MGNN-AL: A Graph Neural Network with Active Learning for Node Importance Prediction in Multilayer Networks

Jiaxun Li, Li Tao

Measuring node importance in multilayer networks is a challenging and relatively under-explored research problem. Although numerous methods have been developed for single-layer networks, they cannot guarantee reliable results when directly applied to multilayer settings. Most existing multilayer network node identification methods overlook the unique structural characteristics inherent in multilayer networks, particularly the importance of layers. Furthermore, with the rapid development of deep learning, which has achieved significant success across various fields, applying deep learning methods to node identification in multilayer networks has become a promising research direction. This paper models the problem of identifying important nodes in multilayer networks as a classical regression task and proposes a multilayer network node influence prediction method that integrates active learning with graph neural networks, called MGNN-AL. The proposed method integrates the node characteristics along with a layer attention mechanism, effectively capturing the distinct structural features of different layers and their varying contributions to node importance. Empirical evaluations across a wide range of real-world multilayer networks consistently validate the superiority of the proposed method, particularly in enhancing node identification accuracy compared to existing methods.

Index Terms—Multilayer Networks, Node Importance Prediction, Graph Neural Networks , Active Learning.

1 Introduction

In the field of complex network research, single-layer network models have been widely employed to represent systems composed of a single type of node and a single type of relationship [1]—for instance, interactions among users in social networks. However, real-world complex systems rarely exist in isolation; rather, they are formed through multiple types of interactions, exhibiting strong heterogeneity and multidimensional attributes. For example, in social networks, user relationships across different platforms often display heterogeneity, and a single user may demonstrate distinct patterns of social connections on multiple platforms. Moreover, interpersonal ties in social networks are inherently multidimensional, encompassing kinship, collegial, and friendship relations, among others. Together, these diverse types of connections yield network structures of greater complexity. Consequently, traditional single-layer models struggle to capture such multifaceted systems comprehensively, motivating the introduction of multilayer networks [2], [3]. A multilayer network consists of multiple interconnected layers, each representing a specific type of relation or interaction pattern. Within this framework, a given node can establish different types of connections across layers, enabling a more precise representation of the multidimensional interactions that characterize real-world systems.

Compared with single-layer networks, the task of evaluating important nodes in multilayer networks is considerably more challenging. This complexity primarily arises from the fact that a node may assume different roles and exert varying levels of influence across distinct layers. For instance, a user may be highly active on WeChat while showing little to no engagement on Weibo. If the assessment of this user's importance relies solely on the structural features of a single platform, it risks overlooking their role in other layers, leading to partial and potentially inaccurate conclusions. Therefore, the problem of identifying important nodes in multilayer networks is of substantial research significance.

Over the past decades, scholars have proposed a variety of methods for ranking node importance in single-layer networks from different perspectives, such as degree centrality [4], betweenness centrality [5], closeness centrality [6], eigenvector centrality [7], and PageRank [8]. These methods have demonstrated strong performance in identifying important nodes within single-layer structures. However, research on node importance identification in multilayer networks remains relatively limited. Most existing methods are direct extensions of single-layer centrality measures and thus fall short in capturing the unique topological properties inherent to multilayer networks, which restricts their effectiveness. Although such methods can identify a subset of important nodes to some extent, they largely overlook the distinctive structural information of multilayer networks.

To address the limitations of existing methods in identifying important nodes within multilayer networks, this paper proposes a graph neural network method enhanced with an active learning mechanism for node importance prediction in multilayer network, referred to as MGNN-AL. Specifically, a fourth-order tensor is first employed to model the multilayer network, from which five categories of structural features are extracted to construct the input matrix. Node representations are then learned through the

• *Jiaxun Li and Li Tao are with the College of Computer and Information Science, Southwest University, Chongqing, China. E-mail:tli@swu.edu.cn*

GraphSAGE [9] model. Subsequently, a layer-wise attention mechanism is introduced to quantify the importance of each layer, enabling effective modeling of the structural characteristics unique to multilayer networks. Based on this design, the node scores from each layer are aggregated through weighted fusion with their corresponding layer weights to generate the final prediction of node influence.Furthermore, an active learning strategy is incorporated to optimize the model. In particular, the Louvain community detection algorithm [10] is applied to partition the network into multiple communities, from which the node with the highest degree centrality in each community is selected as a representative sample for iterative model training. This strategy improves both the predictive performance and the generalization ability of the proposed framework. The main contributions of this paper are as follows:

- We formulate the problem of identifying important nodes in multilayer networks as a classical regression task and propose MGNN-AL, a multilayer influence prediction method that integrates graph neural networks with active learning, thereby improving the accuracy of important node identification in multilayer environments.
- We introduce a layer-wise attention mechanism to capture the heterogeneous structural properties of different layers and their varying contributions to node importance, enhancing the model's ability to characterize complex network structures.
- We incorporate an active learning strategy to optimize the selection of training samples, which strengthens the adaptability and generalization capability of the model in multilayer complex networks.

The remainder of this paper is organized as follows. Section II provides a brief review of existing methods for important node identification. Section III describes the proposed MGNN-AL method in detail. Section IV outlines the experimental setup, including datasets, baseline methods, the SIR model, and evaluation metrics. Section V reports and analyzes the experimental results. Finally, Section VI concludes the paper and discusses potential directions for future research.

2 Related Work

A wide range of methods have been proposed for identifying important nodes in complex networks. According to the type of topological information they rely on, these methods can be broadly classified into two categories: global-information-based methods and local-information-based methods. Global methods exploit the overall network structure to evaluate node importance. Representative approaches include betweenness centrality (BC) and closeness centrality (CC). BC measures a node's contribution to network connectivity by calculating its frequency of occurrence on shortest paths, while CC evaluates the sum of the shortest path lengths from a node to all other nodes, reflecting its global centrality within the network. Although these methods are effective in identifying critical nodes, their high computational complexity limits their applicability to large-scale networks. Local methods, on the other hand, are more computationally efficient and therefore better suited for large-scale networks. Typical examples include degree centrality (DC) and the H-index [11]. DC measures a node's influence by counting its number of neighbors, but it neglects the attributes of those neighbors. Similarly, the H-index incorporates local connectivity but still relies heavily on local topological information, which may cause the identification results to be restricted to locally optimal solutions.

In addition, with the rapid progress of machine learning and deep learning techniques, new research directions have emerged in the field of important node identification. By employing embedding methods and manually designed features, researchers can obtain vectorized representations of network structures and subsequently train machine learning models to predict node importance [12], [13]. The advent of graph neural networks (GNNs) [14] and graph convolutional networks (GCNs) [15] has further broadened this perspective, enabling the integration of advanced learning paradigms into important node identification and improving both performance and accuracy [16], [17], [18]. For instance, Kumar [19] reformulated the influence maximization problem as a classical regression task and solved it through transfer learning using a graph-based long short-term memory network (GLSTM). Yu [20] combined adjacency matrices with convolutional neural networks (CNNs) and proposed a simple yet efficient method, RCNN, for identifying the most important nodes in terms of spreading capability. In RCNN, a feature matrix is generated for each node and subsequently processed with CNNs to train and predict node influence. These approaches have demonstrated excellent effectiveness in terms of both accuracy and computational efficiency. However, most existing GNN-based methods still focus on single-layer networks, while research on important node identification in multilayer complex networks remains relatively scarce. Moreover, deep learning methods typically require large amounts of training data, whereas large-scale real-world datasets are often difficult to obtain, which imposes significant constraints on their application and further development.

