Proceedings of the 5th International Conference on Advanced Science & Engineering ICOASE 2025, University of Zakho and Duhok Polytechnic University, 24th – 25th September 2025

(Special issue for Passer Journal of Basic and Applied Sciences)
Paper No. 10

A Hybrid Deep Learning Approach for Influential Nodes Identification in Complex Networks

Mohammed A. Ramadhan [1], Abdulhakeem O. Mohammed [1]

[1] Department of Computer Science, College of Science, University of Zakho, Zakho 42002, Kurdistan Region, Iraq.
Email: mohammed.abdullah@staff.uoz.edu.krd , a.mohammed@uoz.edu.krd

Abstract
The influential nodes in complex network are the key of high effective information spreading. Several techniques have been developed for the discovery of such nodes, including centrality-based approaches, machine learning-based approaches, and deep learning-based approaches. This paper proposes CNNG, a novel hybrid deep learning model combining Convolutional Neural Networks (CNN) and Graph Attention Networks (GAT) for predicting node influence. CNNG processes each node's local subgraph through parallel CNN and GAT components to capture both structural patterns and relational dependencies. The model is trained in a supervised regression setting, using node influence scores generated via SIR simulations. Experiments on twelve real networks show that CNNG achieves the best average Kendall's Tau coefficient of 0.7510, outperforming the second-best method by 2.85% as well as the highest average Monotonicity Index of 0.9954 and the highest average Jaccard Similarity of 0.7298 over ten networks. Further, CNNG provides these results in a reasonable amount of computing time, which demonstrates the practicability, efficiency and generality of CNNG for the complex network analysis.

Keywords: Complex networks, Influential nodes, Deep learning, SIR model.

1 Introduction

Identifying influential nodes in complex networks is a critical task with a wide range of applications such as biological networks [1], social networks [2], community networks [3], [4], etc. In such networks some nodes may have a preferential role in influencing the information flow or the state of the system, making it important to identify them accurately. Discovering influential nodes in complex networks has been widely studied from traditional centrality-based to the machine learning based methods, recently advanced to the deep learning based methods. Centrality-based approaches attempt to measure the importance of a node by its topological features such as degree [5], closeness [6], and betweenness [7] rank nodes by the local connections, shortest paths, and control over the flow of the information, respectively. Other approaches like K-core [8] split the network into a series of layers, from which a series of core nodes are obtained which are supposed to be more influential, while K-shell iteration factor [9] improves the previous approach by including the dynamic of node removing in the decomposition process. Similarly, voteRank [10] uses an iterative voting mechanism to select multiple key spreaders, other metrics like H-index [11] focus on local neighborhood influence, V-community [12] measures spreading influence by counting the number of communities connected to each node. Another example of centrality-based methods is mixed degree decomposition [13], which combines global and local measurements to improve the accuracy for identifying influential nodes. Recently, Mohammed [14] proposed the Layered Clustering Degree (LCD) method, which hierarchically layers nodes, then clusters them locally, and ranks them based on global degree. While most of these methods are simple and fast to compute, they often struggle or fail to capture the complexity of real-world influence, which can lead to inaccurate identification of the most influential nodes.

To deal with these drawbacks, machine learning-based methods have been developed to transform influential node detection problem into a classification or a regression problem. In such methods, all nodes are characterized by feature vector values, typically computed based on their structural characteristics such as centrality scores, and then a model is trained to predict nodes' influence. For example, Wen et al. [15] presented a hybrid model that integrating Analytic Hierarchy Process (AHP) and the Least-Squares Support Vector Machine (LS-SVM) to identify influential nodes, which adopt AHP to calculate the importance of nodes through multiple centrality measure like closeness and betweenness centrality. Similarly, Zhao et al. [16] considered the problem of identifying influential nodes as a

classification problem and described each of the nodes of the network by using a constructed feature vector based on nine centrality measures. They then applied seven machine learning models to make predictions of influential nodes. In another study, Rezaei et al. [17] used a collective feature engineering framework for identifying influential nodes. The approach proposed primarily aimed to extract node-level features, such as connectivity, degree, and extended coreness to capture structural influence. They used Support Vector Regression (SVR) with a Radial Basis Function (RBF) kernel as their prediction model on the described features to predict the node influence. Additionally, Hajarathaiah et al. [18] proposed a framework to generate feature vector for every node, they used several centrality measures such as: degree, clustering coefficient, katz centrality. They subsequently employed multiple machine learning approaches, including SVM, DT, and MLP, to learn the relationships between nodes and predict vital nodes.

Recently, deep learning based approach has emerged as a promising direction for identifying influential nodes by learning directly from raw graph structures. Deep learning based approach reduce the dependency on manual-crafted features and instead use neural architectures to extract meaningful patterns from the network topology and node feature. For instance, a study by Yu et al. [19] introduced a deep learning based model called RCNN, they reformulate the influential node identification problem into a regression task, and created fixed-size subgraphs around each node and built an adjacency matrix for each subgraph, which was used as input for their CNN-based model to predict node influence. Another deep learning approach, namely InfGCN, was proposed by Zhao et al. [20] to identify influential nodes in complex networks. InfGCN was based on Graph Convolutional Networks (GCNs) and formulates the task as a classification problem. The authors used centrality measures as node features and combined them with the symmetric normalized Laplacian matrix derived from neighborhood subgraphs. These subgraphs were extracted using a Breadth-First Search (BFS) strategy. The resulting information was then fed into the InfGCN model to classify influential nodes. Another study by Ou et al. [21] presented M-RCNN, a multi-channel RCNN framework. Their model constructs subgraphs through BFS and encodes micro, community, and macro-level structural features, allowing for multiscale learning and better prediction. Similarly, Bhattacharya et al. [22] identified the influential nodes in social networks as classification task, first they applied BFS on each node for creating subgraphs from original network. Then for each subgraph an adjacency matrix based on symmetric normalized laplacian, and node feature matrix based on DC, BC, k-shell, and density are created. Finally, they employed GCN for detecting the influential nodes in the network. Kumar et al. [23] proposed SGNN, a graph embedding and GNN-based framework that uses struc2vec embeddings and a GNN regressor to predict node influence for influence maximization tasks. Complex hybrid models have also been introduced to capture both structural and relational patterns, such as Zhang et al. [24] introduced a hybrid model named CGNN, which denote Convolutional Neural Networks (CNN) combined with Graph Neural Networks (GNN). Another work by Wu et al. [25] proposed a GCN method by combining it with the regular equivalence-based similarities, in order to increase the accuracy of the influential nodes identification. Recently, Xiong et al. [26] proposed AGNN, an autoencoder GNN fusion framework. The model employs a GCN based autoencoder to generate latent node representations, which are then passed to a rank prediction module for identifying vital nodes in complex networks. Chen et al. [27] introduced CNT, a Transformer-based model that builds dynamic input sequences from nodes and their neighbors, processed through Transformer encoders to identify influential nodes in complex networks. Rashid and Bhat [28] proposed OlapGN, a multi-layered GCN model. It combines overlapping community detection with graph convolutional layers, and then uses classic centrality features to identify influential nodes in social networks.

