

## Journal Pre-proofs

Key nodes identification in complex networks based on subnetwork feature extraction

Luyuan Gao, Xiaoyang Liu, Chao Liu, Yihao Zhang, Giacomo Fiumara, Pasquale De Meo

PII: S1319-1578(23)00185-4  
DOI: <https://doi.org/10.1016/j.jksuci.2023.101631>  
Reference: JKSUCI 101631

To appear in: *Journal of King Saud University - Computer and Information Sciences*

Received Date: 20 March 2023  
Accepted Date: 16 June 2023

Please cite this article as: Gao, L., Liu, X., Liu, C., Zhang, Y., Fiumara, G., Meo, P.D., Key nodes identification in complex networks based on subnetwork feature extraction, *Journal of King Saud University - Computer and Information Sciences* (2023), doi: <https://doi.org/10.1016/j.jksuci.2023.101631>

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2023 The Author(s). Published by Elsevier B.V. on behalf of King Saud University.



# Key nodes identification in complex networks based on subnetwork feature extraction

Luyuan Gao<sup>a</sup>, Xiaoyang Liu<sup>a,\*</sup>, Chao Liu<sup>a</sup>, Yihao Zhang<sup>b</sup>, Giacomo Fiumara<sup>c</sup> and Pasquale De Meo<sup>d</sup>

<sup>a</sup>*School of Computer Science and Engineering, Chongqing University of Technology, Chongqing, 400054, China*

<sup>b</sup>*School of Artificial Intelligence, Chongqing University of Technology, Chongqing, 401135, China*

<sup>c</sup>*MIFT Department, University of Messina, V.le F. Stagno D'Alcontres, 31, 98166, Messina, Italy*

<sup>d</sup>*Department of Computer Science, University of Messina, V.le F. Stagno D'Alcontres, 31, 98166, Messina, Italy*


---

## ABSTRACT

---

---

\*Corresponding author

 lxy3103@cqut.edu.cn (X. Liu)

ORCID(s): 0000-0002-8619-0356 (X. Liu)

# Key nodes identification in complex networks based on subnetwork feature extraction

## ARTICLE INFO

### Keywords:

Key nodes identification  
Complex network  
Subnetwork feature extraction  
Graph convolutional networks

## ABSTRACT

The problem of detecting key nodes in a network (i.e. nodes with the greatest ability to spread an infection) has been studied extensively in the past. Some approaches to key node detection compute node centrality, but there is no formal proof that central nodes also have the greatest spreading capacity. Other methods use epidemiological models (e.g., the SIR model) to describe the spread of an infection and rely on numerical simulations to find out key nodes; these methods are highly accurate but computationally expensive. To efficiently but accurately detect key nodes, we propose a novel deep learning method called *Rank by Graph Convolutional Network*, *RGCN*. Our method constructs a subnetwork around each node to estimate its spreading power; then RGCN applies a graph convolutional network to each subnetwork and the adjacency matrix of the network to learn node embeddings. Finally, a neural network is applied to the node embeddings to detect key nodes. Our RGCN method outperforms state-of-the-art approaches such as RCNN and MRCNN by 11.84% and 13.99%, respectively, when we compare the Kendall's  $\tau$  coefficient between the node ranking produced by each method with the true ranking obtained by SIR simulations.

## 1. Introduction

Identifying key nodes in complex networks is an important research problem which attracted the interest of researchers from many fields (Newman, 2018; Liu, Ye, Fiumara and De Meo, 2023). These key nodes typically exhibit high accessibility and centrality, exerting decisive influence over network structure and controlling network evolution (Gong, Ji, Xie, Gao and Qin, 2022). Consequently, the identification of key nodes is relevant to address a number of problems, such as identifying critical traffic hubs for efficient and safe transportation (Huang, Chen, Cai, Wang and Hu, 2022), maximizing advertising benefits by identifying opinion leaders in social networks (Yu, Li and Yuan, 2021), and measuring the academic influence of scientists (Zhou, Liang, Wang, Huang and Jin, 2021).

The process of identifying key nodes in a network is strictly tied to the problem of measuring the *degree of importance* of a node, according to some definition of importance. In turn, the degree of importance of a node in a network is often known as *centrality* in the network science literature. Because of the importance of key nodes in regulating the functioning and the growth of a network, it is not surprising the wide number of centrality metrics currently available. Some of the existing centrality metrics leverage the knowledge of network local topology, that is they focus on a node and its neighbourhoods to measure how important such a node is. Examples of local centrality metrics include degree centrality, k-shell, H-index, and semi-local centrality.

Another family of centrality metrics focus, instead, on the knowledge of the full network topology to assess the importance of a node. We call these centrality metrics as *global centrality metrics* and we roughly classify them into *path-based*, *eigenvector-based*. Path-based methods such as eccentricity, closeness (Sariyüce, Kaya, Saule and Catalyirek, 2013), and Information-Rank (Liu, Gao, Fiumara and

De Meo, 2022b) consider the number of paths or the number of times a node is on the shortest path to compute the centrality of such a node. Eigenvector-based methods such as HITS (Liu, Jiang and Zou, 2020), LeaderRank (Li, Zhou, Lü and Chen, 2014), and PageRank (Gleich, 2015) take into account not only the number but also the quality of neighboring nodes in determining node importance.

In addition, the gravity model has been proposed to address the locality of degree centrality. (Zhao, Wen, Jahan-shahi and Cheong, 2022) proposed a gravity model based on random walk, which employs multiple random walkers to explore different parts of the graph and introduces an error rate metric to control the uncertainty of random walks. This method overcomes the problem of high time complexity when computing the shortest distance between all nodes using the gravity model. Zhong *et al.* (Zhong, Zhang and Deng, 2022) proposed a local degree dimension (LDD) method to compute node centrality. Specifically, given a target node  $x$ , the approach of Zhong *et al.* (Zhong *et al.*, 2022) proposes to divide the network into *layers* (the nodes in the  $k$ -th layer are nodes at distance  $k$  from  $x$ ) and they calculate the rate with which the number of nodes in the layers increases or decreases. The increasing and the decreasing rates are multiplied by the degree of  $x$  to obtain a more objective evaluation of the centrality of  $x$  itself.

Approaches above make extensive use of graph topology to compute node centralities. More recently, some authors viewed the problem of computing node centralities as a *learning problem*. In fact, scientific advancements in the field of Deep Learning provided novel and effective representation tools such as Deep Walk (Perozzi, Al-Rfou and Skiena, 2014) and Node2Vec (Grover and Leskovec, 2016). Methods above aim to learn low-dimensional latent representations of graph structure data which are now used for a wide range of tasks, such as classification, clustering, link prediction, and graph visualization.

ORCID(s):

Recently, Yu *et al.* (Yu, Wang, Fu, Chen and Xie, 2020) proposed an algorithm (called RCNN) based on convolutional neural networks (CNN) to detect key nodes in complex networks. Specifically, the RCNN algorithm first generates a subgraph for each node and then it constructs a  $L \times L$  feature matrix for each nodes as the input of the convolutional neural network. Since RCNN only considered the degree of micro-level nodes as the information to construct the feature matrix, it may lead to peripheral nodes being misclassified as diffusion influence nodes. Ou *et al.* (Ou, Guo, Xing and Liu, 2022) proposed the multi-channel RCNN (MR-CNN) algorithm by comprehensively considering micro-level, community-level, and macro-level structural information. The input of the convolutional neural network is a  $3 \times L \times L$  feature matrix.

Graph Convolution Networks (GCNs) are an appealing option to calculate node centralities due to their ability to handle both node features and network structure information. GCN has achieved great success in many fields, including drug design, recommendation systems, and natural language processing (Sun, Zhao, Gilvary, Elemento, Zhou and Wang, 2020). An important advantage of the GCN model is its ability to process non-Euclidean data, which is particularly suited for complex networks. Inspired by the GCN, this paper proposes a method for identifying key nodes in complex networks. Our method, called RGCN, extracts subnetwork features for specific subnetworks surrounding a node and it uses a GCN to learn node representations which incorporate the contribution of the subnetworks above.

In our framework, GCNs effectively combine network topology (encoded through the adjacency matrix of a network) with subnetworks surrounding each node to effectively detect key nodes. In our opinion, in fact, the importance of a node  $x$  should be related with the ability of  $x$  to disseminate information in the network and, in particular, key nodes are those nodes with the largest ability to disseminate information.

However, even if  $x$  would have a high degree, we could not conclude that such node has a high diffusion capacity: in fact, the neighbors of  $x$  could have a small number of neighbors and, thus, an information diffusion process that originates at  $x$  could quickly die due to lack of nodes to infect if we would move only few steps away from  $x$ .

With these premises, it is more appropriate to consider a *subnetwork* surrounding the node  $x$ : if such a subnetwork were dense enough, then the chances that an information propagation process originating at  $x$  immediately dies would be smaller than in the case we discussed first. Furthermore, nodes in real networks often represent complex real world objects and, thus, they are often described by one or more features. According to past studies (Lawyer, 2015), node features can impact on the node's ability to disseminate information. To this end, GCNs are a very elegant yet powerful tool for combining structural properties encoded in subnetworks with node features to learn more expressive node representations. We believe that more refined node

representations are crucial to better identify key nodes in networks.

The main contributions of this paper can be summarized as follows:

- We formulate the problem of predicting the power of a node to spread an infection as a regression problem and, thus, we classify as key nodes the nodes displaying the largest spreading power. We adopted the SIR model to describe the spread of the infection. We considered both network topology and subnetworks surrounding a node to learn node embeddings through a GCN and these embeddings are used to predict the spreading power of each node.
- We evaluate RGCN and five baseline methods on both fully connected and non-fully connected networks. Results demonstrate that RGCN's key node identification approach outperforms state-of-the-art methods, with higher correlation between the centrality scores obtained by RGCN and the true spreading power obtained by numerical simulations.

The plan of the paper is as follows: in Section 2 we review related literature, while Section 3 presents the proposed method, which is divided into five subsections that cover the methodology architecture, representation learning for complex networks, representation learning for node features, and more. Section 4 describes the experimental setup, including the network datasets used, evaluation metrics, and baseline methods. In Section 5, the experimental results are discussed and analyzed in relation to the experimental setup. Section 6 discusses the limitations and shortcomings of the proposed method. Finally, Section 7 summarizes the paper and suggests future research directions.

## 2. Related Work

In this section we review some popular methods to detect key nodes in large complex networks. We can classify existing methods into two main categories, namely: *topology-based* and *deep learning* based methods.

Topology-based methods analyze the network topology to identify key nodes; in some cases, some dynamic models (e.g. the spread of influence in the network) are simulated to detect key nodes.

In contrast, deep-Learning based methods adopt an opposite view point, that is they formulate the task of detecting key nodes as a *learning problem* and, to this end, they apply well-known tools from Machine Learning such as the logistic regression, Support Vector Machines (SVM) and, more recently, Convolutional Neural Networks (CNNs).