With the rapid development of information technology and the explosive growth of data, traditional single-layer networks are no longer sufficient to characterize complex networks or system structures. To better capture multidimensional interaction patterns, the concept of multilayer networks has been introduced. Some researchers have analyzed important nodes by explicitly considering not only node centrality within and across layers but also the relative importance of layers themselves. For example, [21] represented multilayer networks using a third-order tensor and computed both node centrality and layer centrality by solving tensor equations, further proving the uniqueness of multilayer eigenvector centrality. More recently, Lv et al. [22] extended this idea within a tensor-based framework by modeling multilayer networks with a fourth-order tensor and proposed an improved PageRank-based gravity centrality method (PRGC) for identifying key nodes. PRGC builds upon multi-PageRank centrality and operates in two stages. In the first stage, node and layer centrality values are obtained by solving tensor equations. In the second stage, based on the computed centrality, the method introduces a novel distance calculation strategy that accounts for the

influence of different layers on the shortest paths between nodes, providing a more accurate characterization of node importance in multilayer networks. However, since this method involves high-order tensor operations, its computational complexity is substantial, limiting its applicability to large-scale networks. In addition to tensor-based methods, a number of heuristic algorithms have been designed based on rules and empirical observations to identify important nodes in multilayer networks. For instance, Wang et al. [23] proposed a weighted local structure entropy approach, which integrates intra-layer betweenness centrality with the effects of inter-layer connections to comprehensively assess node importance across multiple layers. Similarly, other researchers have extended gravity models to multilayer contexts. Ni et al. [24] developed a weighted gravity center method (WGCM), which incorporates both neighborhood size and social distance, and later proposed a multiple gravity centrality (MGC)-based two-step seeding strategy. This strategy first determines the target layer for seed identification and then selects the corresponding seed nodes. Although these methods have shown promising results in identifying important nodes within multilayer structures, they generally involve global computations with high complexity, which constrains their scalability to large networks.

3 Proposed Method

This section provides a detailed description of the proposed MGNN-AL model, whose overall architecture is illustrated in Fig. 1. In the task of identifying important nodes, the central challenge lies in estimating the influence value of each node in the network. To address this, this study formulates the important node identification problem as a regression task, aiming to predict continuous influence score for every node. The method incorporates the concept of transfer learning. Specifically, the model is first trained on a synthetic network characterized by structural complexity, rich data, and high diversity, yielding model parameters with strong generalization capability. The trained model and its learned parameters are then transferred to the target network, where they are further refined through active learning. With the fine-tuned parameters, the model predicts node influence values in the target network, ranks the nodes based on these predictions, and ultimately identifies the most important ones. To further enhance predictive performance on the target network, the method introduces an active learning mechanism. During prediction, representative node samples are actively selected to dynamically expand the labeled dataset, which is then used to fine-tune the model and improve its generalization ability. The remainder of this section presents detailed descriptions of each key module of the proposed algorithm.

3.1 Feature Extraction

To comprehensively capture the structural characteristics of nodes in the network, this study selects several classical centrality measures as node feature inputs, including Degree Centrality (DC), K-Shell (KS), Eigenvector Centrality (EC), Average Neighbor Degree (AND), and Average Neighbor Clustering Coefficient (ANCC). These indicators characterize node importance from different perspectives and provide effective feature support for the subsequent node prediction task. DC quantifies the number of direct neighbors connected to a node, reflecting its local level of activity. KS assigns shell indices to nodes according to their peeling order, indicating the hierarchical position and structural stability of nodes in the network. EC recursively evaluates the importance of a node's neighbors, considering both its direct connections and the influence of its adjacent nodes. AND measures the mean degree of a node's neighbors, capturing the density of the local neighborhood. ANCC assesses the tightness of connections among a node's neighbors, thus reflecting the local clustering structure.

$$\begin{cases} p = \left\{ \dfrac{DC(v)}{\max(p)}, v \in V \right\} \\ q = \left\{ \dfrac{KS(v)}{\max(q)}, v \in V \right\} \\ r = \left\{ \dfrac{EC(v)}{\max(r)}, v \in V \right\} \\ s = \left\{ \dfrac{AND(v)}{\max(s)}, v \in V \right\} \\ t = \left\{ \dfrac{ANCC(v)}{\max(t)}, v \in V \right\} \end{cases} \tag{1}$$

Let p, q, r, s, t denote the centrality measures introduced above, and to eliminate the adverse effect of heterogeneous feature scales on the model training process, all node features are normalized. Specifically, the max-normalization method is employed, in which each feature value is divided by the maximum value of that feature across all nodes, linearly mapping the feature values into the interval [0,1]. The normalization formula is given as (1):

Let X denote the normalized feature matrix of nodes in the multilayer network, where $X \in R^{N \times 5 \times L}$. Here, N represents the number of nodes in the network, and L denotes the number of layers. Formally, it can be expressed as:

$$X = \left\{ \left[p_u^l, q_u^l, r_u^l, s_u^l, t_u^l \right], u \in V, l \in \{1, \cdots, L\} \right\} \tag{2}$$

3.2 Nodes representation learning with layer attention

After obtaining the input feature matrix of the target network, this study employs GraphSAGE to learn node representations. GraphSAGE is a graph neural network model based on neighborhood sampling and feature aggregation, which effectively captures local structural information and is well-suited for large sparse graphs. The representation of node v at the l-th layer is given by:

$$h_v^{(k,l)} = \sigma\left(W^{(k,l)} \cdot \mathrm{CONCAT}\left(h_v^{(k-1,l)}, \mathrm{LSTM}\left(\left\{ h_u^{(k-1,l)} \mid u \in N_l(v) \right\} \right) \right) \right) \tag{3}$$

where:

