The following image shows a representation of the network, highlighting the first 100 nodes to facilitate visualization.

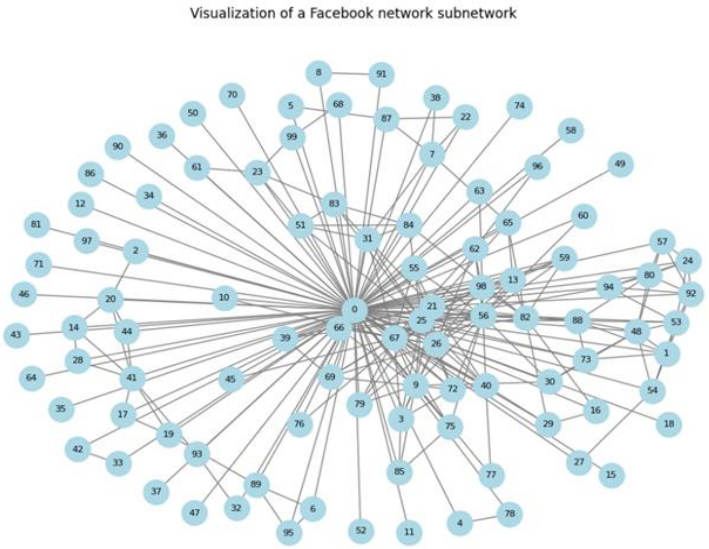

Figure 2: First 100 nodes of the network

The topological analysis reveals fundamental properties that help understand the structure and dynamics of this social network.

1. **Connection density**: With an average degree of approximately 44 connections per user, the network shows high connectivity.
2. **Small-world effect**: The average path length of 3.69 and a diameter of 8 confirm the presence of the "small-world" phenomenon, as defined by Watts and Strogatz, where information can propagate quickly through the network with few intermediate hops. (Newman, 2000)
3. **Community formation**: The average clustering coefficient of 0.6055 indicates a strong tendency for cohesive group formation, so it can be said that a user's friends tend to be friends with each other.
4. **Positive assortativity**: The assortativity value of 0.0636 suggests a slight

[4] **S**tanford **N**etwork **A**nalysis **P**latform

tendency for users to connect with others of similar degree.

Given the **wide range** between the minimum and maximum degrees of the graph, the degree distribution was analyzed. This distribution reveals the presence of a few nodes with very high degrees, hubs, and a large majority of nodes with relatively low degree, a pattern characteristic of many real networks, including social networks.

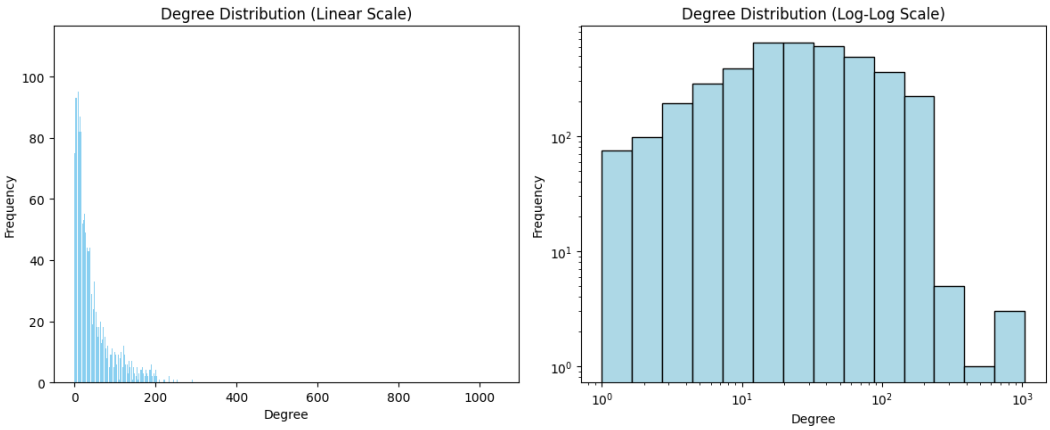

Figure 3: Networks degree distribution

Identifying influential nodes is crucial in social network analysis, to better understand how the network is composed, and which are the most influential nodes. In social networks specifically, understanding central nodes helps explain how information spreads through communities and identifies potential "*influencers*" who might expedite or impede transmission processes.

The centrality analysis was performed using three complementary metrics to capture different aspects of the node influence:

Betweenness Centrality: This metric quantifies the frequency with which a node is on the shortest paths between other nodes in the network (Newman, 2003a). It has been selected because it identifies actors that function as 'bridges' between different communities, exerting control over the flow of information.

Degree Centrality: This metric was chosen for its ability to directly identify nodes with the highest number of connections. In the context of social networks, these nodes represent individuals with numerous social ties, so it can be concluded that these nodes have greater access to information and resources.

Eigenvector Centrality: This measure was included because, unlike the previous ones, it considers not only the quantity of connections but also their quality assigning higher scores to nodes connected to other highly central nodes (Bonacich, 2007). This perspective is crucial for identifying individuals who, while not having the highest number of connections, are linked to particularly influential people.

The analysis of the metrics obtained reveals significant patterns in the influence structure that were useful to better understand the network.

First, it is notable that nodes 107, 1684 and 1912 consistently appear in the top five in at least two of the three metrics analyzed, suggesting that these users have a multidimensional influence on the network.

However, there are nodes that stand out among them. On the one hand, node 107 occupies the first position in both *betweenness* and *degree*, with significantly high values (0.48 and 0.26 respectively), which indicates that it not only has numerous direct connections but also acts as a crucial bridge between different communities in the network. Node 1912, on the other hand, shows an interesting pattern of ranking first in *eigenvector* while ranking in the top 5 of the other measures, suggesting that it is connected to particularly influential users and has a considerable number of direct connections.

Finally, it is noteworthy that nodes with high *betweenness* do not necessarily coincide with those that stand out in *eigenvector*, which reveals the existence of different structural roles in the network, some users function mainly as connectors between communities, while others derive their influence from their associations with prestigious nodes.

3.2 Data generation

3.2.1 Structural feature selection

As a dual approach to node representation was implemented, first the augmented features, which capture various topological aspects of the nodes in the network, were established.

Eleven structural features were extracted for each node, each capturing different dimensions of its position and importance in the network, can be grouped into three main categories:

1. Community and clustering features:
    - Community Feature (greedy modularity algorithm)[5]
    - Clustering Coefficient
    - Core Number (K-core)[6]
2. Direct centrality metrics:
    - Node Degree
    - Degree Centrality
3. Advanced centrality metrics:
    - Betweenness Centrality
    - Closeness Centrality[7]
    - PageRank
    - Eigenvector Centrality

[5] Based on the greedy modularity algorithm (Newman, 2004), this algorithm assigns each node an identifier according to the community it belongs to. This information is fundamental to capture the clustering structure of the network and how nodes are organized into cohesive communities.
[6] This measure reveals the position of the node in terms of the hierarchical structure of the network. (Montresor et al., 2011)
[7] Measures the proximity of a node to all other nodes in the network. (Zhang & Luo, 2017)

- Katz Centrality (implemented with α=0.005, β=1.0)[8]

After extracting these features, a correlation analysis was applied to identify and eliminate redundancies. This step optimizes the model by reducing dimensionality without losing relevant information, avoids multicollinearity problems that could generate predictive instability, and favors better generalization to new data by means of a more compact feature set.

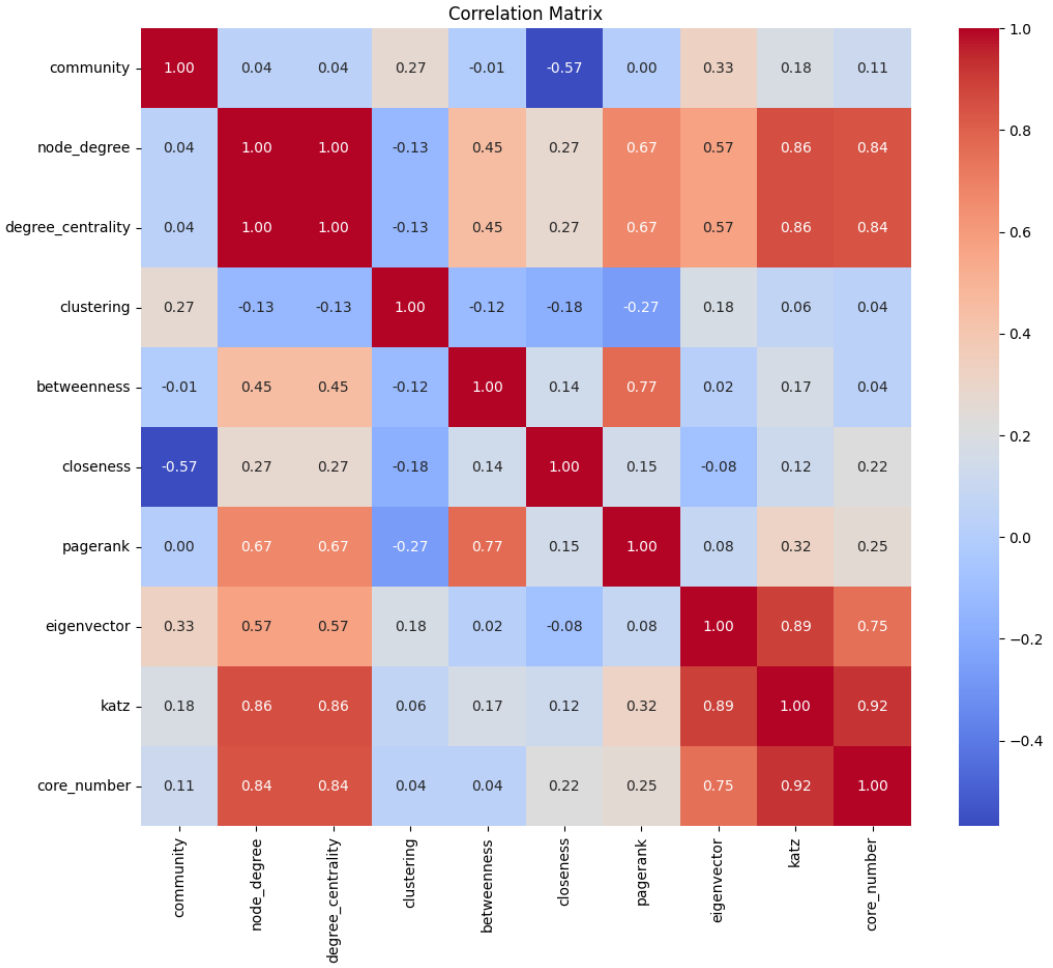

Figure 4: Features correlation matrix

After performing the correlation analysis between the extracted structural features, a significant pattern of redundancies in the data can be observed in the figure above. The correlation matrix revealed **strong** and expected **associations** between various metrics: *node degree* and *degree centrality* showed a perfect correlation (r=1.0); *katz* and *core number* showed a very high correlation with each other (r=0.92) and also with the *node degree* feature (r=0.86 and 0.84 respectively) and *degree centrality* (r=0.86 and 0.84) metrics; while *katz* and *eigenvector* also exhibited a strong association (r=0.89).

Based on these results, an **optimal subset of features** was selected to minimize redundancy while preserving the relevant topological information. The final configuration includes *community features*, *node degree*, *clustering coefficient*,

[8] This centrality (Katz, 1953) measures the importance of a node by taking into account not only its direct connections, but also all indirect routes to other nodes, weighted by their length. (Landherr et al., 2010)

betweenness, *closeness*, *pagerank* and *eigenvector.*

This selection eliminated highly correlated features such as *degree centrality*, *katz* and *core number*, resulting in a more compact and computationally efficient representation that maintains the diversity of perspectives on the structural position of each node.

3.3 Structural labeling based on neighborhood properties

Once the nodes that composed the network were analyzed and the relevant characteristics were assigned to each node, **labeling construction** was performed. The main objective of this process was to generate a set of labels for the nodes based specifically on the characteristics of their neighbors, not on their own individual characteristics.

This labeling strategy allowed the creation of a network where the node labels were determined by their **first-level neighbors' features**, creating structural relationships that a GNN should be able to capture. This approach allowed the structural role of each node within its local network context to be captured, revealing interaction patterns that would not be evident through an isolated analysis of node properties. The fundamental purpose was to examine whether the GNN could effectively identify and exploit these relational patterns during node classification, assessing its ability to capture dependencies that go beyond individual features.

The labeling algorithm that was designed uses **thresholds based on percentiles** of the observed distribution of each metric, rather than arbitrary absolute values. This decision allows automatic adaptation to the specific characteristics of each network, ensuring robust classification regardless of the scale or density of the network.

3.3.1 Calculated neighbor metrics

The features were calculated (Table 2) for the properties of the neighbors of each node, and the following thresholds were applied for the subsequent segmentation:

- **Mean degree of neighbors**: To find out whether the node is mainly connected to nodes with high or low connectivity.
- **Mean clustering coefficient of neighbors**: Indicates whether the node's neighbors are part of densely connected structures.
- **Mean betweenness centrality of neighbors**: Shows whether the neighbors act as bridges in the network.
- **Mean closeness centrality of neighbors**: Determines whether the neighbors have central or peripheral positions.
- **Mean PageRank of neighbors**: Evaluates the relative importance of the neighbors according to this algorithm.
- **Mean eigenvector centrality of neighbors**: Indicates whether the neighbors are connected to other important nodes.
- **Mean betweenness-to-clustering ratio of neighbors**: To identify

those neighboring nodes that connect distinct but not very cohesive communities.

- **Degree rank between neighbors**: Quantifies the heterogeneity of connectivity in the node's local environment, calculated as the difference between the maximum and minimum degree of its neighbors.
- **Normalized variance of community characteristics**: Implemented as the coefficient of variation of neighbors' community values; to identify nodes whose neighbors belong to different communities.

Table 2: Metrics obtained from averages of neighbor node

Feature	Min	Max	Mean	Std	10%	25%	50%	75%	90%
Mean degree (neighbors)	3.5	1045	67.632	74.603	30.000	41.479	52.111	69.800	101.126
Mean clustering (neighbors)	0.042	0.944	0.588	0.098	0.499	0.550	0.596	0.637	0.682
Mean betweenness (neighbors)	0.000	0.481	0.011	0.035	0.000	0.000	0.000	0.008	0.024
Mean closeness (neighbors)	0.178	0.460	0.280	0.030	0.243	0.266	0.271	0.299	0.319
Mean pagerank (neighbors)	0.000	0.007	0.000	0.001	0.000	0.000	0.000	0.000	0.001
Mean eigenvector (neighbors)	0.000	0.076	0.005	0.008	0.000	0.000	0.001	0.006	0.016

3.3.2 Threshold setting procedure

By analyzing the different ranges in which the metrics fall, in order to classify the nodes in a robust way, dynamic thresholds divided into "high" and "low" were established based on the statistical distribution of each metric.

- For the metrics, degree, clustering coefficient and closeness, the 75th percentile was chosen as the high threshold and the 25th percentile as the low threshold.
- For the betweenness centrality with many low values, the 75th percentile was used as the high threshold and the median (50th percentile) as the low threshold.
- For highly skewed metrics where only a few nodes score high (PageRank, eigenvector centrality), the 90th percentile was used as the high threshold and the 40th percentile as the low threshold.
- In addition, the 90th percentile was also used for the degree measure to find those nodes that stood out the most.

These percentile thresholds were defined based on the statistical distributions observed in the network data, degree metric also had an extreme range (3.5–1045), with its 75th percentile at 69.800, highlighting the presence of a few highly connected nodes; betweenness centrality's distribution was dramatically skewed, with its 75th

percentile at only 0.008 while half of all nodes had values below 0.000125; PageRank's highly selective 90th percentile was at 0.001; and eigenvector centrality's 90th percentile at 0.016 confirms that there were a few nodes achieving significant influence.