The work presented in this research falls under the deep learning–based approaches and handle the problem of identifying influential nodes in complex networks as a supervised regression task. A new hybrid deep learning model, named CNNG, is proposed for high ranking accuracy and stable generalization performance. The proposed model combines the strengths of two complementary components: a Convolutional Neural Network (CNN) that captures local structural patterns from transformed adjacency matrices, and a Graph Attention Network (GAT) that learns relational representations through adaptive neighborhood weighting. This parallel design allows CNNG to analyze both the network structure and influence relationships in local subgraphs, without relying on manually created features or processing the whole network.

The proposed CNNG model is evaluated for ranking accuracy, computational efficiency, and generalization. Experimental results demonstrate that it consistently outperforms traditional centrality methods and other deep learning baselines. The main contributions of this work are summarized as follows:

- We introduce CNNG, a parallel hybrid deep learning framework that integrates CNN and GAT to simultaneously capture structural patterns and relational dependencies for influential node identification.

- Unlike existing deep learning-based methods that rely on a single learning stream or sequential layer stacking, CNNG adopts a parallel learning strategy, enabling complementary feature extraction from local subgraphs.
- The CNNG model is trained on a synthetic Barabási Albert (BA) network and evaluated on twelve real world networks, demonstrating strong generalization across diverse network structures.
- Extensive experiments results demonstrate that CNNG consistently outperforms traditional centrality-based methods and representative deep learning baseline, achieving an average Kendall's Tau correlation of 0.7510 (2.85% higher than the second best method), an average Monotonicity Index of 0.9954, and an average Jaccard Similarity of 0.7298, while maintaining a reasonable computational cost.

The rest of this paper is organized as follows: Section 2 presents proposed hybrid model and its methodology. In section 3, the evaluation metrics are introduced. Experimental results and discussions are given in Section 4, and Section 5 concludes by summarizing the study and future direction.

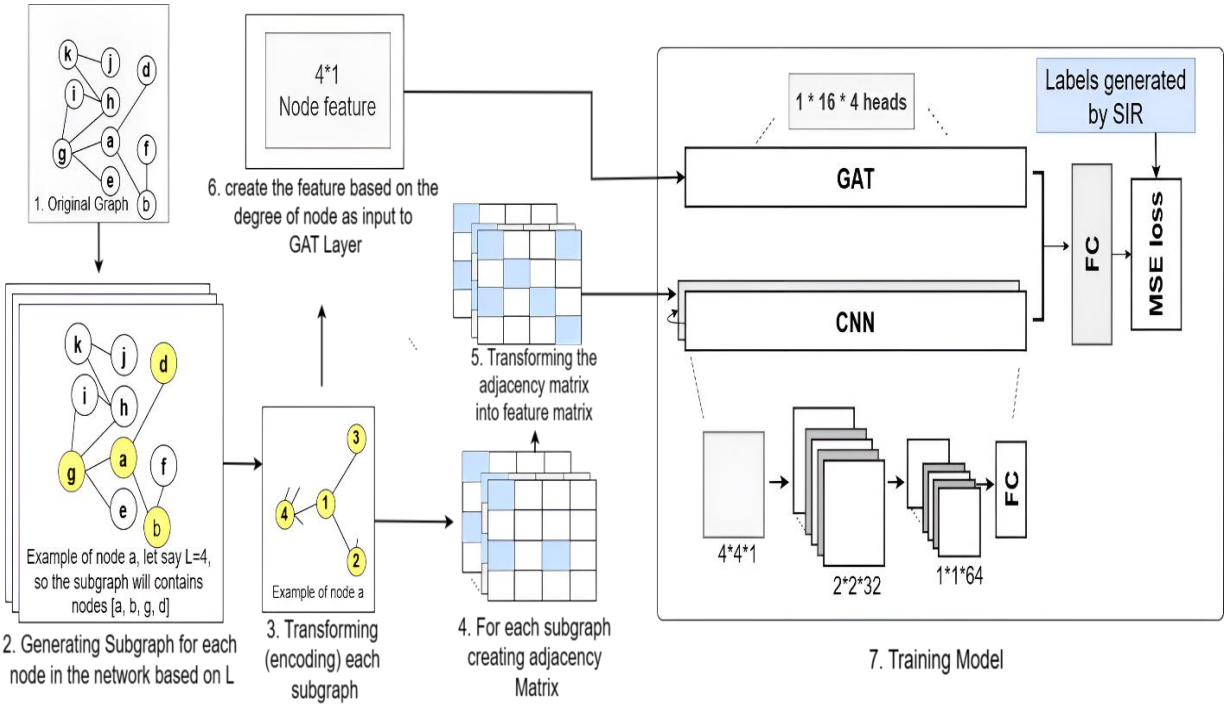


Figure 1: The CNNG model computes node influence as two parallel branches processing each node local subgraph (for example, L = 4). The CNN branch extracts structural patterns from a degree-based adjacency matrix, while the GAT branch utilizes node degree features to capture relational dependencies. The two outputs are fused and fed into a fully connected layer (FC) to output an influence score.

2 Methods and Materials

This paper propose a hybrid deep learning framework using Graph Attention Networks (GAT) [29] and Convolutional Neural Networks (CNN) [19] for ranking influential nodes in complex networks. The main concept is to treat the influence estimation problem as a supervised regression task, where the label for each node is its average spreading ability, obtained from SIR simulations. The overall architecture of the proposed hybrid model is described in Figure 1. It consists of a CNN layer and a GAT layer that capture spatial topological features and relational dependencies between nodes in a local subgraph, and their outputs are subsequently integrated to predict the influence score of each node.

To provide intuition into the influence identification process, the two components of the proposed CNNG framework play complementary roles. The CNN branch captures local structural patterns within each node's extracted subgraph, such as connectivity distribution and neighborhood configuration. In contrast, the GAT branch models relational importance by adaptively weighting neighboring nodes based on their features and structural relevance. By fusing these structural and relational representations, CNNG identifies influential nodes based on both their topological position and the strength of their connections.