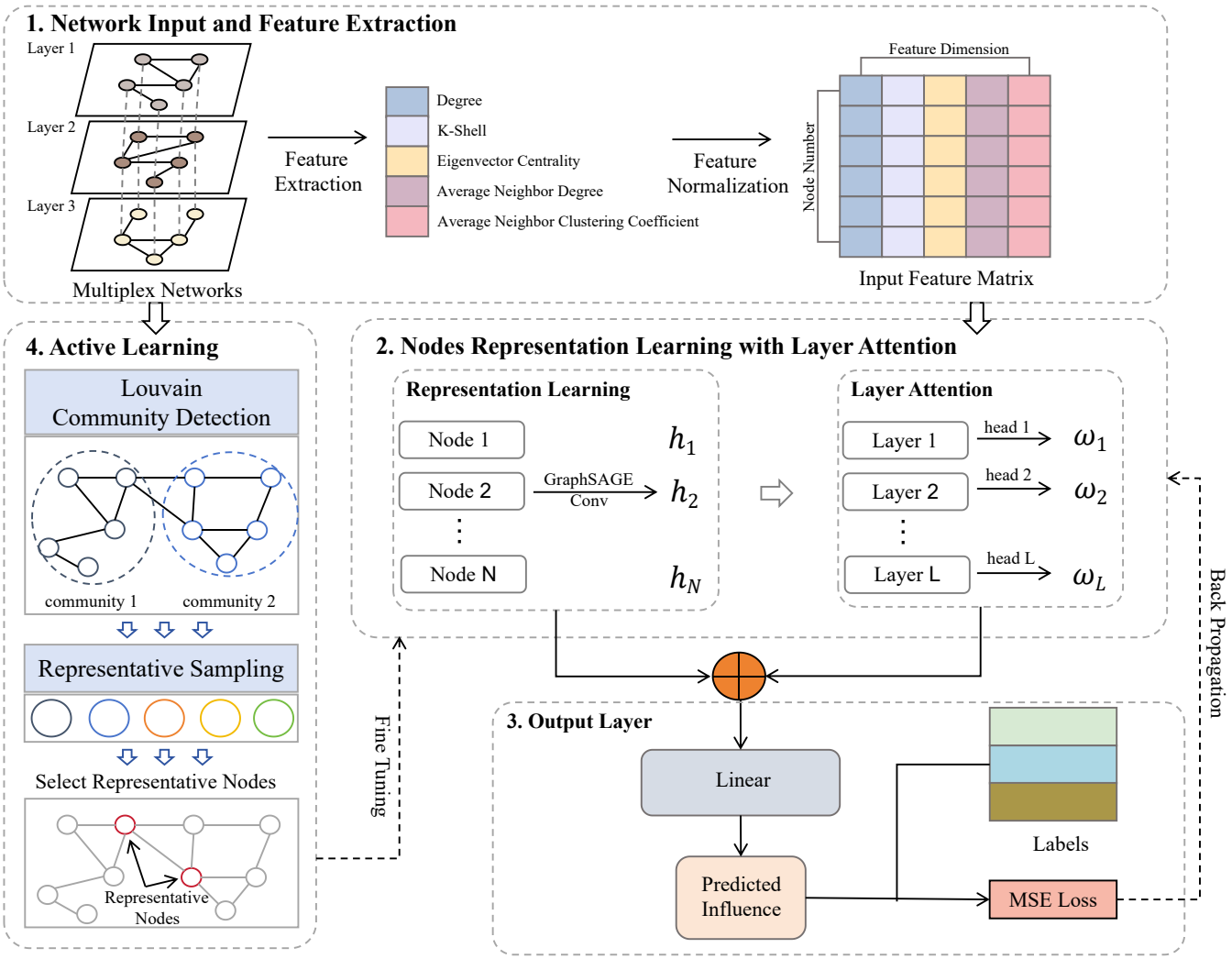


Fig. 1: The Structure of the MGNN-AL Model. The structure of MGNN-AL is divided into four parts: (1)Five types of node features are selected to construct the input feature matrix; (2)Two GraphSAGE convolutional layers with LSTM aggregators and a layer attention mechanism are employed to learn node representations; (3)Output layers for mapping the captured high-dimensional features to node influence values; (4)Active learning conducted on real networks using representative sampling.

- k denotes the number of GNN layers, and l denotes the index of the multilayer network layer;
- $W^{(k,l)}$ is the trainable weight matrix;
- $\mathbf{h}_v^{(0,l)} = X_v^{(l)}$;
- $\mathcal{N}_l(v)$ represents the set of neighbors of node v in the l-th layer of the network;
- σ is the activation function, for which ReLU is adopted in this work.

In graph neural networks, attention mechanisms are typically employed to adaptively assess either the relevance between a node and its neighbors or the contribution of different structural levels to node representation learning. For example, the Graph Attention Network (GAT) [27] dynamically assigns importance weights to neighboring features based on their correlation with the target node's features, enhancing the expressiveness of node representations in complex network structures. In the case of multilayer networks, variations in node connectivity patterns exist across different layers, and the contribution of each layer to the final evaluation of node importance is inherently uncertain. Inspired by this idea, we introduce a layer attention mechanism to dynamically evaluate the relative importance of each layer in multilayer networks. Specifically, we assign a dedicated attention head to each layer and compute its weight for the final node influence prediction task using the hidden states of all nodes within that layer. The layer attention weight is calculated as follows:

$$\omega_l = \frac{\exp\left(\mathrm{ReLU}\left(\mathbf{a}^\top \mathbf{W} \cdot \frac{1}{|V|}\sum_{i\in V}\mathbf{h}_i^{(l)}\right)\right)}{\sum_{m=1}^{L}\exp\left(\mathrm{ReLU}\left(\mathbf{a}^\top \mathbf{W} \cdot \frac{1}{|V|}\sum_{i\in V}\mathbf{h}_i^{(m)}\right)\right)} \tag{4}$$

where:

- $\mathbf{h}_i^{(l)}$ denotes the representation vector of node i at layer l;

- $\mathbf{a}^\top$ and $\mathbf{W}$ are learnable parameters;
- $\exp(\cdot)$ corresponds to the softmax normalization.

After learning node representations with GraphSAGE and extracting layer weights through the attention mechanism, the final embedding vector of node i can be expressed as follows:

$$h_i^{'} = \sum_{l=1}^{L} w_l \cdot h_i^l \tag{5}$$

3.3 Output Layer

The output layer of the model employs a linear transformation, which takes the final node representations obtained above as input features and generates the predicted influence values of the nodes. The prediction loss is computed by comparing the predicted values with the corresponding ground-truth labels, and model parameters are updated through backpropagation to enhance prediction accuracy. The mathematical formulation of the output layer is as follows:

$$\hat{y}_i = W_{out} \cdot h_i^{'} + b_{out} \tag{6}$$

where $\hat{y}_i$ denotes the predicted value, while W_{out}, $h_i^{'}$, and b_{out} represent the learnable weight matrix, the input feature vector, and the bias term, respectively.

To measure the deviation between the predicted results and the ground-truth labels, this study adopts the Mean Squared Error (MSE) as the loss function. By minimizing this loss through backpropagation, the model parameters are iteratively updated, improving the prediction performance. The MSE is defined as follows:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{7}$$

where N denotes the number of nodes, y_i represents the ground-truth influence value of node i, and $\hat{y}_i$ denotes the predicted value.