Table 3: High and low percentile thresholds selected for each metric

Metric	High threshold (percentile)	Value	Low threshold (percentile)	Value
Mean degree (neighbors)	75%	69.800	25%	41.479
Mean clustering coefficient (neighbors)	75%	0.637	25%	0.550
Mean betweenness centrality (neighbors)	75%	0.008	50%	0.000125
Mean closeness centrality (neighbors)	75%	0.299	25%	0.266
Mean PageRank (neighbors)	90%	0.001	40%	0.000265
Mean eigenvector centrality (neighbors)	90%	0.016	40%	0.000609

3.4 Classification methodology

The classification process developed classifies nodes into six distinct structural categories, each defined by specific criteria based on the aggregate properties of their **immediate neighbors**:

1. Structural connectors

These nodes function as **bridges** between structurally heterogeneous environments, allowing information transfer between otherwise disconnected or weakly connected groups. Identification criteria:

- High variance in the metrics of their neighbors, the mean of the variance of the neighbors' metrics exceeds the high threshold set for variance.
- High betweenness or high degree range between neighbors

These nodes **connect different communities**, hierarchies or structural regions of the network, playing a crucial role in the overall integration and information flow between subsystems.

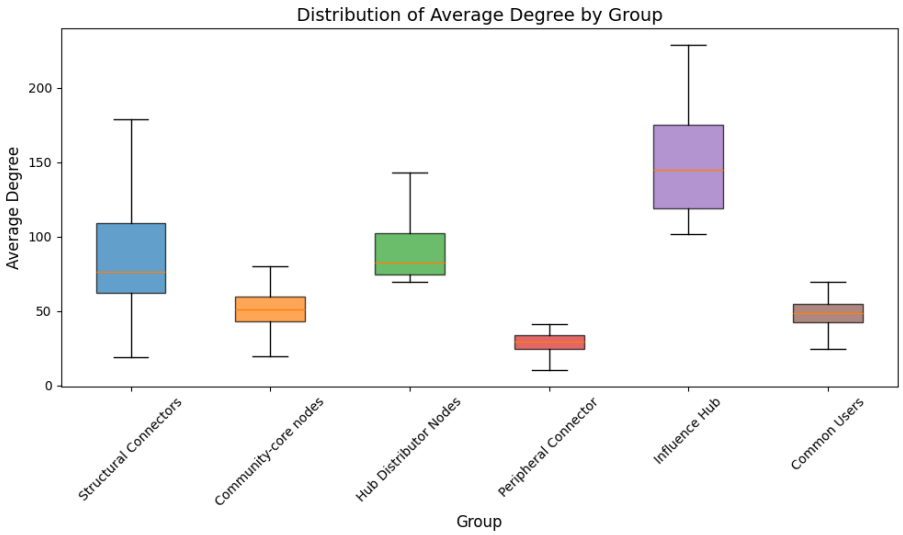

Figure 5: Distribution of neighbor average degree by cluster

2. Community-core nodes

The nodes in this second group constitute the cohesive core of **well-defined communities**, characterized by being surrounded by neighbors with strong community ties and dense triangular structures. Characterized by the metrics:

- A low variance in neighbor community labels, which suggests a more homogeneous local structure.
- High average clustering coefficient among neighbors.

Functionally, these nodes act as points of high local stability, being surrounded by **neighbors with strong community ties** and dense triangular structures, they reinforce the redundancy and robustness of the network.

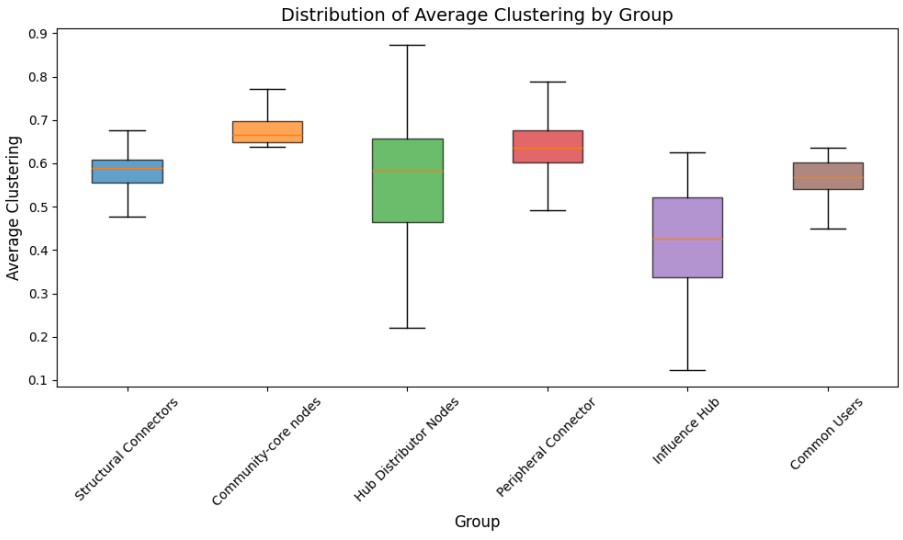

Figure 6: Distribution of average clustering of neighbors by cluster

3. Hub distributor nodes

They function as **information distribution centers**, characterized by **connecting to other high centrality nodes**. Characterized by the metric:

The mean degree among neighbors exceeds the predefined high threshold.

4. **Peripheral connector**

Nodes **connected to other nodes of low influence** and with limited access to the rest of the network. Characterized by the metrics:

- Low average degree between neighbors
- Low average closeness centrality

Together with their neighbors, these nodes may form isolated communities or terminal chains, with **limited participation in the main information flows** of the network.

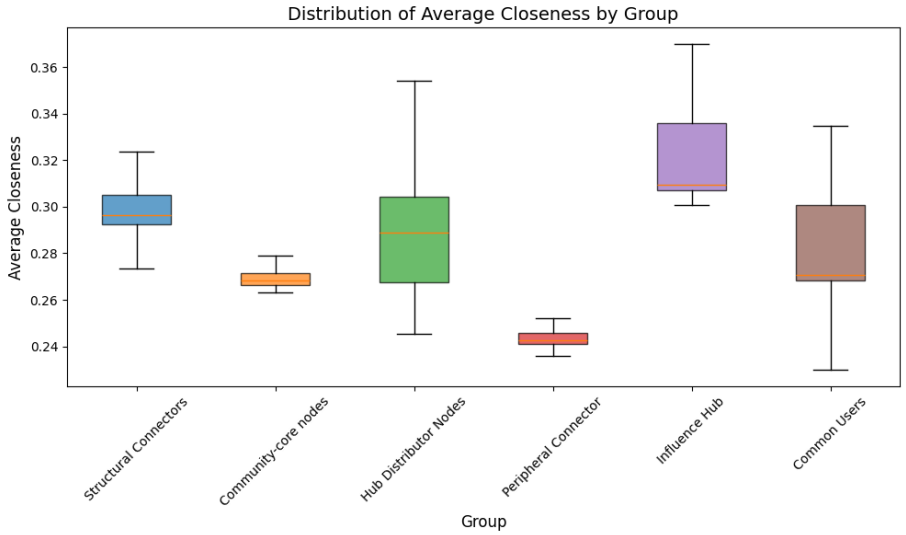

Figure 7: Distribution of average closeness of neighbors by cluster

5. **Community hub neighbor nodes**

These nodes were distinguished by the fact that their neighbors have a **high degree of community homogeneity**, that is, they have a very central position in the network. Identification criteria:

- High average community characteristics among neighbors
- High average closeness centrality
- A degree mean above all the other groups, above the 90 percentiles.

6. **Common users**

In this category the nodes were grouped **that do not meet the specific criteria** of the previous categories, representing ordinary structural positions within the network.

3.4.1 Optimal label assignment for multiclass nodes

During the labeling process, **1,085 nodes** were identified that matched criteria for more than one category. For these nodes, a **distinctiveness scoring system** was implemented that assesses the degree of membership in each candidate category.

First, all relevant metrics were normalized by dividing by their global maximum threshold, with those metrics for which lower raw values imply greater role affinity, such as community variance in cohesive clusters, inverted after normalization so that smaller original values translate into higher scores. Second, these metrics were aggregated using fixed weights. Third, for every node meeting multiple role criterion, the category associated with the **highest aggregate score** was selected.

- For Structural Connectors, distinctiveness weights betweenness centrality (50%) and variance of community characteristics (50%), capturing their bridging function between heterogeneous environments.
- For Community Hubs, inverted normalized variance of neighbor metrics (50%), which favors homogeneity, was combined with high clustering coefficient (50%), prioritizing structural cohesion.
- For Distributor Hubs, the score integrates the average degree (50%) and eigenvector centrality (50%), reflecting their strategic connectivity with other influential nodes. Although the main group definition did not explicitly include eigenvector centrality, it was added to the rule to resolve ties among nodes with similar degrees, ensuring a more robust selection.
- For Peripherals, it was calculated by the inverted degree (40%) and the inverted closeness centrality (60%), where low values in these metrics increase distinctiveness as a peripheral node.
- For Community Ambassadors, the formula combines degree (50%), closeness (30%) and community homogeneity (20%, implemented as variance inversion).

Upon completion of the labeling process, the nodes were distributed as follows.

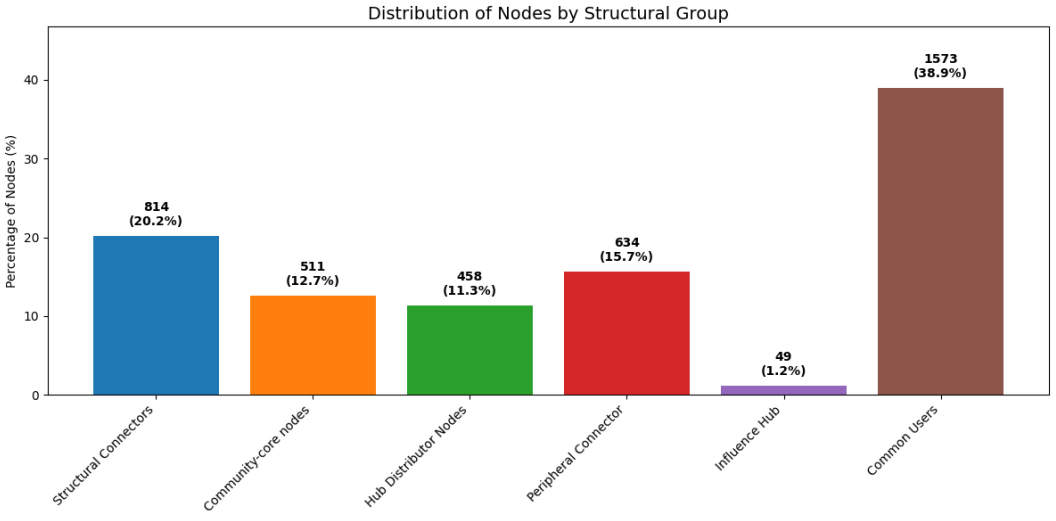

Figure 8: Distribution of nodes by cluster

As can be seen in the above illustration, common users form the largest segment of the network, reflecting a broad base of low-centrality nodes, while influence hubs constitute a very small elite. Structural connectors and peripheral connectors together account for over a third of all nodes, underscoring the dual importance of bridging roles and shaping the network topology.

3.5 Invented feature selection

These synthetic variables were designed based on the exploratory analysis to capture properties **representative** of users in social networks such as Facebook. The chosen distributions seek a balance between diversity and plausibility, generating a dataset that, while synthetic, can reveal patterns similar to those that would be observed in real data.

- **Age:** A normal distribution was chosen, with a mean of **35 years** and a standard deviation of 10 years with boundaries between 15 and 100 years. This distribution reflects a population centered on young and middle-aged adults, thus allowing for the possibility of older users.
- **Gender**: A categorical distribution was implemented with **fixed** probabilities: 45% female, 45% male and 10% unspecified. This distribution represents a balance between male and female users, including a percentage for those who prefer not to declare their gender or identify with other options.
- **Category of Interest:** A categorical distribution with differing probabilities was established to simulate the variety of popular topics on social networks. Categories with higher probability (20%) represent broadly popular interests such as "Entertainment" and "Lifestyle", while categories with lower probability (10%) would represent more specific niches such as "Technology", "Sports" and "Education". The full distribution includes "News" (15%), "Entertainment" (20%), "Technology" (10%), "Travel" (15%), "Lifestyle" (20%), "Sports" (10%) and "Education" (10%).
- **Weekly Posts:** A Poisson distribution with a mean of **5 posts per week** was used to model discrete events such as the number of posts. This distribution captures the variability in posting patterns between users with different levels of activity.
- **Daily Active Hours**: A normal distribution was implemented with a mean of 2 hours and a standard deviation of 1 hour and a boundary between 0 and 12 hours. This distribution models the variability between users, from occasionally inactive to more intensive users.
- **Account Age**: A truncated exponential distribution (parameter λ: 0.5) with a maximum limit of 10 years was chosen. This distribution reflects the historical growth of the platform, with a higher proportion of recent accounts and a lower proportion of old accounts.

3.6 Normalization of the features

Before saving the features and labels for their use, the data was normalized. This step was crucial to improve the performance and stability of the models, especially in cases like this, where features of different scales and distributions were used. Normalization prevents features with large magnitudes from dominating the model. Additionally, normalizing facilitates interpretation, as values now represent proportions relative to their maximums, making it easier to understand the relative significance of each feature.

For the **augmented features**, Min-Max normalization was applied due to their extremely sparse nature. These features have a large variability in their ranges, which can complicate model learning.

Min-Max normalization transforms the data into the range [0,1], compressing all values to a uniform range, facilitating model convergence.

For the invented characteristics, an approach adapted to the nature of the data was applied:

- Continuous numerical variables such as, age, weekly publications, daily hours, and account seniority were normalized to the range [0,1] using their respective known minimum and maximum values.
- Categorical variables (gender and interests) were kept unnormalized, retaining their original discrete values to maintain their interpretability.

4 Architecture of GNN models

This research explores whether GNN models can effectively identify node roles when those roles were defined by the structural properties of their local network environment rather than by the node's individual attributes.

4.1 Data preparation

For the training process, a division of the data into training and test sets was performed, 80% for training and 20% for testing, using the *RandomNodeSplit* transformation from ***PyTorch Geometric***.

This split strategy was specifically implemented to address the significant imbalance present in the classes of the data set. The use of function-specific parameters ensures that, despite the imbalance, nodes of all classes were maintained in the test set, thus allowing a more robust assessment of model performance in each of the categories, including minority groups.

4.2 Implemented architectures

Two fundamental architectures were implemented in the field of graph neural networks:

The first architecture, **GraphSAGE**, was selected for its ability to efficiently handle large-scale graphs like *ego-facebook*, which contains multiple overlapping social circles. The architecture of this model, through its technique of sampling and aggregation of neighborhood characteristics (Hamilton et al., 2017), proves particularly effective for modeling friendship connections and group affiliation patterns that define this social network. This feature proves particularly valuable in this network analysis, as node characteristics may be influenced by connectivity patterns beyond immediate neighbors.

The dimensionality of latent space was set to values of 64 and 128, magnitudes large enough to capture the complexity of the network. Regularization through dropout (0.0 and 0.2) was specifically incorporated to avoid excessive dependence on highly connected nodes.

Simultaneously, the Graph Convolutional Network model was implemented for its ability to efficiently spread information through direct connections, which is particularly useful in social graphs, enabling the model to capture complex patterns in user relationships. (Ying et al., 2018), (Kipf & Welling, 2016)

Both implementations share a similar structure that includes initial projection layers, possible configurable intermediate layers and an end layer that projects to the class space. **ReLU activations** between layers were used because of their proven effectiveness in mitigating the problem of gradient vanishing (Lu et al., 2020), particularly relevant in social graphs where signals must propagate through multiple connections. The final output was processed with a **logarithmic softmax** function to obtain a **probability distribution** over the classes, facilitating interpretation.

The exploration of different network depths (1-3 layers) is justified by the particular social network structure, where communities can manifest at different levels of proximity to the central node. The deeper layers allow to capture

relationships between distant communities, while the more superficial architectures focus on more direct relationships, both aspects relevant in the characterization of social circles.

4.3 Design of experiments

As previously discussed, both GraphSAGE and GCN architectures were implemented with varying depths, hidden dimensions, and regularization strategies. Building on this foundation, the experimental design focused on a systematic hyperparameter search to assess their impact on model performance.

