A comparative analysis of Community Detection, Link Prediction, and Role Discovery in networks

Adriano Machado
Faculty of Science, University of Porto
up202105352@up.pt

Francisco da Ana
Faculty of Science, University of Porto
up202108762@up.pt

João Lima
Faculty of Science, University of Porto
up202108891@up.pt

ABSTRACT

This study presents a comparative analysis of classical algorithms and modern Graph Neural Network (GNN) architectures across three core network analysis tasks: community detection, link prediction, and role discovery. By evaluating these methods on a variety of network datasets, including citation and social networks, the research highlights the critical trade-offs between topology-based and feature-based approaches. For community detection, GNNs excel when rich node features are present, whereas traditional methods are effective for structurally-defined networks. In link prediction, a trade-off emerges between the superior classification accuracy of GNNs and the strong computing efficiency and ranking performance of feature-based Random Forest models. For role discovery, engineered graphlet features proved to be exceptionally powerful, often matching or outperforming GNN-based clustering. The findings underscore that the optimal choice of algorithm is highly dependent on network characteristics and specific task requirements, demonstrating that even with modern deep learning methods, feature engineering remains crucial.

1. INTRODUCTION

Networks provide a powerful framework for modeling complex systems where relationships between entities are as important as the entities themselves. From citation networks that reveal the evolution of scientific knowledge to social platforms connecting billions of users, network analysis has become indispensable for understanding the structural patterns that govern our interconnected world.

Three fundamental computational challenges lie at the heart of network analysis. **Community detection** seeks to identify densely connected groups of nodes that represent natural clusters or functional modules within the network. **Link prediction** addresses the problem of inferring missing connections or forecasting future relationships based on existing network topology and node attributes. **Role discovery** focuses on identifying nodes that occupy similar structural positions or perform analogous functions within the network, regardless of their community membership, a task that captures the concept of structural equivalence across different network regions.

The primary motivation for this work is to perform a systematic, hands-on comparison between classical, well-established algorithms and modern Graph Neural Network (GNN) architectures across these three tasks. By implementing and evaluating a wide range of methods on diverse datasets, our aim is to provide a clear perspective on their relative strengths, weaknesses, and the impact of different feature engineering strategies.

Our key contributions include:

- Implementation and evaluation of distinct algorithms
- Analysis on seven diverse network datasets, including citation networks (Cora, Citeseer, PubMed), social networks (Actor, Twitch), and synthetic benchmarks (KarateClub, CLUSTER)
- Examination of how different feature engineering strategies impact algorithm performance across various network types

This paper is structured as follows: Section 2 details the methodology, including the datasets and the specific algorithms implemented for each task. Section 3 presents and analyzes the experimental results. Finally, Section 4 concludes with a summary of our findings and finally we discuss potential avenues for future work.

2. METHODOLOGY

To ensure a fair comparison between models, we studied community detection, link prediction, and role discovery as independent tasks. Each task was evaluated on a curated set of datasets using task-specific metrics within a reproducible experimental pipeline.

2.1 Datasets

We evaluated our algorithms on eight network datasets from different domains: three citation networks (Cora, Citeseer, PubMed), three social networks (Twitch-EN, Twitch-DE, Actor), and two specialized graphs (CLUSTER, Karate Club).

2.1.1 Dataset Selection by Task

- **Community detection**: Karate Club, Cora, and PubMed. These networks have established ground-truth communities and node features.
- **Link prediction**: Cora, Citeseer, Twitch-EN, and Twitch-DE. We used both citation and social networks to test performance across different graph types.

- **Role discovery**: CLUSTER, Actor, and Cora. The synthetic CLUSTER dataset was designed for role discovery, while Actor and Cora contain nodes with distinct structural roles.

Cora appears in all three tasks, allowing cross-task performance comparison.

2.1.2 *Dataset Characteristics*

The citation networks [5] represent academic papers as nodes with citations as directed edges. Node features are extracted from paper abstracts and content. The social networks capture user interactions: Twitch-EN and Twitch-DE represent follower relationships [4], while Actor represents film collaboration networks [3].
We included two specialized datasets: the synthetic CLUSTER dataset with known structural roles [2], and the Karate Club network [7], a standard benchmark with documented ground-truth communities.
Table 1 shows dataset statistics. The networks range from 34 nodes (Karate Club) to 19,717 nodes (PubMed). The datasets vary in structural properties: citation networks are sparse with low average degrees, social networks are denser with higher clustering, and synthetic graphs have controlled characteristics.

Table 1: Statistics of the Datasets Used in the Experiments

Dataset	Nodes	Edges	Avg. Degree	Density
Karate Club	34	78	4.59	0.1390
Cora	2,708	5,278	3.90	0.0014
Citeseer	3,327	4,552	2.74	0.0008
PubMed	19,717	44,324	4.50	0.0002
Twitch-EN	7,126	77,774	21.83	0.002
Twitch-DE	9,498	315,774	66.49	0.003
Actor	7,600	26,659	7.02	0.0009
CLUSTER	12,000	36,000	6.00	0.0005

2.2 Community Detection Methods

For the community detection task, our methodology was structured to provide a robust comparison between traditional, topology-based algorithms and modern, feature-aware Graph Neural Network (GNN) models. The primary objective was to partition the nodes of a network into distinct groups, or communities, where connections within groups are denser than connections between them. We implemented and evaluated a range of algorithms within a reproducible experimental pipeline, as you can see in Figure 1.

2.2.1 *Traditional Algorithms*

These methods serve as foundational baselines, operating purely on the graph's topology to identify community structures. We implemented three widely-used algorithms from the `networkx` and `community` libraries.

- **Louvain**: A fast and highly popular greedy optimization method that aims to maximize the modularity of a network's partitions. It operates in two phases: first, it assigns each node to its own community and then iteratively moves nodes to neighboring communities to find the largest increase in modularity. Second, it creates a new, coarser-grained network where the communities from the first phase become the new nodes, and the process is repeated until modularity can no longer be improved.