2. 1 Subgraph construction and feature engineering

For each node $v \in V$ in the network, a fixed-size subgraph is constructed, denoting the subgraph with $G_v = (V_v, E_v)$ containing node v and its $L-1$ nearest neighbors, in order to perform localized learning and scalability over a large network. In the selection process, nearest neighbors to v are selected (hop_1 neighbors, hop_2 neighbors, etc.), and in the condition where more than one node have the same hop, nodes with the higher degree will be selected. The resulting subgraph is then used to build two parallel inputs for CNNG model:

Adjacency Matrix. The input to the CNN branch is a transformed matrix of the corresponding subgraph's adjacency matrix A_v, which denote as B_v. This matrix has dimensions $L * L$, where L is the number of nodes in the extracted subgraph, consisting of the target node and its $L-1$ nearest neighbors [19]. In this case, node degree information from the original graph is embedded into the adjacency structure. The transformation is defined as follows:

$$B_v[i,j] = \begin{cases} A_v[0,j] \cdot k_j & \text{if } i = 0, j = 1,2,3,..,L-1 \\ A_v[i,0] \cdot k_i & \text{if } j = 0, i = 1,2,3,..,L-1 \\ k_i \cdot A_v[i,j] & \text{if } i = j = 0,1,2,3,..,L-1 \\ A_v[i,j] & \text{otherwise} \end{cases} \quad (1)$$

Where, A_v is the binary adjacency matrix of the subgraph, and k_i represents the degree of node in the original network. The resulting matrix B_v is then treated as input for CNN to extract high-level structural patterns from the local topology.

Node Feature Matrix. The GAT layer takes in a node feature matrix X_v in parallel with the CNN, with one scalar input feature per node in the subgraph. The features are organized in a $L * 1$ matrix with L equal to the number of nodes in the subgraph. The degree of a node in the original network is represented by each entry in the matrix. The feature for node $u_i \in V_v$ is defined as:

$$X_v[i] = \deg_G(u_i) \quad (2)$$

The attention mechanism adaptively assigns weights to neighboring nodes depending on their features and structural positions, enabling the model to focus more on topologically relevant nodes. The nodes in each subgraph are ordered deterministically to ensure that the input is consistent between samples. The target node v is placed at index 0, followed by its selected neighbors based on nearest neighbors and degree. If the subgraph contains fewer than L nodes such as disconnected components, zero-padding is applied to both B_v and X_v to preserve fixed dimensions.

2. 2 SIR Model: Label generation

After constructing the subgraph inputs described in Section 2.1, the Susceptible–Infected–Recovered (SIR) model [30] is employed to generate ground-truth influence labels for each node. The SIR model is a widely used epidemic diffusion model for evaluating spreading processes in complex networks, and it has been extensively adopted in influential node identification studies [19]–[22], [24]–[27]. In the SIR model, each node can be in one of three states: susceptible (S), infected (I), or recovered (R). Initially, all nodes are in the susceptible state, except for a single node, which is selected as the initial infected source. During the diffusion process, infected nodes attempt to transmit the infection to their susceptible neighbors with a fixed infection probability μ, and then transition to the recovered state with a recovery probability β. Once a node reaches the recovered state, it no longer participates in the spreading process. The simulation terminates when no infected nodes remain.

To ensure effective diffusion, the infection probability μ is selected relative to the epidemic threshold μ_c, which represents the minimum infection rate required for spreading in the network. In this study, diffusion simulations are conducted for values of μ/μ_c ranging from 1.0 to 2.0, allowing node influence to be evaluated under different spreading conditions.

The spreading influence of a node v_i is defined as the total number of nodes that become infected and subsequently recover when the diffusion process originates from v_i. Due to the stochastic nature of the SIR process, the simulation is repeated 1000 times for each node, and the average number of affected nodes across these runs is computed. This average value is assigned as the influence label for node v_i and is used to train the proposed CNNG model.

2. 2 Training Model

The proposed CNNG model is trained on an artificial Barabási–Albert (BA) network [31] consisting of n nodes with an average degree of k. After preparing and constructing the subgraph inputs and generating labels using the SIR model, the model is trained to predict node influence through a combination of convolutional and attention-based layers.

The CNN takes as input the degree-weighted adjacency matrix B_v[1], this matrix passed through 2 convolutional layers. The first convolutional (conv2d) layer takes 1 input channel, gives 32 output channels, with kernel size of 5, stride 1, and padding of 2. The 32 input channels and 64 output channels are used in the same kernel size, stride, and padding of the second convolutional (conv2d) layer. A ReLU activation function is applied after each convolution and then add max pooling layer with kernel size of 2. The last feature map is then flattened and processed by a fully connected (FC) layer of input size 64×(L/4)×(L/4) and output size 1.

On the other hand, the GAT component takes the node feature matrix X_v (explained in Section 2.1). It uses 1 input channel and 16 output channels per head with 4 attention heads and a single attention layer. It produces a node embedding of size 64 by concatenating the outputs of all attention heads and averaging them across all the nodes to create a single vector of length 64. The output of CNN and GAT are combined and passed to a fully connected layer to obtain the final prediction. CNNG model is trained with Adam optimizer with a learning rate of 0.001 for 800 epochs. All hyperparameters were finalized based on extensive experiments optimizing performance across network types. Finally, the MSE loss function is used to measure the difference of predicted and ground truth values generated form SIR model.

3 Evaluation Metrics

This paper evaluates how well the proposed model ranks node influence based on three metrics: Jaccard Similarity [32], Kendall's Tau Correlation [33] and Monotonicity Relation [34]. These metrics reflect various properties of the predicted ranking like overlap with the ground truth, ordinal consistency and rank uniqueness. In this paper, all comparisons are on the basis of ground truth rankings that are obtained from SIR model.

Jaccard similarity. Jaccard similarity measures the overlap between two sample sets. Let A and B be the sets of nodes that fall within the top-k nodes of the predicted and ground truth rankings respectively, then the jaccard similarity calculated as:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

The score is between 0 and 1, where larger values indicate that the predicted top nodes exhibit strong agreement with their corresponding ground truth top nodes.

Kendall's Tau (τ). Kendall's Tau is a rank-based measure of correlation that indicates the degree to which two rankings agree. It works on the basis of concordant and discordant pairs of elements in the ranks. A pair of items are concordant when the rankings agree on their order, and discordant otherwise. The metric is defined as:

$$\tau = \frac{C - D}{\frac{1}{2}n(n-1)} \tag{4}$$

Where C and D are the concordant and discordant pairs respectively, and n the ranked nodes. They range from -1 (perfect inverse correlation) to $+1$ (perfect agreement).

Monotonicity Index (MI). This metric is used to evaluate how unique the ranks of the predicted list are, which is calculated as:

$$MI = \left(1 - \frac{\sum_{r \in R} N_r \cdot (N_r - 1)}{N_u \cdot (N_u - 1)}\right)^2 \tag{5}$$

Where R is the ranks assigned, N_r is the number of nodes which has the rank r, and N_u is the total number of unique ranks in the list. MI give a value from 0 to 1, where value close to 1 means most nodes were assigned unique ranks.