3.4 Active Learning

To further enhance the prediction performance of the proposed model in multilayer complex networks, this study introduces an active learning strategy aimed at optimizing the training process by selecting representative node samples. The detailed procedure is as follows: First, the Louvain community detection algorithm is applied independently to each layer of the multilayer network to obtain the community partition results. Let the partition of the l-th layer network be denoted as $\mathcal{C}^{(l)} = \{C_1^{(l)}, C_2^{(l)}, \ldots, C_{m_l}^{(l)}\}$, where m_l is the number of detected communities in layer l. Subsequently, within each community $C_j^{(l)}$, the node with the highest degree centrality is selected as the representative node, denoted as

$$v_j^{(l)} = \arg\max_{v \in C_j^{(l)}} \text{Deg}(v), \tag{8}$$

where $\text{Deg}(v)$ represents the degree centrality of node v. All representative nodes selected across all communities and all layers are then aggregated to form the active learning sample set:

$$\mathcal{S}_{AL} = \bigcup_{l=1}^{L} \bigcup_{j=1}^{m_l} \{v_j^{(l)}\}. \tag{9}$$

Finally, the representative node samples in $\mathcal{S}_{AL}$ are fed into the model for additional training or fine-tuning, further improving the prediction accuracy and generalization ability of the model.

4 Experimental Setup

This section primarily aims to explain the experimental setup used to validate the proposed method, MGNN-AL. First, the datasets employed in the experiments are introduced. Next, the model training setup is described. Then, the baseline methods used for comparison with this approach are presented. Following that, the SIR model used to simulate the propagation process is outlined. Finally, the performance evaluation metrics for the proposed method are provided.

4.1 Datasets

This paper selects 10 real-world multilayer network datasets, as shown in the table 1, to validate the effectiveness of the proposed method, MGNN-AL. These datasets can be roughly divided into three categories, with the number of nodes ranging from several dozen to over 8,000, and network layers spanning from 3 to 7. They exhibit varying scales and structural characteristics, enabling a comprehensive evaluation of the method's performance across diverse network environments.

TABLE 1: BASIC INFORMATION OF MULTILAYER NETWORKS

Network	L	N	E	Type
CS-Aarhus [29]	5	61	620	Social
Lazega-Law-Firm [30]	3	71	2223	Social
HepatitusCVirus [31]	3	105	137	Genetic
CKM-Physicians-Innovation [32]	3	246	1551	Social
C.Elegans-Connectome [32]	3	279	5863	Neural
HumanHIV1 [32]	4	1005	1355	Genetic
Rattus [33]	6	2640	4267	Genetic
C.Elegans [33]	6	3879	8181	Genetic
Arabidopsis [33]	7	6980	186547	Genetic
Drosophila [33]	7	8215	43366	Genetic

4.2 Model Training Setup

In the pre-training phase, our model is first trained on a set of synthetic multilayer networks. The structures of these synthetic networks are defined as follows: the total number of nodes is 10,000; the average degree takes values of 2, 4, 6, and 8; and the number of layers ranges from 3 to 8. To introduce randomness, each layer is generated with varying numbers of nodes and edges, resulting in a total of 24 distinct structures. The goal is to enable the model to acquire an initial capability for predicting node influence across different networks. The model takes the network topologies and node features as input and outputs predicted node influence values. The input dimension is $N \times 5 \times L$

(where N is the number of nodes and L is the number of layers). Convolutions are performed using GraphSAGE, producing an output of dimension $N \times 32 \times L$. A layer-attention mechanism is then applied to assign weights across layers, followed by two fully connected layers for regression (32–16–1). The mean squared error (MSE) is used as the loss function. During training, the learning rate is set to 0.005, with 500 epochs, and the Adam optimizer is employed. For prediction tasks on real networks, we adopt an active learning strategy to fine-tune the pre-trained model parameters, improving prediction accuracy and enhancing the model's generalization capability.

4.3 Methods to Compare

To effectively validate the effectiveness of the proposed method, we compare it with seven representative approaches, including:

- **Centrality-based methods**: DC [4] and K-shell [28]
- **Heuristic-based methods**: ED [23]
- **Deep learning-based methods**: RCNN [20] and GLSTM [19]
- **Tensor decomposition-based methods**: F-e [21] and PRGC [22]

It should be noted that the DC, K-shell, RCNN, and GLSTM methods cannot directly identify important nodes in multilayer networks. Therefore, for these methods, we first evaluate each layer of the network separately, then aggregate the results by summing and averaging the scores across layers to obtain the final importance of each node within the entire multilayer network.

4.4 SIR Model

The SIR model [34] describes the process of epidemic spreading within a population, where nodes transition among three states: susceptible (S), infected (I), and recovered (R). Initially, only one node is infected, while all other nodes remain susceptible. At each time step, an infected node transmits the disease to its susceptible neighbors with probability β, and recovers with probability $\lambda = 1$. This process continues until no infected nodes remain. At time t, the total number of infected and recovered nodes is denoted as $F(t)$, which increases over time and eventually stabilizes. To reduce randomness, the spreading influence is averaged over 1000 independent runs. The epidemic threshold of the network [35] is given by:

$$\beta_c \approx \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle} \tag{10}$$

where $\langle k \rangle$ denotes the average degree of the network, and $\langle k^2 \rangle$ represents the second moment of the degree distribution.

4.5 Evaluation Metrics

(1) Kendall's tau value (τ):

Kendall's Tau [36] measures the correlation between two ordered sequences. One sequence is obtained from the node ranking produced by the identification method, and the other is derived from the SIR model. The consistency of the two rankings is evaluated by counting the number of concordant and discordant pairs. A pair of elements is concordant if their relative order is the same in both sequences, and discordant otherwise. The Kendall's Tau coefficient is defined as:

$$\tau = \frac{2(n_c - n_d)}{N(N-1)} \tag{11}$$

where n_c and n_d denote the number of concordant and discordant pairs, respectively. The value of τ ranges from -1 to 1, with larger values indicating greater similarity between the two rankings.

(4) Final infected scale ($F(t)$):

The SIR diffusion model is employed to simulate the information spreading process. A seed set of size k is selected as the initial infected nodes, and the diffusion proceeds with infection probability β and recovery probability λ. For simplicity, we set $\lambda = 1$, which means that each infected node recovers in the next time step. The spreading process terminates at time t_c when no infected nodes remain. The final infected scale is defined as the proportion of nodes infected during the SIR process, mathematically expressed as:

$$F(t) = \frac{R(t)}{N} \tag{12}$$

where $R(t)$ denote the numbers of recovered nodes at time t, and N is the total number of nodes in the network.

5 Results and Discussion

This section provides a comprehensive analysis of the experimental results obtained on the aforementioned datasets, focusing on the performance of the proposed method in comparison with other approaches. Specifically, the analysis includes experiments on Kendall's tau, infection scale, and an ablation study, which collectively validate the effectiveness and advantages of the proposed method from multiple perspectives.

5.1 Comparison of Identification Accuracy

Table. 2 reports the Kendall's Tau results of eight methods across ten datasets. The experiments are conducted based on the SIR model, with the infection probability set to β_c. As shown in the table, the proposed method MGNN-AL consistently achieves the best performance on all datasets.