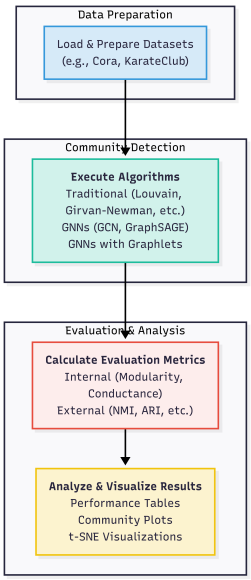


Figure 1: The experimental workflow for the community detection task.

- **Girvan-Newman**: This is a classic divisive algorithm that identifies community boundaries by progressively removing the edges with the highest "betweenness centrality", a measure of how often an edge lies on the shortest path between other pairs of nodes. By removing these "bridge-like" edges, the underlying community structure is revealed. Due to its computational intensity ($O(m^2n)$), its application was limited to our smaller datasets (KarateClub and Cora Subset) to ensure feasibility.

- **Label Propagation (LPA)**: An efficient and intuitive algorithm where nodes iteratively update their community label to match the one held by the majority of their neighbors. This process continues until no node can improve its situation by changing its label, resulting in a stable community partition. Its near-linear time complexity makes it highly scalable for large networks.

2.2.2 *Graph Neural Networks (GNNs)*

Representing the state-of-the-art, GNN-based methods learn low-dimensional node embeddings that capture both the graph's topology and the nodes' inherent features. These learned embeddings are then used to predict community assignments. Our framework includes two leading GNN architectures from `PyTorch Geometric` for this semi-supervised task:

- **Graph Convolutional Network (GCN)**: A model that learns node representations by aggregating feature information from a node's immediate neighbor-

hood. The core of the GCN is its layer-wise propagation rule, where node features are combined and transformed through a series of graph convolutional layers, allowing the model to learn complex structural patterns.

- **GraphSAGE (Graph Sample and AGgregatE)**: Unlike GCN, which uses the full neighborhood for aggregation, GraphSAGE employs a sampling strategy. For each node, it uniformly samples a fixed number of neighbors at each layer, making the aggregation process more efficient and scalable to larger graphs.

Furthermore, we explored the impact of feature engineering by running experiments where the GNNs were trained on **graphlet-enhanced features**. This involved augmenting or replacing the original node features with triangle counts, a basic but powerful structural feature that describes a node's local clustering behavior.

2.2.3 Evaluation Metrics

We then evaluated the performance of each algorithm using a suite of both internal and external validation metrics, implemented with the help of `scikit-learn` and `networkx`.

- **Internal Metrics**: These metrics evaluate the quality of a partition based solely on the graph's structure, without any ground-truth information.
  - **Modularity**: Measures the density of edges within communities compared to the density of edges between them. Scores closer to 1 indicate strong community structure.
  - **Average Conductance**: Measures the "cut" quality of the communities. A lower conductance score is better, as it indicates that communities are well-isolated with few inter-community connections.
- **External Metrics**: These metrics compare the detected communities against known ground-truth labels.
  - **Normalized Mutual Information (NMI)**: An information-theoretic measure of the similarity between the predicted and true community assignments. A score of 1 indicates a perfect match.
  - **Adjusted Rand Score (ARI)**: Measures the similarity between two partitions while correcting for chance. It ranges from -1 (disagreement) to 1 (perfect agreement).
  - **Fowlkes-Mallows Score (FMS)**: The geometric mean of precision and recall, measuring the similarity between the predicted and true partitions.

2.3 Link Prediction Methods

To predict links in a network, we implemented and compared a wide range of models, which can be grouped into three main categories: heuristic, traditional machine learning and Graph Neural Network (GNN) based methods. The entire experimental pipeline, from data handling to evaluation, is systematized to ensure reproducibility and fair comparison, as depicted in the workflow diagram (Fig. 2)

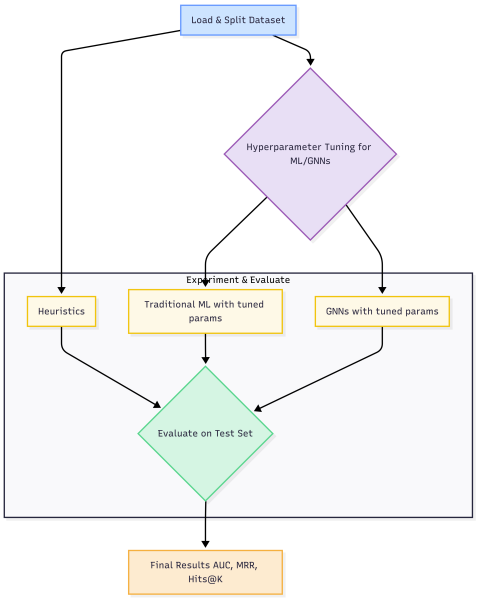


Figure 2: The experimental workflow for the link prediction task.

The process begins with the `DatasetManager` module, which loads datasets such as Cora, Citeseer, and Twitch (EN and DE). It then performs a fixed, reproducible train/validation/test split using a specified random seed, partitioning the edges of the graph. For our learning-based models, a dedicated hyperparameter tuning step is performed. Once optimized, the models are instantiated in the main experiment script, trained on the training data, and evaluated on the held-out test set.

2.3.1 Heuristic Methods

These models serve as fundamental baselines that are parameter-free and rely on graph topology to score potential links. We implemented four widely-used heuristics:

- **Common Neighbors:** The score is the number of shared neighbors between two nodes.
- **Jaccard Index:** Normalizes the number of common neighbors by the total number of neighbors of the two nodes.
- **Adamic-Adar:** A variant of common neighbors that gives more weight to shared neighbors with a lower degree.
- **Preferential Attachment:** Assumes that nodes with high degrees are more likely to form new connections, scoring links based on the product of the nodes' degrees.

2.3.2 Traditional Machine Learning

This approach transforms the link prediction task into a standard binary classification problem. Instead of working on the graph directly, models like **Decision Trees**, **Random Forest**, **Logistic Regression**, and **K-Nearest Neighbors** are trained on a feature space derived from the graph, extracting similarity scores. Our `FeatureExtractor`

module generates a feature vector for each candidate link using measures inspired by heuristic scores, such as Common Neighbors, Jaccard Index, Adamic-Adar, Preferential Attachment, Resource Allocation, and Community Sharing.