### 2.1. Topology-based methods

Topology-based methods can be divided into *local* (if they only consider the neighbourhood of a node) and *global* (if, conversely, they consider the whole network topology). An intermediate category of approaches is called *hybrid*:

here we extend the neighbourhood of a node, but we impose an upper bound on the horizon of such a node (e.g. only nodes that are up to  $k$  hops away from a target node contribute to the computation of the node itself).

Most of the methods we describe in this section compute node centrality and, thus, they select the largest centrality nodes as key nodes.

The most well-known method based on local topology is the *degree centrality* (Freeman et al., 2002), which uses the number of first-order neighbors of a node as an indicator of node centrality. However, degree centrality is easily affected by network interruptions, and it ignores the differences in the importance of nodes and their neighboring nodes.

In addition, the degree distribution in real networks is often highly skewed (Adamic, Lukose, Puniyani and Huberman, 2001; Newman, 2018), that is most edges touch only a few hub nodes; in this case, the vast majority of network nodes would have the same degree centrality, making such a centrality measure incapable of distinguishing key nodes from the remaining ones.

To address some of the shortcomings of degree centrality, some authors have suggested using the entire network topology to define more refined centrality metrics.

An interesting class of global methods makes use of *paths*: a path in a network is understood as a sequence of non-repeating nodes, such that successive pairs of nodes are connected by an edge. The length of a path is equal to the number of nodes composing it minus one, and such a definition is easy to generalise if we manage weighted networks (that is, if we assume that edges are provided with weights). In this case, the path length is the sum of the weights associated with the edges forming a path. The length of the shortest path connecting two nodes  $x$  and  $y$  in a network is used to measure the distance of  $x$  and  $y$ , and, thus, several metrics based on node distances have been proposed so far. For example, the *closeness centrality* (Newman, 2018) of a node  $x$  is equal to the inverse of the distance of  $x$  to all other nodes connected to  $x$ . Closeness centrality has been widely applied in the past but, unfortunately, it lacks discriminative power for nodes located at the edge of the network or in ring-shaped network structures similar to concentric circles. Another centrality metric based on the shortest paths is the *betweenness centrality* (Leydesdorff, 2007; Newman, 2018) which counts the frequency a node  $x$  is located on shortest paths in the network. The betweenness centrality is attractive because it provides a more comprehensive evaluation of a node's influence and control but, because of its high computational complexity, it does not scale well on large-scale networks.

DynamicRank (Chen, Sun, Tang, Tian and Xie, 2019) introduces *propagation dynamics models* to compute node centrality. Unlike methods merely based on the network topology, DynamicRank associates each node with an infection probability and it simulates an information diffusion process to compute node centralities. DynamicRank has linear time complexity in the size of the network. Consequently, DynamicRank works well on large-scale and sparse networks, but it is applicable only to undirected and unweighted

networks and it cannot consider multilayer networks. The InformationRank algorithm (Liu et al., 2022b) introduces information propagation probability and reflects local and global topological information of the network by considering path diversity. The computational time complexity of InformationRank may become unacceptable as the network size increases.

Ullah et al. (Ullah, Wang, Sheng, Long, Khan and Sun, 2021) propose the *Local-Global Centrality* method (LGC), which combines local and global topological information; the LGC method is more effective than methods above because it combines node degree centrality and shortest path information between nodes. However, the LGC method suffers from some limitations. Specifically, it requires calculations for the entire network, so it is inefficient when dealing with very large networks.

In general, topology-based methods are poorly scalable: for example, computing all pairs of shortest paths in a network takes cubic time in the number of nodes, so centrality metrics based on shortest paths (e.g., closeness or betweenness) are not applicable to networks with billions of nodes/edges. The scalability problem is exacerbated if we are in charge of calculating the centrality of all nodes in the network, or if we are dealing with dynamic networks whose topology changes significantly even in short time intervals. One possible solution is to develop scalable algorithms capable of providing an accurate approximation of the centrality values (Bader, Kintali, Madduri and Mihail, 2007; Brandes and Pich, 2007; Cohen, Dellinger, Pajor and Werneck, 2014). For this purpose, *sampling methods* have been widely used in the past. In sampling methods, a given number of *seed nodes* are selected from the network and the *single source shortest path* (SSSP) is computed (i.e. we compute the shortest path from any seed node to any other node in the network). If a node is a seed, then we can compute the exact value of the closeness/betweenness centrality; otherwise, the centrality of a non-seed node is approximated by averaging the results generated from the available SSSP trees.

Despite their elegance and the good trade-off between accuracy and scalability, sampling-based methods raise non-trivial questions; for example, how many nodes should be selected as seeds? How should we choose the seed nodes? Most of these questions are still unresolved, and therefore techniques from machine learning and deep learning have recently been considered as good alternatives to sampling-based methods.

## 2.2. Deep learning-based methods

More recently, machine learning models and artificial neural networks have been used to approximate the true value of some centrality metrics (we refer the reader to Grando et al (Grando, Granville and Lamb, 2019) for an excellent survey).

To address the computational complexity of traditional methods, Wen et al. (Wen, Tu, Wu and Jiang, 2018) proposed an algorithm based on the least-squares support vector machine. First, four indicators reflecting the importance of



nodes comprehensively are selected, and then the Analytic Hierarchy Process (AHP) is applied to obtain the importance scores of nodes. However, this method also has some limitations. Although the comprehensive indicators and AHP method are used to obtain the global and comprehensive evaluation of nodes, AHP itself may also be subjective and uncertain.

Park *et al.* (Park, Kan, Dong, Zhao and Faloutsos, 2019) proposed GENTI to identify important nodes in knowledge graphs using supervised learning. By modeling the complex relationship between entities and their importance using the rich information in knowledge graphs, this method performs well in experiments, but it requires a large amount of labeled data. Huang *et al.* (Huang, Sun, Du, Liu, Lv and Xiong, 2021) proposed a new node importance evaluation algorithm called RGTN-NIE, which combines Graph Convolutional Networks GCN and graph representation learning techniques and uses both structural and attribute features of nodes to more accurately evaluate node importance. However, the disadvantage of this method is that some parameters need to be manually adjusted, such as the number of layers and attention mechanism parameters, requiring some domain knowledge and experience. The common form of complex networks is the attribute-free graph, so the idea of key node identification based on knowledge graphs is not applicable to the identification of key nodes in complex networks. Fan *et al.* (Fan, Zeng, Sun and Liu, 2020) proposed a deep reinforcement learning framework called FINDER to find groups of key nodes, where the graph state and action are learned through inductive graph representation learning. However, although the FINDER method can solve the key node problem, it is not suitable for all types of network optimization problems.

Zhao *et al.* (Zhao, Jia, Zhou and Zhang, 2020) proposed a graph deep learning framework called InfGCN to transform the problem of identifying key nodes in complex networks into a classification problem; the InfGCN method uses some indicators for measuring node importance and the neighbor graph as input. However, other types of node features or higher-order network structures may not have been fully utilized.

Convolutional neural networks (CNNs) and Graph Neural Networks (GCNs) are excellent tools that can be successfully applied to approximate centrality values. In particular, CNNs have been applied in other important areas, such as the analysis of the human neuromuscular system and the evaluation of rehabilitation programmes *et al.* (Zhang, Zhao, Shone, Li, Frangi, Xie and Zhang, 2022a).

Liu *et al.* (Liu, Cao and Zhou, 2022a) proposed an adaptive learning graph convolutional neural network model to rank node importance, using multi-task learning to combine the ranking task with the regression task to complete the node ranking in complex networks. However, there was no analysis of the effectiveness and necessity of sub-supervised pre-training tasks, nor were any methods or indicators given to balance the weights and effects of different tasks. Zhang *et al.* (Zhang, Wang, Jin, Song and Li, 2022b) proposed

a method called CGNN to identify key nodes, obtaining the feature matrix of nodes through contraction algorithm. Although the CGNN method can automatically learn the feature of node importance from multiple indicators, it requires a large amount of training data and computing resources, and it may have risks of overfitting or underfitting. Also, there was no detailed analysis and tuning of the parameters and structure of the deep learning model.

The main problems of key nodes identification in complex networks based on deep learning methods are their interpretability and the need for a large amount of training data and computing resources. As for the scalability issue, in fact, we point out that the training process of standard deep learning architectures is often time-consuming due to the large number of parameters to be learned, as well as the fact that some approaches implement a hierarchical structure consisting of several stacked layers. More recently, Extreme Learning Machine (ELM) theory has been proposed to speed up the training process (we refer to Zhang *et al.* (Zhang, Li, Xiao and Zhang, 2020) for a detailed review of the state of the art); in particular, when training ELMs, hidden layer parameters are randomly drawn from a given interval and the output weights are computed by the least squares method (Cao, Hao, Lai, Vong and Luo, 2016), making ELMs much faster and easier to implement than many other existing deep learning methods.

Taking into account the common issues in both traditional methods and deep learning methods, we propose the RGCN framework, which introduces the concept of subnetworks. Only a subset of nodes in each subnetwork are assigned node features. The entire network topology and the node features of the subnetworks are then used as input for the model, allowing GCN to capture high-order neighbor information of each node. This approach not only simplifies the computation but also reduces the demand for large amounts of training data.

### 3. Proposed Method

#### 3.1. Methodology architecture

In this section, we describe our approach, called Rank by Graph Convolution Network *RGCN*, to identifying key nodes in a network.

For this purpose, let us assume that an infection propagates in the starting from an arbitrary node, say  $u$ . Intuitively, the key nodes are the nodes having the greatest *spreading power*, i.e., the nodes capable of infecting the largest number of nodes in the network. Many off-the-shelf mathematical models are available to describe the spread of an infection in a network. The *Susceptible Infected Recovery* (SIR) model is one of the most widely studied models; despite its simplicity, the SIR model is well suited to describe the spread of epidemics in the real world.

