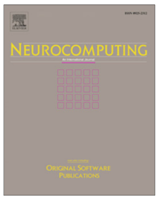
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



InfGCN: Identifying influential nodes in complex networks with graph convolutional networks

Gouheng Zhao[a], Peng Jia[a,*], Anmin Zhou[a], Bing Zhang[b]

[a] College of Cybersecurity, Sichuan University, Chengdu, China
[b] Beijing Institute of Technology, Beijing, China

ARTICLE INFO

Article history:
Received 18 April 2020
Revised 6 July 2020
Accepted 16 July 2020
Available online 22 July 2020
Communicated by Wenguan Wang

Keywords:
Influential nodes identification
Deep learning
Graph convolution networks
Complex networks

ABSTRACT

Identifying influential nodes in a complex network is very critical as complex networks are ubiquitous. Traditional methods, such as centrality based methods and machine learning based methods, only consider either network structures or node features to evaluate the significance of nodes. However, the influential importance of nodes should be determined by both network structures and node features. To solve this problem, this paper proposes a deep learning model, named InfGCN, to identify the most influential nodes in a complex network based on Graph Convolutional Networks. InfGCN takes neighbor graphs and four classic structural features as the input into a graph convolutional network for learning nodes' representations, and then feeds the representations into the task-learning layers, comparing the ground truth derived from Susceptible Infected Recovered (SIR) simulation experiments with quantitative infection rate. Extensive experiments on five real-world networks of different types and sizes demonstrate that the proposed model significantly outperforms traditional methods, and can accurately identify influential nodes.

1. Introduction

Identifying influential nodes is a subject with great significance in network science for its wide applications including finding influential spreaders in social networks [1], predicting essential proteins [2], quantifying scientific influences [3], detecting financial risks [4], and predicting career movements [5], etc. Network science is a novel interdisciplinary research field and can be utilized to describe systems with graph structures. Previous researches have revealed various properties of complex networks such as rich-club [6], scale-free [7] and core/periphery structures [8]. These properties show that only a small number of nodes play the important roles in a complex network. Thus it is essential to identify the most influential nodes. Many previous works have been done to describe the importance of nodes from certain aspects such as centrality based methods in terms of physical structural importance, and machine learning based methods in terms of node features. These methods have their limitations. For Centrality based methods, there is a mismatch between the importance of structures and the importance of functions. The reason is that many existing methods measure the importance of nodes from the perspective of network structures, so the importance of nodes based on single structure information cannot fully reflect the functional importance in the influence scenario. For machine learning based methods, they depend too much on feature engineering. Feature selection can significantly affect the performance of machine learning based methods. In fact, the influential importance of nodes is determined by not only their features but the links between their neighbors.

Graph Convolutional Networks (GCN) [9] are recently prevailing deep learning models, which can process both node features and the links between nodes. The application of GCN has achieved great success in many research fields including chemistry and drug design [10], social networks [11], as well as natural language processing [12]. One of the important advantages of GCN models is the ability to process graph-structured data which can be used as representations of a large number of systems across various areas. These characteristics make it possible to apply GCN models to handle some tasks in complex networks.

In this paper, the top 5% influential nodes are considered as the most influential nodes, and other nodes as the less influential nodes. By doing this, the evaluation of nodes importance is transformed into a classification task and a deep learning framework named InfGCN is proposed, which could process both node features and links between nodes. In the framework, four classic

* Corresponding author.
E-mail address: pengjia@scu.edu.cn (P. Jia).

https://doi.org/10.1016/j.neucom.2020.07.028
0925-2312/© 2020 Elsevier B.V. All rights reserved.

structural features are first selected, and fixed-size neighbor networks of nodes are got through Breadth First Search (BFS), and then graph convolution technology is utilized to learn latent predictive signals. Finally, the predictive signals are fed into task-learning layers, comparing the ground truth derived from Susceptible Infected Recovered (SIR) [13] simulation experiments with a quantitative infection rate. The effectiveness and efficiency of our proposed framework are demonstrated on 5 real-world networks of different types and different sizes, by comparing InfGCN with some traditional methods such as centrality based methods and machine learning based methods. Experimental results suggest that InfGCN can significantly improve prediction performance.

The rest of this paper is organized as follows. In Section 2, the related works are presented systematically. Section 3 introduces the proposed framework in detail. In Section 4, a method to construct datasets are proposed. In Sections 5 and 6, extensive experiments are set up and conducted. Finally, Section 7 concludes this paper.

2. Related work

Researchers have introduced a lot of methods to identify influential nodes in complex networks such as information entropy [14], propagation probability dynamics [15], etc. In general, these methods can be categorized into two classes: centrality based methods and machine learning based methods. Besides, as a hot subject in the deep learning community, Graph Convolutional Networks (GCN) have begun to be studied by more and more researchers. Therefore, Our work refers to a large amount of literature on centrality based methods, machine learning based methods and graph convolutional networks.