- **Architecture**: GraphSAGE and GCN, two approaches with different information propagation mechanisms.
- **Depth**: 1, 2 and 3-layer models were used to capture dependencies at different distances in the graph.
- **Learning rate**: Values of 0.005, 0.01 and 0.02 were tested to determine the optimal convergence speed.
- **Regularization**: The effect of dropout was evaluated with rates of 0.0 and 0.2.
- **Representative capacity:** Dimensionalities of 64 and 128 were evaluated in the hidden layers.

This scan generated **72 different setups** that were thoroughly evaluated with a **500** times training process for each setup.

4.3.1 Training and Evaluation Process

For each configuration, the following protocol was executed:

- Initialization of the model with the corresponding hyperparameters.
- Optimization by Adam with the specific learning rate.
- Training for 500 epochs, saving the loss at each iteration.
- Monitoring every 10 epochs of the performance on the training set.
- Final evaluation on the test set.

During the evaluation, the calculation and storage of gradients were disabled to optimize computational resources and ensure deterministic predictions. The evaluation function calculates:

- Global performance metrics included *accuracy*, *macro F1-score*, precision score, balanced accuracy, and the *confusion matrix*. These metrics were selected to address the inherent class imbalance in the network, ensuring a robust evaluation that does not favor majority classes.
- A class analysis was also conducted, tracking correctly and incorrectly classified nodes along with their indexes, true labels, and predicted labels.

In addition, a comprehensive tracking and selection system was established during the execution of the experiments. This system enabled the identification of the best-performing GraphSAGE and GCN models for each architectural depth. Model selection was primarily based on the highest *balanced accuracy score* achieved on the test index. In cases where multiple models exhibited identical *balanced accuracy score*, the model with higher *macro F1-score* and *accuracy* was selected to ensure a more reliable overall performance. However, all metrics were considered in the comparison.

Selected models were stored to facilitate subsequent comparative analyses. Furthermore, visualizations of the **training loss curves** were generated, providing an analytical tool to examine the convergence patterns and training stability across different experimental configurations.

As already mentioned, the experiments were performed on two representations of the graph. The first graph includes the augmented features derived from the network structure itself. The second graph, on the other hand, with hybrid characteristics, combining real attributes with synthetic variables designed to emulate typical properties of users in social networks.

4.4 GNN performance using augmented features only

After performing all the experiments corresponding to this network, the best models selected.

Table 4: Summary of Top GNN Models (Augmented Features Only)

	Num layers	lr	dropout	Hidden channels	Balance acc	f1 macro	test acc
SAGE	1	0.02	0	64	0.626	0.639	0.763
SAGE	2	0.02	0	128	0.628	0.645	0.764
SAGE	3	0.02	0	128	0.683	0.671	0.768
GCN	1	0.02	0	128	0.577	0.573	0.726
GCN	2	0.02	0	128	0.554	0.533	0.721
GCN	3	0.02	0	128	0.594	0.595	0.734

As shown, optimal performance was achieved with a three-layer architecture GraphSAGE model and a learning rate of 0.02, with a *balance accuracy* of **68%**, an *accuracy* of approximately **77%** and a *macro F1-score* of **67%** on the test set.

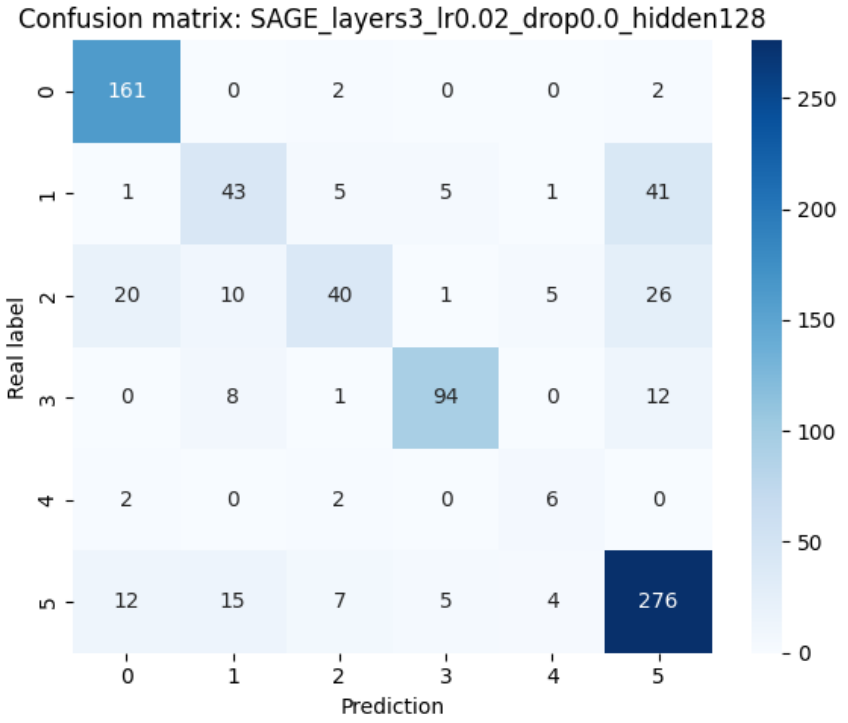

Figure 9: Confusion Matrix for the three-layer GraphSAGE model (lr = 0.01, dropout = 0.0, hidden = 128) using augmented features only

In **Figure 9**, the confusion matrix shows particularly high true positive rates for class 0 and class 5. In contrast, classes 2 and 3 exhibit elevated misclassification rates: many true class 2 nodes were assigned to class 5 or class 0, and a notable proportion of class 1 nodes were predicted as class 5.

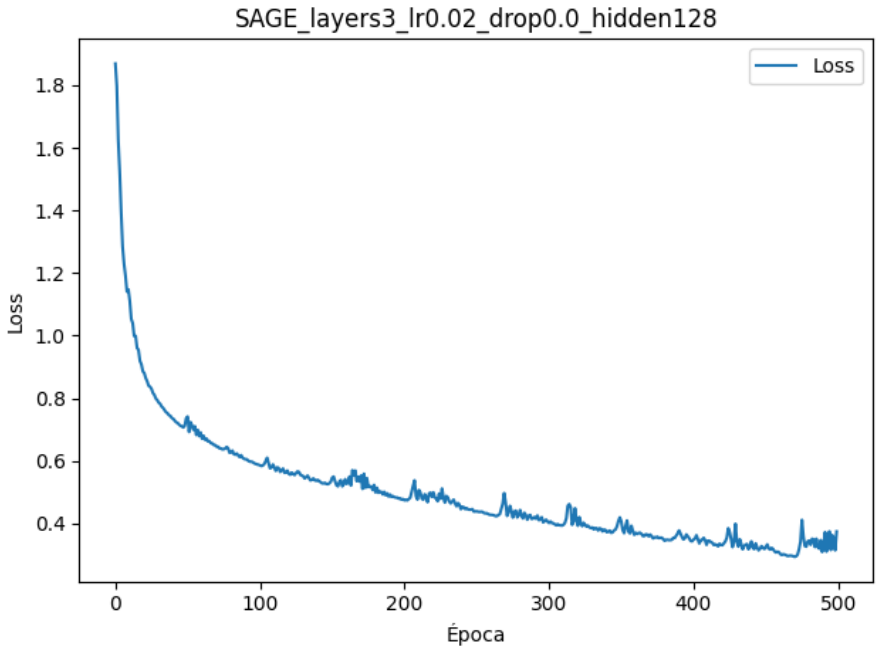

Figure 10: Training loss curve over 500 epochs for the three-layer GraphSAGE model (lr = 0.01, dropout = 0.0, hidden = 128) using augmented features only

Figure 10 shows the training loss declining sharply in the first 50 epochs and then decreasing more slowly, stabilizing around 0.35–0.40. Small fluctuations appear after epoch 200, but no sudden spikes happened.

4.5 GNN performance using combined features

In the second figure, the following results were obtained after performing all the experiments:

Table 5: Summary of Top GNN Models (Mixed Features)

	Num layers	lr	dropout	Hidden channels	Balance acc	f1 macro	test acc
SAGE	1	0.02	0.0	64	0.6070	0.6090	0.7423
SAGE	2	0.02	0.2	128	0.6127	0.6102	0.7385
SAGE	3	0.01	0.0	64	0.6531	0.6375	0.7509
GCN	1	0.02	0.0	128	0.5695	0.5723	0.7249
GCN	2	0.02	0.2	128	0.5811	0.5902	0.7385
GCN	3	0.01	0.0	64	0.5755	0.5567	0.7299

As shown in **Table 5**, the three-layer GraphSAGE model with a learning rate of 0.01 achieved the best scores with a *balanced accuracy* of **65.3%**, a standard *accuracy* of **75.1%**, and a *macro F1-score* of **64%**

Figure 11 presents the confusion matrix for this best model. The strong diagonal entries for classes 0 and 5 indicate that these frequent classes are well separated. However, medium-frequency classes still overlap many true class 5 nodes are misclassified as class 1 or 2, and class 2 nodes occasionally appear as class 1 or 5.

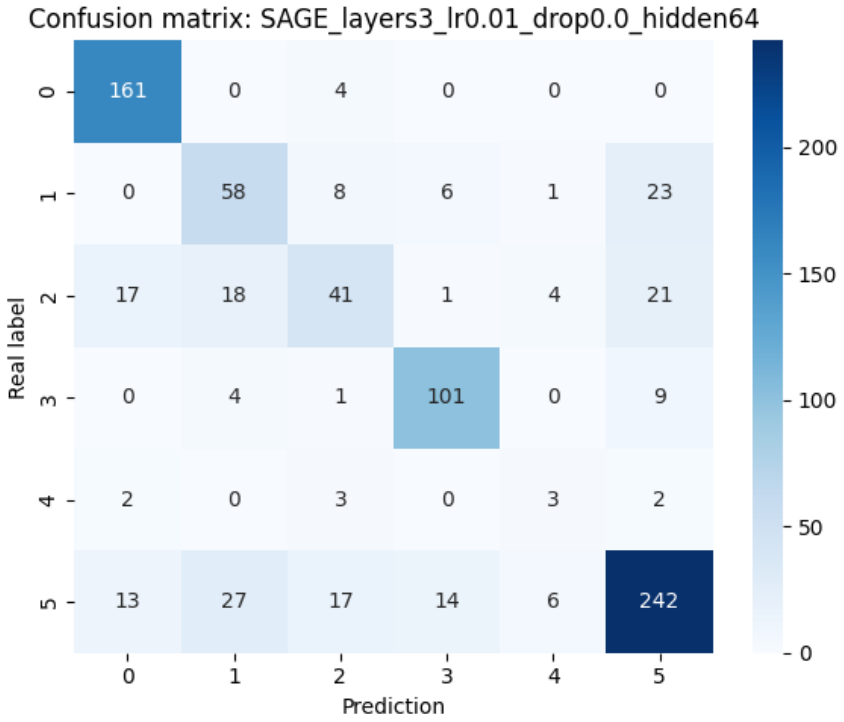

Figure 11: Confusion Matrix for the three-layer GraphSAGE model (lr = 0.01, dropout = 0.0, hidden = 64) using mixed features

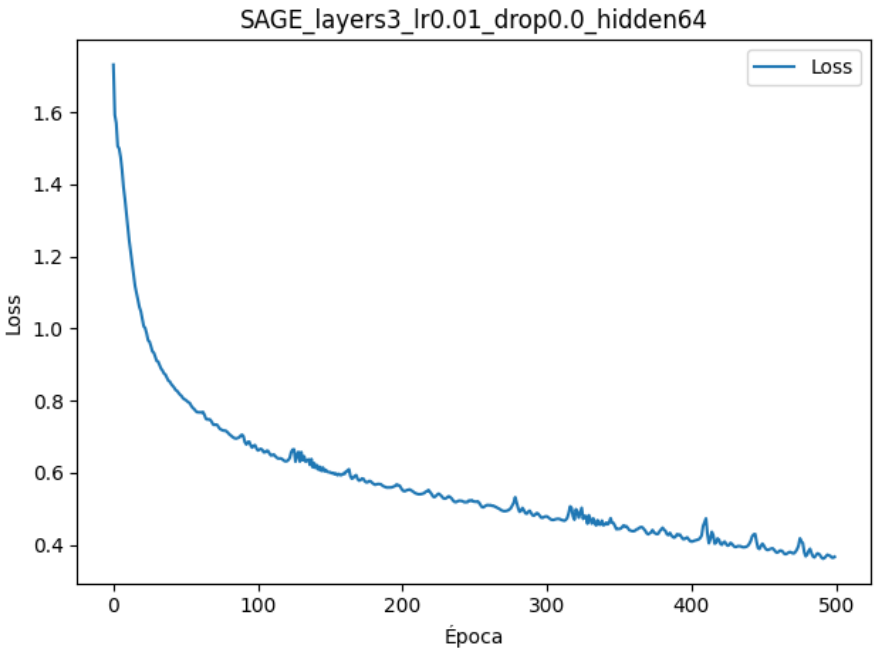

Figure 12: Training loss curve over 500 epochs for the three-layer GraphSAGE model (lr = 0.01, dropout = 0.0, hidden = 64) using mixed features

Figure 12 shows the training loss curve for the observed epochs. It follows a pattern similar to the previous model, a fast decline in the first 50 epochs and then a more gradual decrease, stabilizing around 0.35–0.45. Small oscillations after epoch 200 reflect minor parameter adjustments but no major instabilities.

5 GNN model explainability analysis

Following the selection of optimal models, a thorough explanatory and **comparative analysis** of various explainers was conducted. This evaluation was complemented by the implementation of specific metrics designed to objectively quantify and justify the obtained results. This section details the explainability methods applied, allowing not only understanding of the internal behavior of the models but also **validation of their performance** from an interpretative perspective.

5.1 Explainers used for model interpretability

Three different explanation methods were employed to analyze and interpret the optimized models: *GNNExplainer*, *PGExplainer*, and *GraphMask*. Each explainer was configured with specific parameters to ensure effective interpretation of node classification results.

5.1.1 GNNExplainer

GNNExplainer was implemented as a local *post-hoc* explanation method to **indetify important subgraph structures**.

The explainer was trained for 500 epochs to optimize the edge mask weights, focusing on edge importance rather than node features. The ***phenomenon*** explanation type was selected to identify the subgraph most relevant to the model's prediction. The model configuration parameters specify that the explanation task involves **multiclass classification** at the **node level**, with the model's outputs interpreted as log probabilities, ensuring appropriate handling of the classification predictions during the explanation process.

5.1.2 PGExplainer

PGExplainer, a parametric and inductive explanation model, was configured to learn explanations across multiple instances. Unlike other explainers, it requires a **specific training phase** before it can generate explanations, which allows it to learn patterns across multiple instances rather than explaining each prediction in isolation. Because of it and for optimizing it, a **specialized sampling strategy** was employed.

This implementation was configured with 100 training epochs and 400 sample nodes. The epoch count was determined through empirical testing, which revealed that loss curves typically plateaued after approximately 100 epochs, with minimal or no improvement in explanation quality beyond this point. The computational cost of training was another significant consideration, as the method is considerably more resource-intensive than other explainers.

A key optimization in the sampling strategy was maintaining a **60% ratio** of nodes from the same class as the target node. This balanced approach ensured that the explainer received sufficient examples of both positive class nodes and for other nodes of the network, leading to more discriminative explanations.

5.1.3 GraphMask

GraphMask works by learning differential gates for each message between

nodes, which allows to determine which connections in the network have the greatest impact on node predictions (Schlichtkrull et al., 2020).

GraphMask was specifically configured with 3 layers and trained for 50 epochs. The decision to limit training to 50 epoch was made after considering the **significant computational demands** of this explainer, which requires substantial resources for each training iteration. Empirical testing indicated that the explanation quality showed **minimal improvement beyond this point**, making additional epochs inefficient from a computational perspective.

The layer parameter was set to 3 to match the architecture of the *GraphSAGE* model being explained, which also contained 3 message-passing layers.

5.2 Metrics used for model interpretability

5.2.1 Fidelity metrics

Fidelity metrics evaluate the contribution of the explanatory subgraph to the prediction (Yuan et al., 2020). Two types of fidelity were measured:

- *Fidelity*+: Measures the impact of removing the explanatory subgraph from the entire graph.
- *Fidelity*-: Measures model performance when only the explanatory subgraph is provided to the model.