In addition, to further investigate the stability of each method under different infection probabilities, we analyze the Kendall's Tau values across varying infection settings. Fig. 2 illustrates the variation of Kendall's Tau when the infection probability ranges from $0.5 * \beta_c$ to $1.5 * \beta_c$ with a step size of 0.25. It can be observed that MGNN-AL maintains consistently high Kendall's Tau values across all ten datasets, with relatively small fluctuations, demonstrating strong stability and consistency. In particular, in networks such as *Arabidopsis_Genetic*, *C-elegans_Connectome*, *CS-Aarhus*, and *Drospophila_Genetic*, MGNN-AL shows a significant advantage over the other methods. Even under large variations of the infection probability, MGNN-AL is able to preserve high ranking consistency, indicating its robustness against changes in infection parameters.

TABLE 2: Kendall's tau values (τ) obtained for various datasets using different algorithms. The value of the infection probability (β) is taken as β_c under the SIR model. The best-performing results (ranked first) are highlighted in bold.

Dataset	DC	Kshell	GLSTM	RCNN	F-e	PRGC	ED	MGNN-AL
Arabidopsis	0.5090	0.5520	0.5703	0.2147	0.4821	0.3265	0.5355	**0.8091**
C.Elegans-Connectome	0.6245	0.6366	0.4896	0.6201	0.6837	0.6576	0.6182	**0.8670**
C.Elegans	0.4403	0.4359	0.5120	0.0307	0.1760	0.2301	0.5570	**0.7206**
CKM-Physicians-Innovation	0.7066	0.6664	0.4882	0.6865	0.3749	0.3407	0.7136	**0.8190**
CS-Aarhus	0.7475	0.7148	0.6339	0.6251	0.6514	0.5377	0.7005	**0.8415**
Drosophila	0.7013	0.6942	0.6653	0.4210	0.4529	0.4639	0.7034	**0.8341**
HepatitusCVirus	0.5344	0.4648	0.5081	0.4344	0.1388	-0.0564	0.4227	**0.5828**
HumanHIV1	0.3666	0.3740	0.2722	0.2594	0.2920	0.1914	0.3198	**0.5506**
Lazega-Law-Firm	0.8599	0.6901	0.4535	0.5662	0.8262	0.8101	0.8672	**0.8801**
Rattus	0.6140	0.4297	0.6067	0.0442	0.0271	0.1154	0.3576	**0.6464**

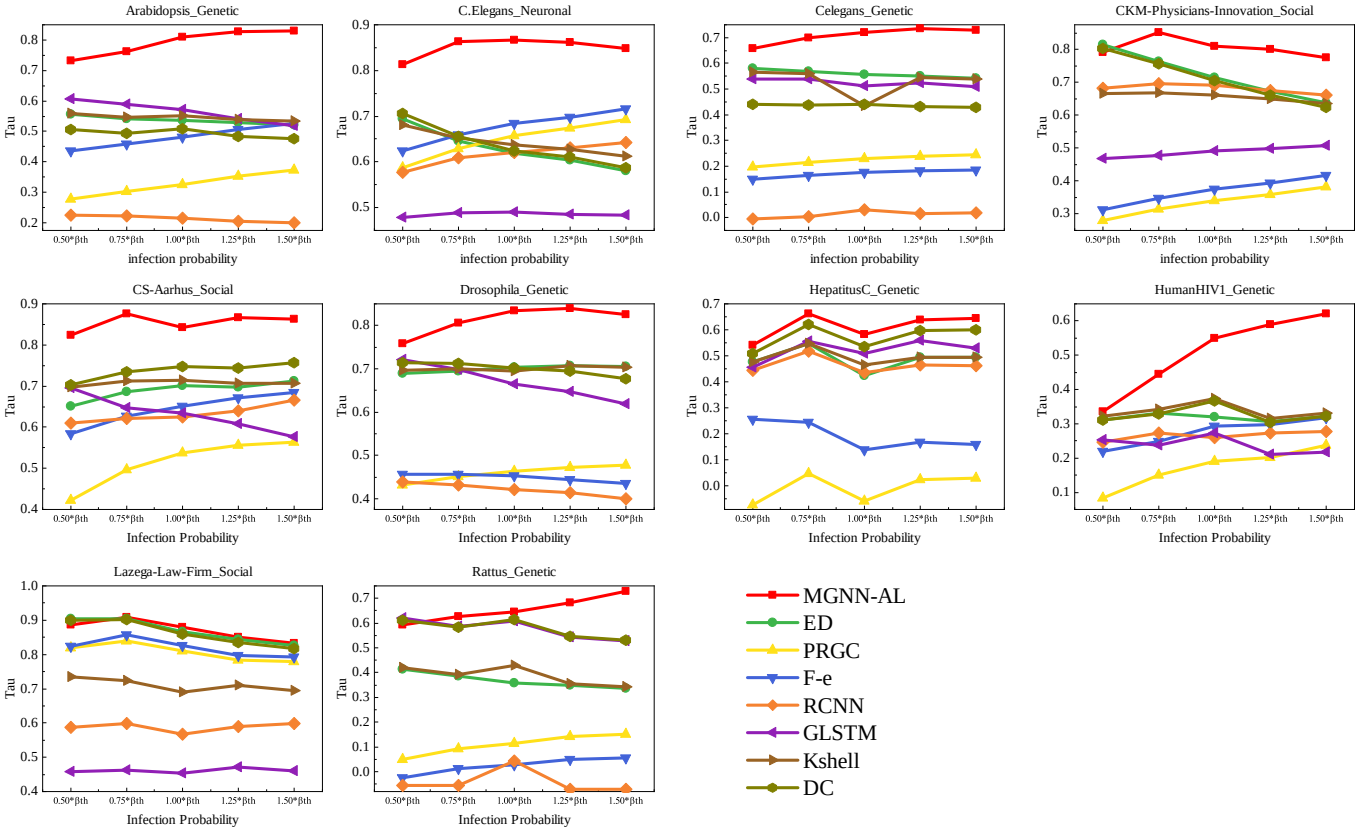


Fig. 2: Kendall's tau value Vs. infection probability (β) for different chosen datasets obtained for different algorithms.

In summary, MGNN-AL not only achieves superior predictive performance under a single infection probability setting, but also demonstrates high stability across different infection probability parameters, validating its practicality and robustness in the task of identifying important nodes in multilayer networks.

5.2 Comparison of Spreading Ability

This section presents the performance of different methods in identifying nodes with strong spreading capabilities, discussing the experimental results under two scenarios: varying the proportion of initial infected nodes and varying the infection probability.

5.2.1 *Infection Scale vs Initial Infected Nodes*

Figure. 3 illustrates the variation of the spreading scale with different numbers of initial infected nodes under the infection probability β_c. The horizontal axis represents the proportion of initially infected nodes, ranging from 0.01 to 0.1 with a step size of 0.01, while the vertical axis indicates the final spreading scale.