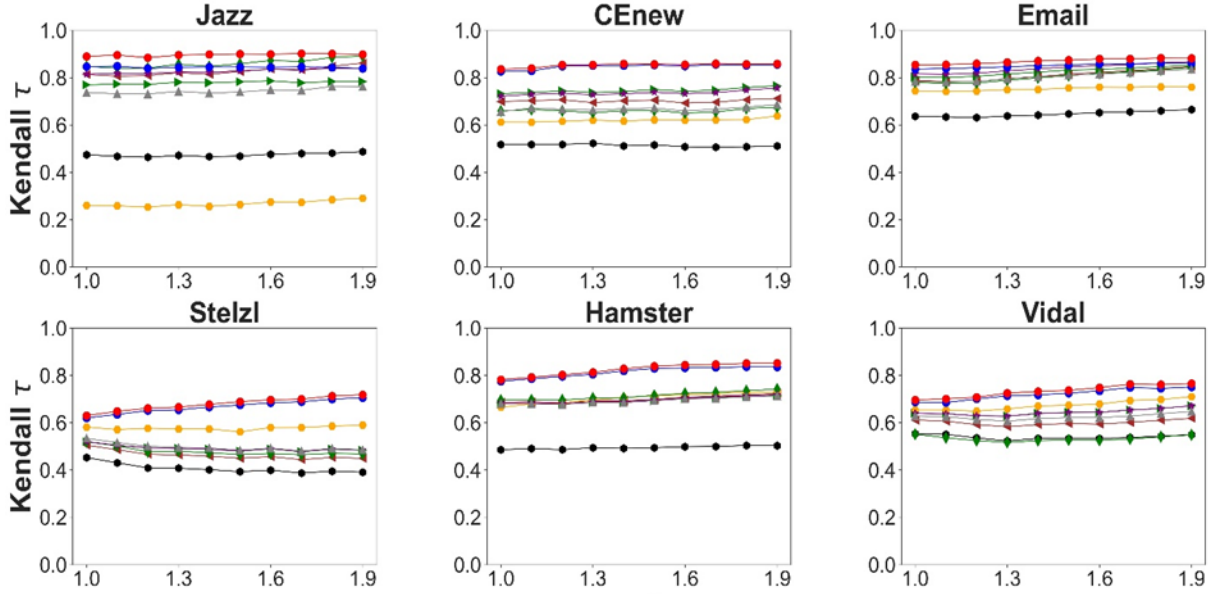
4 Results and Discussions

To evaluate the effectiveness of the proposed CNNG model in identifying influential nodes, experiments were conducted using twelve real-world network datasets. The performance of CNNG was compared against a several set of baseline methods, including Betweenness Centrality (BC) [7], Degree Centrality (DC) [5], H-index (HI) [11], K-core [8], V-Community (VC) [12], K-shell Iteration Factor (KSIF) [9], Mixed Degree Decomposition (MDD) [13], and the deep learning-based model RCNN [19].

Table 1: The basic topological characteristics of twelve real world network datasets.

Networks	Nodes	Edges	$\langle K \rangle$	K_{max}	μ_c	μ
Jazz	198	2742	27.7	100	0.026	0.039
CEnew	453	2025	8.94	237	0.025	0.037
Email	1133	4351	9.622	71	0.056	0.084
Stelzl	1702	3155	7.27	189	0.063	0.094
Hamster	2426	16630	13.71	273	0.024	0.036
Vidal	3023	6149	4.1	129	0.069	0.103
Facebook	4039	88234	43.691	1045	0.009	0.014
EPA	4271	8909	4.172	175	0.036	0.054
Router	5022	6258	2.492	106	0.078	0.117
GrQ	5242	14484	5.52	81	0.063	0.094
LastFM	7624	27806	7.29	216	0.040	0.060
PGP	10680	24316	4.554	205	0.055	0.083

Dataset. The proposed CNNG model is trained on a synthetic network generated using the famous BA model, as explained in Section 2.3. Training was explored by varying the number of nodes {1000, 2000, 3000, 4000, 8000} and average degree {4, 10, 20} to find an effective configuration. The experimental results show that the proposed CNNG model can achieve the best performance when experimental training is applied to a network with 1000 nodes and the average degree is 4, which is used as the default setting in the rest of the study. The infection probabilities used in SIR simulations are set to $\mu = 1.5\mu_c$ where μ_c is the SIR model spreading threshold which defined as: $\mu_c = \langle k \rangle / (\langle k^2 \rangle - \langle k \rangle)$, recover probability is set to β =1, resulting in an infection threshold. These settings are applied consistently during the generation of ground-truth influence labels for training. In order to assess the performance of the model, tests were conducted on twelve real-world network datasets, including Jazz [35], CEnew [36], Email [37], Stelzl [38], Hamster [39], Vidal [40], Facebook [41], EPA [42], Router [43], GrQ [44], LastFM [45], and PGP [46]. These networks are summarized in Table 1 with their respective topological characteristics (number of nodes, edges, average degree $\langle K \rangle$, maximum degree (K_{max}), infection probability ($\mu * 1.5$), and infection probability threshold (μ_c)). Same preprocessing pipeline as the training data is applied to each test network dataset.



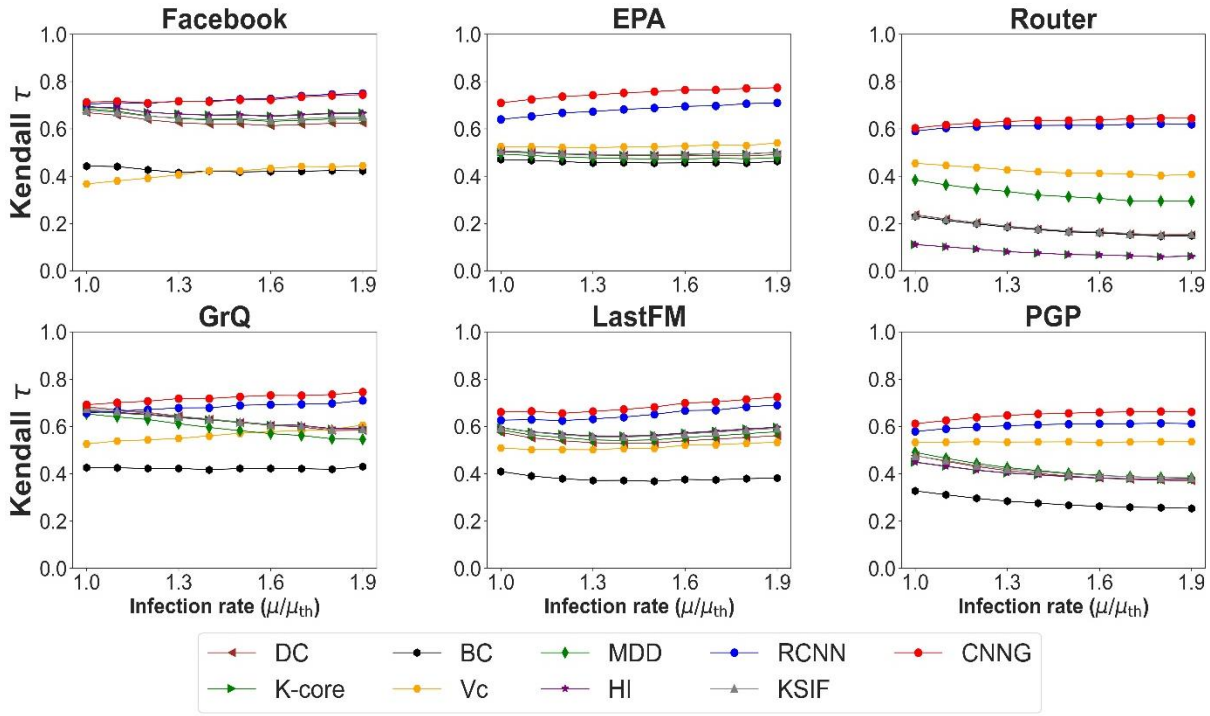


Figure 2: Kendall's τ between model rankings and ground-truth generated by SIR model with varying infection rate.