5.2.2 Characterization score

The characterization score combines positive and negative fidelity values using equal weights. For this evaluation, a balanced **50/50 weighting** was maintained between the two fidelity components. This metric provides a unified assessment of explanation quality (Amara et al., 2022).

5.2.3 Unfaithfulness

Unfaithfulness measures the disparity between a GNN model's prediction behavior on the original graph versus its behavior on the explanation subgraph. This metric quantifies how accurately an explanation represents the model's decision-making processes. Lower unfaithfulness scores indicate explanations that more faithfully capture the model's internal reasoning (Agarwal et al., 2022).

5.3 Comparative analysis of explainers

In the following sections, the performance of the three explainers is assessed using the four key metrics across each defined cluster. Initially, the aggregated metric results are presented to offer a comparative perspective on their capacity to recover the structural labeling logic for each cluster. Then, representative examples from each cluster are examined to show how each method allocates importance to connections, contrasting the resulting explanations with the predefined labeling rules.

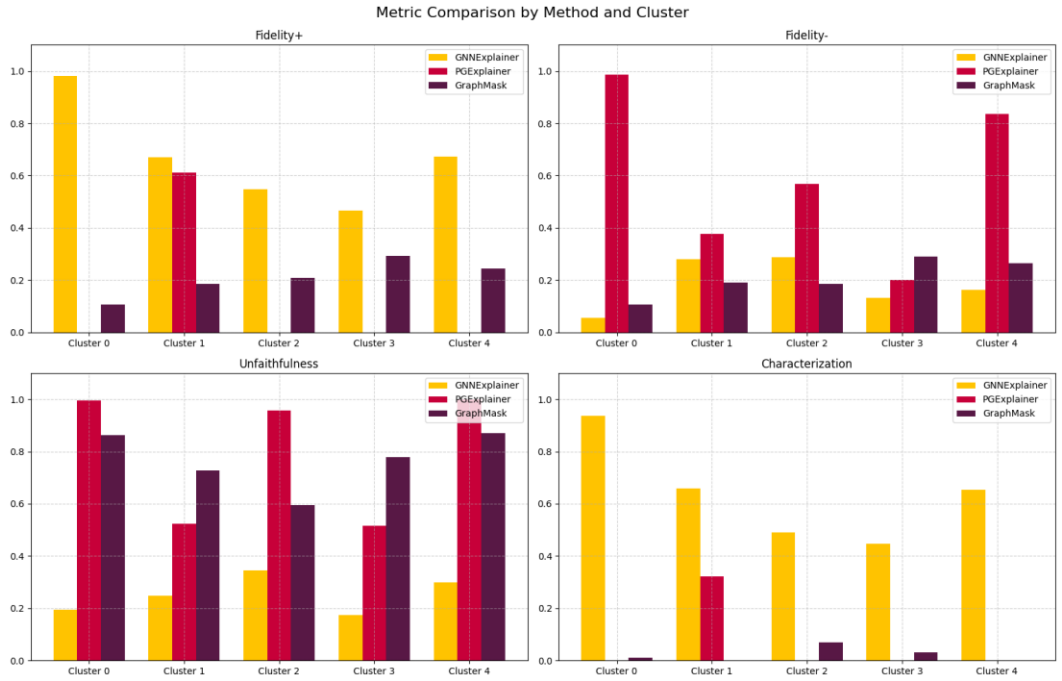

Figure 13: Metric comparison by meyhod and cluster for the augmented only model

Overall, **GNNExplainer** shows the best performance. It presents consistently high *Fidelity+* and *Characterization* values in all clusters, indicating that its explanations correctly highlight the **most relevant connections** for prediction and that it manages to capture well the patterns defined for each group. For example, in cluster 0 shows a *Fidelity+* of 0.9816 and a *Characterization* of 0.9361, while in cluster 4 it maintains high values, 0.6735 and 0.6531, respectively. Moreover, their *Unfaithfulness* values are low in all clusters, between 0.17 and 0.34, suggesting **stable explanations** in the face of small perturbations in the network structure.

On the other hand, **PGExplainer** obtains good *Fidelity-* scores, especially in clusters 0 and 4, indicating that it adequately **penalizes irrelevant connections**. However, its *Fidelity+* scores are zero in all clusters except cluster 1, and its *Characterization* values are very low or zero, revealing an inability to correctly identify the patterns that define each cluster. In addition, it shows the highest values of *Unfaithfulness* in all cases, reflecting a high sensitivity to the removal of elements from the network and, therefore, a **lower explanatory robustness**.

GraphMask, on the other hand, shows deficient performance on all metrics. Its *Fidelity+* and *Fidelity-* scores remain at low and similar values, suggesting that it fails to correctly differentiate relevant connections from irrelevant ones. In *Characterization*, values are consistently close to zero, indicating an **inability to identify meaningful structural rules or patterns**. In addition, it presents high values of *Unfaithfulness* in all clusters, which supports its limited explanatory capacity.

5.4 Cluster-based analysis

Following the aggregated metric results, the focus shifts to an examination of individual nodes. For each selected cluster, two representative nodes were examined, **one correctly classified and one misclassified**. This approach allows for the

assessment of how each explainer captured the underlying decision patterns, especially in relation to the rule-based labeling strategy. The analysis is organized in a consistent format. For each node, the edge masks produced by three explainability methods are visualized and compared. Each explanation is then interpreted in the context of the labeling criteria. The goal is to determine whether the highlighted **subgraphs align with its rule**.

Following the case-based analysis, the metrics presented above are used to support the qualitative observations.

5.4.1 Cluster 0 analysis

As previously established, a node is classified into Cluster 0 when either the mean betweenness centrality of its neighbors exceeds the **high betweenness threshold** or the range of degree values among its neighbors surpasses the **high degree range** threshold.

This section presents the subgraphs highlighted by three explainability methods when analyzing node 420, which was **correctly classified** as belonging to Cluster 0.

Examination of node 420's statistics reveals, a 1030 value in the neighbor degree range feature. This high value satisfies the high degree range criterion, explaining why node 420 is correctly classified in Cluster 0. The large degree range is primarily due to the presence of node 107 among its neighbors, creating significant disparity between highest and lowest degree neighbors.

Table 6: Comparison of Explainers for Node 420 (Cluster 0, Correctly Classified)

Explainer	Highlighted nodes	Observations
GNNExplainer	107, 348, 376, 353	Selective, aligns well with the rule. Highlights node 107 (most influential)
PGExplainer	107, 348, 376, 353	Matches GNNExplainer in structure, but with lower magnitude values.
GraphMask	All 34	Uniform attribution (0.6667); fails to capture relevant relations.

- ***Analysis of explainers' performance***

GNNExplainer demonstrates high alignment with the classification rule by highlighting only 4 out of 34 neighbors, with particular emphasis on those relevant to the classification criterion:

- **Node 107**: This neighbor is critical to the classification as it has:
  - Extremely high degree centrality (1045)
  - Elevated betweenness centrality (0.4805)
  - Highest pagerank value (0.0069)

As the highest-degree neighbor, node 107 is the primary contributor to node 420's high neighbor degree range value.

- **Node 348** (Importance: 0.8153): Has significant topological importance with:
    - High degree centrality (229)
    - Considerable betweenness centrality (0.0380)
    - Acts as an intermediary in the network
- **Node 376** (Importance: 0.7578): Displays:
    - High degree centrality (133)
    - Intermediate clustering coefficient (0.2632)
    - Moderate betweenness centrality (0.0062)
- **Node 353** (Importance: 0.7254): Shows:
    - High degree centrality (102)
    - Clustering coefficient of 0.3172

The *GNNExplainer* correctly identified the neighbor with the highest degree and other nodes with significant degree values, demonstrating sensitivity to the network properties that directly influence the classification criterion.

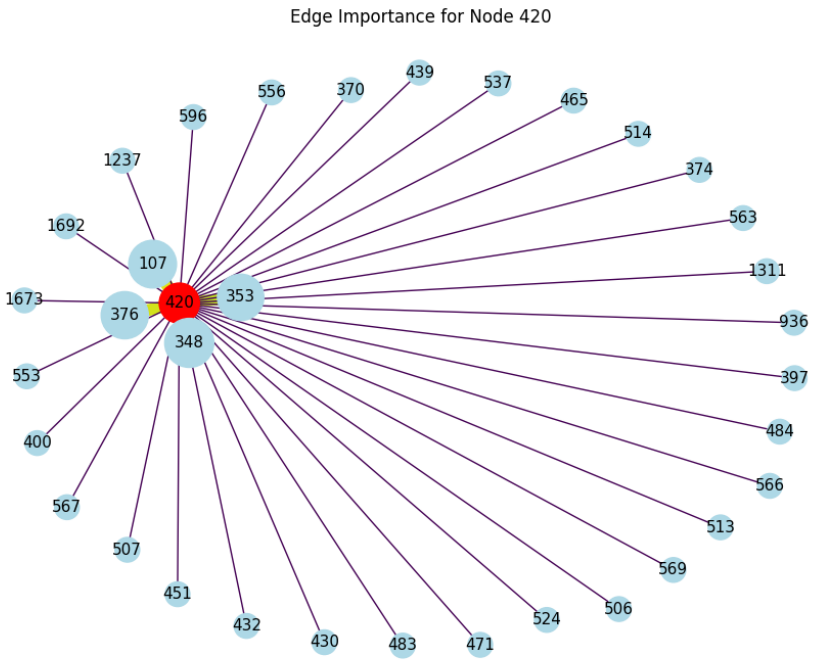

Figure 14: Explanation of Node 420 Using GNNExplainer

1. **PGExplainer:** This explainer identified **the same 4 connections** as GNNExplainer, but with substantially smaller importance magnitudes:
    - Edge (420, 348): 3.61e-18
    - Edge (420, 107): 1.74e-22

- Edge (420, 376): 9.76e-20
- Edge (420, 353): 8.03e-21

PGExplainer shows strong concordance with *GNNExplainer*, identifying the identical set of 4 key neighbors. Despite assigning numerically different importance values, with much smaller magnitudes, *PGExplainer* captures the same underlying structure relevant to the classification decision, as shown in Figure 25

GraphMask: In contrast to the selective behavior of the previous methods, GraphMask assigned uniform importance (0.6667) to all 34 connections of node 420, showing no discrimination between edges, clearly visible in Figure 26. This undiscriminating approach fails to highlight node 107's special relevance to the neighbor degree range criterion. The uniform attribution provides limited interpretable information regarding the model's decision-making process and shows poor alignment with the known classification rule.

- ***Misclassified node***

Node 911 was classified as Cluster 0 by the model, while its **true class is Cluster 2**. This node has only one neighbor, node 107, and has a high mean degree, 1045.0 and more importantly a high betweenness 0.480518, but, due to its only neighbor, a zero neighbor degree range.

The classification rule for Cluster 0 requires either high metric variance with high betweenness or high neighbor degree range. Cluster 2 simply requires a high mean degree of neighbors.

GNNExplainer reveals that the misclassification is from the node's unusual structure. With only one neighbor, node 911 does not have the neighborhood diversity needed for proper evaluation of variance or degree range, automatically failing to meet Cluster 0 criteria. However, it strongly satisfies the Cluster 2 criterion through its neighbor's extremely high degree.

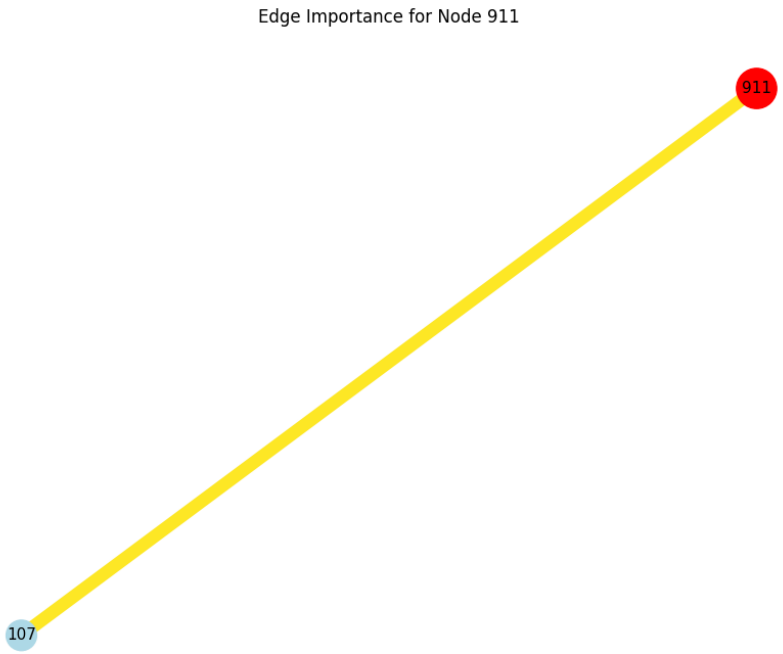

Figure 15: Explanation of Node 911 using GNNExplainer

Despite correctly identifying the importance of the single connection, the explainer shows that the model overemphasizes betweenness while failing to properly account for the structural implications of having a single neighbor, leading to the classification error.

5.4.2 Cluster 1 analysis

As previously stated, a node is classified in Cluster 1, when the variance of its neighbors' metrics is less than or equal to the high variance threshold and the average clustering coefficient of its neighbors is greater than or equal to **the high clustering threshold**.

For this cluster, **node 143**, which was correctly classified as belonging to Cluster 1, is analyzed.

The analysis of the neighborhood statistics for node 143 shows a **high average clustering coefficient** (0.7411), which satisfies one of the key classification criteria for Cluster 1: a high level of local cohesiveness among neighboring nodes. This structural property indicates that node 143 is embedded in a tightly connected subgraph, supporting its correct classification in Cluster 1 according to the rule-based labeling strategy.

Table 7: Comparison of Explainers for Node 143 (Cluster 1, Correctly Classified)

Explainer	Highlighted nodes	Observations
GNNExplainer	68,46,35,99,131,0	Selective, aligns well with the rule. Highlights nodes with high clustering

PGExplainer	68,46,35,99,131,0	Matches GNNExplainer in structure, but with lower magnitude values.
GraphMask	All 12 nodes	Mixed/binary attribution (0.6667 or 1.0); includes non-relevant nodes, reducing interpretability.

- ***Analysis of explainers' performance***

The ***GNNExplainer*** correctly identifies the neighbors with **high clustering coefficients**. Highlighting 6 out of 12 neighbors, with particular emphasis on those relevant to the classification criterion:

- Node 68 (Importance: 0.8703): High clustering coefficient (0.8333)
- Node 46 (Importance: 0.8210): Perfect clustering coefficient (1.0)
- Node 35 (Importance: 0.8178): Perfect clustering coefficient (1.0)
- Node 99 (Importance: 0.7646): Moderately high clustering coefficient (0.6410)
- Node 131 (Importance: 0.7638): Perfect clustering coefficient (1.0)
- Node 0 (Importance: 0.5768): Low clustering coefficient (0.0420); but GNNExplainer assigns it the lowest importance among the highlighted nodes

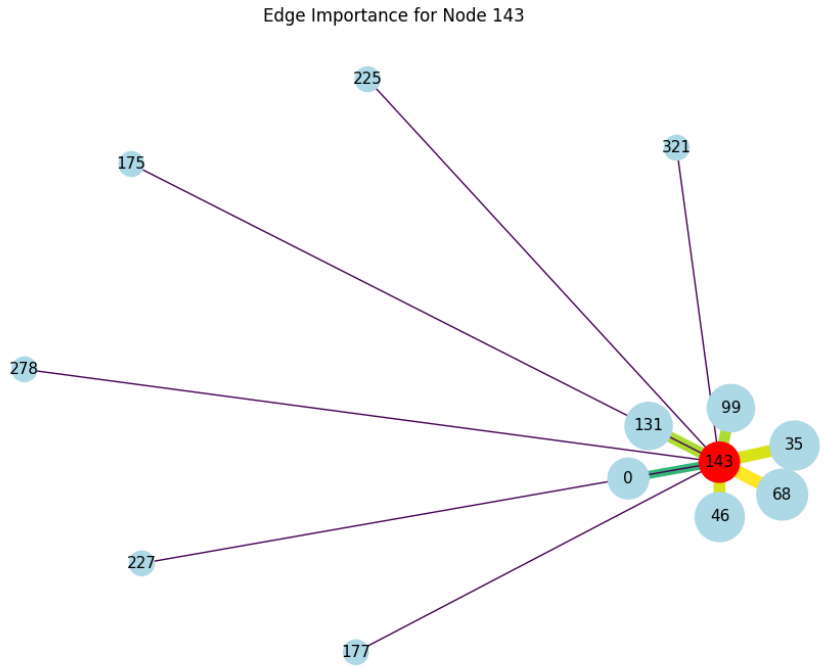

Figure 16: Explanation of Node 143 using GNNExplainer

PGExplainer shows structural agreement with *GNNExplainer*, identifying approximately the same set of key neighbors. Despite assigning numerically different values of importance, with much smaller magnitudes, it captures the same underlying structure relevant to the classification decision, illustrated in Figure 27.