2.3.3 Graph Neural Networks (GNNs)

These models learn low-dimensional embeddings for each node from the graph structure and features, then use these embeddings to predict link likelihood. Our framework includes models based on Graph Convolutional Networks (GCN) and GraphSAGE encoders. For each encoder, we explored two different decoder architectures to compute the final link score: a simple **Dot Product Decoder** that calculates the dot product of the two node embeddings, and a **MLP Decoder** that concatenates the embeddings and passes them through a multi-layer perceptron.

2.3.4 Evaluation Metrics

To assess model performance, we use a set of standard metrics that capture different aspects of prediction quality. The metrics are:

- **AUC (Area Under the ROC Curve):** A measure of the model's ability to discriminate between positive and negative (existing vs. non-existing) links. It evaluates overall binary classification performance, with 1.0 being a perfect classifier and 0.5 a random classifier.
- **MRR (Mean Reciprocal Rank):** A ranking-based metric that evaluates how high up the list of predictions the true links are found. It is the average of the reciprocal of the ranks of all correct predictions.
- **Hits@K:** Another ranking metric that measures the proportion of true links that are found within the top-K highest-scored predictions.

While AUC measures general binary classification power, MRR and Hits@K are crucial for recommendation-oriented scenarios where the ranking of the most likely links is important. All these metrics indicate better performance for higher values.

2.4 Role Discovery Methods

For the role discovery task, the objective is to partition nodes into groups that are structurally equivalent. This is distinct from community detection, which groups nodes based on dense internal connections. Our methodology compares a traditional approach based on clustering engineered features, and a deep learning approach that clusters GNN-learned node embeddings. The entire experimental workflow is designed for reproducibility, from hyperparameter tuning to final analysis and reporting. The two primary pipelines for discovering roles are illustrated in Figure 3.

2.4.1 Feature-Based Clustering

First, we engineer a vector of structural properties for each node. Then, we apply the K-Means clustering algorithm to this feature space to group nodes into roles. We implemented two variants based on the features used:

- **FeatureBasedRoles**: Uses a vector of standard network centrality measures (degree, betweenness, closeness, eigenvector, PageRank) and the clustering coefficient.

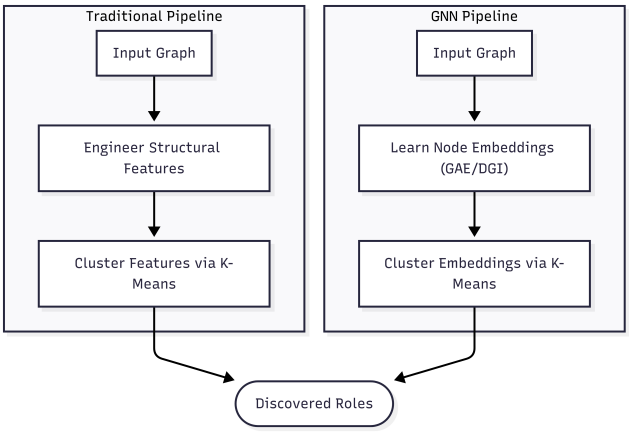


Figure 3: The two distinct pipelines implemented for role discovery: a traditional approach using engineered features and a GNN-based approach using learned embeddings

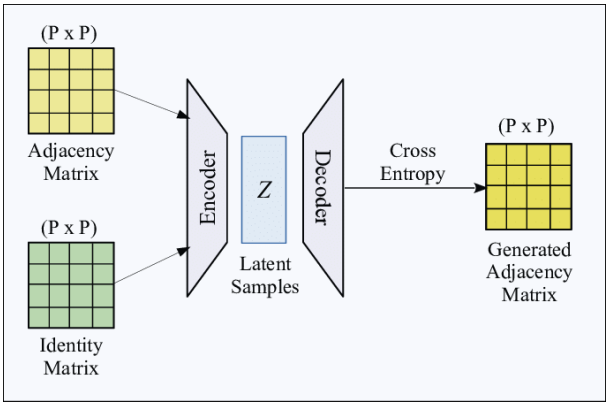
- **FeatureBasedRolesGraphlets**: Uses a feature set derived from graphlet degree vectors, which count the occurrences of small (2 to 4-node) induced subgraphs connected to each node.

2.4.2 GNN-Based Clustering

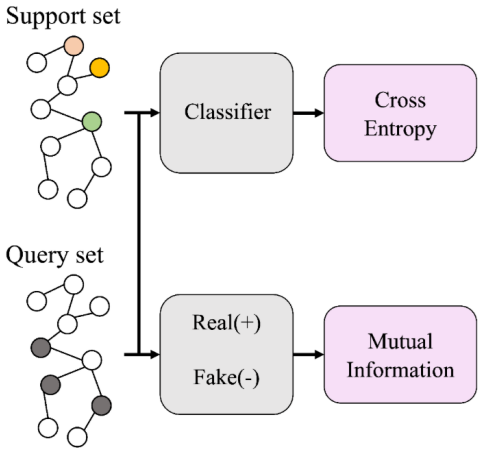
We used modern deep learning techniques to learn node representations automatically. An unsupervised Graph Neural Network (GNN) learns a low-dimensional embedding for each node directly from the graph structure. These embeddings, which inherently capture structural information, are then clustered using K-Means. We implemented and compared two leading unsupervised GNN architectures, illustrated in Figure 4.
To investigate the impact of input features on these GNNs, we tested both architectures with two types of initial node features: a) simple centrality metrics and b) the pre-computed graphlet degree vectors.

- **Graph Auto-Encoder (GAE)**: As shown in Figure 4(a), the GAE is an encoder-decoder framework. The GCN-based encoder maps nodes to a low-dimensional latent space (Z). A simple decoder then attempts to reconstruct the original graph's adjacency matrix from these latent embeddings. The model is trained by minimizing the cross-entropy loss between the original and reconstructed adjacency matrices, forcing the embeddings in Z to capture the graph's topological structure.

- **Deep Graph Infomax (DGI)**: The DGI framework operates on the principle of contrastive learning, as conceptualized in Figure 4(b). It learns node embeddings by training a discriminator to distinguish between "real" pairs (a node embedding and a summary vector of the entire graph) and "fake" pairs (a node embedding from a corrupted graph and the same summary vector). By maximizing the mutual information between the local node representations and the global graph summary, the encoder learns embeddings that are both locally aware and globally consistent.