Kendall's Tau result. The Kendall's Tau results are presented in Table 2, which reports the average rank correlation values under varying infection rates μ ranging from 1.0 to 2.0. Kendall's Tau metric captures the agreement between the predicted node rankings and the ground truth rankings derived from SIR simulations. The proposed CNNG model consistently demonstrates strong correlation across diverse network structures. For instance, CNNG achieves an average τ score of 0.8700 on the Email network, surpassing not only the traditional methods but also the RCNN (0.8467). In addition, on both Jazz and Hamster, it gets scores of 0.8975 and 0.8250 respectively, which confirms that its ability to preserve rank consistency with small networks. In the Facebook network, CNNG achieves an average τ of 0.7239, ranking as the second best method, very close to RCNN (0.7244). This indicates that CNNG generalizes well even on densely connected networks, maintaining a high level of ranking agreement with the SIR ground truth. CNNG also performs well on large networks like LastFM and PGP with τ scores equal to 0.6843 and 0.6495, respectively. Moreover, as illustrated in Figure 2, CNNG achieves the highest Kendall's Tau correlation among all infection rates within the range of 1.0 and 2.0, confirming its robustness under diverse infection rate.

Table 2: Average results of Kendall's tau correlation of rankings generated by proposed model and different methods.

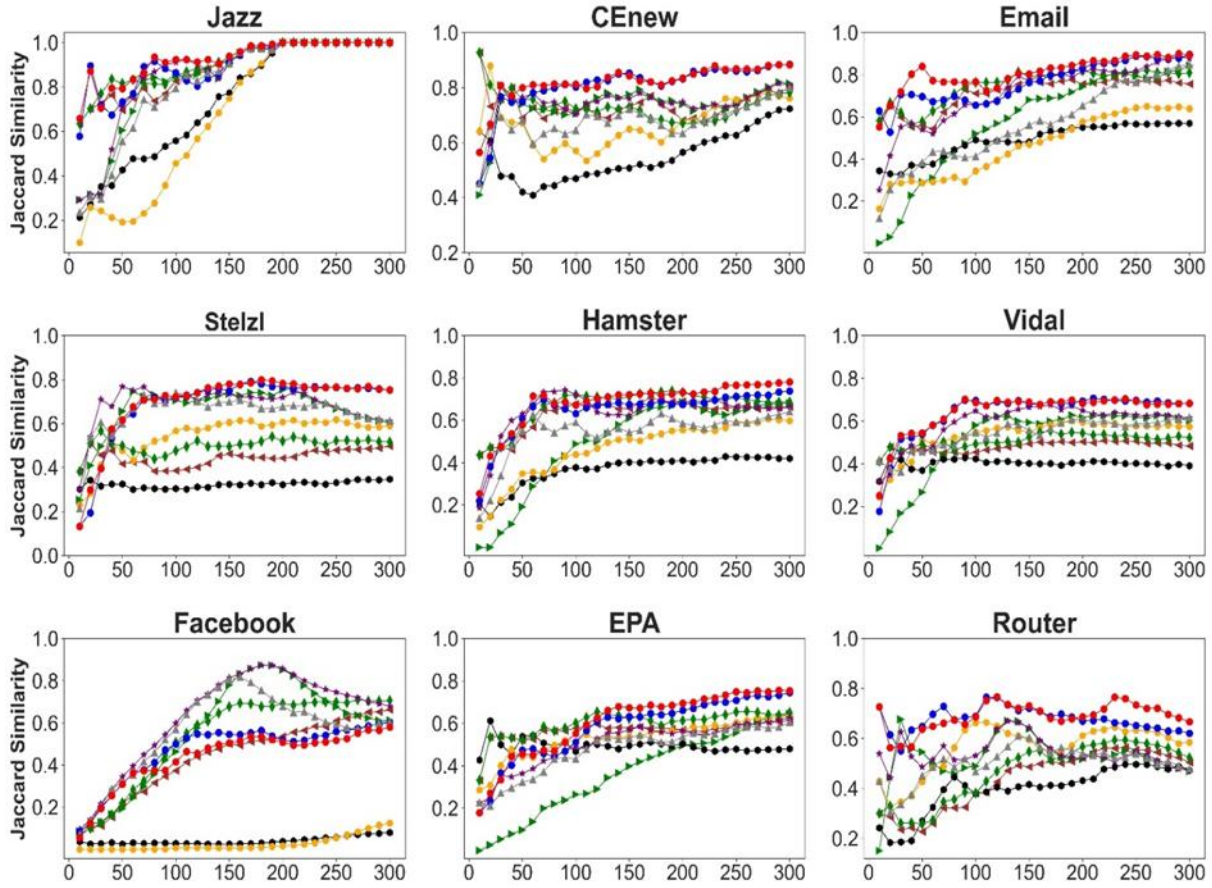
Networks	BC	DC	HI	Kcore	Vc	KSIF	MDD	RCNN	CNNG
Jazz	0.4733	0.8268	0.8268	0.7787	0.2748	0.7435	0.8607	0.8435	**0.8975**
CEnew	0.5135	0.7015	0.7354	0.7453	0.5705	0.6685	0.6606	0.8461	**0.8514**
Email	0.6460	0.8106	0.8393	0.8252	0.7556	0.8057	0.8048	0.8467	**0.8700**
Stelzl	0.4065	0.463	0.4919	0.491	0.5772	0.497	0.4786	0.6432	**0.6788**
Hamster	0.4945	0.8203	0.6962	0.7001	0.7001	0.6927	0.7140	0.8142	**0.8250**
Vidal	0.5390	0.6003	0.6442	0.6445	0.6803	0.6246	0.5322	0.7188	**0.7344**
Facebook	0.4250	0.6310	0.6671	0.6678	0.4126	0.6487	0.6487	**0.7244**	0.7239
EPA	0.4598	0.4918	0.4953	0.4939	0.5269	0.4962	0.4785	0.6832	**0.7441**
Router	0.1770	0.1810	0.0779	0.0776	0.4117	0.1819	0.3251	0.6126	**0.6329**
GrQ	0.4227	0.6271	0.6264	0.6237	0.5666	0.6276	0.5936	0.6848	**0.7204**
LastFM	0.3801	0.5454	0.5749	0.5764	0.5257	0.5725	0.5589	0.6507	**0.6843**
PGP	0.2794	0.4065	0.3998	0.3989	0.5341	0.4143	0.4188	0.6029	**0.6495**