- Edge (143, 46): 7.80e-15
- Edge (143, 35): 7.93e-15
- Edge (143, 99): 5.06e-15
- Edge (143, 68): 4.69e-15
- Edge (143, 131): 2.66e-16
- Edge (143, 0): 3.07e-16.

As with the remaining connections, all received values of 0.

Unlike the selective behavior of *GNNExplainer* and *PGExplainer*, **GraphMask** assigns importance to all neighbors with a binary pattern, as can be seen in Figure 28. This less discriminative approach includes nodes that are not relevant for the ranking decision according to the rule set. The uniform assignment provides limited interpretable information about the decision-making process of the model.

- ***Misclassified node***

Node 3557 was classified as Cluster 1 by the model, while its true class is Cluster 3. The *GNNExplainer* results show different importance values across the node's 31 neighbors, with only 8 neighbors receiving non-zero importance values:

- **Highest importance values:** (3557, 3527): 0.675; (3557, 3488): 0.629; (3557, 3514): 0.618; (3557, 3518): 0.585
- **Medium importance values**: (3557, 3437): 0.267; (3557, 3456): 0.264; (3557, 3528): 0.238; (3557, 3545): 0.202

The classification rule for Cluster 1 requires low metric variance combined with high mean clustering coefficient, while Cluster 3 requires low mean degree and low mean closeness centrality of neighbors.

Examining the node's neighborhood statistics shows that it has an average degree of 59.35, an average clustering coefficient of 0.532, an average closeness of 0.243, and a neighbor degree range of 532. The two neighbors with the highest importance values, 3527 and 3518, share similar characteristics, a high clustering coefficient (above the 90th percentile), 0.708 and 0.706 respectively, and moderate degrees, 23 and 33.

GNNExplainer's attribution pattern reveals why the model misclassified this node. The explainer emphasizes connections to neighbors with high clustering coefficients, which aligns with the Cluster 1 criterion requiring high mean clustering. The model appears to have learned this pattern effectively, as evidenced by the high importance assigned to nodes 3527 and 3518.

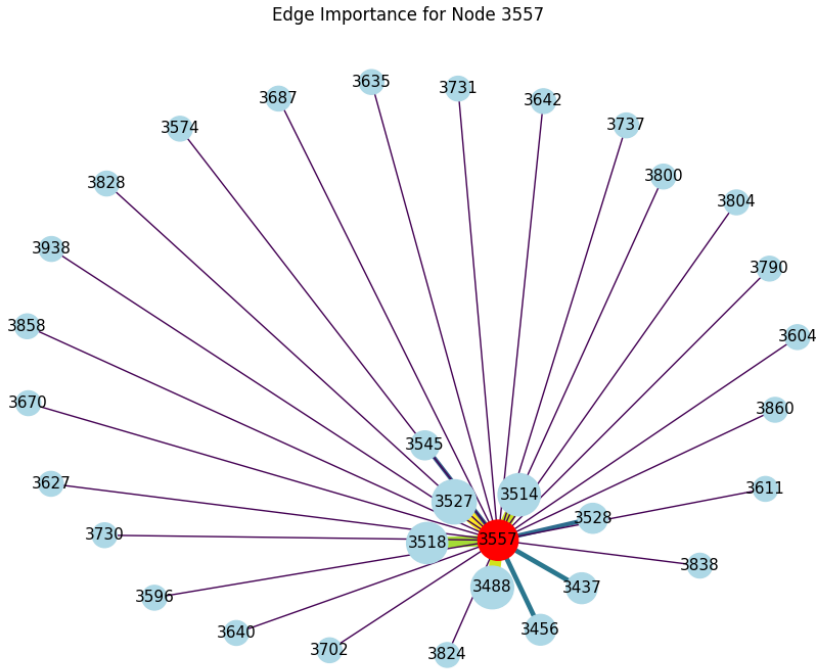

Figure 17: Explanation of Node 3557 using GNNExplainer

However, this focus forgets the more low-degree neighbors that would reduce the mean degree value and align with Cluster 3 criteria. The explainer assigns zero importance to most of the neighbors, many of which have low degree and closeness values that would satisfy Cluster 3 requirements.

5.4.3 Cluster 2 analysis

As previously described, a node is classified into Cluster 2 when the mean degree of its neighbors exceeds the **high degree threshold**. For the analysis of this criterion, node 27 will be analyzed, which was correctly classified as belonging to Cluster 2. This node statistics reveal a mean degree of 94.6, which exceeds the high degree threshold, satisfying this criterion.

Table 8: Comparison of Explainers for Node 27 (Cluster 2, Correctly Classified)

Explainer	Highlighted nodes	Observations
GNNExplainer	0	Assigns nearly all importance to the only high-degree neighbor, perfectly aligning with the mean-degree rule.
PGExplainer	0	Identifies the same single key neighbor with a much smaller magnitude, preserving structural agreement with GNNExplainer.
GraphMask	54, 119, 0, 324, 329	Bad discrimination, gives highest weight (1.0) to lower-degree nodes (54, 119) and only 0.6667 to node 0, contradicts the classification rule.

- ***Analysis of explainers' performance***

GNNExplainer demonstrates strong alignment with the classification rule by highlighting only one key neighbor:

Node 0 (Importance: 0.9765): Very high degree (347)

This node has a significantly higher degree than all other neighbors, making it the primary contributor to node 27's high mean degree of neighbors. By assigning almost 100% of the importance to this node, *GNNExplainer* correctly identifies **the most influential connection** for the classification criterion.

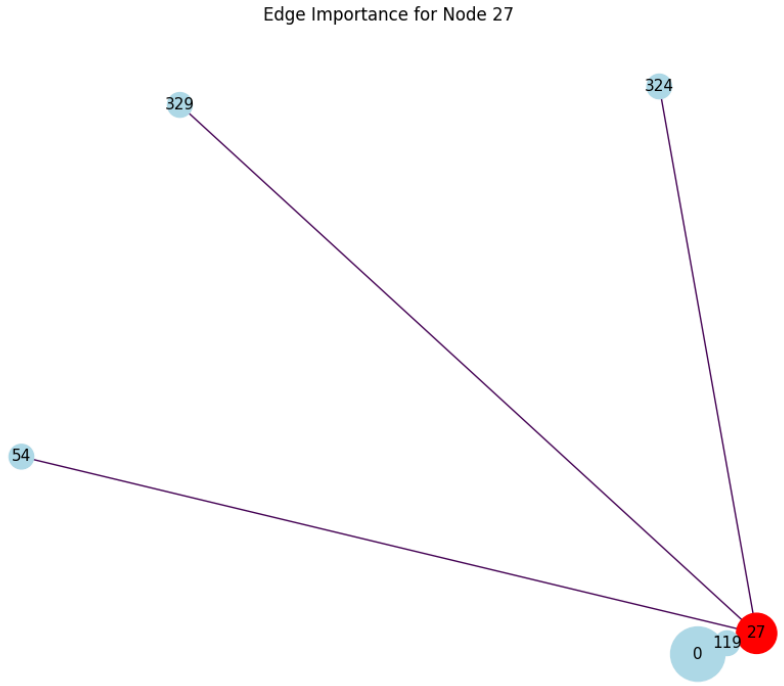

Figure 18: Explanation of Node 27 using GNNExplainer

As in the previous cases, **PGExplainer** shows perfect agreement with GNNExplainer in terms of structure, identifying node 0 as the only important neighbor, Figure 29. Despite assigning a numerically different importance value, it captures the same underlying structure relevant to the classification decision.

Once again, **GraphMask** shows a lack of discrimination in its attributions, giving importance to all connections rather than focusing on the most relevant ones, illustrated in Figure 30. Moreover, it assigns the highest importance to nodes with relatively modest degrees, node 54 with degree 8 and node 119 with degree 62, while giving less importance to the node 0, with only 0.6667 importance. This attribution pattern contradicts the mean degree classification rule, as it fails to emphasize the connection that most significantly impacts the node's classification criterion.

- ***Misclassified node***

While 1710 node's true class is Cluster 0, the model assigned it to Cluster 2. The *GNNexplainer* highlights several connections as particularly influential in the classification decision. The connections to nodes 925 (0.948), 526 (0.905), and 1688

(0.837) received the highest importance scores, with five neighbors receiving scores above 0.6 and five neighbors receiving scores of zero.

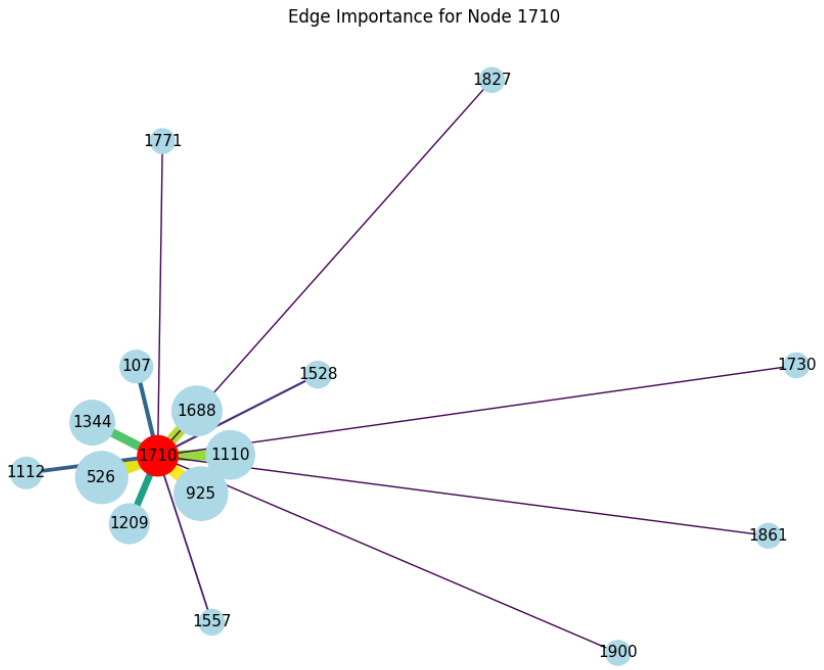

Figure 19: Explanation of Node 1710 using GNNExplainer

The node's neighborhood exhibits a mean degree of 165.93 and a high degree range of 1025.0. According to the defined rules, Cluster 2 membership requires only a high mean neighbor degree, while Cluster 0 requires either the combination of high variance and high betweenness or simply a high neighbor degree range.

The **focus** on the average degree ends up **overlooking** the large variation in node degrees within the neighborhood. The explainer helps uncover this bias in the model's internal representations. It shows that the model tends to **give more weight** or important to neighbors that support the average degree rule, while ignoring how degrees are spread out.

5.4.4 Cluster 3 analysis

The classification criterion for Cluster 3 requires that both the mean degree of a node's neighbors is less than or equal to the **low degree threshold** and the mean closeness centrality of its neighbors is less than or equal to the **low closeness threshold**.

To analyze this rule, **node 3664** was chosen based on its correct assignment to the cluster. The statistics for node 3664 confirm its correct classification with a mean degree of neighbors of 7.03 and a mean closeness centrality of neighbors of 0.2469.

Table 9: Comparison of Explainers for Node 3664 (Cluster 3, Correctly Classified)

Explainer	Highlighted nodes	Observations

GNNExplainer	3500, 3537, 3615, 3437	Selectively highlights neighbors with low degree and low closeness centrality, directly aligning with the criteria.
PGExplainer	3500, 3537, 3615, 3437	The same values as *GNNExplainer*, but with smaller importances.
GraphMask	All nodes	Assigns uniform importance (0.6667) to every connection, failing to highlight the low degree, low closeness nodes critical for classification.

- ***Analysis of explainers' performance***

The **GNNExplainer** results demonstrate strong alignment with the classification rule by identifying neighbors with properties that directly satisfy the classification criteria:

- Node 3537 (Importance: 0.6992) exhibits a very low degree (4) and low closeness centrality (0.2393).
- Node 3500 (Importance: 0.7004) has a low degree (20) and low closeness centrality (0.2395).
- Node 3615 (Importance: 0.6961) similarly presents a low degree (20) and low closeness centrality (0.2395).

The similar importance values, around 0.7, assigned to these three nodes indicates recognition of their similar contribution to satisfying the classification criteria. A lower importance value assigned to node 3437 (0.2811) indicates that while this node contributes to the classification, its properties may be less aligned with the classification criteria.

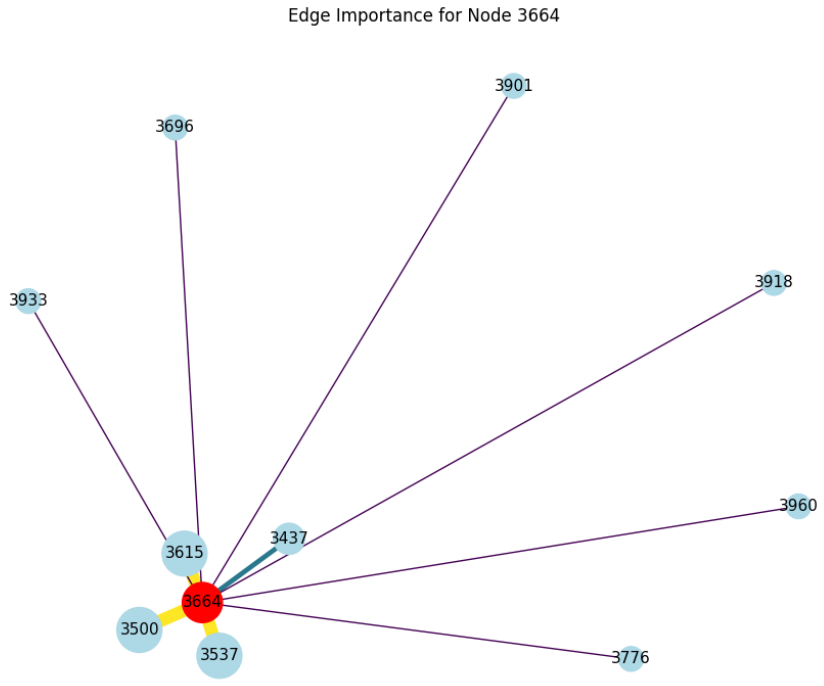

Figure 20: Explanation of Node 3664 using GNNExplainer

PGExplainer demonstrates structural consistency with *GNNExplainer* by identifying the identical set of 4 key neighbors, Figure 31. The extremely small magnitude of the importance values, however, shows challenges for interpretation. Despite this numerical difference, the structural agreement between GNNExplainer and PGExplainer remains revealing.

GraphMask exhibits consistent behavior with previous analyses by assigning uniform importance to all connections, as seen in Figure 32. This same attribution offers **limited insight** into which specific neighbors influence the classification decision. The approach fails to highlight the neighbors with the low degree and low closeness properties that are critical to the Cluster 3 classification criteria.

- ***Misclassified node***

The misclassification of node 2262 illustrates how the model can rely too heavily on one structural feature while neglecting another. Although the node belongs to Cluster 5, the model incorrectly assigns it to Cluster 3. *GNNExplainer* provides insight into this error by highlighting specific neighbor connections that influenced the decision.