Overall, the MGNN-AL method demonstrates superior spreading effectiveness compared to the baseline methods in most networks. In networks such as *C.Elegans_Neuronal*, *Drosophila_Genetic*, *HepatitisC_Genetic*, *Lazega-Law-Firm_Social*, and *Rattus_Genetic*, the spreading scale achieved by MGNN-AL is significantly higher, showing a distinct advantage. In the *Arabidopsis_Genetic* and *C.Elegans_Genetic* networks, when the proportion of initial

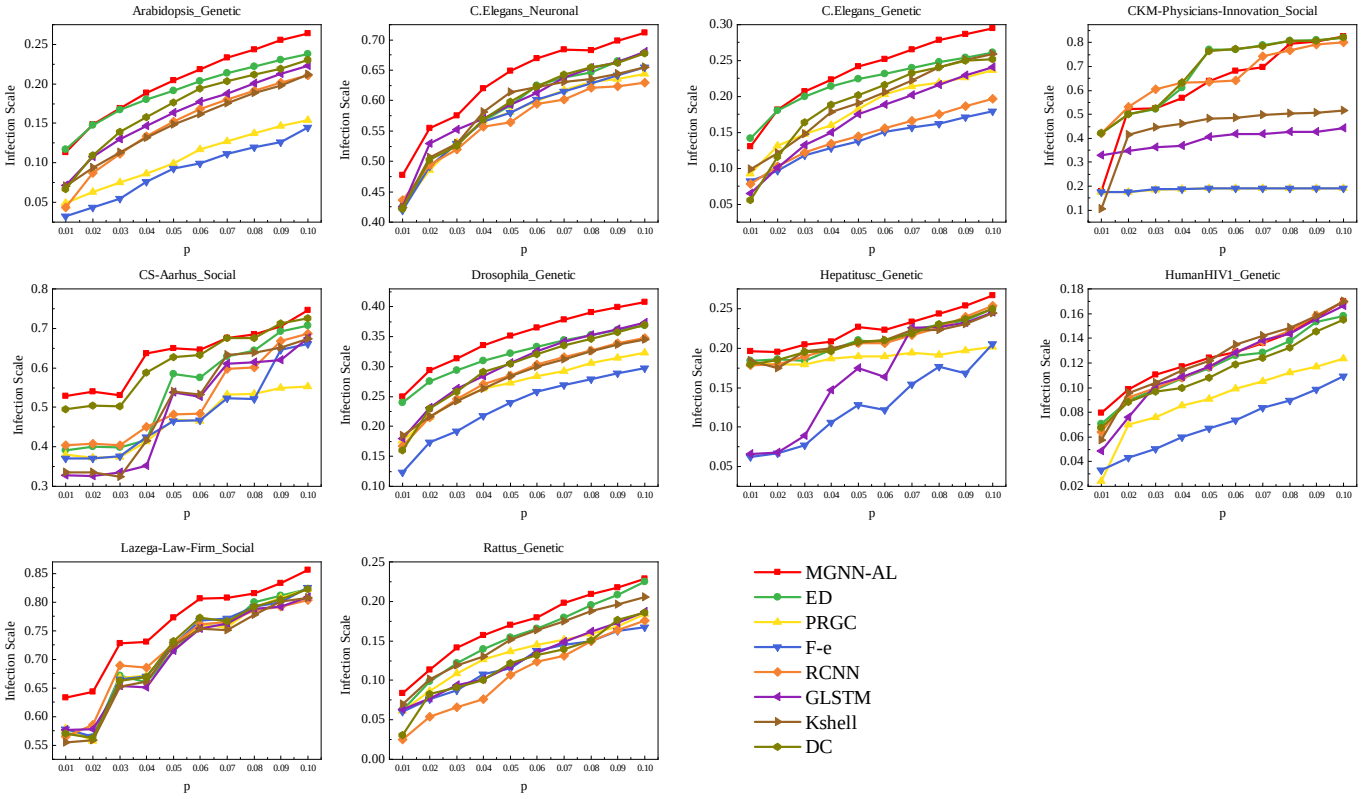


Fig. 3: Final infected scale Vs. initial infected nodes obtained by various methods on different datasets. The results are obtained using 1000 independent simulations of the SIR model with infection probability as β_c for the respective dataset.

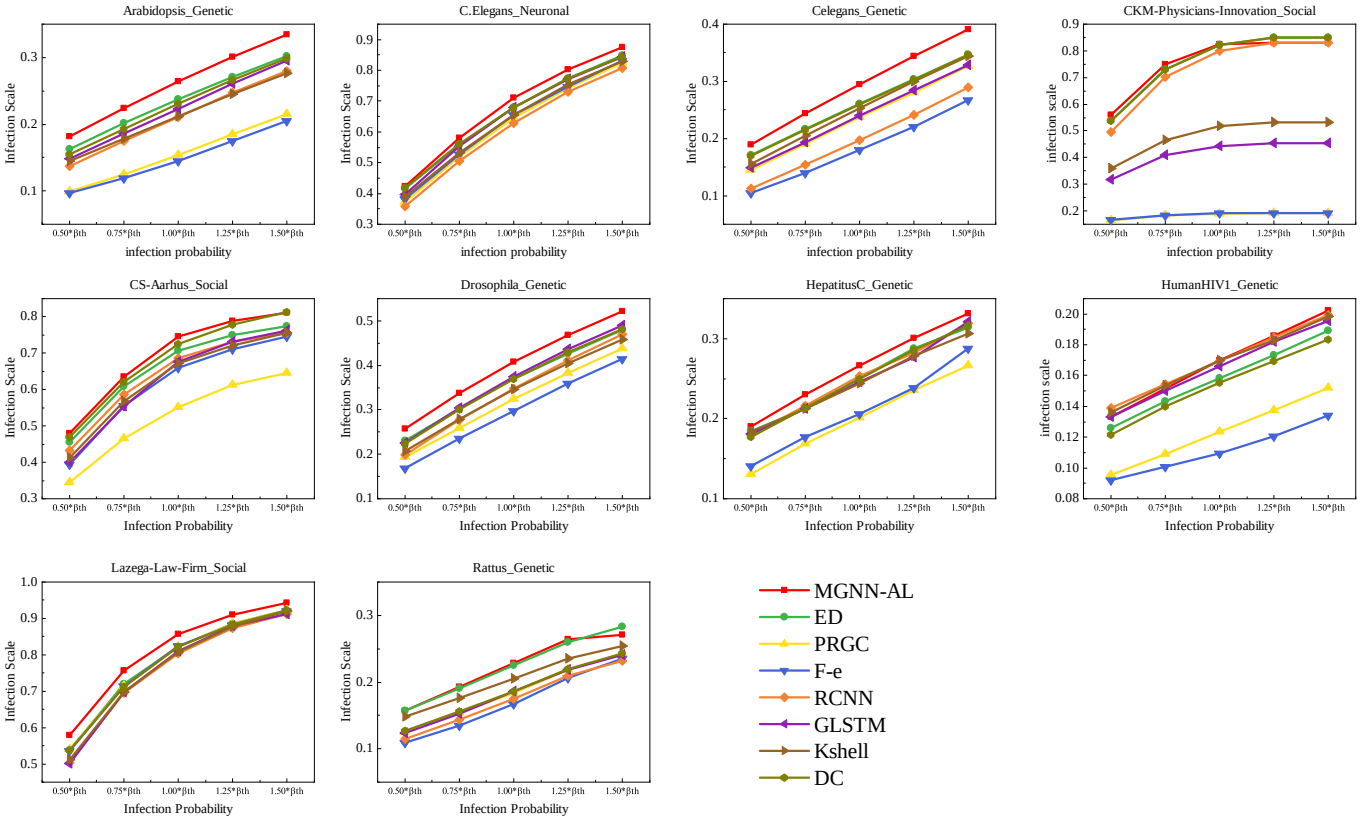
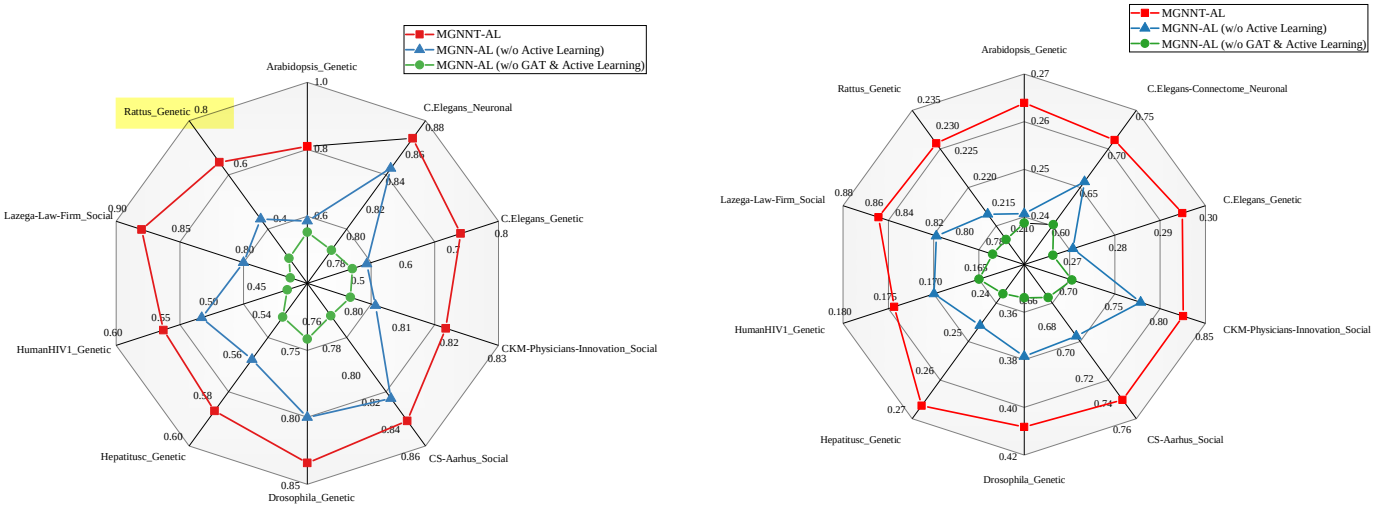