Jaccard similarity results. The average Jaccard Similarity from top-10 up to top-300 nodes is shown in Table 3. This measure is used in this study to indicates the overlap between the top influencers predicted and the ground truth

ranking derived through the SIR simulations. The CNNG model outperforms others on almost all network datasets. It achieves a top Jaccard score of 0.9271 on Jazz, with 0.8207 on CEnew, 0.8092 on Email, and 0.7707 on PGP. CNNG also gives good performance on Hamster and Stelzl (with a score of 0.6861 and 0.5774, respectively) as well, outperforming traditional methods and RCNN model. In the GrQc network, CNNG reaches a score of 0.5565, being the second best performance achieved after RCNN model, on the dens networks like Facebook, CNNG achieves a score of 0.4387, yet remains superior against traditional measures. To further illustrate the model's ability to maintain high consistency across different top-k thresholds, Figure 3 presents a detailed comparison of Jaccard Similarity trends across different method.

Table 3: Average results of Jaccard Similarity generated by different methods.

Networks	BC	DC	HI	Kcore	Vc	KSIF	MDD	RCNN	CNNG
Jazz	0.7378	0.8896	0.8587	0.8410	0.6420	0.8202	0.9071	0.9021	**0.9271**
CEnew	0.5502	0.7374	0.7408	0.7354	0.6681	0.6796	0.7342	0.8068	**0.8207**
Email	0.4883	0.7129	0.7140	0.5774	0.4955	0.5617	0.7509	0.7643	**0.8092**
Stelzl	0.3232	0.4463	0.6864	0.6427	0.5543	0.6500	0.5027	0.6861	**0.6972**
Hamster	0.3671	0.6321	0.6321	0.4978	0.4096	0.4096	0.5428	0.6463	**0.6861**
Vidal	0.4020	0.4801	0.5915	0.5138	0.5168	0.5412	0.5082	0.6383	**0.6427**
Facebook	0.0398	0.4398	0.6325	0.5609	0.0279	0.5599	0.5373	**0.4707**	0.4387
EPA	0.4933	0.5644	0.5038	0.3655	0.5368	0.4762	0.5991	0.5809	**0.6152**
Router	0.3979	0.4380	0.5438	0.5263	0.5263	0.5031	0.4678	0.6765	**0.6894**
GrQ	0.0941	0.4217	0.3994	0.3929	0.1120	0.4024	0.4217	**0.5304**	0.4767
LastFM	0.1686	0.3829	0.5778	0.5281	0.1728	0.4902	0.4209	0.6185	**0.6399**
PGP	0.0976	0.4835	0.6356	0.6267	0.2431	0.5652	0.5710	0.7068	**0.7707**



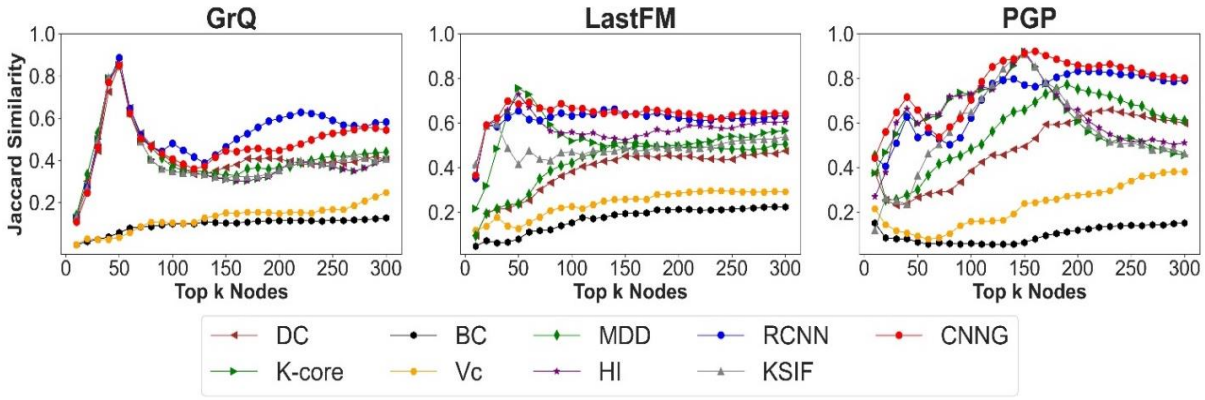


Figure 3: Jaccard Similarity between top-k nodes from various methods and ground truth.

Uniqueness of ranking. The Monotonicity Index results in Table 4 measure how uniquely each method ranks nodes. CNNG model shows high MI values across all networks, CNNG achieves 0.9998 on Facebook, LastFM and EPA networks. It also reaches 0.9991 on Jazz and Email and 0.9988 on CEnew, confirming its capacity of yielding unique influence scores on diverse network types. Even with sparse networks like Router, CNNG retains high MI values of 0.9972, which are much higher than the corresponding values from other methods. RCNN often appears as the second-best method in terms of MI scores. In general, CNNG indicates a higher diversity of rankings compared to other methods.

Table 4: Monotonicity indices of methods across twelve real world network datasets.

Networks	BC	DC	HI	Kcore	Vc	KSIF	MDD	RCNN	CNNG
Jazz	0.9885	0.9659	0.9659	0.7944	0.2086	0.2086	0.9919	**0.9991**	**0.9991**
CEnew	0.8741	0.7921	0.7310	0.6961	0.6974	0.8436	0.8755	0.9975	**0.9988**
Email	0.9400	0.8873	0.8582	0.8088	0.7701	0.8852	0.9227	**0.9998**	**0.9998**
Stelzl	0.5995	0.5287	0.4509	0.4260	0.5479	0.5373	0.5373	0.9956	**0.9957**
Hamster	0.7123	0.8979	0.8835	0.8714	0.7835	0.8997	0.9268	0.9856	**0.9857**
Vidal	0.6656	0.6507	0.5865	0.5569	0.6238	0.6666	0.6666	0.9923	**0.9925**
Facebook	0.9855	0.9739	0.9664	0.9419	0.5979	0.9794	0.9914	**0.9998**	**0.9998**
EPA	0.5344	0.5285	0.5080	0.4940	0.4512	0.5283	0.5393	0.9895	**0.9898**
Router	0.3043	0.2886	0.0875	0.0691	0.3023	0.2919	0.3030	0.9965	**0.9972**
GrQ	0.3822	0.7460	0.6906	0.6631	0.5460	0.7599	0.7935	0.9871	**0.9874**
LastFM	0.8346	0.7977	0.7570	0.7570	0.5976	0.8114	0.8400	**0.9998**	**0.9998**
PGP	0.5120	0.6193	0.5172	0.4805	0.4220	0.6449	0.6651	**0.9997**	**0.9997**