(a) Graph Auto-Encoder (GAE) [1]



(b) Deep Graph Infomax (DGI) concept [6]

Figure 4: Two unsupervised Graph Neural Network architectures.

2.4.3 *Hyperparameter Tuning*

To ensure a fair and robust comparison, we performed hyperparameter tuning for all GNN-based models on each dataset. We conducted a grid search to find the optimal combination of parameters that maximized the Silhouette Score. For each set of hyperparameters, the model was evaluated for a range of role counts ($k \in \{3, 4, 5, 6, 7\}$), and the best score achieved across this range was used to select the winning parameter set. The search space for the key hyperparameters is detailed in Table 2. The best configuration found for each model and dataset was then used for the final evaluation.

Table 2: Hyperparameter search space for GNN-based models.

Hyperparameter	**Model(s)**	**Values Searched**
Learning Rate	GAE-based DGI-based	{0.01, 0.005} {0.001, 0.0005}
Hidden Channels	All GNNs	{128, 256}
Embedding Dimension	GAE-based	{32, 64}

2.4.4 *Evaluation Metrics*

Since role discovery is an unsupervised task without ground-truth labels, we use internal clustering validation metrics to assess the quality of the discovered roles:

- **Silhouette Score**: Measures how similar a node is to its own cluster (cohesion) compared to other clusters (separation). Ranges from -1 to 1, where a high value indicates dense and well-separated clusters.
- **Davies-Bouldin Index**: Computes the average similarity between each cluster and its most similar one. Lower values indicate better separation.
- **Calinski-Harabasz Index**: The ratio of the sum of between-cluster dispersion to the sum of within-cluster dispersion. Higher scores indicate better-defined clusters.

3. EXPERIMENTS AND RESULTS

3.1 Community Detection

We evaluated community detection algorithms by comparing traditional topology-based methods with modern GNN architectures. We first examine the datasets, then compare algorithm performance quantitatively across all datasets. Finally, we analyze results on the Cora network to understand the nature of discovered communities.

3.1.1 *Datasets and Their Impact*

The performance of community detection algorithms is heavily influenced by the underlying structure and characteristics of the network. We selected four standard datasets, each presenting a unique challenge:

- **Karate Club:** A small, well-studied social network with 34 nodes and 78 edges. Its primary characteristic is the absence of rich node features; the graph is defined purely by its topology.
- **Cora & PubMed:** These are large citation networks with thousands of nodes and edges. Critically, each node (a research paper) is endowed with high-dimensional features derived from its abstract's word vector. This semantic information provides a rich signal for feature-aware algorithms.
- **Cora Subset:** A 150-node subgraph of Cora, created to allow for the execution of computationally expensive algorithms like Girvan-Newman on a feature-rich network.

This selection was deliberate, as it allowed us to test a central hypothesis: **traditional algorithms, which rely on topology, will perform well on graphs defined by structure, while GNNs, which leverage features, will excel on networks where semantic information is available.** As expected, our results confirm this hypothesis. The poor performance of GNNs on the Karate Club dataset can be directly attributed to the lack of informative features, which are essential for the training and representation learning process. Conversely, their superior performance on Cora and PubMed is a direct consequence of their

ability to effectively utilize the rich textual features to identify semantically coherent communities.

3.1.2 General Algorithm Performance Analysis

The results across all datasets show distinct patterns for each algorithm type. Traditional methods focus on graph structure, while GNNs use node features to capture semantic relationships. Table 3 presents the complete results.

- **Louvain Algorithm:** Across all datasets, the Louvain method consistently achieved the highest Modularity scores, a result that is entirely expected as its objective function is to greedily optimize this exact metric. For instance, on the full Cora dataset, it reached a modularity of 0.8137, significantly outperforming all other models in this regard. However, this structural optimization came at a cost. Louvain consistently identified a very large number of communities (e.g., 101 on Cora), leading to a highly fragmented partition. While this yields low (good) Average Conductance scores, it results in poor performance on all external validation metrics (NMI, ARI, FMS) when compared to GNNs on feature-rich datasets. This shows that maximizing modularity does not necessarily equate to finding semantically meaningful, ground-truth-aligned communities.

- **Girvan-Newman Algorithm:** This algorithm performed very well on the small **Karate Club** graph, achieving the second-highest modularity (0.4013) and strong external validation scores (e.g., ARI of 0.8077). This performance is characteristic of the algorithm on small, well-defined networks. However, due to its high computational complexity, we could not run it on the larger Cora and PubMed datasets, which confirms its well-known limitation in scalability.

- **Label Propagation (LPA):** As expected, LPA was extremely fast, but its performance was generally disappointing. It often produced a massive number of communities (e.g., 454 on Cora), leading to lower modularity scores than Louvain and poor alignment with ground-truth labels. Its simple, neighbor-based label updating mechanism makes it susceptible to forming trivial or noisy communities in large, complex networks.

- **GNN Models (GCN & GraphSAGE):** The GNNs tell a story of two extremes. On the small, feature-poor **Karate Club** dataset, they struggled to outperform traditional methods, with GraphSAGE even yielding a negative modularity score (-0.0865), indicating a partition worse than random. This is expected, as GNNs require sufficient data and informative features for effective training. In sharp contrast, on the large, feature-rich **Cora** and **PubMed** datasets, GCN and GraphSAGE were the undisputed top performers on all external metrics. For example, on Cora, GCN achieved an ARI of 0.612, far surpassing Louvain (0.226) and Label Propagation (0.0761). This success is directly attributable to their ability to leverage the rich node features to learn semantically meaningful node embeddings, leading to communities that align almost perfectly with the ground-truth academic subjects. Their performance on both architectures was remarkably similar, suggesting that for these tasks, the choice between GCN and GraphSAGE was not critical.

- **Graphlet-Enhanced GNNs:** The introduction of graphlet features (triangle counts) had a surprisingly minor, and sometimes negative, impact. On the feature-rich datasets like Cora, augmenting the existing features with this structural information yielded only marginal performance changes. This suggests that the rich textual features already provided a strong enough signal, making the additional structural information redundant. On the feature-poor Karate Club dataset, replacing the original features with only graphlet features significantly worsened the performance of both GCN and GraphSAGE.