Accordingly, we can associate each node with a score that quantifies its spreading power, and, thus key nodes are the nodes with the highest scores. We can compute the scores of all nodes through numerical simulations; however,

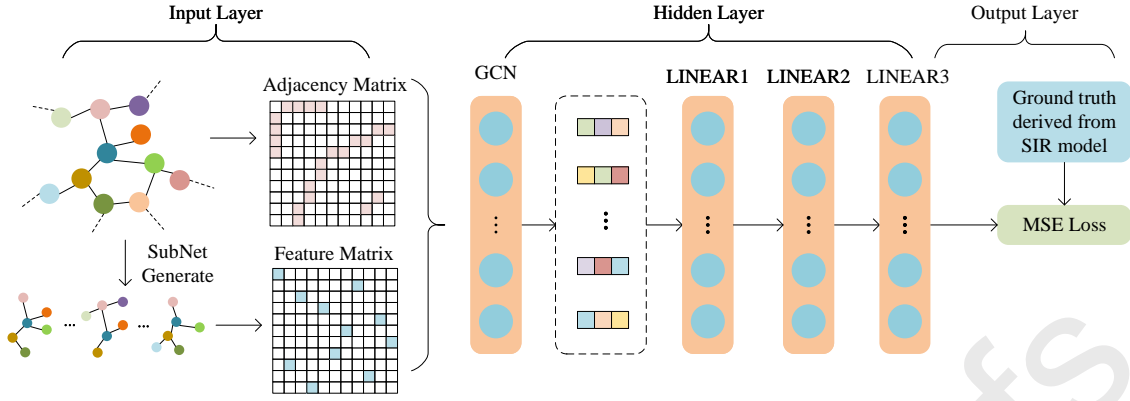


Fig. 1: The framework of RGCN method

to obtain accurate results, we need to run thousands of simulations, which is computationally very expensive if the input network is moderately large.

In this paper, we propose to formulate the problem of estimating the diffusion power of a node as a *regression problem*. To determine the features of a node, we propose to consider the topology of the input network, encoded by its adjacency matrix; furthermore, for each node, we extract a *subnetwork* of fixed dimension surrounding that node. Such a subnetwork is relevant to estimate the spreading power of a node  $u$ : in fact, if the subnetwork surrounding  $u$  would contain many high degree nodes, then  $u$  would be more likely to infect other nodes.

Network topology as well as the subnetworks associated with each node are the input of a *Graph Convolutional Network* (GCN), which allows an efficient node representation. Node representations are the input of a neural network that estimates the spreading score of a node.

To train our neural network, we minimize the Mean Square Error (MSE) between the estimated spreading scores and the true ones obtained from simulations. The proposed RGCN system architecture is illustrated in Fig.1.

### 3.2. Representation of complex networks

A complex network is a system consisting of an assembly of entities, called nodes or vertices, interacting through pairwise connections named edges. Notwithstanding these simple ingredients, what makes complex networks so attractive is the emergence of some intriguing phenomena emerge that cannot be derived from the analysis of the components. Among the others, here we only cite the small-world phenomenon, and the heavy tail of probability distribution. The interested reader can find a comprehensive discussion in (Barabási and Pósfai, 2016). A complex network can be represented as a graph  $G = (V, E)$ , in which  $V = \{v_1, v_2, \dots, v_n\}$  is the set of nodes and  $E \subseteq V \times V$  is the set of edges. Let  $n$  be the number of nodes and  $m$  be the number of edges. A network can be represented by an adjacency matrix  $\mathbf{A}$  of size  $n \times n$ , where  $\mathbf{A}_{i,j} = 1$  if there is an edge from node  $i$  to node  $j$ , and  $\mathbf{A}_{i,j} = 0$  otherwise.

A walk from node  $u$  to node  $v$  is a sequence of nodes starting with  $u$  and ending with  $v$ , such that between any consecutive nodes there is a link. A path is a special walk in which all nodes are distinct. A *shortest path* is a path spanning over the minimum number of edges. If the element  $(u, v)$  of the matrix  $\mathbf{A}^k$  (with  $k \geq 2$ ) is one, nodes  $u$  and  $v$  are connected by a walk having length  $k$ .

An important concept in complex networks analysis is *centrality*. It is a measure of the importance of a node (or an edge) in a network. In the last decades several centrality measures have been introduced that emphasize some topological or relational aspects of nodes or edges. Probably the most widely used is the *degree centrality*, defined as:

$$DC(u) = \frac{d_u}{n-1} \quad (1)$$

$d_u$  being the *node degree* of node  $u$ , namely the number of edges incident to  $u$ . The degree centrality is a real number in the range 0 (an isolated node) and 1 (a node connected to all other nodes).

### 3.3. Representation learning of nodes

We employ the GCN for node representation learning of complex networks and partial node features. The architecture of GCN is analogous to that of a typical convolutional neural network, but its convolutional layer operates on graphs, where nodes act as convolution kernels, rather than in a continuous image space. This process can be formalized as follows:

$$\mathbf{H}^{l+1} = GCN(\mathbf{H}^l, \mathbf{A}) = \sigma(\hat{\mathbf{D}}^{-1/2} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-1/2} \mathbf{H}^l \mathbf{W}^l) \quad (2)$$

where  $\mathbf{A}$  refers to the adjacency matrix that converts the complex network into an unweighted and undirected representation.  $\hat{\mathbf{D}}$  is a diagonal matrix with self-loops added, which is represented as the identity matrix.  $\mathbf{W}^l$  represents the trainable weight matrix for the  $l$ -th layer's linear transformation. GCN is effective in learning the relationships between nodes and can extract complex features from the graph. It can quickly converge on complex graph structures and has good robustness to input data.

### 3.4. Susceptible-Infected-Recovered(SIR) Model

The infection capability value of each node is obtained label by simulating the SIR model 1000 times. The infection capability allows us to distinguish key nodes from the remaining nodes in the network, i.e. we assume that key nodes are those nodes with the greatest infection capability. The ultimate goal of RGCN is to predict the infection capability of any target node in the network.

The SIR model is a classic epidemic model in which  $S$  stands for the susceptible state,  $I$  denotes the infected state, and  $R$  implies the recovered state. In the SIR model, the probability an infected node will transmit the infection to a susceptible neighbor is  $\beta$ , and the recovery probability is  $\mu$ . The differential equations governing the SIR model are as follows:

$$\begin{cases} \frac{dS(t)}{dt} = -\beta S(t)I(t) \\ \frac{dI(t)}{dt} = \beta S(t)I(t) - \mu I(t) \\ \frac{dR(t)}{dt} = \mu I(t) \end{cases} \quad (3)$$

The recovery rate  $\mu$  is generally set to 1, and the infection rate threshold  $\beta$  is set to  $\frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle}$ .

### 3.5. Key Nodes Identification

In this section we describe our RGCN approach and, in detail, we first illustrate the main components of the RGCN architecture (see Section 3.5.1 and, subsequently, we detail the procedure to extract a subnetwork for each node (see Section 3.5.2).

#### 3.5.1. The RGCN architecture

The RGCN model proposed in Fig.1 consists of an input layer, a hidden layer, and an output layer.

The input layer learns new node features by extracting subnetworks for each node (see Section 3.5.2). The hidden layer consists of a GCN layer and three linear layers. The GCN layer is defined as follows:

$$\mathbf{H}^{i+1} = \sigma(\mathbf{A}\mathbf{H}^i\mathbf{W}^i + \mathbf{b}^i) \quad (4)$$

where  $\mathbf{A}$  is the normalized adjacency matrix of the complex network,  $\mathbf{H}^i$  represents the node feature representation at the  $(i + 1)$ -th GCN layer. In addition,  $\mathbf{W}^i$  and  $\mathbf{b}^i$  are trainable weight and bias parameters, and  $\sigma$  denotes the non-linear activation function, specifically the exponential linear unit (ELU) function in our case.

The linear layer is defined as follows:

$$\hat{y} = \mathbf{H}^{i+1}\mathbf{A}^T + \mathbf{b} \quad (5)$$

We uses three linear layers for regression tasks.

The output of the last layer  $\hat{y}$  is compared with the value  $y$  obtained by simulating the SIR model 1000 times. Then, we used the mean squared error loss (see Equation 6).

$$Loss = \frac{1}{2}(y - \hat{y})^2 \quad (6)$$

The Adam optimizer is chosen to update the model parameters. The process of learning new node features is shown in Algorithm 1.

---

#### Algorithm 1: Feature Update in RGCN

---

**Data:** Network  $G$ , Node feature matrix  $\mathbf{X}$ , Weight matrix  $\mathbf{W}$

**Result:** New feature matrix of nodes  $\mathbf{Z}$

```

1 Initialize: GCN model;
2 for node  $i$  in  $G$  do
3    $\mathbf{x}_i \leftarrow \mathbf{X}[i]$ ;
4    $N_i \leftarrow$  Get  $i$  neighbors;
5   Initialize an empty  $\mathbf{h}_i$ ;
6   for neighbor  $j$  in  $N_i$  do
7      $\mathbf{x}_j \leftarrow \mathbf{X}[j]$ ;
8      $\mathbf{h}_i \leftarrow \mathbf{x}_j + \mathbf{h}_i$ 
9   end
10   $\mathbf{a}_i \leftarrow \mathbf{x}_i + \mathbf{h}_i$ ;
11   $\mathbf{z}_i \leftarrow \mathbf{x}_i \times \mathbf{W}$ ;
12   $\mathbf{Z} \leftarrow \mathbf{z}_i$ 
13 end
14 return  $\mathbf{Z}$ 

```

---

In Algorithm 1, the node feature matrix  $\mathbf{X}$  is obtained by calculating the node features of the subnetworks using Information Rank, while the remaining nodes without feature calculation rely on the degree of nodes during the GCN learning process.

The adaptive weight matrix  $\mathbf{W}$  is used to adjust the strength of information transmission between different nodes, which can preserve local structural information as features.

#### 3.5.2. Constructing a subnetwork

We are now able to describe the process of constructing the subnetwork associated with a node  $u$ . The pseudocode for constructing such a subnetwork is given in Algorithm 2; an example detailing the output of Algorithm 2 is reported in Figure 2. Algorithm 2 takes as input a network  $G$ , a node  $u$  (from which we start constructing the subnetwork) and a positive integer  $L$  (that is the number of nodes the subnetwork has to contain). With reference to Figure 2, suppose to choose  $u = a$  and  $L = 9$ .

Algorithm 2 must construct a candidate list of nodes of size at least equal to  $L - 1$  and, then, it selects  $L - 1$  nodes from such a list. The output subnetwork will contain the node  $a$  plus  $L - 1$  nodes from the candidate list and, thus, it will have exactly  $L$  nodes, as we required. To determine candidate nodes, Algorithm 2 first extracts the first-order neighbors of  $a$  (that is, nodes which are one-hop away from  $a$ ), that is nodes  $b$ ,  $e$ , and  $l$ .

Since Algorithm 2 discovered only three nodes which is less than  $L - 1 = 9 - 1 = 8$ , then it has to continue exploring the network  $G$  by adding second-order neighbors (that is, nodes which are two hops away from  $a$ ). Thus, Algorithm 2 considers nodes  $d, f, g, h, i, c$ , and  $k$ ; overall, Algorithm 2 manages  $10 > 8 = L - 1$  nodes and, thus, the network exploration steps ends. Each node is inserted in the candidate list only once. To determine which nodes have to be included, we sort the list of candidate nodes by descending degree and



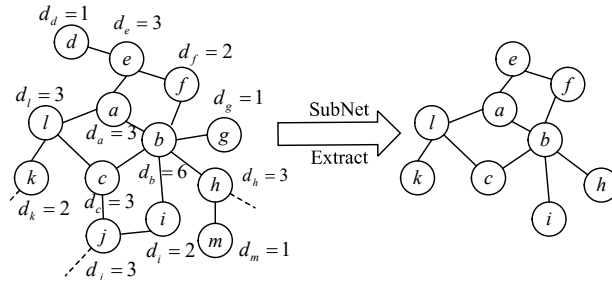


Fig. 2: Subnetwork generation process

we pick the top  $L - 1$  elements; in our case, we get the list:  $b, l, c, e, f, h, i$ , and  $k$ .