Practical Running Time. In Table 5, the testing time (in seconds) on twelve real-world networks is summarized to measure the computational efficiency of each method. All experiments are implemented with Python 3.8.3 in a 64-bit Windows environment on a computer where Intel Core i5-8250U CPU and 8GB RAM. From the results, it is clear that traditional methods like DC, Kcore, and HI have the lowest runtimes, which usually are less than a second per dataset. However, BC is noticeably more computationally expensive among traditional methods, particularly on larger networks, due to its higher time complexity. In contrast, deep learning-based models like RCNN and proposed CNNG involve higher computational cost due to the complexity of subgraph-based processing and neural network inference. However, this makes CNNG still a practical choice and scales reasonably up with the size of the network. For instance, the running time of CNNG is 2.5366s on Jazz, 6.3683s on CEnew, and 12.2446s on Email. On a large networks size like LastFM and PGP, it takes 76.9998s and 90.4563s respectively. These results demonstrate that CNNG model indeed finds a trade-off between effectiveness and efficiency.

In general, the experiments show that the CNNG is able to extract structural and relation patterns effectively, outperforming the compared methods. The model demonstrates robustness of ranking performance over a wide range of networks, from small to large networks and it performs efficient computation. By feeding local subgraphs into parallel feature extraction paths, the CNNG is capable of robust generalization, achieving better performance

compared to the state-of-the-art methods. These results confirm the efficiency of hybrid deep learning in network analysis.

Table 5: The running time in seconds of several methods on different network datasets.

Network	BC	DC	HI	Kcore	Vc	KSIF	MDD	RCNN	CNNG
Jazz	0.6874	0.0000	0.0156	0.0000	0.0000	0.0781	0.0781	1.8133	2.5366
CEnew	1.8738	0.0001	0.0157	0.0156	0.0012	0.0937	0.0625	4.6729	6.3683
Email	16.2224	0.0002	0.0312	0.0312	0.0016	0.2812	0.2343	7.5060	12.2446
Stelzl	21.3825	0.0001	0.0312	0.0156	0.0156	0.1567	0.1327	6.5067	6.2418
Hamster	60.4655	0.0003	0.0937	0.0625	0.0158	0.5624	0.5467	23.9504	31.7388
Vidal	45.6283	0.0002	0.0312	0.0256	0.0156	0.1093	0.0937	12.5993	16.2076
Facebook	305.2806	0.0002	0.2037	0.2195	0.0469	2.0106	3.1297	76.8970	114.558
EPA	186.8062	0.0050	0.0935	0.0645	0.0469	0.5354	0.3561	21.6288	33.1897
Router	231.1309	0.0156	0.0469	0.0469	0.0312	0.2343	0.2346	19.8599	36.0517
GrQ	178.9475	0.0021	0.0313	0.0312	0.0156	0.3270	0.2346	26.7077	36.0854
LastFM	456.3282	0.0034	0.0941	0.2187	0.0156	1.0362	0.4865	56.0079	76.9998
PGP	1231.3799	0.0169	0.1718	0.1249	0.0469	1.1404	1.0935	61.5906	90.4563

Conclusion

This work proposes a novel efficient hybrid deep learning model named CNNG to identify influential nodes in complex networks by combining Convolutional Neural Networks (CNN) with Graph Attention Networks (GAT). In contrast to previous approaches which possibly adopt a small set of isolated structural metrics, CNNG adapts to both local topology and relational dependencies by feeding subgraph of each node into two parallel constituents that concentrate on structural patterns and node relationships respectively. To measure the performance and robustness of CNNG, Kendall's Tau (τ) and Monotonicity Index (MI) and Jaccard Similarity are utilized as evaluation metrics as evaluation metrics. The model is compared to nine baselines over twelve real-world networks, demonstrating that CNNG improves (τ) scores, high MI values closer to 1 and stronger overlap in top-ranked nodes. While CNNG exhibits strong performance across a range of datasets, it is constrained to unweighted networks. However, the results illustrate that CNNG can be an effective and efficient tool for measuring node importance in various types of complex networks.

Funding

This research received no external funding.

Authors contribution

All authors contributed equally to the conception, design, implementation, and writing of this work. All authors read and approved the final manuscript.

Conflict of interests

The authors declare that there is no conflict of interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Zareie, A. Sheikhahmadi, M. Jalili, and M. S. K. Fasaei, "Finding influential nodes in social networks based on neighborhood correlation coefficient," *Knowledge-based Syst.*, **194**, 105580, 2020.

[2] U. Ishfaq, H. U. Khan, and S. Iqbal, "Identifying the influential nodes in complex social networks using centrality-based approach," *J. King Saud Univ. Inf. Sci.*, **34**(10), 9376–9392, 2022.

[3] B. Song, Y.-R. Song, G.-P. Jiang, X. Wang, L.-L. Xia, and others, "Identifying influential nodes in community networks," in *2023 42nd Chinese Control Conference (CCC)*, 832–836, 2023.

[4] S. Rajeh, M. Savonnet, E. Leclercq, and H. Cherifi, "Comparative evaluation of community-aware centrality measures," *Qual. \& Quant.*, **57**(2), 1273–1302, 2023.