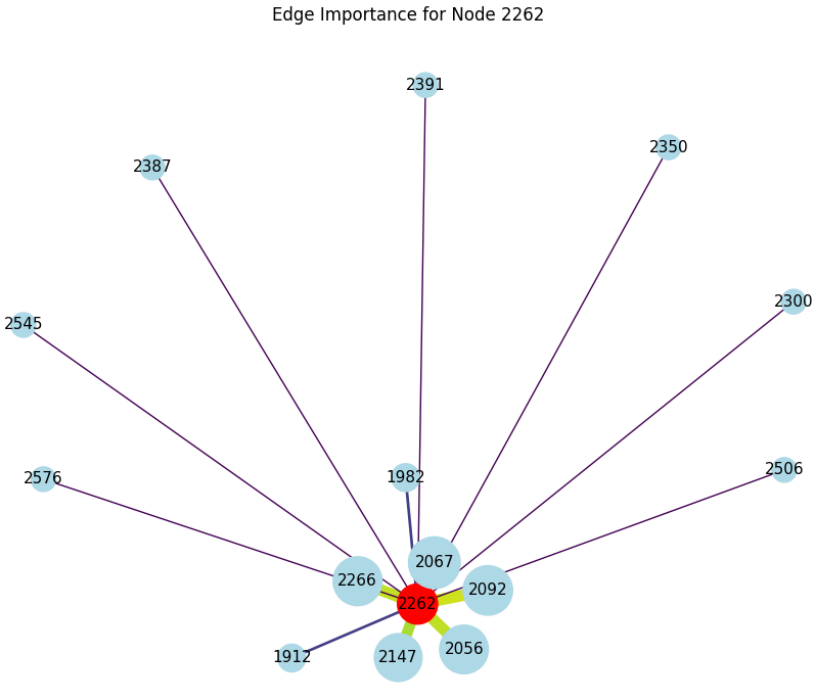

Figure 21: Explanation of Node 2262 using GNNExplainer

The explanation reveals that the model assigns high importance to neighbors like nodes 2067, 2092, 2266, 2056, and 2147, all of which received importance scores between 0.78 and 0.90. These neighbors share a common trait: **low closeness values**, aligning with one of the two conditions for Cluster 3.

However, the node's neighborhood contradicts the second condition. With a mean degree of 120.71, it far exceeds the low-degree requirement for Cluster 3. Despite this, the model overlooks degree information, as even highly connected neighbors like node 2266 (degree 234) and node 2056 (degree 115) receive high importance scores.

5.4.5 Cluster 4 analysis

The classification criterion for Cluster 4, as mentioned before, requires two main conditions, **high closeness centrality** of neighbors and **high mean degree** of neighbors. Node **160** meets these conditions, with a neighboring mean degree of **177.5** and a mean closeness centrality of **0.3073**, both exceeding the respective thresholds and thereby justifying its assignment to Cluster 4. The three explainability methods subsequently reveal distinct attribution patterns when interpreting this classification.

Table 10: Comparison of Explainers for Node 160 (Cluster 4, Correctly Classified)

Explainer	Highlighted nodes	Observations
GNNExplainer	0	Between the two nodes, the one that meets the cluster criteria, high-degree and high-closeness neighbor, receives greater importance.
PGExplainer	0	Identifies the same single key neighbor with a much smaller importance value
GraphMask	0, 260	Assigns maximum importance to both neighbors, failing to discriminate between the high-impact and low-impact nodes.

- ***Analysis of explainers' performance***

The **GNNExplainer** results demonstrate alignment with the classification rule by identifying the connection to node 0 as significant. This neighbor exhibits properties that directly contribute to satisfying the classification criteria, a very high degree (347) and a high closeness centrality (0.353343), whereas node 260 fails to meet either criterion.

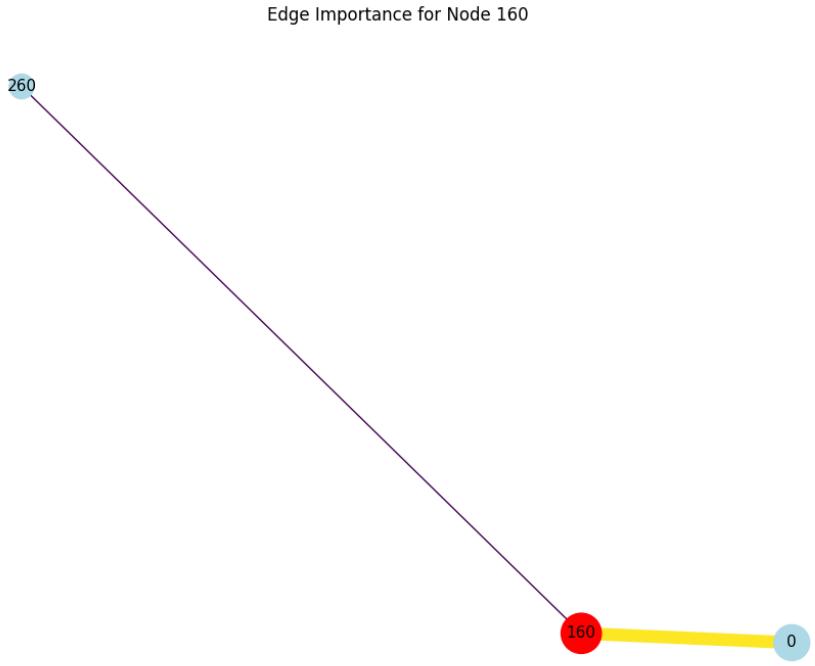

Figure 22: Explanation of Node 160 using GNNExplainer

PGExplainer mirrors *GNNExplainer's* focus on the connection to node 0 but assigns it to an extremely low importance value, Figure 33. as observed in previous clusters, limiting its interpretability in this context.

GraphMask, again, exhibits a uniform importance attribution by assigning maximum values to both connections, as illustrated in Figure 34. This approach fails to differentiate between the high-impact neighbor and the low-impact neighbor.

- ***Misclassified node***

The misclassification of **node 365**, predicted as Cluster 4 but truly in Cluster 2, is caused by the form of the model's overreliance on a couple of neighbors that partially fit Cluster 4's rules, while ignoring all the other neighborhood statistics.

GNNExplainer highlights nodes 348 and 360 with high importance scores, 0.91 and 0.81, since both have high closeness scores, 0.37 and 0.30, and node 348 even has a large degree. These two connections give the impression that node 365 meets Cluster 4's closeness and degree requirements.

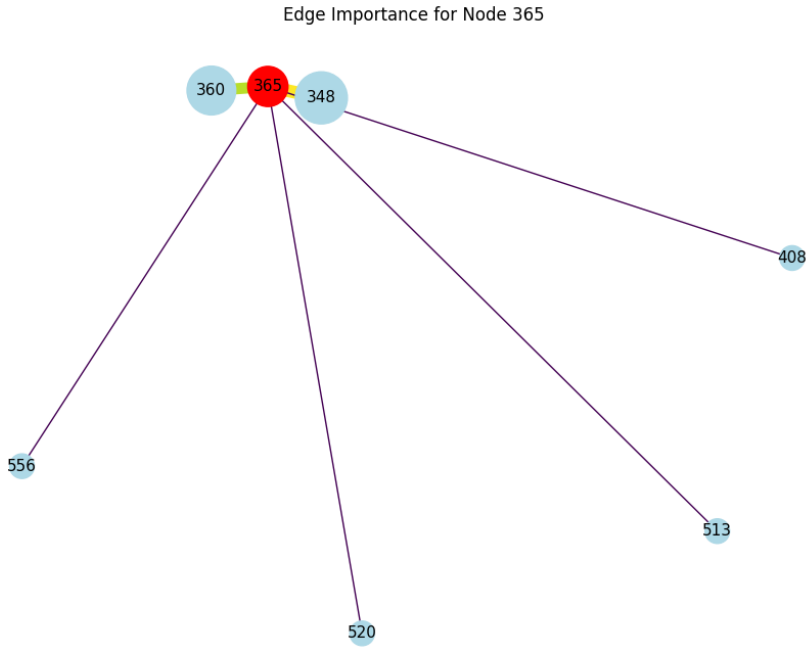

Figure 23: Explanation of Node 365 using GNNExplainer

However, node 365's overall neighborhood does not satisfy clusters rule for high mean degree, its mean degree is only 85.7, below the top degree threshold. In fact, it only fulfills Cluster 2's single condition of having a high mean degree.

6 Evaluation of GNNExplainer's robustness to artificial noise

Since the *GNNExplainer* has proven to be the most consistent and effective method in identifying explanations aligned with the structural rules of the clusters, its robustness to the **introduction of noise** in the data was evaluated. Specifically, whether the incorporation of non-informative dummy variables affected the assignment of importance made by the explainer was analyzed. This analysis makes it possible to **assess the robustness of the explanatory** model in the face of irrelevant information and to verify whether it continues to correctly highlight the characteristics that are decisive for the prediction.

To ensure a fair comparison, the analysis was conducted on the same set of nodes across all clusters. This approach allowed for a consistent evaluation of the explainer's behavior, isolating the effect of the introduced noise from variations in node selection.

- In Cluster 0, node 420 kept the same top connections: 107 (0.78), 348 (0.25), 353 (0.18) and 376 (0.24).
- In Cluster 1, node 143 again highlighted edges to 0 (0.71), 35 (0.80), 46 (0.83), 68 (0.83), 99 (0.80) and 131 (0.81).
- In Cluster 2, node 27 still selected the edge to 0 with a score of 0.39.
- In Cluster 3, node 3664’s strongest links were 3437 (0.61), 3500 (0.80), 3537 (0.85) and 3615 (0.81), the same.
- In Cluster 4, node 160 continued to favor edges to 0 (0.39)

In all clusters, *GNNExplainer* identified the same connections, although the importance of these relationships varied, whether only augmented features were used or artificial noise was added.

Additionally, a comparison between the metrics obtained in each cluster was made.

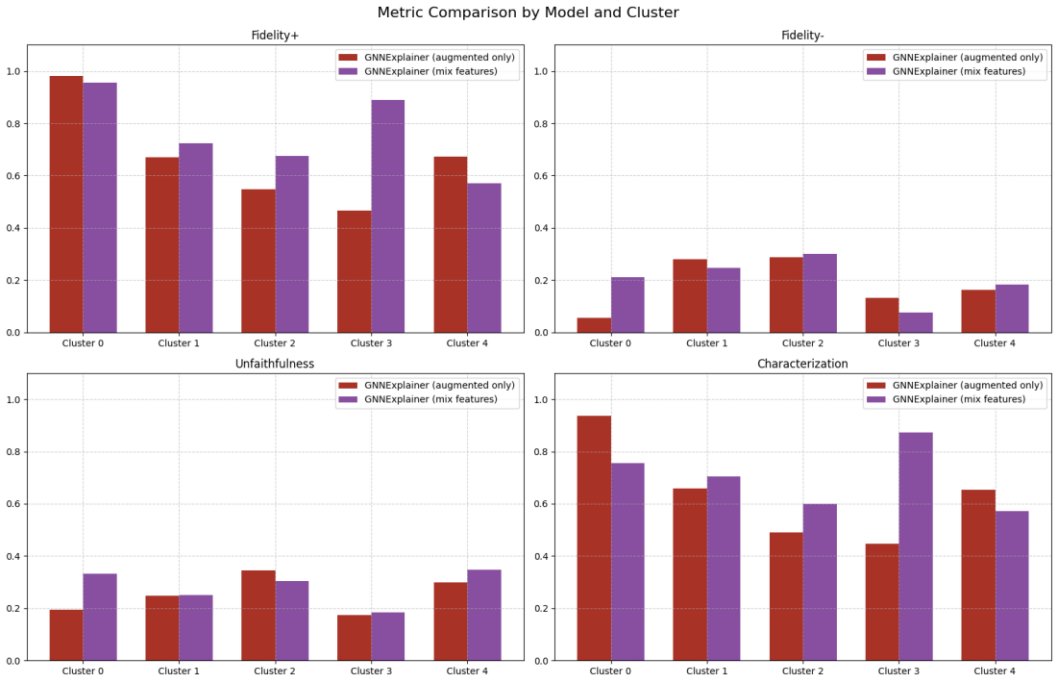

Figure 24: GNNExplainer metric comparison by model and cluster

The comparison shows that, despite minor fluctuations, overall **results remain consistent**. The *invented features* did not substantially affect the explainability of the predictions, key patterns remain intact, and the models continue to capture the underlying relationships used to assign labels.

7 Study limitations

Before interpreting these findings, several limitations of the study must be acknowledged.

First, the clustering strategy was based mostly on the mean values of each node's neighbors, which **may have hidden critical variations** when a node's neighborhood contained highly heterogeneous neighbors. Second, a substantial subset of approximately 1500 nodes were not classified by any cluster-defining rule and were therefore assigned to an unanalyzed "common-user neighbors" group, within which latent substructures or patterns may have existed but remained unexplored.

Finally, nodes whose statistics aligned with multiple clusters were forced into a single category based on distinctness criteria, which may have introduced assignment inconsistencies by overlooking equally valid alternative memberships.

Together, these factors may have limited the detail of the cluster insights and reduced the reliability of the predictive explanations.

8 Conclusions

This section presents the conclusions derived from the research objectives and findings of the study. The analysis of the results allows for an evaluation of the extent to which each objective was achieved and the overall implications of the work.

In terms of feature and label design, both invented (noise) features and the augmented (structural) features were successfully constructed, **capturing the structural information** of both the graph and individual nodes. The labeling procedure generated coherent clusters that accurately reflect each node's neighborhood structure. These **ground-truth labels** provided a reliable basis for evaluating model behavior and explanations. While this approach proved effective, it is important to acknowledge certain limitations as detailed in the limitations section of this paper.

The combination of these carefully designed features and labels provided a solid foundation for the graph analysis, enabling accurate representation of the complex relationships within the network structure.

Concerning the second objective of developing GNN architecture for node prediction, the results demonstrate significant success across different feature configurations. The SAGE model using only augmented features, achieved optimal performance with a three-layer architecture and a *learning rate* of 0.02, resulting in a *balanced accuracy* of **68%**, an *accuracy* of approximately **77%**, and a *macro F1-score* of **67%** on the test set. Comparatively, the mixed-feature SAGE model with a three-layer architecture and a *learning rate* of 0.01 achieved a *balanced accuracy* of **65.3%**, an *accuracy* of **75.1%**, and a macro F1-score of **64%.**

These results suggest that the developed GNN models effectively capture the general relationships within the graph rather than focusing exclusively on individual node features. The models demonstrate a **strong capability** to recognize broader graph characteristics and effectively integrate the influence of neighboring nodes in the prediction process. This confirms the GNN's ability to use the relational structure of the data, where the state of one node is influenced by and influences its neighbors within the network.

With respect to model explainability, it was determined that **GNNExplainer** was **the most effective method** that recovered ground-truth relationships. High average *Fidelity+,* low *Unfaithfulness* across all clusters and strong *Characterization* scores demonstrated that its edge importance attributions coincide with the true neighborhood based rules. By contrast, **PGExplainer** achieves high *Fidelity-*, indicating that penalizes irrelevant connections, but exhibits very high *Unfaithfulness* and *Characterization* scores near zero in most clusters. **GraphMask**, on the other hand, performs poorly across all metrics, demonstrating a general failure to distinguish meaningful relationships from noise and that differ significantly from the ground truth.

These results confirm the initial expectation that a robust explainer should highlight the specific edges that support the rule-based labeling strategy across the different clusters, and they validate *GNNExplainer's* ability to do so across clusters and noise features. At the same time, the analysis of the misclassified nodes reveals that the explainer tends to assign importance to nodes that conform to the criteria of

the predicted cluster, while overlooking connections that would be truly important for correct classification. This observation highlights a limitation in how the model interprets complex network structures. Additional findings indicated that the models **underperform** when nodes have **limited neighbors** and tend to prioritize certain structural patterns over others.

Furthermore, a comparative analysis between the Augmented Features and Mixed Features models revealed similar explainability metrics, suggesting that the GNNs consistently relied on the underlying structural patterns despite the presence of additional noisy features. This supports the robustness of the models in prioritizing relevant information for classification.