3.1.3 Interpreting Communities on the Cora Dataset

To better understand the practical implications of these results, we performed a qualitative analysis of the communities discovered in the Cora citation network. The t-SNE plot of the GCN-learned node embeddings, shown in Figure 5, provides a visual confirmation of the GNN's ability to learn semantically meaningful representations. The plot shows distinct clusters of nodes that correspond remarkably well with the ground-truth paper subjects.
This strong performance is a direct result of the GNN's capacity to integrate both the network's topology and the rich textual features of the research papers. By learning embeddings that place papers on similar topics close to each other in the latent space, the GNN is able to identify communities that are not only structurally dense but also thematically coherent. In conclusion, for networks with rich node attributes, GNN-based methods provide a superior approach for discovering meaningful communities in the real world. Although traditional topology-based algorithms remain valuable for their speed and ability to optimize structural metrics such as modularity. Besides that, their inability to leverage feature information limits their effectiveness in semantic community detection tasks.

3.2 Link Prediction

This analysis is structured to evaluate models not just on their ability to classify links, but also on their capacity to rank potential links effectively. The objective is to identify robust models that perform well across both classification metrics (AUC) and ranking metrics (MRR, Hits@K), as the latter is crucial for practical applications such as recommendation systems.

3.2.1 Datasets

The performance of link prediction models is highly dependent on the underlying structure of the network. Our results show a clear distinction between the performance on citation networks versus social networks, highlighting how different network topologies favor different aspects of the prediction task.

Table 3: Summary of Community Detection Results Across All Datasets and Algorithms. The best performing model for each metric category (Internal vs. External) is highlighted in bold for each dataset.

Dataset	Algorithm	# Communities	Modularity ↑	Avg. Conductance ↓	NMI ↑	ARI ↑	Fowlkes-Mallows ↑
KarateClub	Louvain	4	**0.4188**	0.4421	**0.8932**	**0.8653**	**0.9029**
	Label Propagation	3	0.3251	**0.4315**	0.5510	0.5115	0.6859
	Girvan-Newman	5	0.4013	0.5761	0.8301	0.8077	0.8609
	GNN (GCN)	4	0.3476	0.4869	0.6361	0.4745	0.6282
	GNN (GraphSage)	4	0.2772	0.6514	0.5058	0.4474	0.6219
	GNN (GCN) + Graphlets	4	-0.0639	0.8139	0.2762	0.1135	0.3648
	GNN (GraphSage) + Graphlets	4	-0.0865	0.8526	0.3879	0.2050	0.4444
Cora Subset	Louvain	9	**0.6566**	**0.3579**	0.1211	0.0064	0.3032
	Label Propagation	17	0.5821	0.4919	0.1476	-0.0039	0.3013
	Girvan-Newman	11	0.6391	0.4008	0.1358	0.0038	0.2754
	GNN (GCN)	6	0.0285	0.7376	0.5790	**0.6728**	0.9242
	GNN (GraphSage)	6	0.0052	0.8024	**0.6092**	0.6606	**0.9256**
	GNN (GCN) + Graphlets	5	0.0103	0.7349	0.4627	0.5543	0.9044
	GNN (GraphSage) + Graphlets	6	0.0053	0.7879	0.5673	0.6201	0.9189
Cora	Louvain	101	**0.8137**	**0.0470**	0.4498	0.2260	0.3447
	Label Propagation	454	0.6722	0.4600	0.4143	0.0761	0.2014
	GNN (GCN)	7	0.6967	0.2619	0.6036	**0.6120**	**0.6791**
	GNN (GraphSage)	7	0.6861	0.2821	0.5889	0.5892	0.6596
	GNN (GCN) + Graphlets	7	0.6998	0.2593	**0.6052**	0.6044	0.6726
	GNN (GraphSage) + Graphlets	7	0.6634	0.3204	0.5711	0.5607	0.6357
PubMed	Louvain	43	**0.7675**	0.2304	0.1982	0.1094	0.2869
	Label Propagation	1740	0.6185	0.5768	0.1805	0.0389	0.1683
	GNN (GCN)	3	0.5001	0.2636	**0.3799**	**0.4301**	**0.6322**
	GNN (GraphSage)	3	0.4938	**0.2575**	0.3493	0.3966	0.6093
	GNN (GCN) + Graphlets	3	0.5003	0.2753	0.3720	0.4244	0.6294
	GNN (GraphSage) + Graphlets	3	0.4563	0.2737	0.3335	0.3742	0.5978

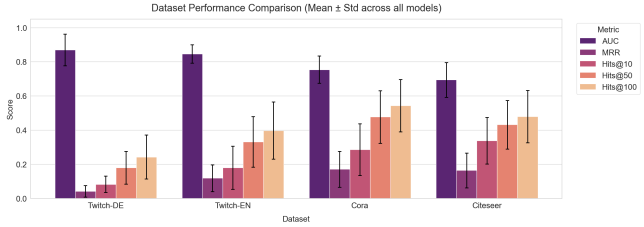


Figure 6: Average link prediction performance (mean ±std) across all models for each dataset.

As illustrated in Figure 6, for social networks, models are good at classifying links (high AUC scores around 0.85) but poor at ranking them (low MRR and Hits@K). Conversely, on Cora and Citeseer citation networks, models are better at ranking potential links (higher MRR and Hits@K) but less effective at simple classification (lower AUC). This demonstrates that the network's structure is a critical factor, and the choice of models and evaluation metrics should be aligned with the specific network and the desired outcome, such as classification versus ranking.

3.2.2 *Heuristic Methods*

As parameter-free baselines, the heuristic methods provide an essential benchmark for link prediction performance based purely on graph topology. As shown in Table 4, all four heuristics achieved comparable classification performance, with AUC scores ranging from 0.71 to 0.75.

However, a clear distinction emerges in their ranking capabilities. The **Adamic-Adar** model significantly outperformed the others in ranking, achieving the highest MRR (0.1726) and Hits@100 (0.3764) scores. This indicates that its method of weighting common neighbors by their degree of rarity is particularly effective for identifying and prioritizing true links.