[5] L. C. Freeman, “Centrality in social networks conceptual clarification,” *Soc. Networks*, **1**(3), 215–239, doi: 10.1016/0378-8733(78)90021-7 (1978).
[6] G. Sabidussi, “The centrality index of a graph,” *Psychometrika*, **31**(4), 581–603, 1966.
[7] L. C. Freeman, “A set of measures of centrality based on betweenness,” *Sociometry*, 1977.
[8] M. Kitsak *et al.*, “Identification of influential spreaders in complex networks,” *Nat. Phys.*, **6**(11), 888–893, 2010.
[9] Z. Wang, Y. Zhao, J. Xi, and C. Du, “Fast ranking influential nodes in complex networks using a k-shell iteration factor,” *Phys. A Stat. Mech. its Appl.*, **461**, 171–181, 2016.
[10] J. X. Zhang, D. B. Chen, Q. Dong, and Z. D. Zhao, “Identifying a set of influential spreaders in complex networks,” *Sci. Rep.*, **6**(June), 1–10, doi: 10.1038/srep27823 (2016).
[11] L. Lü, T. Zhou, Q.-M. Zhang, and H. E. Stanley, “The H-index of a network node and its relation to degree and coreness,” *Nat. Commun.*, **7**(1), 10168, 2016.
[12] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *J. Stat. Mech. theory Exp.*, **2008**(10), P10008, 2008.
[13] A. Zeng and C.-J. Zhang, “Ranking spreaders by decomposing complex networks,” *Phys. Lett. A*, **377**(14), 1031–1035, 2013.
[14] A. O. Mohammed, “Lcd: Identifying Influential Nodes in Complex Networks With Layered Clustering and Degree,” *Sci. J. Univ. Zakho*, **13**(2), 235–244, doi: 10.25271/sjuoz.2025.13.2.1483 (2025).
[15] X. Wen, C. Tu, M. Wu, and X. Jiang, “Fast ranking nodes importance in complex networks based on LS-SVM method,” *Phys. A Stat. Mech. its Appl.*, **506**, 11–23, doi: 10.1016/j.physa.2018.03.076 (2018).
[16] G. Zhao, P. Jia, C. Huang, A. Zhou, and Y. Fang, “A machine learning based framework for identifying influential nodes in complex networks,” *IEEE Access*, **8**, 65462–65471, 2020.
[17] A. Asgharian Rezaei, J. Munoz, M. Jalili, and H. Khayyam, “A machine learning-based approach for vital node identification in complex networks,” *Expert Syst. Appl.*, **214**(October 2022), 119086, doi: 10.1016/j.eswa.2022.119086 (2023).
[18] K. Hajarathaiah, M. K. Enduri, S. Anamalamudi, A. Abdul, and J. Chen, “Node Significance Analysis in Complex Networks Using Machine Learning and Centrality Measures,” *IEEE Access*, **12**(November 2023), 10186–10201, doi: 10.1109/ACCESS.2024.3355096 (2024).
[19] E.-Y. Yu, Y.-P. Wang, Y. Fu, D.-B. Chen, and M. Xie, “Identifying critical nodes in complex networks via graph convolutional networks,” *Knowledge-Based Syst.*, **198**, 105893, 2020.
[20] G. Zhao, P. Jia, A. Zhou, and B. Zhang, “InfGCN: Identifying influential nodes in complex networks with graph convolutional networks,” *Neurocomputing*, **414**, 18–26, 2020.
[21] Y. Ou, Q. Guo, J. L. Xing, and J. G. Liu, “Identification of spreading influence nodes via multi-level structural attributes based on the graph convolutional network,” *Expert Syst. Appl.*, **203(**May), 117515, doi: 10.1016/j.eswa.2022.117515 (2022).
[22] R. Bhattacharya, N. K. Nagwani, and S. Tripathi, “Detecting influential nodes with topological structure via Graph Neural Network approach in social networks,” *Int. J. Inf. Technol.*, **15**(4), 2233–2246, 2023.
[23] S. Kumar, A. Mallik, A. Khetarpal, and B. S. Panda, “Influence maximization in social networks using graph embedding and graph neural network,” *Inf. Sci. (Ny).*, **607**, 1617–1636, doi: 10.1016/j.ins.2022.06.075 (2022).
[24] 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*, **497**, 13–27, doi: 10.1016/j.neucom.2022.05.010 (2022).
[25] Y. Wu, Y. Hu, S. Yin, B. Cai, X. Tang, and X. Li, “A graph convolutional network model based on regular equivalence for identifying influential nodes in complex networks,” *Knowledge-Based Syst.*, **301**(June), 112235, doi: 10.1016/j.knosys.2024.112235 (2024).
[26] Y. Xiong, Z. Hu, C. Su, S. M. Cai, and T. Zhou, “Vital node identification in complex networks based on autoencoder and graph neural network,” *Appl. Soft Comput.*, **163**, (June), 111895, doi: 10.1016/j.asoc.2024.111895 (2024).
[27] L. Chen *et al.*, “Identifying influential nodes in complex networks via Transformer,” *Inf. Process. Manag.*, **61**(5), 103775, doi: 10.1016/j.ipm.2024.103775 (2024).
[28] Y. Rashid and J. I. Bhat, “Knowledge-Based Systems Olap GN : A multi-layered graph convolution network-based model for locating influential nodes in graph networks,” **283**(August 2023), 2024.
[29] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *arXiv Prepr. arXiv1710.10903*, 2017.
[30] R. Pastor-Satorras and A. Vespignani, “Epidemic dynamics and endemic states in complex networks,” *Phys. Rev. E*, **63(**6), 66117, 2001.
[31] A. L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science (80-.)*, **286**(5439), 509–512, doi: 10.1126/science.286.5439.509 (1999).
[32] G. I. Ivchenko and S. A. Honov, “On the jaccard similarity test,” *J. Math. Sci.*, **88**, 789–794, 1998.
[33] M. G. Kendall, “A new measure of rank correlation,” *Biometrika*, vol. 30, no. 1–2, pp. 81–93, 1938.
[34] J. Bae and S. Kim, “Identifying and ranking influential spreaders in complex networks by neighborhood coreness,” *Phys. A Stat. Mech. its Appl.*, **395**, 549–559, 2014.
[35] P. M. Gleiser and L. Danon, “Community structure in jazz,” *Adv. complex Syst.*, **6**(04), 565–573, 2003.

[36] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, “The large-scale organization of metabolic networks,” *Nature*, **407**(6804), 651–654, 2000.
[37] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, “Self-similar community structure in a network of human interactions,” *Phys. Rev. E*, **68(**6), 65103, 2003.
[38] U. Stelzl *et al.*, “A human protein-protein interaction network: a resource for annotating the proteome,” *Cell*, **122**(6), 957–968, 2005.
[39] J. Kunegis, “Konect: the koblenz network collection,” in *Proceedings of the 22nd international conference on world wide web*, 1343–1350, 2013.
[40] J.-F. Rual *et al.*, “Towards a proteome-scale map of the human protein--protein interaction network,” *Nature*, **437**(7062), 1173–1178, 2005.
[41] J. Leskovec and J. Mcauley, “Learning to discover social circles in ego networks,” *Adv. Neural Inf. Process. Syst.*, **25**, 2012.
[42] R. A. Rossi and N. K. Ahmed, “The Network Data Repository with Interactive Graph Analytics and Visualization,” in *AAAI*, 2015. [Online]. Available: https://networkrepository.com
[43] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, “Measuring ISP topologies with Rocketfuel,” *IEEE/ACM Trans. Netw.*, **12**(1), 2–16, 2004.
[44] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graph evolution: Densification and shrinking diameters,” *ACM Trans. Knowl. Discov. from Data*, **1**, (1), 2--es, 2007.
[45] B. Rozemberczki and R. Sarkar, “Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models,” in *Proceedings of the 29th ACM international conference on information \& knowledge management*, 1325–1334, 2020.
[46] M. Boguná, R. Pastor-Satorras, A. D\’\iaz-Guilera, and A. Arenas, “Models of social networks based on social distance attachment,” *Phys. Rev. E—Statistical, Nonlinear, Soft Matter Phys.*, **70**(5), 56122, 2004.