Finally, the information propagation ability of each node in the subnetwork is calculated by means of the Information-Rank algorithm (Liu et al., 2022b) as its feature.

---

**Algorithm 2:** Subnetwork Construction
 

---

**Data:** Network  $G$ , Node  $u$ , Subnetwork Size:  $L$

**Result:**  $SubNet$  of node  $u$

```

1 Initialize: Empty list  $N^u$ ;
2  $N^u \leftarrow$  Get  $u$  neighbors;
3 while  $|N^u| < N - 1$  do
4   Initialize an empty list  $neb$ ;
5   for  $n$  in  $N^u$  do
6      $neb \leftarrow$  Get  $n$  neighbors;
7   end
8    $N^u \leftarrow$  Unique  $neb$ 
9 end
10 if  $|N^u| > L - 1$  then
11    $SubNetNodes \leftarrow$  Sort  $N^u$  by degree;
12    $SubNetNodes \leftarrow$  Get top  $N - 1$  nodes;
13    $SubNetNodes \leftarrow u + SubNetNodes$ ;
14    $SubNet \leftarrow$  Extract from  $G$  corresponding to
      $SubNetNodes$ ;
15 else
16    $SubNetNodes \leftarrow u + N^u$ ;
17    $SubNet \leftarrow$  Extract from  $G$  corresponding to
      $SubNetNodes$ ;
18 end
19 return  $SubNet$ 

```

---

## 4. Experimental setup

### 4.1. Network Datasets

To accurately verify the accuracy and universality of the RGCN model, the test set contains 12 real networks, including 6 fully connected networks and 6 non-fully connected networks. In this experiment, these networks are processed as undirected and unweighted networks. The 12 networks included: (1) Arenas (Duch and Arenas, 2005): The metabolic network of the nematode *Caenorhabditis elegans*; (2) Email (Guimera, Danon, Diaz-Guilera, Giralt and Arenas, 2003): The email communication network

among students at the University of Rovira i Virgili; (3) Jazz (Gleiser and Danon, 2003): the collaboration network of jazz musicians; (4) PGP (Rossi and Ahmed, 2015): The largest connected component of the network of users of the Pretty-Good-Privacy algorithm for secure communication; (5) Power-Grid (Watts and Strogatz, 1998): The electric power grid of the western United States; (6) Router (Spring, Mahajan and Wetherall, 2002): A network of Autonomous Systems that form the Internet; (7) Stelzl (Stelzl, Worm, Lalowski, Haenig, Brembeck, Goehler, Stroedicke, Zenkner, Schoenherr, Koeppen et al., 2005): The protein-protein interaction network of yeast; (8) Vidal (Rual, Venkatesan, Hao, Hirozane-Kishikawa, Dricot, Li, Berriz, Gibbons, Dreze, Ayivi-Guedehoussou et al., 2005): The initial version of a protein-protein interaction map at the proteome scale for humans; (9) Chicago (Eash, Chon, Lee and Boyce, 1979): The transportation network of the Chicago metropolitan area; (10) NS (Newman, 2006): The collaboration network of researchers in the field of networking; (11) Figeys (Ewing, Chu, Elisma, Li, Taylor, Climie, McBroom-Cerajewski, Robinson, O'Connor, Li et al., 2007): The protein-protein interaction network of human cells, derived from the first large-scale study of protein interactions in human cells using mass spectrometry-based methods; and (12) Hamster (Kunegis, 2013): The friendship network of users of the Hamster social networking site.

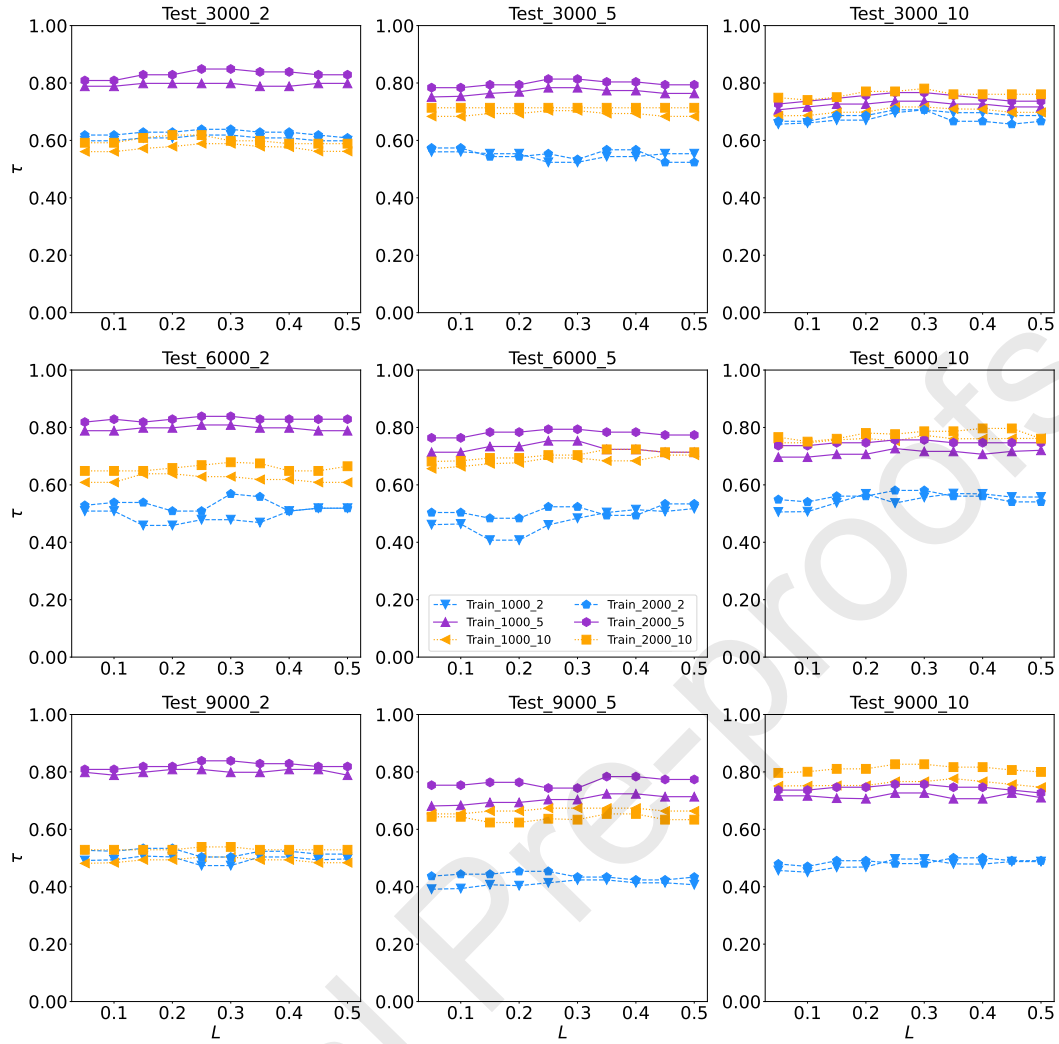
Table 1 lists some statistical characteristics of the 12 real-world networks, which represent network representations of different social systems and have different topological structures and features. All experiments are implemented on a server with two Intel 64-bit Gold 6226R 2.90 GHz CPUs and two NVIDIA RTX A6000 GPUs.

### 4.2. Evaluation metrics

**Pearson correlation coefficient:** The Pearson correlation coefficient  $\sigma_{XY}$  is used as an evaluation metric to observe the difference between the node scores obtained by RGCN and the traditional method IR that obtains partial node scores as features. The calculation of  $\sigma_{XY}$  is as follows:

$$\rho_{XY} = \frac{cov(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (7)$$

where  $X$  and  $Y$  represent the score sequences of any two algorithms for evaluating the importance of each node in



**Fig. 3:** The Kendall  $\tau$  coefficient changes between the node ranking obtained by RGCN using different sub-network sizes  $L$  for node feature and the ground truth ranking

the complex network.  $\mu_X$  and  $\mu_Y$  are the means of the two sequences,  $\sigma_X$  and  $\sigma_Y$  are the standard deviations of the two sequences. This coefficient is used to indicate the correlation between the two methods for computing node importance in the complex network. A lower coefficient indicates that the two methods have less consistency in their calculation methods.

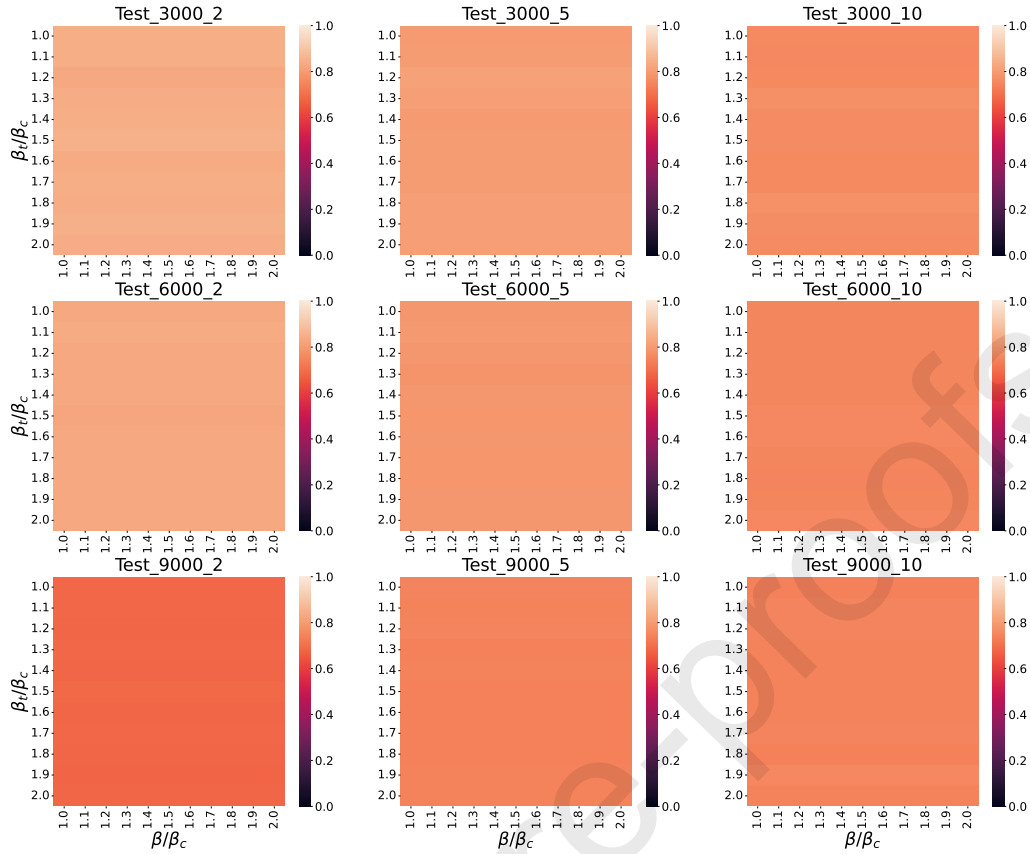
**Kendall  $\tau$  correlation coefficient:** The Kendall  $\tau$  correlation coefficient is defined as a measure of the association between two rankings of the same set of items. Given two rankings  $X$  and  $Y$  of length  $n$ , the Kendall  $\tau$  coefficient is denoted as  $\tau$  and is defined as:

$$\tau = \frac{P - Q}{P + Q + S} \quad (8)$$

where  $P$  is the number of concordant pairs (i.e., pairs that appear in the same order in both  $X$  and  $Y$ ),  $Q$  implies the number of discordant pairs (i.e., pairs that appear in opposite orders in  $X$  and  $Y$ ), and  $S$  expresses the number of pairs that have the same value in both  $X$  and  $Y$ . The combination

of  $X$  and  $Y$  as  $(x_i, y_i)$  and  $(x_j, y_j)$ , where if  $x_i = x_j$  or  $y_i = y_j$ , then  $(x_i, y_i)$  and  $(x_j, y_j)$  are not considered as concordant or discordant pairs. When  $x_i > x_j$  and  $y_i > y_j$ , or  $x_i < x_j$  and  $y_i < y_j$ , and for all other cases,  $(x_i, y_i)$  and  $(x_j, y_j)$  are considered as discordant pairs. The Kendall  $\tau$  coefficient ranges from  $-1$  to  $1$ , with values closer to  $1$  indicating a stronger positive correlation between the two rankings, values closer to  $-1$  indicating a stronger negative correlation, and values close to  $0$  indicating no correlation.

**Monotonicity index:** The Monotonicity Index (MI) (Bae and Kim, 2014) is used to measure the overall distinguish ability of node importance ranking sequences obtained from different key node identification algorithms, which reflects the extent to which different algorithms differentiate the importance of nodes. The value of MI ranges from  $0$  to  $1$ , with a value closer to  $1$  indicating a better distinguish ability of the node importance ranking sequence obtained by the algorithm. The calculation formula of MI is as follows:



**Fig. 4:** The impact of changing infection rate  $\beta$  on the Kendall  $\tau$  coefficient between the RGCN ranking and the ground truth ranking of identified nodes with fixed sub-network size  $L$ .

**Table 1**

The statistical properties of the 12 real networks.  $n$  denotes the number of nodes,  $m$  denotes the number of edges,  $\langle k \rangle$  is the average degree,  $k_{max}$  is the maximum degree,  $c$  is the average clustering coefficient of the network.  $H$  is the degree heterogeneity, defined as  $H = \langle k^2 \rangle / \langle k \rangle^2$ .  $G_{cv}\%$  and  $G_{ce}\%$  represent the proportions of nodes and edges in the largest connected component of the network, respectively.  $d$  is the network density.

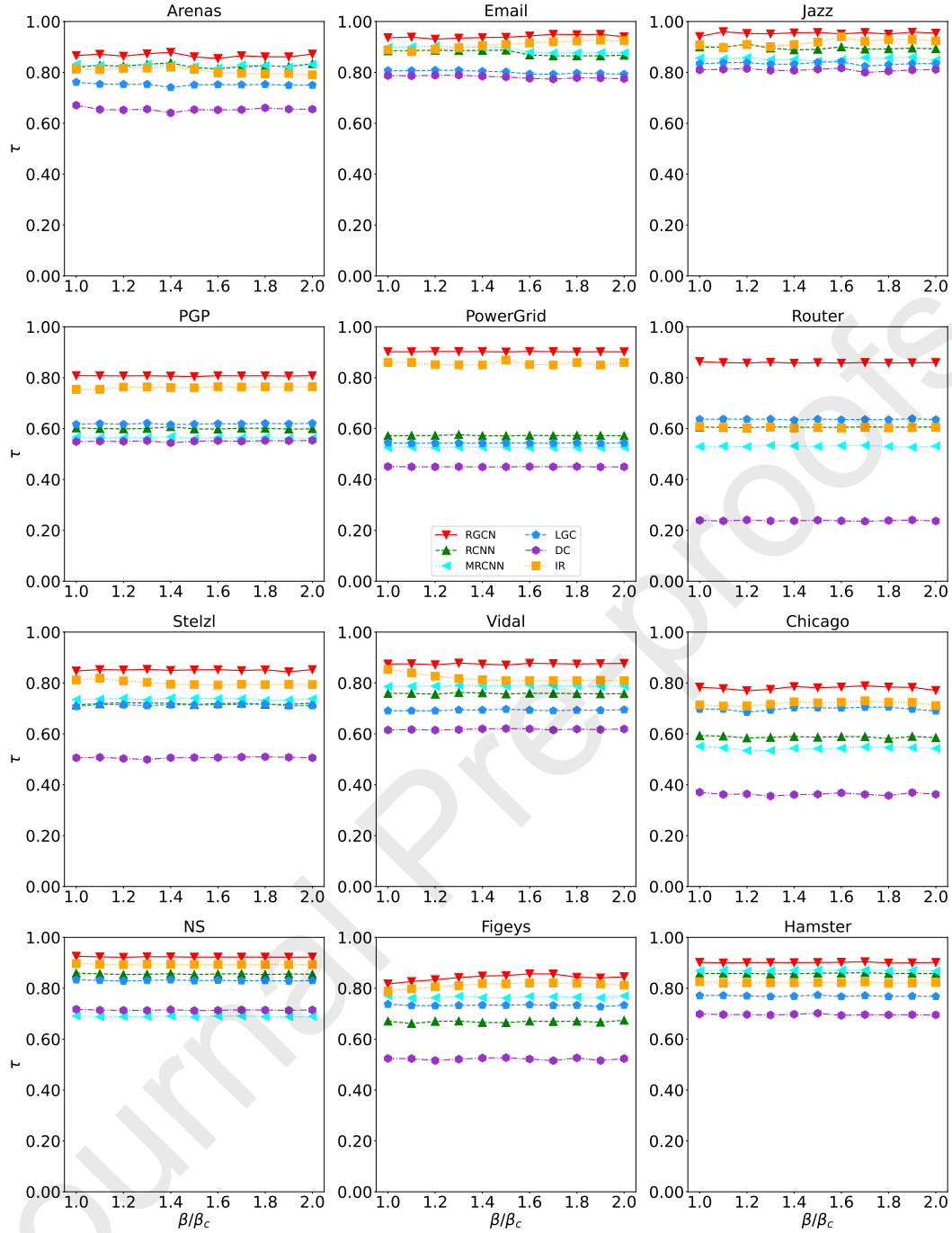
	$n$	$m$	$\langle k \rangle$	$k_{max}$	$c$	$H$	$G_{cv}\%$	$G_{ce}\%$	$d$
Arenas	453	2025	8.9	237	0.6465	4.5258	100	100	0.0198
Email	1133	5451	9.6	71	0.2202	1.9511	100	100	0.0085
Jazz	198	2742	8.9	237	0.6468	1.3948	100	100	0.0198
PGP	10680	24316	8.9	237	0.6468	4.0631	100	100	0.0198
PowerGrid	4941	2025	8.9	237	0.6468	1.4174	100	100	0.0198
Router	5022	2025	8.9	237	0.6468	5.4690	100	100	0.0198
Stelzl	1702	2025	8.9	237	0.6468	4.5557	100	100	0.0198
Vidal	3023	2025	8.9	237	0.6468	3.7373	100	100	0.0198
Chicago	1467	2025	8.9	237	0.6468	2.9568	100	100	0.0198
NS	1461	2025	8.9	237	0.6468	1.8037	100	100	0.0198
Figeys	2239	2025	8.9	237	0.6468	9.9034	100	100	0.0198
Hamster	2426	2025	8.9	237	0.6468	3.1058	100	100	0.0198

### 4.3. Baselines

(1) RCNN. Convolutional Neural Network (CNN) is commonly used in the field of image data convolution. The inspiration for the proposal of RCNN comes from the convolution of image data. For each node, a two-dimensional feature matrix  $L \times L$  is constructed, on which two convolutions and two maximum pooling operations are performed.

$$MI = \left( 1 - \frac{\sum_{v \in R_v} N_v(N_v - 1)}{N(N - 1)} \right)^2 \quad (9)$$

where  $N$  represents the number of nodes in the network,  $R_v$  is the list of sorted evaluation scores for all nodes, and  $N_v$  denotes the number of nodes that have a score of  $v$ .