Fig. 4: Final infected scale Vs. infection probability values obtained by different methods for various chosen algorithms. The results are obtained using 1000 independent simulations of the SIR model. The number of initial infected nodes is taken as ten percent of the total nodes in each network.



(a) Ablation study of Kendall's tau coefficient.

(b) Ablation study of final infection scale.

Fig. 5: Ablation study of Kendall's tau coefficient and final infection scale for the MGNN-AL method across different datasets, with the infection probability set to $\beta = \beta_c$ and the initial infected nodes proportion set to $p = 0.1$

infected nodes $p \in [0.01, 0.03]$, the ED method exhibits a spreading performance comparable to that of MGNN-AL. However, as p increases, MGNN-AL gradually surpasses all comparison methods, achieving stronger spreading capability. In the *CKM-Physicians-Innovation_Social* network, MGNN-AL performs relatively worse at $p = 0.01$; nevertheless, as p increases, its spreading scale consistently grows, eventually matching the performance of the ED and DC methods, jointly achieving the best results. In the *HumanHIV1_Genetic* network, MGNN-AL maintains an advantage when $p \in [0.01, 0.05]$; however, as p further increases, its spreading scale becomes slightly inferior to that of the GLSTM and K-shell methods.

In summary, the experimental results indicate that the nodes identified by the MGNN-AL method possess stronger spreading capabilities, validating its effectiveness in detecting highly important nodes in multilayer complex networks.

5.2.2 *Infection Scale vs Infection Probability*

Figure. 4 illustrates the variation of the spreading scale under different infection probabilities when the top 10% of nodes are selected as initial infected nodes. The horizontal axis represents the infection probability, ranging from $0.5*\beta_c$ to $1.5 * \beta_c$ with a step size of 0.25, while the vertical axis indicates the final spreading scale. The results show that, as the infection probability increases, the spreading scale of all methods exhibits an upward trend. This demonstrates that infection probability is a key factor driving the diffusion process, and this pattern is universal across different network topologies, such as genetic, neural, and social networks.

In the majority of datasets (e.g., *Arabidopsis_Genetic*, *Celegans_Connectome*, *C.elegans_Genetic*, *Drosophila_Genetic*, *HepatitisC_Genetic*, and *Lazega-Law-Firm*), the spreading scale achieved by the proposed MGNN-AL method is significantly higher than that of other comparison methods. In the *HumanHIV1_Genetic* dataset, when the infection probability β ranges from $0.5 * \beta_c$ to $1.0 * \beta_c$, the performance of MGNN-AL is comparable to that of RCNN and K-shell. However, when $\beta \geq 1.25 * \beta_c$, MGNN-AL rapidly outperforms the others. In the *CKM-Physicians-Innovation* and *Rattus_Genetic* datasets, MGNN-AL performs best when β is between $0.5 * \beta_c$ and $1.0 * \beta_c$, but as β increases further, its performance declines to the second position.

In summary, these results indicate that under varying infection probabilities, the MGNN-AL method consistently identifies nodes with stronger spreading capabilities, achieving larger spreading scales and validating the stability of the proposed method.

5.3 Ablation Study

To validate the contribution of each module in the proposed MGNN-AL method, we conducted an ablation study. Specifically, we removed the active learning module and the layer attention module, respectively, to assess the impact of each component on accuracy and final spreading scale. The Kendall's tau coefficient was adopted to measure prediction accuracy, while the spreading scale was used as the indicator of propagation effectiveness. All experiments were carried out on the same datasets under identical settings.

As shown in Fig. 5 , the radar charts reveal that removing the active learning module leads to a significant drop in both Kendall's tau and spreading scale, confirming the critical role of active learning in enhancing model performance and generalization. Furthermore, the exclusion of the layer attention module causes a remarkable degradation in performance, demonstrating that the attention mechanism effectively captures the heterogeneous contributions of different layers to the final node importance.

In summary, the ablation study verifies that each component of the MGNN-AL method is indispensable for performance improvement. In particular, the graph attention mechanism and the active learning strategy substantially enhance both the effectiveness and the generalization ability of the proposed model.