In summary, the experimental findings validate the premise that, when labels are defined by neighbor metrics, a faithful explainer will prioritize those very neighbor relations. **GNNExplainer meets this criterion** more consistently than the other alternatives, while *PGExplainer* and *GraphMask* require additional enhancements to align their attributions with true structural logic. This work thus not only confirms the hypothesis but also highlights clear directions for future improvements in graph explainability methods.

9 Future lines of research

Building on the findings of this research, several promising directions for future work can be identified. Firstly, the current clustering approach could be enhanced through the incorporation of additional investigation and analysis. This **analysis should focus on Cluster 5**, the "common-user neighbors" group, as this group may contain subtle substructures or patterns that were not fully captured in the present study. A more intensive examination of this cluster could reveal valuable insights about network organization and node relationships.

Secondly, although *GNNExplainer* was effective in this study, future work should compare it with other post-hoc methods such as **SubgraphX**, which may deliver more precise or detailed insights into model behavior. Such a comparative analysis would enhance understanding of the GNN's decision-making processes and support the development of more transparent, interpretable models.

10 Ethical considerations

The ethical implications of data usage and synthetic data generation were considered. By using augmented structural features for node labeling while introducing realistic but synthetic personal attributes, the study maintained the integrity of the research while avoiding privacy concerns.

Although the synthetic features were designed to mimic real world characteristics, they remain artificial. Therefore, these results should be viewed as a technical evaluation of GNN and the explainers' capabilities, with the understanding that performance may differ when applied to networks that exhibit real attribute relationships.

Any future application of similar methodologies to real personal data would require additional ethical review, particularly regarding potential biases considerations. These distinctions help frame the appropriate context for interpreting the research contribution.

References

Agarwal, C., Queen, O., Lakkaraju, H., & Zitnik, M. (2022, August 19). *Evaluating explainability for graph neural networks*. arXiv.org. https://arxiv.org/abs/2208.09339

Amara, K., Ying, R., Zhang, Z., Han, Z., Shan, Y., Brandes, U., Schemm, S., & Zhang, C. (2022, June 20). *GRAphFramEX: Towards Systematic Evaluation of Explainability Methods for Graph Neural Networks*. arXiv.org. https://arxiv.org/abs/2206.09677

BarabáSi, A., & Albert, R. (1999). *Emergence of scaling in random networks*. *Science*, *286*(5439), 509–512. https://doi.org/10.1126/science.286.5439.509

Biggs, N., Lloyd, E. K., & Wilson, R. J. (1986). *Graph Theory*, 1736-1936. Oxford University Press.

Bollobás, B. (1981). THE DIAMETER OF RANDOM GRAPHS. 267(1).

Bollobás, B. (1998). *Modern graph theory*. Springer-Verlag.

Bonacich, P. (2007). *Some unique properties of eigenvector centrality. Social Networks*, 29(4), 555–564. https://doi.org/10.1016/J.SOCNET.2007.04.002

Bondy, J. A., & Murty, U. S. R. (1976). *Graph Theory with Applications*. North-Holland.

Bugueño, M., Biswas, R., & De Melo, G. (2024, July 23). *Graph-Based Explainable AI: A comprehensive survey*. https://hal.science/hal-04660442/

Christoph, M. (2020). *Interpretable machine learning: A guide for making black box models explainable.*

Chung, F., & Lu, L. (2006). *Complex Graphs and Networks. En Regional conference series in mathematics*. https://doi.org/10.1090/cbms/107

Chung, F. (2014). A brief survey of PageRank algorithms. *IEEE Transactions on Network Science and Engineering*, *1*(1), 38–42. https://doi.org/10.1109/tnse.2014.2380315

Diestel, R. (2000). *Graph theory* (2nd ed.). Springer.

Diestel, R. (2006). *Graph Theory*. Springer Verlag.

Easley, D., & Kleinberg, J. (2010). *Networks, crowds, and markets*. https://doi.org/10.1017/cbo9780511761942

Escobar, B. M., Redrovan, D., Villeda, E., & Santana, A. H. (2021). ¿Somos conscientes del efecto de las redes sociales en nuestra nutrición? *Innovare Revista De Ciencia Y Tecnología*, *10*(3), 178–180. https://doi.org/10.5377/innovare.v10i3.12990

Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., & Yin, D. (2019, February 19). Graph neural networks for social recommendation. arXiv.org. https://arxiv.org/abs/1902.07243

Fang, J., Liu, W., Zhang, A., Wang, X., He, X., Wang, K., & Chua, T. (2023, February). *On Regularization for Explaining Graph Neural Networks: An Information Theory Perspective*. OpenReview. https://openreview.net/forum?id=5rX7M4wa2R_

Fortunato, S. (2010). Community detection in graphs. Physics Reports, 486(3–5), 75–174. https://doi.org/10.1016/J.PHYSREP.2009.11.002

Gross, J. L., Yellen, J., & Anderson, M. (2018). Graph Theory and its applications. In *Chapman and Hall/CRC eBooks*. https://doi.org/10.1201/9780429425134

Hamilton, W. L., Ying, R., & Leskovec, J. (2017, June 7). *Inductive representation learning on large graphs*. arXiv.org. https://arxiv.org/abs/1706.02216

Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., & Chang, Y. (2020, January 17). GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks. arXiv.org. https://arxiv.org/abs/2001.06216

Kakkad, J., Jannu, J., Sharma, K., Aggarwal, C., & Medya, S. (2023, June 2). A survey on Explainability of graph Neural Networks. arXiv.org. https://arxiv.org/abs/2306.01958

Katz, L. (1953). A New Status Index Derived from Sociometric Analysis. Psychometrika, 18(1), 39–43. https://doi.org/10.1007/BF02289026

Kipf, T. N., & Welling, M. (2016, September 9). *Semi-Supervised Classification with Graph Convolutional Networks*. arXiv.org. https://arxiv.org/abs/1609.02907

Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. International Journal of Advance Research in Computer Science and Management Studies, 1(6). www.ijarcsms.com

Krapivsky, P. L., Rodgers, G. J., & Redner, S. (2001). Degree distributions of growing networks. *Physical Review Letters*, *86*(23), 5401–5404. https://doi.org/10.1103/physrevlett.86.5401

Kwak, H., Lee, C., Park, H., & Moon, S. (2010, April). What is Twitter, a social network or a news media?. In *Proceedings of the 19th international conference on World wide web* (pp. 591-600). https://doi.org/10.1145/1772690.1772751

Landherr, A., Friedl, B., & Heidemann, J. (2010). A critical review of centrality measures in social networks. *Business & Information Systems Engineering*, *2*(6), 371–385. https://doi.org/10.1007/s12599-010-0127-3

Leskovec, J., & Mcauley, J. (2012). Learning to Discover Social Circles in Ego Networks. Advances in Neural Information Processing Systems, 25. http://snap.stanford.edu/data/

Li, J., Pang, M., Dong, Y., Jia, J., & Wang, B. (2024, June 5). *Graph Neural Network Explanations are Fragile*. arXiv.org. https://arxiv.org/abs/2406.03193

Likas, A., Vlassis, N., & J. Verbeek, J. (2003). The global k-means clustering algorithm. Pattern Recognition, 36(2), 451–461. https://doi.org/10.1016/S0031-3203(02)00060-2

Lu, L. L. L., Shin, Y. S. Y., Su, Y. S. Y., & Karniadakis, G. E. K. G. E. (2020). Dying ReLU and Initialization: Theory and Numerical examples. *Communications in Computational Physics*, *28*(5), 1671–1706. https://doi.org/10.4208/cicp.oa-2020-0165

Lucca, A. M. T. (2000). Teoría de Grafos (Segunda Parte). *Revista De Educación Matemática*, *15*(2). https://doi.org/10.33044/revem.10922

Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., & Zhang, X. (2020, November 9). *Parameterized Explainer for Graph Neural network*. arXiv.org.

https://arxiv.org/abs/2011.04573

Mao, G., & Zhang, N. (2013). Analysis of Average Shortest-Path Length of Scale-Free Network. *Journal of Applied Mathematics*, *2013*, 1–5. https://doi.org/10.1155/2013/865643

Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., & Bhattacharjee, B. (2007). Measurement and analysis of online social networks. *In Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, 29–42. https://doi.org/10.1145/1298306.1298311

Montresor, A., Francesco, D. P., & Miorandi, D. (2011, March 28). *Distributed K-Core Decomposition*. arXiv.org. https://arxiv.org/abs/1103.5320

Newman, M. E. J. (2000, January 10). *Models of the Small World: a review*. arXiv.org. https://arxiv.org/abs/cond-mat/0001118

Newman, M. E. J. (2002). Assortative mixing in networks. *Physical Review Letters*, *89*(20). https://doi.org/10.1103/physrevlett.89.208701

Newman, M. E. J. (2003a). A measure of betweenness centrality based on random walks. Social Networks, 27(1), 39–54. https://doi.org/10.1016/j.socnet.2004.11.009

Newman, M. E. J. (2003b). The Structure and Function of Complex Networks. SIAM Review, 45(2), 167–256. http://www.jstor.org.sabidi.urv.cat/stable/25054401

Newman, M. E. J., & Park, J. (2003). Why social networks are different from other types of networks. Physical Review. E, Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics, 68(3). https://doi.org/10.1103/physreve.68.036122

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, *69*(6). https://doi.org/10.1103/physreve.69.066133

Newman, M. (2010). *Networks: An introduction* (1st ed.). Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199206650.001.0001

Patel, H., & Sahni, S. (2022, November 3). *Exploring explainability methods for graph neural networks*. arXiv.org. https://arxiv.org/abs/2211.01770

Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., & Prokhorenkova, L. (2023, February 22). *A critical look at the evaluation of GNNs under heterophily: Are we really making progress?* arXiv.org. https://arxiv.org/abs/2302.11640

Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E., & Hoffmann, H. (2019). Explainability methods for graph convolutional neural networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2019.01103

Rodrigues, F. A. (2019, January 22). *Network centrality: an introduction*. arXiv.org. https://arxiv.org/abs/1901.07901

Royat, A., Moghadas, S. M., Lesley, D. C., & Munteanu, A. (2024, September 17). *GINTRIP: Interpretable Temporal Graph Regression using Information bottleneck and Prototype-based method*. arXiv.org. https://arxiv.org/abs/2409.10996

Saramäki, J., Kivelä, M., Onnela, J., Kaski, K., & Kertész, J. (2007). Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, *75*(2). https://doi.org/10.1103/physreve.75.027105

Schlichtkrull, M. S., Nicola, D. C., & Titov, I. (2020, October 1). *Interpreting graph neural networks for NLP with differentiable edge masking*. arXiv.org. https://arxiv.org/abs/2010.00577

Shu, K., Cui, L., Wang, S., Lee, D., & Liu, H. (2019, July). defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 395-405). https://doi.org/10.1145/3292500.3330935

Tabassum, S., Pereira, F. S. F., Fernandes, S., & Gama, J. (2018). Social network analysis: An overview. Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery, 8(5). https://doi.org/10.1002/widm.1256

Wang, H., Liu, T., Sheng, Z., & Li, H. (2024). Explanatory subgraph attacks against Graph Neural Networks. *Neural Networks*, *172*, 106097. https://doi.org/10.1016/j.neunet.2024.106097

Wang, X., He, X., Cao, Y., Liu, M., & Chua, T. (2019). KGAT. *In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 950–958. https://doi.org/10.1145/3292500.3330989

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph convolutional neural networks for Web-Scale recommender systems. *In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 974–983. https://doi.org/10.1145/3219819.3219890

Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019, March 10). *GNNExplainer: Generating Explanations for Graph Neural Networks*. arXiv.org. https://arxiv.org/abs/1903.03894

Yuan, H., Tang, J., Hu, X., & Ji, S. (2020a). XGNN: Towards Model-Level Explanations of Graph Neural Networks. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 430 – 438. https://doi.org/10.1145/3394486.3403085

Yuan, H., Yu, H., Gui, S., & Ji, S. (2020b, December 31). Explainability in Graph Neural Networks: A Taxonomic survey. arXiv.org. https://arxiv.org/abs/2012.15445

Yuan, H., Yu, H., Wang, J., Li, K., & Ji, S. (2021, February 9). *On explainability of graph neural networks via subgraph explorations*. arXiv.org. https://arxiv.org/abs/2102.05152

Zhang, H., Wu, B., Yuan, X., Pan, S., Tong, H., & Pei, J. (2024). Trustworthy Graph Neural Networks: aspects, methods, and trends. Proceedings of the IEEE, 112(2), 97–139. https://doi.org/10.1109/jproc.2024.3369017

Zhang, J., & Luo, Y. (2017). Degree Centrality, Betweenness Centrality, and Closeness Centrality in Social Network. 300–303. https://doi.org/10.2991/MSAM-17.2017.68

Zhang, J., Chen, Z., Mei, H., Da, L., Luo, D., & Wei, H. (2023, July 15). *RegExplainer: Generating Explanations for graph neural networks in regression tasks*.

arXiv.org. https://arxiv.org/abs/2307.07840

Zhang, M., & Chen, Y. (2018, February 27). *Link prediction based on graph neural networks*. arXiv.org. https://arxiv.org/abs/1802.09691

Appendix A: Centrality measures

Table 11: Top 5 Betweenness centrality nodes

Node	Betweenness Centrality
107	0.480518
1684	0.337797
3437	0.236115
1912	0.229295
1085	0.149015

Table 12: Top 5 Degree centrality nodes

Node	Degree Centrality
107	0.258791
1684	0.196137
1912	0.186974
3437	0.135463
0	0.085934

Table 13: Top 5 eigenvector centrality nodes

Node	Eigenvector Centrality
1912	0.095407
2266	0.086983
2206	0.086052
2233	0.085173
2464	0.084279

Appendix B: PGExplainer and GraphMask explanation figures

In this appendix, the edge-importance explanations produced by PGExplainer and GraphMask are presented for each node cluster