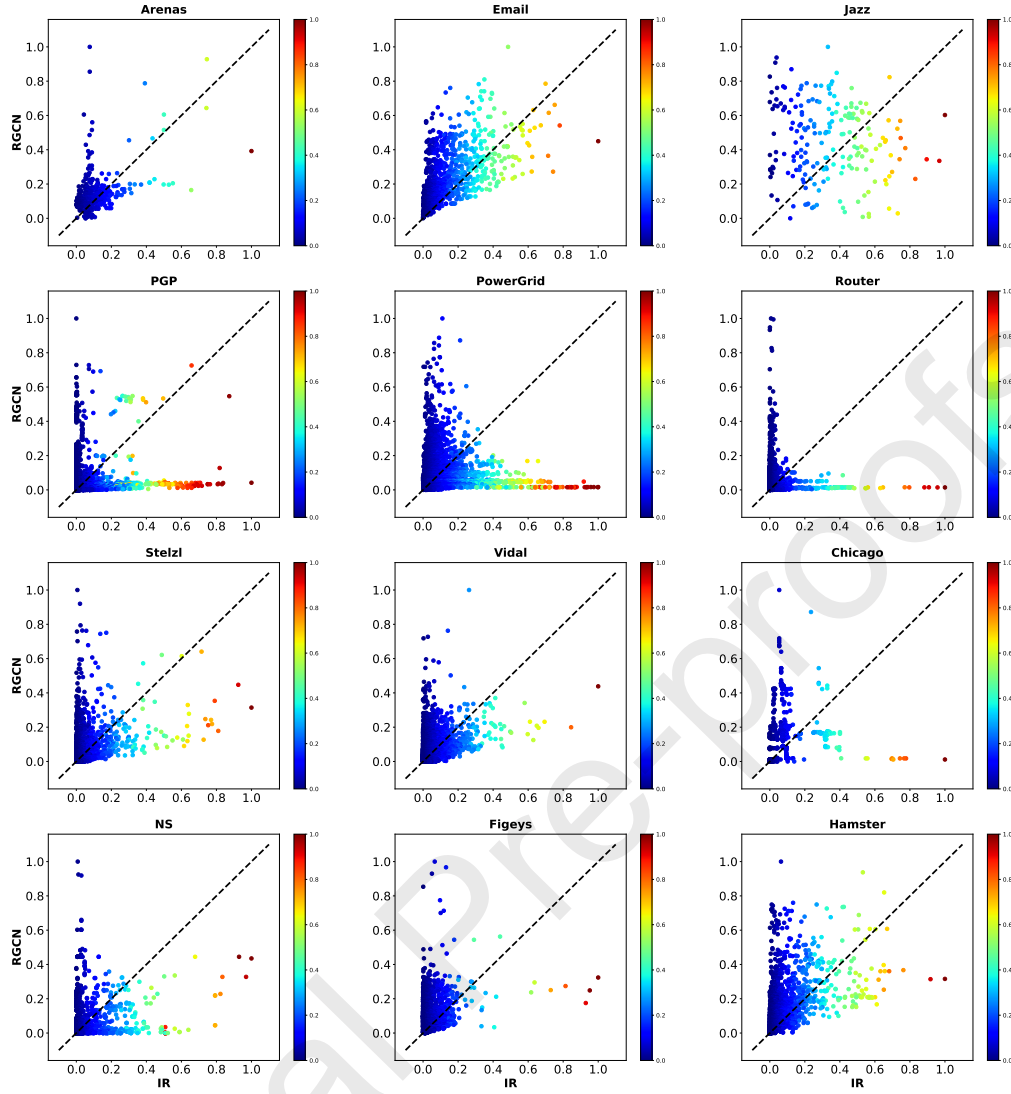
**Fig. 5:** The impact of changing infection rates on the Kendall  $\tau$  coefficient of RGCN in recognizing node rankings compared to baseline facts with a fixed sub-network size  $L$ .

Finally, a fully connected (FC) layer is used to complete the regression and realize the assessment of the importance of each node. The composition of its feature matrix is shown as follows in the formula.

$$\mathbf{B}_{i,j}^u = \begin{cases} \mathbf{A}_{0,j}^u d_{u_j} & i = 0, j = 1, 2, \dots, L-1 \\ \mathbf{A}_{i,0}^u d_{u_i} & i = 1, 2, \dots, L-1, j = 0 \\ d_{u_i} & i, j = 0, 1, \dots, L-1 \\ \mathbf{A}_{i,j}^u & \text{other case} \end{cases} \quad (10)$$

where  $\mathbf{A}^u$  is the adjacency matrix constructed from the set of neighbors of node  $u$ , including the first-order neighbors, second-order neighbors, and so on until enough nodes are selected.  $\mathbf{B}^u$  implies the feature matrix extracted from the adjacency matrix of node  $u$ . Where  $d_u$  denotes the degree of node  $u$  in the original network ( $u$  itself is the first element, followed by the degrees of the first-order neighbor nodes arranged in descending order and then the second-order neighbor nodes), and the input feature matrix size is  $L \times L$ .





**Fig. 6:** The impact of changing infection rates on the Kendall  $\tau$  coefficient of RGCN in recognizing node rankings compared to baseline facts with a fixed sub-network size  $L$ .

(2) MRCNN. MRCNN is an improved algorithm proposed based on the RCNN approach with added features of community and macroscopic level, which describes a feature matrix of nodes. In contrast to RCNN's single-channel feature, MRCNN has multiple levels of feature matrices of  $3 \times L \times L$  dimensionality. Each layer of the feature matrix is constructed in a similar manner to RCNN.

(3) Local-and-Global Centrality (LGC). This method takes into account both local and global topological information of complex networks to calculate node centrality. The LGC of a node  $u$  is defined as follows:

$$LGC(u) = \frac{d_u}{N} \times \sum_{u \neq v} \frac{\sqrt{d_v + \alpha}}{d(u, v)} \quad (11)$$

where  $d_u$  represents the degree of node  $u$ ,  $N$  is the total number of nodes in the network,  $\alpha$  is a tunable parameter between 0 and 1, which controls the contribution of nodes  $v$  outside of node  $u$  in measuring centrality value based on their

degrees. Finally,  $d(u, v)$  is the shortest path length between nodes  $u$  and  $v$ .

(4) Degree Centrality (DC) describes the local centrality of a node in a network. It is defined in Eq. 1.

(5) InformationRank (IR) is an algorithm for evaluating the importance of nodes in complex networks that takes into consideration path diversity. The algorithm calculates path lengths based on each specific network and considers information propagation probability to rank node importance. The main formula is as follows:

$$IR(u) = \sum_{w \in \mathcal{N}(u)^L} (1 - \prod_{l=1}^L (1 - \mu^l))^{|P(u, w)^l|} \quad (12)$$

where  $\mathcal{N}(u)^l$  is the set of neighboring nodes that can be accessed by node  $u$  with a path length of  $L$ . The  $\mu$  implies probability of information transmission between two nodes.  $|P(u, w)^l|$  expresses the number of paths from  $u$  to  $w$  with a path length of  $l$ . In this paper, we use this algorithm to

calculate the information propagation ability of node  $u$  in its subnetwork as a node feature.

## 5. Results and Discussion

### 5.1. Parameter Analysis

As a preliminary experiment we investigated how parameters influence the ability of RGCN to detect key nodes. Firstly, we are interested in quantifying the impact of the subgraph size  $L$  and, to this end, we artificially generated 15 networks using the Barabasi-Albert (BA) model; in our experiments we varied the number of network nodes  $N$  as well as the average degree  $D$ .

Nine of these networks are used as training sets (Train-N-D), while the remaining six are used as test sets (Test-N-D). Additionally, the embedding matrix is evaluated for fluctuations in identifying key nodes under varying infection rates. Figure 3 illustrates the changes in Kendall's  $\tau$  coefficient between the ranking of nodes obtained using different node features of RGCN with varying subgraph sizes ( $L$ ) and the ground truth ranking.

In our experiments we varied the size  $L$  of the subnetwork from 5% to 50% of the size of the original network, with an increment equal to 5% of network size. In our experiments we maintained a fixed infection rate  $\beta_i/\beta_c = \beta/\beta_c = 1.5$ . The propagation capacity information of the subnetwork is obtained as a feature for key node identification. Our experimental results suggest that  $L$  slightly affects the identification of key nodes. We achieved the best performance in the training set when  $L$  ranges from 0.2 to 0.4. Specifically, we observed that the best performance on the test datasets occurred if we trained RGCN on the Train\_2000\_5 dataset. When the fixed subnetwork node quantity was equal to 25% of the size of the original network.

We then explored the variation of the Kendall  $\tau$  coefficient between the RGCN-identified key nodes ranking and the true ranking with changing infection rate. Obtained results are reported in Figure 4.

If we opt for Train\_2000\_5 as the training set, RGCN performs well in most cases and is less affected by changes in infection rate and subnetwork size.

Given these results, we set  $L$  equal to 25% of the (input) network size and we comparatively analysed how the ratio of  $\beta/\beta_c$  affects the ability of the RGCN method and its competitors to generate correct node rankings. In particular, we examined how the Kendall  $\tau$  coefficient between the true ranking and the ranking produced by each method varied as a function of  $\beta/\beta_c$  for each of the 12 real datasets. In our tests we used the Train\_2000\_5 training parameters. Obtained results are plotted in Figure 5.

We report a high correlation at any position on the heatmap at any infection rate. Overall, changing the infection rate has little effect on RGCN. The Kendall  $\tau$  coefficients of RGCN on the five datasets Test\_3000\_2 to Test\_6000\_5 are all above 0.8, and the change is not significant. On the other four datasets, the Kendall  $\tau$  coefficient is above 0.6, and the change corresponds to the change in infection rate which is also not significant.

Table 2 presents the performance of RGCN and baseline methods in terms of MI index on twelve datasets.

From Table 2, it can be observed that RGCN shows better discrimination performance than the other five baseline methods on Arenas, Jazz, and Stelzl datasets in terms of score rankings. On Email and PowerGrid datasets, RGCN's discrimination is comparable to RCNN, MRCNN, and IR. While on the remaining five datasets, RGCN's discrimination is not optimal but still comparable to the best-performing algorithms. DC as the most primitive node importance identification method, its performance and now the optimal traditional method LGC and IR compared to the important node identification results of differentiation lags behind a lot. At the same time, it can also be seen that the node importance results identified by deep learning methods are well differentiated.