Table 4: Performance of heuristic models on link prediction, averaged across all datasets. Adamic-Adar shows the strongest ranking performance (MRR, Hits@100).

Heuristic Model	AUC	MRR	Hits@100
Adamic-Adar	0.7446	0.1726	0.3764
Common Neighbors	0.7413	0.1144	0.3234
Jaccard Index	0.7196	0.1404	0.2471
Preferential Attachment	0.7134	0.0543	0.3105

3.2.3 *Traditional Machine Learning*

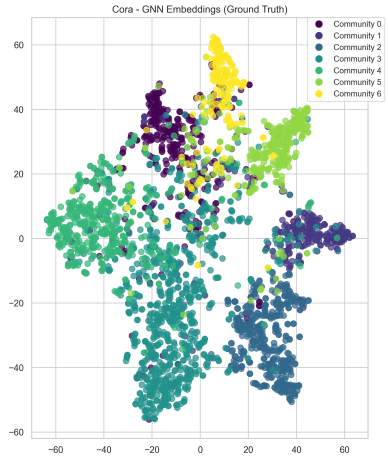
For the traditional machine learning models, each was trained using the optimal hyperparameters found during the tuning phase, which prioritized ranking performance (MRR). The results, summarized in Table 5, show a clear divergence in performance between the models.

While all models achieved respectable classification scores (AUC 0.79), the Decision Tree and KNN models proved to be less effective for ranking tasks. Both models yielded low MRR scores, and the high variance observed in their performance suggests they struggled to consistently learn ranking-oriented patterns in all datasets.

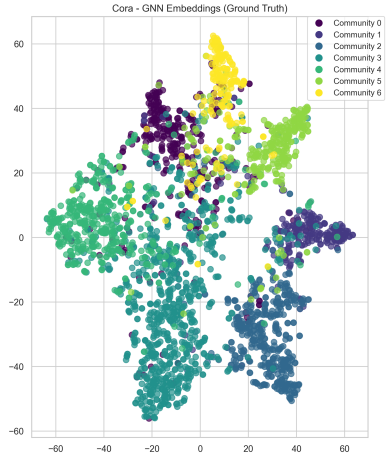
In contrast, Logistic Regression and Random Forest emerged as robust and effective solutions. Logistic Regression achieved the highest MRR (0.2064), and Random Forest posted the best Hits@100 score (0.5469). Their strong performance across all ranking metrics indicates a superior ability to generalize and effectively prioritize potential links.

3.2.4 *Graph Neural Networks (GNNs)*

Graph Neural Networks were trained by minimizing the bi-



(a) t-SNE visualization colored by ground-truth communities.



(b) t-SNE visualization colored by GCN-predicted communities.

Figure 5: A side-by-side comparison of t-SNE visualizations of GCN embeddings on the Cora dataset.

nary cross-entropy loss between predicted and true edges. While this objective function is designed to optimize link classification performance (AUC), it does not explicitly guarantee superior ranking performance (MRR, Hits@K). Our analysis, therefore, focuses on this potential trade-off and the impact of architectural choices, such as the decoder mechanism.

The results, presented in Table 6, reveal two key findings. First, GNNs achieved the highest classification scores of all model families, with `GCNModel` reaching an impressive AUC of 0.9003. This confirms their powerful ability to learn effective representations for discriminating between links and non-links.

Second, the choice of decoder has a significant impact on performance. For both GCN and GraphSAGE architectures, the models using a simple **Dot Product decoder** outperformed their counterparts using a more complex **MLP decoder** across all metrics. For instance, the GCN with a dot product decoder achieved an MRR of 0.1779, more than double that of the MLP version (0.0668). This suggests that the primary challenge lies in the GNN encoder learning high-quality node embeddings; once effective embeddings are learned, a simpler, non-parametric decoder is more effective and less prone to overfitting than a more complex one.

Table 5: Performance of traditional machine learning models, averaged across all datasets. Random Forest and Logistic Regression show the most robust performance, particularly for ranking metrics.

Model	**AUC**	**MRR**	**Hits@100**
Decision Tree	0.8359	0.1026	0.5159
Random Forest	0.8111	0.1972	0.5469
K-Nearest Neighbors	0.8055	0.0275	0.2888
Logistic Regression	0.7912	0.2064	0.4918

Table 6: Performance of GNN models, averaged across all datasets. Models with a Dot Product decoder consistently outperform their MLP counterparts for both GCN and GraphSAGE encoders.

Model Architecture		**AUC**	**MRR**	**Hits@100**
Encoder	**Decoder**			
GCN	MLP	0.8514	0.0668	0.4900
GCN	Dot Product	0.9003	0.1779	0.5619
GraphSAGE	MLP	0.7908	0.0588	0.3941
GraphSAGE	Dot Product	0.7936	0.1264	0.4107

However, exploring these models revealed experimental challenges. All GNN models were trained for a maximum of 500 epochs with an early stopping patience of 50, based on performance on the validation set. We found that this fixed setup might not be optimal for achieving a robust balance across all metrics. The primary challenge is that performance on the binary cross-entropy loss, classification AUC, and ranking MRR do not always correlate. A model trained for longer might achieve a high AUC but could begin to overfit on ranking-specific patterns, degrading its MRR score. The need to find a single stopping point that balances these competing objectives sometimes led to compromises in the final model performance. Future work could focus on multi-objective optimization during the tuning phase to more effectively navigate this complex trade-off and produce models that are strong at both classification and ranking.

3.2.5 Overall model comparison

To provide a clear summary of the trade-offs between the different modeling paradigms, we compare the best-performing model from each category: Adamic-Adar (Heuristic), Random Forest (Traditional ML), and GCNModel2 (GNN). The results, shown in Figure 7, highlight a clear trade-off between predictive power, ranking effectiveness, and computational cost.

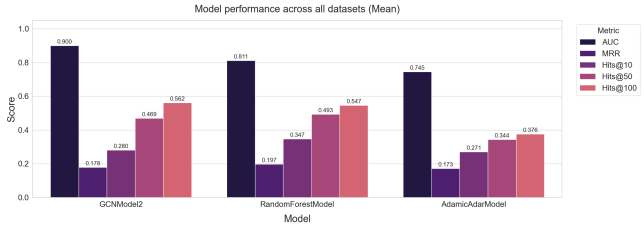


Figure 7: Performance comparison of the best model from each category: Heuristic (Adamic-Adar), Traditional ML (Random Forest), and GNN (GCNModel2 has the dot product decoder). While the GNN excels at classification (AUC), the Random Forest model offers the strongest overall ranking performance.