1. Cluster 0

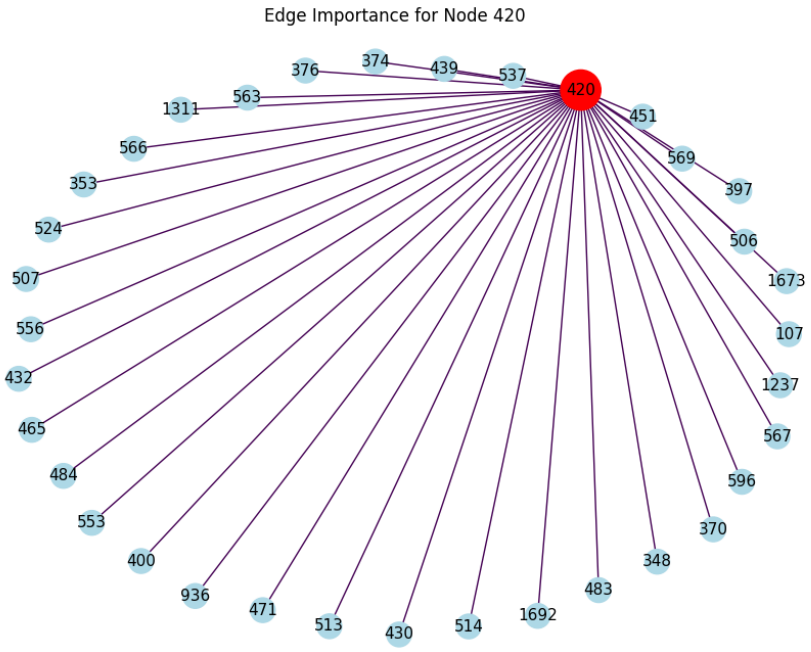

Figure 25: Explanation of Node 420 using PGExplainer

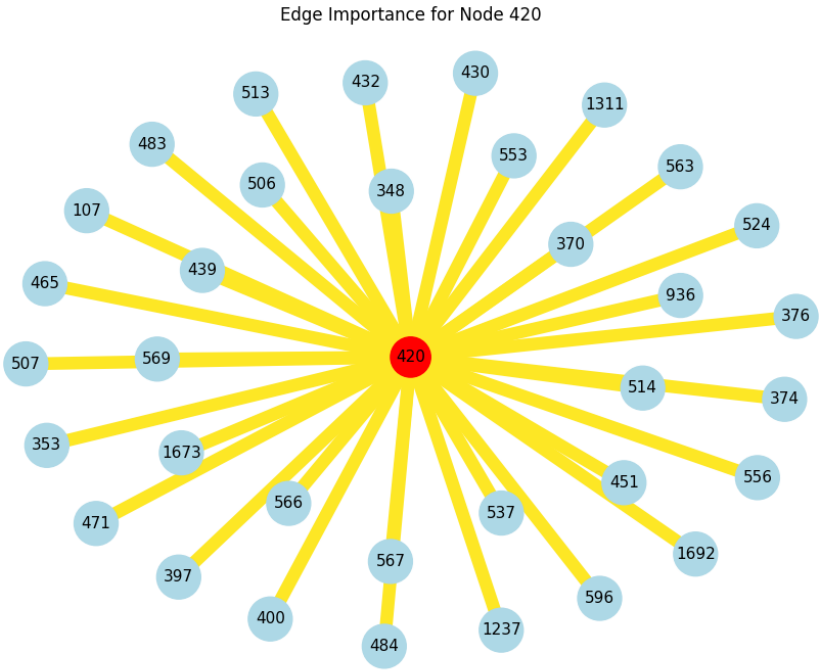

Figure 26: Explanation of Node 420 using GraphMask

2. Cluster 1

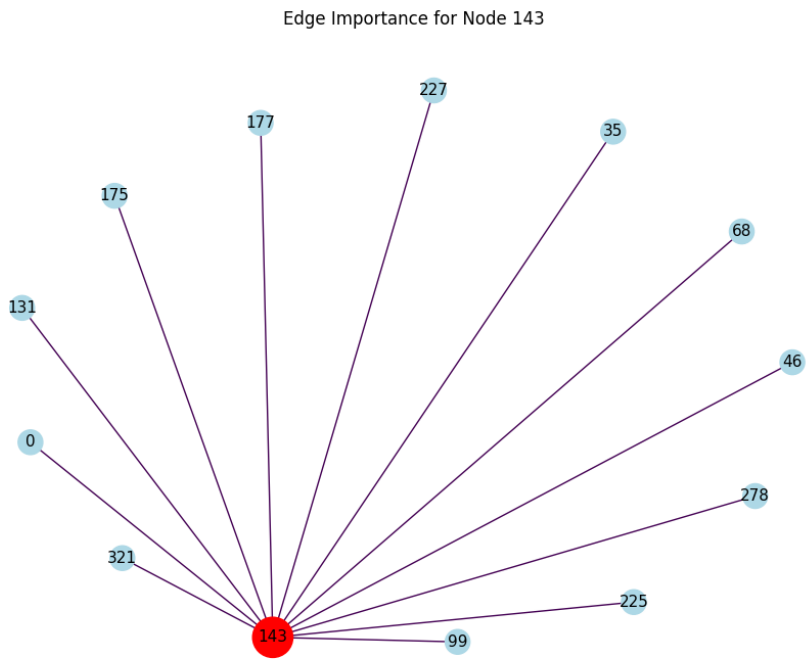

Figure 27: Explanation of Node 143 using PGExplainer

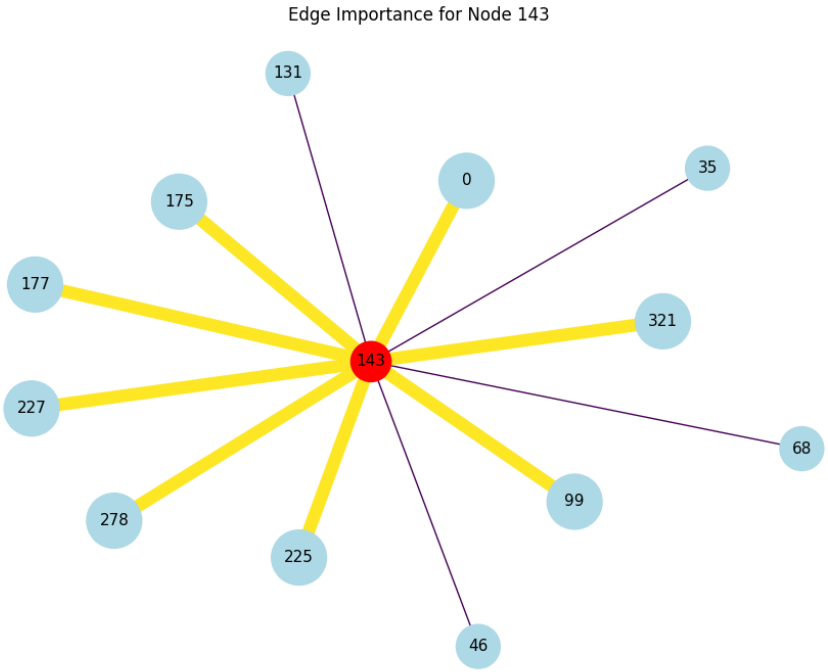

Figure 28: Explanation of Node 143 using GraphMask

3. Cluster 2

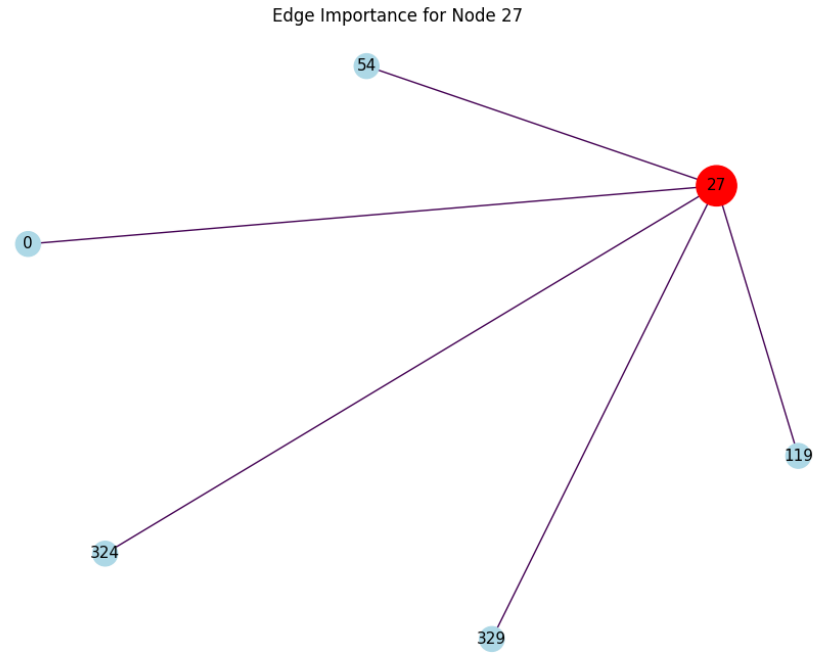

Figure 29: Explanation of Node 27 using PGExplainer

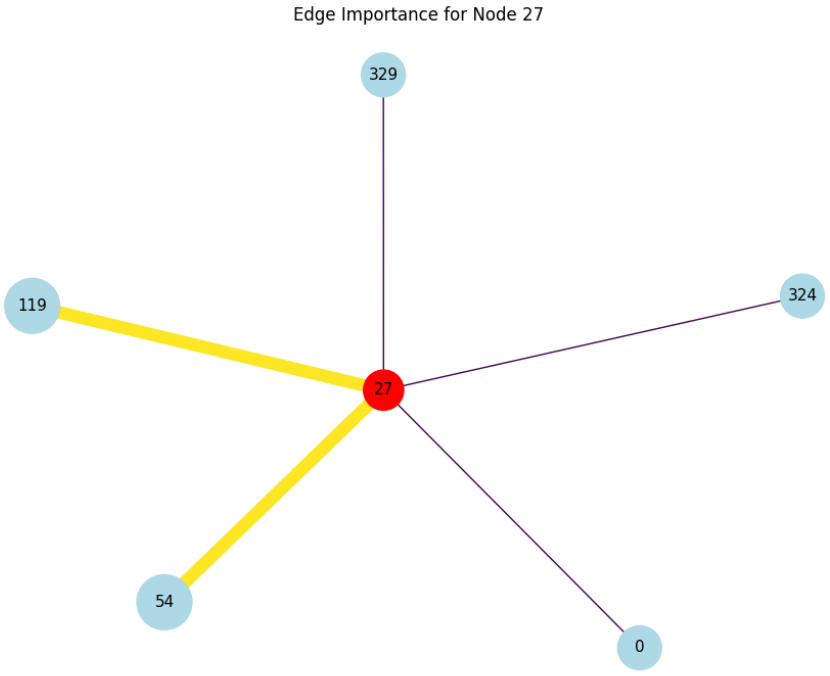

Figure 30: Explanation of Node 27 using GraphMask

4. Cluster 3

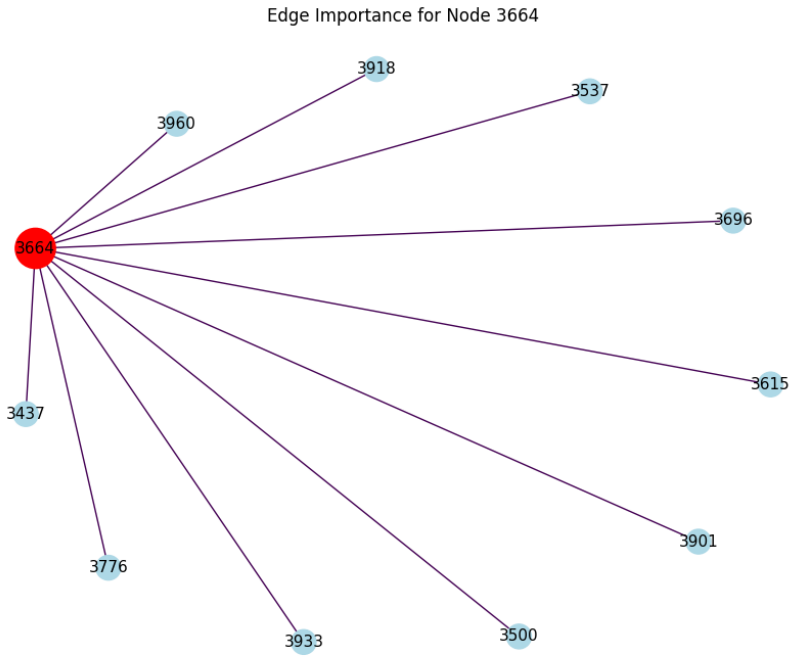

Figure 31: Explanation of Node 3664 using PGExplainer

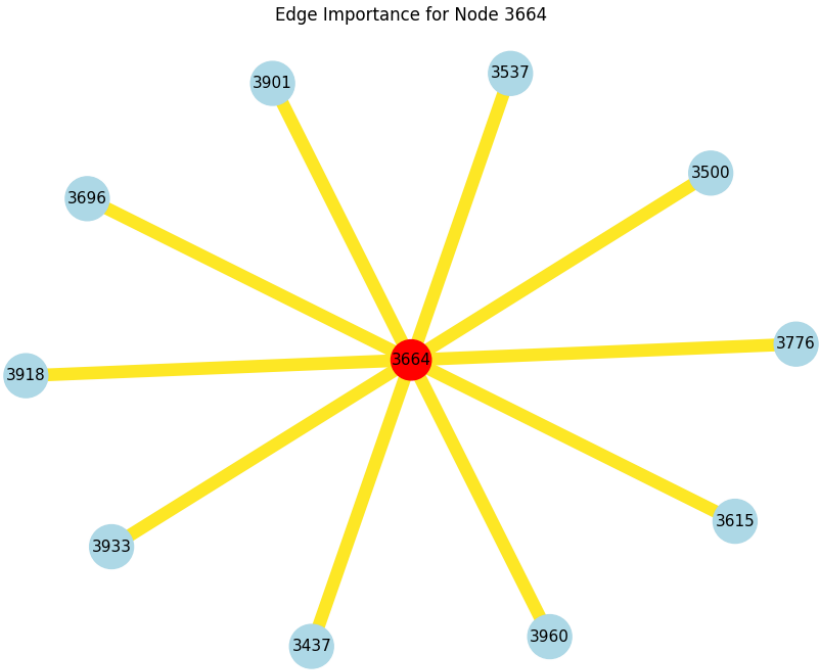

Figure 32: Explanation of Node 3664 using GraphMask

5. Cluster 4

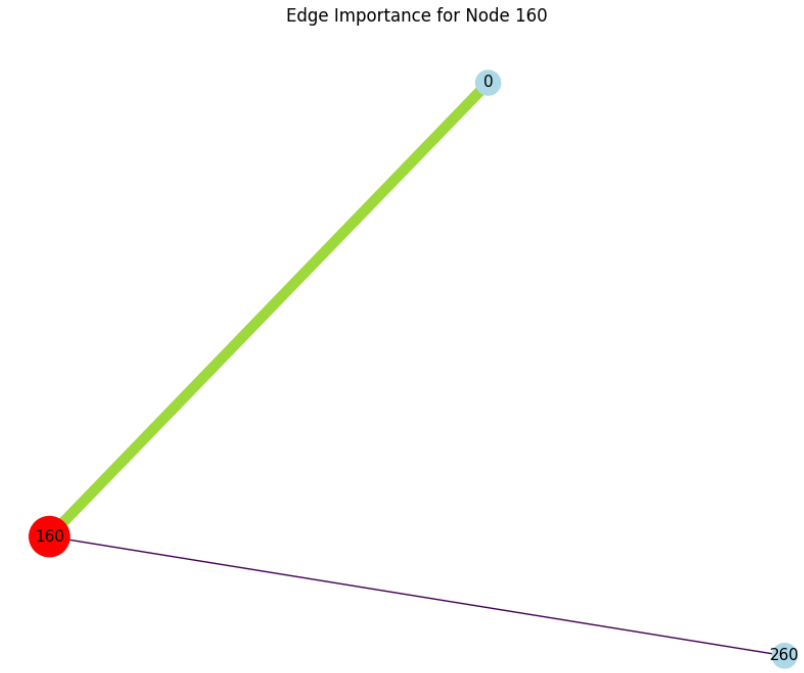

Figure 33: Explanation of Node 160 using PGExplainer

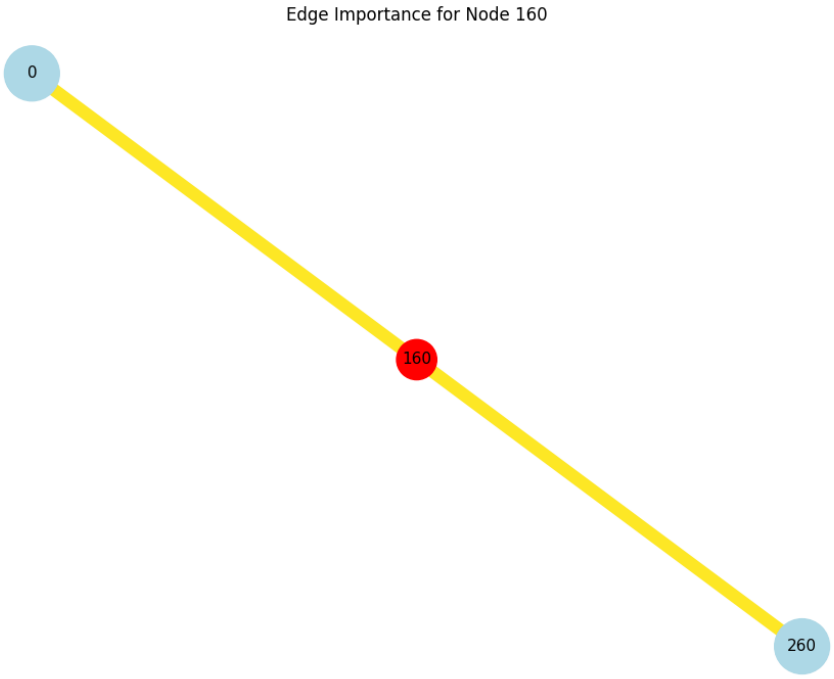

Figure 34: Explanation of Node 160 using GraphMask