6 Conclusion

In summary, this paper focuses on the problem of identifying important nodes in multilayer complex networks, formulating it as a regression task and proposing an efficient node influence prediction method, MGNN-AL. The proposed method integrates the graph neural networks with a layer attention mechanism, and further incorporates an active learning strategy to improve the model's generalization ability. Extensive experiments on multiple real-world multilayer network datasets demonstrate that MGNN-AL outperforms various baseline methods in terms of identification accuracy, while maintaining strong stability. The superior performance of MGNN-AL can be mainly attributed to two key design advantages. First, the graph neural network architecture fully exploits both local and global structural features of the network, effectively capturing complex topological dependencies among nodes. By introducing the layer attention mechanism, the model adaptively adjusts the contribution weights of different layers to the final node influence, enhancing its ability to characterize the structural complexity of multilayer networks. Second, the active learning strategy dynamically selects the most representative training samples, further strengthening the model's generalization capability across diverse network structures.

Despite the promising results, certain limitations remain. The current active learning strategy is relatively basic, as it only considers representative nodes within communities without fully leveraging latent structural information. Future work may focus on optimizing the active learning mechanism by incorporating graph structural entropy or graph sampling strategies to design more refined sample selection algorithms, improving labeling efficiency and model performance. Moreover, the interpretability of the model still requires further enhancement. Future research could explore more explainable architectures to improve the transparency and credibility of important node identification results.

References

[1] A. S. Mata, "Complex networks: A mini-review," *Braz. J. Phys.*, vol. 50, pp. 658–672, 2020.

[2] F. Battiston, V. Nicosia, and V. Latora, "Structural measures for multiplex networks," *Phys. Rev. E*, vol. 89, no. 3, p. 032804, 2014.

[3] M. A. Porter, "What is... a multilayer network," *Notices Amer. Math. Soc.*, vol. 65, no. 11, pp. 1419–1423, 2018.

[4] P. Bródka, K. Skibicki, et al., "Degree centrality in multi-layered social networks," in *Proc. Int. Conf. Computational Aspects of Social Networks (CASoN)*, 2011, pp. 237–242.

[5] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[6] L. C. Freeman, "Centrality in social networks: Conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1979.

[7] B. Ruhnau, "Eigenvector-centrality: A node-centrality?" *Social Networks*, vol. 22, no. 4, pp. 357–365, 2000.

[8] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Comput. Netw. ISDN Syst.*, vol. 30, no. 1–7, pp. 107–117, 1998.

[9] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 1024–1034.

[10] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech.*, vol. 2008, no. 10, p. P10008, 2008.

[11] J. E. Hirsch, "An index to quantify an individual's scientific research output," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 46, pp. 16569–16572, 2005.

[12] X. Wen, C. Tu, M. Wu, and X. Jiang, "Fast ranking nodes importance in complex networks based on LS-SVM method," *Physica A*, vol. 506, pp. 11–23, 2018.

[13] G. Zhao, P. Jia, C. Huang, A. Zhou, and Y. Fang, "A machine learning based framework for identifying important nodes in complex networks," *IEEE Access*, vol. 8, pp. 65462–65471, 2020.

[14] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, 2009.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[16] C. Fan, L. Zeng, Y. Sun, and Y.-Y. Liu, "Finding key players in complex networks through deep reinforcement learning," *Nat. Mach. Intell.*, vol. 2, pp. 317–324, 2020.

[17] M. Zhang, X. Wang, L. Jin, M. Song, and Z. Li, "A new approach for evaluating node importance in complex networks via deep learning methods," *Neurocomputing*, vol. 497, pp. 13–27, 2022.

[18] L. Chen, et al., "Identifying important nodes in complex networks via Transformer," *Inf. Process. Manag.*, vol. 61, art. no. 103775, 2024.

[19] S. Kumar, A. Mallik, and B. S. Panda, "Influence maximization in social networks using transfer learning via graph-based LSTM," *Expert Syst. Appl.*, vol. 212, p. 118770, 2023.

[20] E. Y. Yu, Y. P. Wang, Y. Fu, D. B. Chen, and M. Xie, "Identifying critical nodes in complex networks via graph convolutional networks," *Knowl.-Based Syst.*, vol. 198, p. 105893, 2020.

[21] A. Solé-Ribalta, M. De Domenico, N. E. Kouvaris, A. Díaz-Guilera, S. Gómez, and A. Arenas, "Node and layer eigenvector centralities for multiplex networks," *Phys. Rev. E*, vol. 88, no. 3, p. 032807, 2013.

[22] L. Lv, T. Zhang, P. Hu, D. Bardou, S. Niu, Z. Zheng, G. Yu, and H. Wu, "An improved gravity centrality for finding important nodes in multi-layer networks based on multi-PageRank," *Expert Syst. Appl.*, vol. 238, art. no. 122171, 2023.

[23] D. Wang, F. Tian, and D. Wei, "A new centrality ranking method for multilayer networks," *J. Comput. Sci.*, vol. —, art. no. 101924, Dec. 2022.

[24] C. Ni, J. Yang, Z. Pang, and Y. Gong, "Seeding strategy based on weighted gravity centrality in multiplex networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 1, pp. 331–345, 2023.

[25] H. Wu, R. G. James, et al., "Correlated structural evolution within multiplex networks," *J. Complex Netw.*, vol. 8, no. 2, 2020.

[26] S. Caschili, F. R. Medda, and A. Wilson, "An interdependent multilayer model: Resilience of international networks," *Netw. Spat. Econ.*, vol. 15, no. 2, pp. 313–335, 2015.

[27] P. Velicković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[28] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir, "A model of Internet topology using k-shell decomposition," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 27, pp. 11150–11154, 2007.

[29] M. Magnani, B. Micenková, and L. Rossi, "Combinatorial analysis of multiple networks," *arXiv preprint arXiv:1303.4986*, 2013.

[30] T. A. B. Snijders, P. E. Pattison, G. L. Robins, and M. S. Handcock, "New specifications for exponential random graph models," *Sociol. Methodol.*, vol. 36, no. 1, pp. 99–153, 2006.

[31] M. De Domenico, M. A. Porter, and A. Arenas, "MuxViz: A tool for multilayer analysis and visualization of networks," *J. Complex Netw.*, vol. 3, no. 2, pp. 159–176, 2015.

[32] J. Coleman, E. Katz, and H. Menzel, "The diffusion of an innovation among physicians," *Sociometry*, vol. 20, no. 4, pp. 253–270, 1957.

[33] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, "Structural reducibility of multilayer networks," *Nature Commun.*, vol. 6, p. 6864, 2015.

[34] H. W. Hethcote, "The mathematics of infectious diseases," *SIAM Rev.*, vol. 42, no. 4, pp. 599–653, 2000.

[35] C. Castellano and R. Pastor-Satorras, "Thresholds for epidemic spreading in networks," *Phys. Rev. Lett.*, vol. 105, no. 21, p. 218701, 2010.

[36] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1–2, pp. 81–93, 1938.