### 5.2. Comparative Analysis

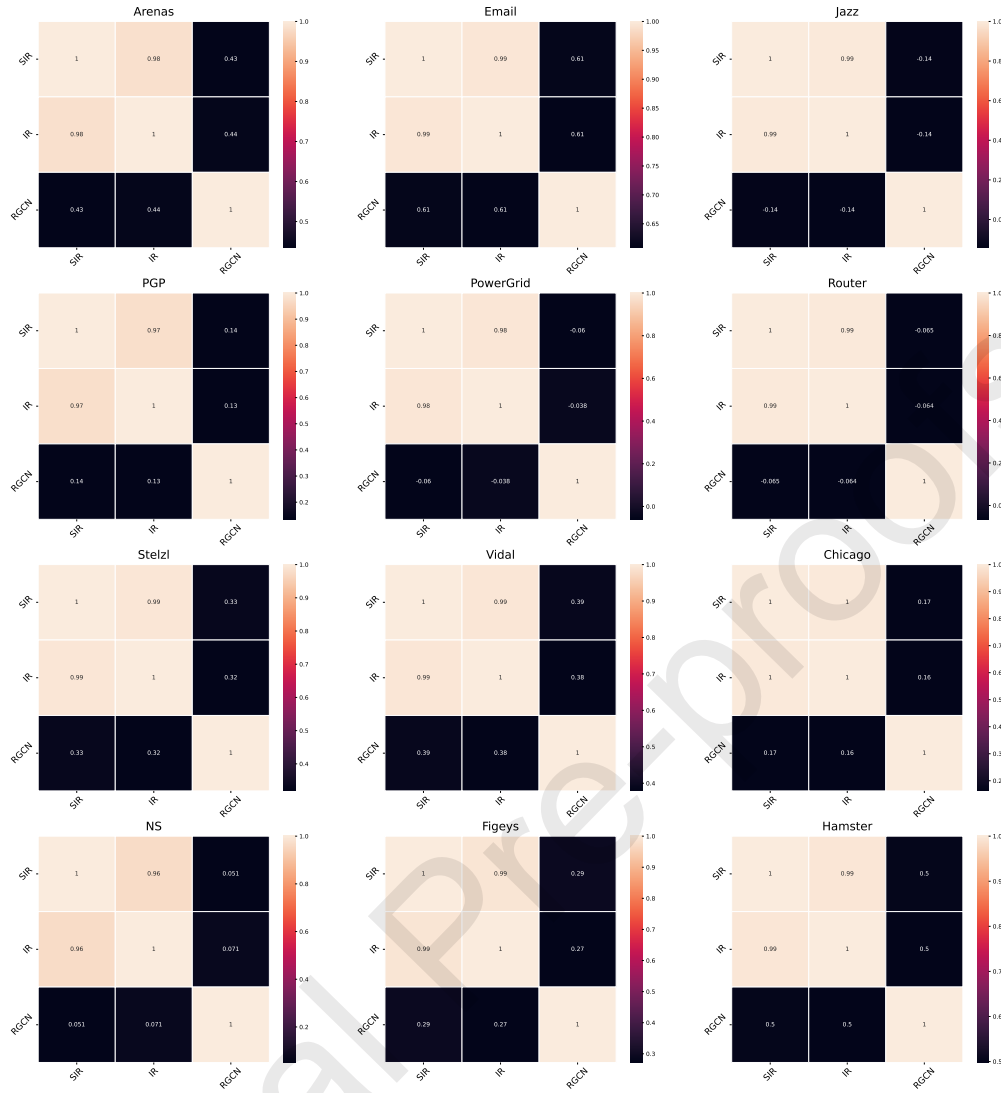
We conducted experiments to compare the node prediction scores of RGCN and IR on 12 networks with baseline SIR simulation results. As traditional IR method is used to calculate the sub-network node features and the results are taken as input layer's raw features for complex network key node identification. The distribution and correlation between the predicted scores of RGCN and IR to that of SIR simulated ground truth are evaluated. Fig. 6 and 7. shows the distributions of node score predictions by RGCN and IR respectively, along with their Pearson correlation coefficients with the ground truth scores obtained from SIR simulation performed 1000 times.

When observing the combination of Fig.6. and Fig.7., it can be seen that on the Email and Hamster datasets, the correlations between RGCN and IR are as high as 0.61 and 0.5, respectively. Also, the score distributions of RGCN and IR in the Email and Hamster datasets in Fig.6. are very close to the middle dashed line. On the other hand, the correlations on Jazz, Power Grid, and Router datasets are negative. The remaining datasets show relatively low correlations. When combined with Figs.5. and 6., it can be seen that although the Kendall  $\tau$  coefficient of IR's ranking of nodes importance shows good performance compared to the baseline truth, there is a high linear correlation between IR's node score and the true score, proving that its calculation method is strongly related to the calculation method of the baseline facts.

In contrast, the node ranking produced by RGCN has a good performance in terms of Kendall  $\tau$  coefficients. However, the node scores obtained from the RGCN algorithm have a very weak linear correlation with the true score and IR score, proving that when calculating the importance of nodes, RGCN used some traditional methods to calculate part of the initial features in IR, but it is completely different from the subsequent processing of node features and the regression of node scores, that is, it used new scoring systems to obtain results closer to the baseline facts.

### 5.3. Complexity Analysis

We thus present the worst-case time complexity analysis of our RGCN method. The time complexity is  $O((n-1)k^2)$



**Fig. 7:** The impact of changing infection rates on the Kendall  $\tau$  coefficient of RGCN in recognizing node rankings compared to baseline facts with a fixed sub-network size  $L$ .

where  $k$  is the average degree of nodes. And obtaining neighboring nodes requires traversing each node's neighbors, so it takes  $O(k)$  time, and after  $n - 1$  times, the total time is  $O((n - 1)k)$ . The operation of selecting the top  $n - 1$  nodes based on their degree requires computing the degree on all nodes at worst, and sorting  $n - 1$  nodes, and both operations have a complexity of  $O(k) + O(n \log n)$ . Regarding space complexity, creating a subgraph depends on its size  $L$  and the degree  $k$  of nodes. Therefore, the worst-case space complexity could be  $O(n^2)$ , where  $n$  is the number of nodes in the original network. In this paper, we select a subgraph size of 25% of the original network nodes, which is not very large, so the space complexity can be considered relatively low.

## 6. Main Limitations of the RGCN method

In this section, we discuss some potential limitations of our approach and outline some possible research avenues to address these limitations.

First, we have implicitly assumed that key nodes coincide with the nodes with the highest capacity to disseminate information in a network, under the assumption that the information dissemination process follows the SIR model. In particular, key nodes could be related to network vulnerability, i.e. we could define key nodes as those nodes that, when removed from the network, lead to a rapid loss of connectivity (e.g. they induce the fragmentation of a connected network into several isolated subcomponents or they lead to an unbinding of the network diameter). It would therefore be interesting to investigate whether the nodes that our method identifies as key nodes are also the nodes that have the greatest impact on the network's vulnerability.

**Table 2**

The MI index performance of the ranking results obtained by RGCN and the five baseline methods on 12 datasets.

	RGCN	RCNN	MRCNN	LGC	DC	IR
Arenas	0.9989	0.9977	0.9984	0.9980	0.7922	0.9991
Email	0.9999	0.9999	0.9999	0.9999	0.8874	0.9999
Jazz	0.9999	0.9992	0.9994	0.9994	0.9659	0.9995
PGP	0.9996	0.9997	0.9997	0.9997	0.6193	0.9997
PowerGrid	0.9999	0.9999	0.9999	0.9999	0.5927	0.9999
Router	0.9965	0.9965	0.9969	0.9969	0.2886	0.9969
Stelzl	0.9961	0.9956	0.9956	0.9958	0.5269	0.9957
Vidal	0.9944	0.9953	0.9944	0.9946	0.6299	0.9944
Chicago	0.8958	0.9098	0.9200	0.8974	0.0530	0.8939
NS	0.9178	0.9238	0.9171	0.9172	0.7069	0.9172
Figeys	0.9940	0.9946	0.9938	0.9953	0.5928	0.9939
Hamster	0.9857	0.9859	0.9857	0.9857	0.8980	0.9858

A second interesting aspect to investigate relates to the procedure used to construct the subnetworks. We followed a greedy approach to the construction of a subnetwork: in fact, starting from node  $x$ , we progressively include the neighbours of  $x$  and, in the construction process, we favour high-degree nodes; this choice is motivated by the fact that the higher the degree of a node, the greater the ability of that node to infect other nodes. Consequently, high-degree nodes should be preserved to maximise the chances of spreading information. However, real networks typically have a small diameter (i.e. we only need to move a few steps away from node  $x$  to cover a significant portion of the entire network) and few high-degree nodes (*hubs*); due to the small diameter value, a hub node could appear simultaneously in several subnetworks. From this reasoning, many of the subnetworks generated by the algorithm 2 could be similar because they contain the same nodes.

Finally, in our experiments we used a large number (twelve) of real networks to get a fair evaluation of the performance of our method.

Our approach heavily depends on the representation ability of a GCN but we do not know whether some specific network features, such as the existence of a modular structure, can affect the performance of our approach. It may be interesting to consider synthetic network generation systems, such as Google's recently released GraphWorld (Palowitch, Tsitsulin, Mayer and Perozzi, 2022), in order to maximise the variety of datasets to be experimented with, and thus provide a more detailed experimental evaluation.

In our experiments, we used the *Kendall's  $\tau$  coefficient* to assess the degree of correlation between the rankings generated by a given method and the true ranking (i.e. the ranking obtained from the SIR model). As for future work, it might be interesting to determine whether several, independent, ranking methods agree on the importance they assign to a *specific subset of nodes*  $N' \subseteq N$ ; more specifically if  $m$  ranking methods are available, a future research avenue consists to determine if there exists a group  $N'$  of nodes that are *consistently* ranked high by each of the  $m$  methods above. If this were true, then  $N'$  would provide a

reliable list of candidates from which we could choose our key nodes. Friedman's test (Friedman, 1937) is an excellent way to perform such an analysis, and to this purpose, we are thinking of applying the procedure outlined in some very recent papers (Kasihmuddin, Jaludin, Mansor, Wahab and Ghadzi, 2022; Zamri, Azhar, AsyrafMansor, Alway and Kasihmuddin, 2022) to gain a better understanding of the accuracy of our method in correctly identifying key nodes.

## 7. Conclusions

This paper introduced the RGCN (Rank by Graph Convolutional Network) method to identify key nodes in complex networks with limited node feature information and training data.

RGCN introduces a new feature construction method by means of information diffusion probabilities and sub-networks. Experiments on 15 synthetic networks and 12 real-world networks (6 fully connected and 6 non-fully connected) demonstrate that RGCN can improve accuracy even with reduced feature matrix dimensions, outperforming traditional manual feature engineering methods and two other deep learning approaches in terms of accuracy judged by Kendall  $\tau$  coefficient. Meanwhile, Pearson coefficient illustrates the significant differences between RGCN and the Information Rank, IR method that computes sub-network features. Good performance in distinguishing key nodes is also shown through MI indices. Parameter analysis indicates that RGCN can achieve good results under fixed sub-network sizes and infection rates with low computational complexity suitable for large-scale networks. As an approach based on deep learning, RGCN also has poorer interpretability.

In the future, we will try to combine RGCN with RNN networks to handle key node identification in dynamic complex networks.

## References

- Adamic, L.A., Lukose, R.M., Puniyani, A.R., Huberman, B.A., 2001. Search in power-law networks. *Physical Review E* 64, 046135.