The GNN model (`GCNModel2`) unequivocally achieved the best link classification performance, with an average AUC of 0.900. This demonstrates the power of deep learning methods to learn effective node representations directly from graph structure for classification tasks.

However, for ranking-oriented metrics, the `RandomForestModel` proved to be the most effective solution, balancing performance and computing efficiency. It achieved the highest MRR (0.197) and the best performance on Hits@10 and Hits@50, indicating its superiority in accurately ranking true potential links. While the GNN's ranking scores were strong, they did not significantly surpass the feature-based Random Forest model.

This presents a crucial trade-off. The GNN provides better classification performance but is the most computationally expensive to tune and train. The Adamic-Adar heuristic offers a respectable baseline with zero training cost. The **Random Forest** model, however, represents the most practical and balanced approach. It delivers the best ranking performance while being significantly cheaper to train than a GNN, making it an excellent choice for applications where both high performance and computational efficiency are important.

3.3 Role Discovery

We first compare the models quantitatively across all datasets, then perform a qualitative analysis on the Cora dataset to interpret the nature of the discovered roles.

3.3.1 Model Performance

We evaluated six models across the Actor, CLUSTER, and Cora datasets. The optimal number of roles k, for each model was selected based on the Silhouette Score ($k \in [3, 7]$), which provides the most robust measure of cluster cohesion and separation.

Our experimental approach evolved through two phases. Initially, we compared traditional feature-based methods against graph neural networks using standard network centrality measures. This preliminary comparison revealed GNN superiority in learning meaningful node representations. However, introducing graphlet-based features dramatically transformed our findings.

Graphlet-based models consistently achieved the highest Silhouette Scores across all datasets, with other internal validation metrics (Davies-Bouldin and Calinski-Harabasz Indices) confirming this trend. Notably, when equipped with

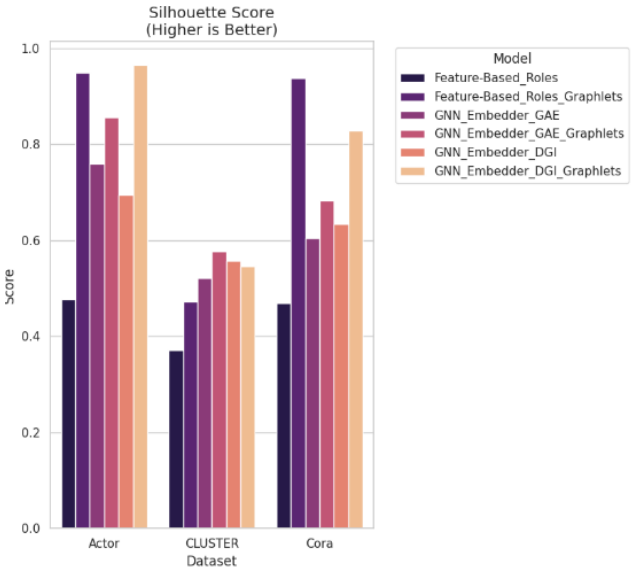


Figure 8: Comparison of the best Silhouette Score achieved by each model across the Actor, CLUSTER, and Cora datasets. Methods using graphlet features consistently produce roles that are more cohesive and better separated

graphlet features, the Feature-Based_Roles_Graphlets model often matched or exceeded GNN performance, suggesting that high-quality structural features can be as powerful as learned representations for role discovery.

3.3.2 Interpreting Roles on Cora

To move beyond quantitative scores and understand the practical utility of our methods, we perform a qualitative analysis of the roles discovered in the Cora citation network. We focus on the top-performing `Feature-Based Graphlets` model, which identified an optimal structure of three distinct roles ($k = 3$). This analysis reveals not only what the roles *are* structurally, but what they *do* functionally and semantically within the network.

First, we visualize the learned role assignments using t-SNE, which projects the high-dimensional graphlet feature vectors into a 2D space. As shown in Figure 9, the model produces clean and well-separated clusters. This visual evidence aligns with the model's high Silhouette Score (0.94) and confirms that the discovered roles are statistically robust and structurally distinct. The plot reveals a large, central "starburst" cluster (Role 0) and two smaller, perfectly isolated satellite clusters (Roles 1 and 2).

To interpret these roles, we first analyze their structural "fingerprints" using the radar plot in Figure 10. This chart displays the normalized average of key structural properties for each role. We then examine their interaction patterns using the role-to-role connectivity matrix in Figure 11, which shows how the different roles connect to one another.

The dominant role (**Role 0**, 2694 nodes) contains the vast majority of papers and exhibits the structural characteristics of a **periphery**. Its nodes have **low centrality scores** but **high clustering coefficients**, indicating they form tight, topically-focused communities with limited global influence. The interaction patterns support this interpretation, with

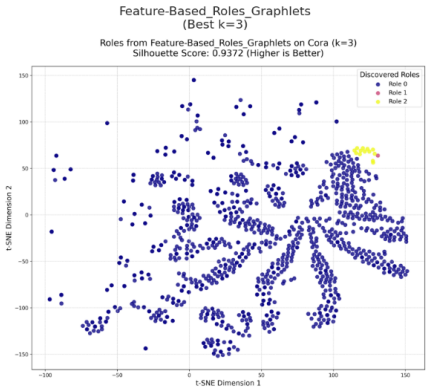


Figure 9: t-SNE visualization of the three roles discovered by the `Feature Graphlets` model on Cora. The clear separation of clusters corresponds to a high Silhouette Score of 0.9372, indicating a high-quality partition.