- Bader, D.A., Kintali, S., Madduri, K., Mihail, M., 2007. Approximating betweenness centrality, in: Proc. of the International Workshop on Algorithms and Models for the Web-Graph (WAW 2007), Springer, San Diego, CA, USA. pp. 124–137.
- Bae, J., Kim, S., 2014. Identifying and ranking influential spreaders in complex networks by neighborhood coreness. *Physica A: Statistical Mechanics and its Applications* 395, 549–559.
- Barabási, A.L., Pósfai, M., 2016. *Network science*. Cambridge University Press, Cambridge. URL: <http://barabasi.com/networksciencebook/>.
- Brandes, U., Pich, C., 2007. Centrality estimation in large networks. *International Journal on Bifurcation and Chaos* 17, 2303–2318.
- Cao, J., Hao, J., Lai, X., Vong, C., Luo, M., 2016. Ensemble extreme learning machine and sparse representation classification. *Journal of the Franklin Institute* 353, 4526–4541.
- Chen, D.B., Sun, H.L., Tang, Q., Tian, S.Z., Xie, M., 2019. Identifying influential spreaders in complex networks by propagation probability dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29, 033120.
- Cohen, E., Dellling, D., Pajor, T., Werneck, R.F., 2014. Computing classic closeness centrality, at scale, in: Proc. of the ACM conference on Online social networks, (COSN 2014), ACM, Dublin, Ireland. pp. 37–50.
- Duch, J., Arenas, A., 2005. Community detection in complex networks using extremal optimization. *Physical review E* 72, 027104.
- Eash, R., Chon, K., Lee, Y., Boyce, D., 1979. Equilibrium traffic assignment on an aggregated highway network for sketch planning. *Transportation Research* 13, 243–257.
- Ewing, R.M., Chu, P., Elisma, F., Li, H., Taylor, P., Climie, S., McBroom-Cerajewski, L., Robinson, M.D., O'Connor, L., Li, M., et al., 2007. Large-scale mapping of human protein–protein interactions by mass spectrometry. *Molecular systems biology* 3, 89.
- Fan, C., Zeng, L., Sun, Y., Liu, Y.Y., 2020. Finding key players in complex networks through deep reinforcement learning. *Nature machine intelligence* 2, 317–324.
- Freeman, L.C., et al., 2002. Centrality in social networks: Conceptual clarification. *Social network: critical concepts in sociology*. Londres: Routledge 1, 238–263.
- Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association* 32, 675–701.
- Gleich, D.F., 2015. Pagerank beyond the Web. *Siam Review* 57, 321–363.
- Gleiser, P.M., Danon, L., 2003. Community structure in jazz. *Advances in complex systems* 6, 565–573.
- Gong, M., Ji, S., Xie, Y., Gao, Y., Qin, A.K., 2022. Exploring temporal information for dynamic network embedding. *IEEE Transactions on Knowledge and Data Engineering* 34, 3754–3764.
- Grando, F., Granville, L.Z., Lamb, L.C., 2019. Machine learning in network centrality measures: Tutorial and outlook. *ACM Computing Surveys* 51, 102:1–102:32.
- Grover, A., Leskovec, J., 2016. node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864.
- Guimera, R., Danon, L., Diaz-Guilera, A., Giral, F., Arenas, A., 2003. Self-similar community structure in a network of human interactions. *Physical review E* 68, 065103.
- Huang, H., Sun, L., Du, B., Liu, C., Lv, W., Xiong, H., 2021. Representation learning on knowledge graphs for node importance estimation, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 646–655.
- Huang, X., Chen, J., Cai, M., Wang, W., Hu, X., 2022. Traffic node importance evaluation based on clustering in represented transportation networks. *IEEE Transactions on Intelligent Transportation Systems* 23, 16622–16631.
- Kasihmuddin, M.S.M., Jaludin, S.Z.M., Mansor, M.A., Wahab, H., Ghadzi, S.M.S., 2022. Supervised learning perspective in logic mining. *Mathematics* 10, 915.
- Kunegis, J., 2013. Konect: the koblenz network collection, in: Proceedings of the 22nd international conference on world wide web. pp. 1343–1350.
- Lawyer, G., 2015. Understanding the influence of all nodes in a network. *Scientific Reports* 5, 8665.
- Leydesdorff, L., 2007. Betweenness centrality as an indicator of the interdisciplinaryity of scientific journals. *Journal of the American Society for Information Science and Technology* 58, 1303–1319.
- Li, Q., Zhou, T., Lü, L., Chen, D., 2014. Identifying influential spreaders by weighted leaderrank. *Physica A: Statistical Mechanics and its Applications* 404, 47–55.
- Liu, B., Jiang, S., Zou, Q., 2020. Hits-pr-hhblits: protein remote homology detection by combining pagerank and hyperlink-induced topic search. *Briefings in bioinformatics* 21, 298–308.
- Liu, C., Cao, T., Zhou, L., 2022a. Learning to rank complex network node based on the self-supervised graph convolution model. *Knowledge-Based Systems* 251, 109220.
- Liu, X., Gao, L., Fiumara, G., De Meo, P., 2022b. Key node identification method integrating information transmission probability and path diversity in complex network. *The Computer Journal*.
- Liu, X., Ye, S., Fiumara, G., De Meo, P., 2023. Influential spreaders identification in complex networks with TOPSIS and k-shell decomposition. *IEEE Transactions on Computational Social Systems* 10, 347–361.
- Newman, M., 2018. *Networks*. Oxford university press.
- Newman, M.E., 2006. Finding community structure in networks using the eigenvectors of matrices. *Physical review E* 74, 036104.
- Ou, Y., Guo, Q., Xing, J.L., Liu, J.G., 2022. Identification of spreading influence nodes via multi-level structural attributes based on the graph convolutional network. *Expert Systems with Applications* 203, 117515.
- Palowitch, J., Tsitsulin, A., Mayer, B., Perozzi, B., 2022. Graphworld: Fake graphs bring real insights for gnns, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 3691–3701.
- Park, N., Kan, A., Dong, X.L., Zhao, T., Faloutsos, C., 2019. Estimating node importance in knowledge graphs using graph neural networks, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 596–606.
- Perozzi, B., Al-Rfou, R., Skiena, S., 2014. Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710.
- Rossi, R., Ahmed, N., 2015. The network data repository with interactive graph analytics and visualization, in: Proceedings of the AAAI conference on artificial intelligence. pp. 4292–4293.
- Rual, J.F., Venkatesan, K., Hao, T., Hirozane-Kishikawa, T., Dricot, A., Li, N., Berriz, G.F., Gibbons, F.D., Dreze, M., Ayivi-Guedehoussou, N., et al., 2005. Towards a proteome-scale map of the human protein–protein interaction network. *Nature* 437, 1173–1178.
- Sariyüce, A.E., Kaya, K., Saule, E., Catalyürek, U.V., 2013. Incremental algorithms for closeness centrality, in: IEEE International Conference on Big Data. pp. 487–492.
- Spring, N., Mahajan, R., Wetherall, D., 2002. Measuring isp topologies with rocketfuel. *ACM SIGCOMM Computer Communication Review* 32, 133–145.
- Stelzl, U., Worm, U., Lalowski, M., Haenig, C., Brembeck, F.H., Goehler, H., Stroedicke, M., Zenkner, M., Schoenherr, A., Koeppen, S., et al., 2005. A human protein–protein interaction network: a resource for annotating the proteome. *Cell* 122, 957–968.
- Sun, M., Zhao, S., Gilvary, C., Elemento, O., Zhou, J., Wang, F., 2020. Graph convolutional networks for computational drug development and discovery. *Briefings in bioinformatics* 21, 919–935.
- Ullah, A., Wang, B., Sheng, J., Long, J., Khan, N., Sun, Z., 2021. Identifying vital nodes from local and global perspectives in complex networks. *Expert Systems with Applications* 186, 115778.
- Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of ‘small-world’ networks. *nature* 393, 440–442.
- Wen, X., Tu, C., Wu, M., Jiang, X., 2018. Fast ranking nodes importance in complex networks based on ls-svm method. *Physica A: Statistical Mechanics and its Applications* 506, 11–23.
- Yu, E.Y., Wang, Y.P., Fu, Y., Chen, D.B., Xie, M., 2020. Identifying critical nodes in complex networks via graph convolutional networks.

- Knowledge-Based Systems 198, 105893.
- Yu, L., Li, G., Yuan, L., 2021. Compatible influence maximization in online social networks. *IEEE Transactions on Computational Social Systems* 9, 1008–1019.
- Zamri, N.E., Azhar, S.A., AsyrafMansor, M., Alway, A., Kasihmuddin, M.S.M., 2022. Weighted random k satisfiability for  $k=1, 2$  (r2sat) in discrete Hopfield neural networks. *Applied Soft Computing* 126, 109312.
- Zhang, J., Li, Y., Xiao, W., Zhang, Z., 2020. Non-iterative and fast deep learning: Multilayer extreme learning machines. *Journal of the Franklin Institute* 357, 8925–8955.
- Zhang, J., Zhao, Y., Shone, F., Li, Z., Frangi, A.F., Xie, S.Q., Zhang, Z.Q., 2022a. Physics-informed deep learning for musculoskeletal modelling: Predicting muscle forces and joint kinematics from surface emg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*.
- Zhang, M., Wang, X., Jin, L., Song, M., Li, Z., 2022b. A new approach for evaluating node importance in complex networks via deep learning methods. *Neurocomputing* 497, 13–27.
- Zhao, G., Jia, P., Zhou, A., Zhang, B., 2020. Infgc: Identifying influential nodes in complex networks with graph convolutional networks. *Neurocomputing* 414, 18–26.
- Zhao, J., Wen, T., Jahanshahi, H., Cheong, K.H., 2022. The random walk-based gravity model to identify influential nodes in complex networks. *Information Sciences* 609, 1706–1720.
- Zhong, S., Zhang, H., Deng, Y., 2022. Identification of influential nodes in complex networks: A local degree dimension approach. *Information Sciences* 610, 994–1009.
- Zhou, X., Liang, W., Wang, K.I.K., Huang, R., Jin, Q., 2021. Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data. *IEEE Transactions on Emerging Topics in Computing* 9, 246–257.

### **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.

## Author Agreement

Submission of work requires that the piece to be reviewed has not been previously published. Upon acceptance, the Author assigns to the Journal of King Saud University – Computer and Information Sciences (JKSUCI) the right to publish and distribute the manuscript in part or in its entirety. The Author's name will always be included with the publication of the manuscript.

The Author has the following nonexclusive rights: (1) to use the manuscript in the Author's teaching activities; (2) to publish the manuscript, or permit its publication, as part of any book the Author may write; (3) to include the manuscript in the Author's own personal or departmental (but not institutional) database or on-line site; and (4) to license reprints of the manuscript to third persons for educational photocopying. The Author also agrees to properly credit the Journal of King Saud University – Computer and Information Sciences (JKSUCI) as the original place of publication.

The Author hereby grants the Journal of King Saud University – Computer and Information Sciences (JKSUCI) full and exclusive rights to the manuscript, all revisions, and the full copyright. The Journal of King Saud University – Computer and Information Sciences (JKSUCI) rights include but are not limited to the following: (1) to reproduce, publish, sell, and distribute copies of the manuscript, selections of the manuscript, and translations and other derivative works based upon the manuscript, in print, audio-visual, electronic, or by any and all media now or hereafter known or devised; (2) to license reprints of the manuscript to third persons for educational photocopying; (3) to license others to create abstracts of the manuscript and to index the manuscript; (4) to license secondary publishers to reproduce the manuscript in print, microform, or any computer-readable form, including electronic on-line databases; and (5) to license the manuscript for document delivery. These exclusive rights run the full term of the copyright, and all renewals and extensions thereof.

I hereby accept the terms of the above Author Agreement.

Author :- Wuyuan Gao xiangyang Liu

Date :- May 20, 2023

Francisco J. M. de M. e

Editor in Chief:- Nasser-Eddine Rikli

Date:-