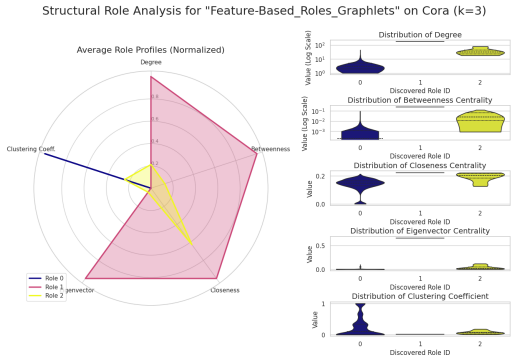


Figure 10: Structural 'fingerprints' and property distributions of the three roles discovered on Cora. The radar plot (left) shows the normalized average of key centrality metrics, revealing the unique signature of each role, while the violin plots (right) show the distribution of these properties within each role.

93% of connections remaining internal to this role.

At the opposite extreme, **Role 1** contains only a single node that acts as the network's **global hub**. This paper exhibits exceptionally high centrality metrics combined with a very low clustering coefficient, signature of a bridge node that connects otherwise disparate network regions without belonging to any particular community.

The intermediate role (**Role 2**, 13 nodes) represents a small but influential group of papers. These nodes maintain centrality scores well above the periphery but below the global hub, suggesting they function as important **secondary hubs or bridges** within major subfields.

Both **Role 1 and 2** direct over **96% of their connections** toward the periphery, with minimal interconnection between them, confirming their **satellite-like relationship**.

3.3.3 Bridging Structure and Semantics

When we map these purely structural roles to the semantic labels (paper subjects) of the nodes (information to which the model was never exposed), the results shown in Figure

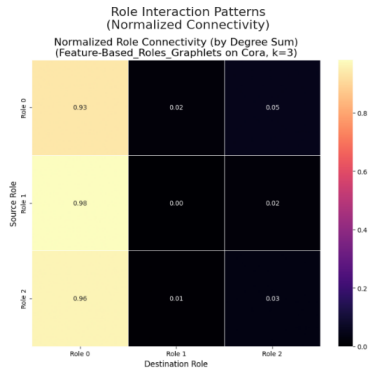


Figure 11: Role interaction patterns on Cora. The heatmap shows the normalized connectivity (by degree sum) from a source role (y-axis) to a destination role (x-axis). High values on the diagonal indicate insular, community-like behavior, while high off-diagonal values reveal connector or satellite roles.

12 reveal a clear link between the structural function of a paper and its academic field.

The analysis reveals that **Role 1**, the global hub, is a paper on **"Genetic Algorithms."** **Role 2** is predominantly composed of papers on **"Neural Networks"** and **"Reinforcement Learning"**. This suggests that the model identified a shared structural signature among these closely related AI fields, grouping them into a "connector" role. In contrast, the large "Periphery" role (Role 0) contains a diverse mix of all subjects, as expected from a core body of literature.

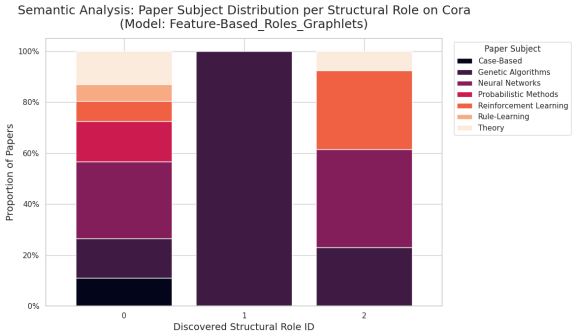


Figure 12: Paper subject distribution within each structurally-defined role. The model, blind to semantics, discovered roles that correspond to distinct academic fields.

4. CONCLUSION AND FUTURE WORK

Our systematic evaluation across community detection, link prediction, and role discovery reveals key insights about the relative strengths of classical and modern approaches. The results demonstrate that algorithm selection must be guided by network characteristics and application requirements rather than universal superiority assumptions.

For **community detection**, traditional methods like Louvain excel at structural optimization on topology-only networks, while GNNs achieve superior semantic alignment when rich node features are available. The high performance dif-

ference between GNNs on Karate Club versus citation networks highlights the critical importance of feature availability for deep learning approaches.

In **link prediction**, we observe a fundamental trade-off between classification accuracy and ranking effectiveness. While GNNs achieve the highest AUC scores, Random Forest demonstrates slightly superior ranking performance), suggesting that classification-focused applications should favor GNNs while recommendation systems can consider simpler models based on traditional ML approaches. The consistent superiority of dot product over MLP decoders suggests that learning high-quality embeddings is more important than complex decoding mechanisms for this task.

For **role discovery**, graphlet-based features consistently outperformed both traditional centrality measures and learned GNN embeddings, achieving exceptional clustering quality. The qualitative analysis reveals that purely structural methods can discover semantically meaningful roles without any semantic input.

Several limitations suggest directions for future work. The fixed early stopping criteria and single-objective optimization may be suboptimal across diverse datasets, suggesting the need for multi-objective training frameworks and adaptive stopping criteria. The limited exploration of higher-order graphlets presents opportunities for more sophisticated structural feature engineering.

Our findings establish that feature engineering remains crucial even in the deep learning era, with carefully designed structural features often matching or exceeding learned representations. This work provides a comprehensive benchmark for network analysis tasks while revealing fundamental trade-offs that should guide both algorithmic development and practical deployment decisions.

5. REFERENCES

[1] N. A. Asif, Y. Sarker, R. Chakrabortty, M. Ryan, M. H. Ahamed, D. Saha, F. Badal, S. Das, F. Ali, S. Moyeen, M. Robiul Islam, and Z. Tasneem. Graph neural network: A comprehensive review on non-euclidean space. *IEEE Access*, 9:75553–75572, 03 2021.

[2] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, C. Faloutsos, B. A. Prakash, and L. Akoglu. Rolx: structural role extraction & mining in large graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1231–1239, 2012. Introduced the synthetic CLUSTER dataset designed for role discovery evaluation.

[3] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang. Geom-gcn: Geometric graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, pages 476–486, 2020. Source for the Actor dataset as used in many modern GNN benchmarks.

[4] B. Rozemberczki, C. Allen, and R. Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021. Source for the Twitch-EN and Twitch-DE social network datasets.

[5] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–106, 2008. Source for the Cora, Citeseer, and PubMed datasets.

[6] X. Xu, J. Du, J. Song, and Z. Xue. InfoMax classification-enhanced learnable network for few-shot node classification. *Electronics*, 12(1):239, 2023.

[7] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977. Source for the Karate Club dataset.