2.1. Centrality based methods

A lot of works have been done to design the centrality based methods based on part of topological structures, which have the limitations on performance and flexibility. On the whole, existing methods can be categorized into four types [16]: (1) distance based methods, which are based on the shortest distance related to nodes, such as betweenness centrality [17] and closeness centrality [18]; (2) neighbor-relationship based methods, which are based on the number of neighbor nodes, such as degree centrality [19] and semi-local centrality [20]; (3) iteration based methods, whose essence is to calculate the importance of neighbors to measure nodes themselves, such as Eigenvector centrality [21] and PageRank centrality [22]; (4) nodes-operation based methods, which are to observe the change after removing or merge nodes. Theses methods just consider a certain structure in general.

2.2. Machine learning based methods

Researches of these methods mainly focus on the feature engineering [23], and selecting better features to get better performance [24,25]. The machine learning algorithms commonly used in this field include Logistic Regression (LR) and Support Vector Machine (SVM), etc. These methods have been applied to many fields to find important nodes in specific scenarios [26,27], which has laid a solid foundation for the development of this filed. However, the machine learning based methods depend too much on the feature selection of the target objects and ignore the relationships between the targets.

2.3. GCN based methods

Graph convolutional Networks have been a hot topic in research communities. Researchers around the world have proposed many GCN methods. Some of these works are done to identify critical nodes in complex networks [28], and address learning the relative importance or attention between nodes [29,30]. These methods can be categorized into two types: (1) spectral-based GCN methods: these methods introduce the filters to define the convolutional kernel from the perspective of signal processing, such as Bruna et al. [31] and Henaff et al. [32]; (2) spatial based GCN methods: these methods define graph convolutional kernel as the representations of aggregating feature information from the neighbors, such as ChebNet [33] and GraphSAGE [34]. In general, the essence of these methods is to aggregate the information of neighbors iteratively, meaning these methods considering both node features and the links between their neighbor nodes.

3. Deep learning model

In this section, we formally propose the framework of a deep learning model, InfGCN, to identify the most influential nodes in complex networks. Firstly, we sample fixed-size neighbor networks (see Section 3.1) and construct feature vectors consisting of 4 classic centralities for each node. Next we feed the neighbor network and feature vector of each node into a Graph Convolutional Networks (GCN) layer for representation learning and then three Fully Connected (FC) layers for task learning with mini-batch learning (see Section 3.2). Finally, the outputs of the model are compared with the ground truth derived from SIR simulation experiments to minimize the negative log-likelihood loss. Meanwhile, we pre-train the model to overcome the problem of insufficient data on small networks (see Section 3.3).

3.1. Constructing neighbor networks

The influential capability of a node tends to be determined by its neighbor nodes. Besides, previous works [35] have revealed that in GCN models the $(k+1)$th representation of a node is related to the kth representation of the neighbors, and this local property causes the kth representation of the node to be only related to its k-step network, namely the neighbor network. Therefore, it is essential to get the neighbor network of a node to measure the importance of the node. One effective way to construct the neighbor network of a node is to perform Breadth First Search (BFS) starting from it to get its neighbors, and then induce the neighbor network by these neighbors. However, the sizes of neighbor networks of different nodes may be different, which is unsuited to mini-batch learning of the most deep learning models. To remedy these issues, we specify a fixed number F as the size of the neighbor network of each node in a network. For each step i of BFS, we get the i th step neighbors of the target node, and then calculate the total number (named t_i) of neighbors that are not longer than ith step from the target. If t_i is smaller than the fixed number F, then go to the next step. If t_i is bigger than F, then filter the ith step neighbors according to their betweenness centrality, discard the ones with smaller betweenness and hold the ones with bigger betweenness. The reason for selecting betweenness centrality as the criterion is that betweenness centrality reflects the bridge role of nodes, and the bridge nodes tend to play a more important role in the process of information spreading.

3.2. Graph convolutional networks based model

With retrieving the neighbor network of each node, we design an effective Graph Convolutional Networks based model to process graph-structured data. As shown in Fig. 1, the proposed deep learning model consists of an input layer, a GCN layer, three Fully Connected (FC) layers, and an output layer. Next, we introduce these layers one by one and build the model step by step.

3.2.1. Input layer

The input layer constructs symmetric normalized Laplacian of the neighbor network and feature vector of each node. To avoid depending on the feature engineering too much, this paper just selects some classic centralities as node features to reflect the structural properties of nodes: (1) degree; (2) closeness centrality; (3) betweenness centrality; (4) clustering coefficient [36]. The descriptions of the selected features are presented in Table 1. Meanwhile, to avoid over-fitting, we normalize the features. For each feature k, the process of the normalization is:

$$f_k = \frac{P_k}{S} - 0.5 \tag{1}$$

where f_k denotes the ranking position in the network according to the value of the centrality feature k, and S denotes the number of nodes in the network. Through the normalization, the value of each feature is transformed between -0.5 and 0.5. **GCN layer** Graph Convolutional Networks (GCN) [37] are semi-supervised algorithms to use the graph structures and feature vectors to learn representation vectors of nodes. GCN layer is defined as follows:

$$H^{i+1} = \sigma(AH^iW^i + b^i) \tag{2}$$

where A is the symmetric normalized Laplacian of the neighbor network, H^i denotes the representations of nodes at the ith GCN layer, W^i and b^i are trainable weights and bias parameters respectively, and σ is the nonlinear activation function which we set as the Exponential Linear Units (ELU) [38] function. H^0 is the feature vectors of neighbor nodes in the input layer. Besides, to make better use of the node features, we add the skip connection [39] in the GCN layer. Meanwhile, to avoid over-fitting, we apply Dropout [40] technology in this layer.

3.2.2. FC layers

The down-stream layers of the GCN layer are three fully connected (FC) layers for the task learning. Each of the FC layers is followed by Elu non-linear function. The first two FC layers are applied to Dropout technology to avoid over-fitting.

3.2.3. Output layer and loss function

Finally, the outputs of the fully connected layers are fed into the LogSoftMax classifier. We compare the classification results with the ground truth derived from SIR simulation experiments mentioned in Section 4 and optimize the negative log-likelihood loss.

3.3. Pre-training the model

Deep learning requires a lot of data. For small networks, there are a few nodes, meaning not enough data for deep learning models to classify nodes. To overcome this problem and make our model fit with small networks, we learn from the ideas of transfer learning [41] technology and pre-train our model. Transfer learning refers to the setting where a model, initially trained on some tasks, is re-purposed on different but related tasks. Deep transfer learning has been immensely successful in computer vision and natural language processing. In this paper, we generalize pre-training to graph data. Although different networks are of different types and different sizes, they all are of network structures, which makes it possible to apply pre-training. Firstly, our model is pre-trained with the datasets of large networks and then fine-tuned with the training data of small networks. Finally, the model can predict nodes in small networks.

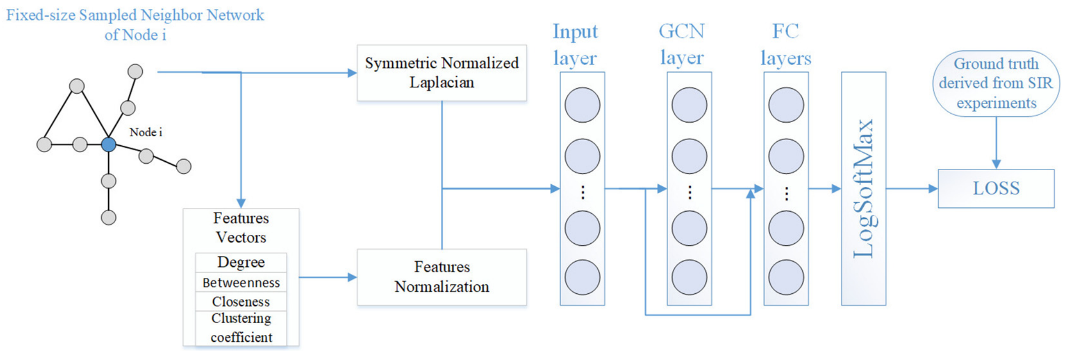


Fig. 1. Framework of InfGCN. (1) The raw input consists of symmetric normalized laplacian and node features; node features consist of four classic structural properties which are normalized before being fed into the deep learning model; (2) GCN layer is for representation learning using graph structures and node features; Meanwhile, skip connection is added in this layer to make better use of node features; (3) Three Fully Connected (FC) layers and LogSoftMax classifier are for task learning; (4) Compare the model output and the ground truth, we get the negative log likelihood loss.

Table 1
The descriptions of the selected features.

Features	Descriptions
Degree	Measuring the number of the neighbors of a node
Betweenness	Measuring the number of the shortest paths through a node, and it describes the bridge role of a node in a network.
Closeness	Measuring the total length of the shortest path between a node and other nodes, and it describes the broadcaster role of a node in a network.
clustering coefficient	Measuring the degree of clustering between the neighbors of a node

4. Method to construct datasets

In practical work, most complex networks lack nodes' labels for influential capability. Therefore, some researchers have used SIR model to simulate the influence of nodes, which is what this paper adopts. However, previous works have not quantitatively set the infection rate in SIR experiments. In this paper, we introduce the concept of discrimination and the metric to select the most appropriate infect rate. In this section, we present the relevant definition and formulation to measure the influential capabilities of nodes in a network.

Definition 1. Influential Capability We use SIR model to simulate the influential capabilities of nodes in a network. SIR model is an abstract description of the process of information transmission. It divides all nodes into three states: susceptible, infectious, and recovered. Infectious nodes can infect susceptible neighbors with infection rate β, and also can turn into a recovered state with recovery rate λ. Once recovered, the nodes can't be infected or infect others again. The whole process is dynamic and random. The simulation experiment is to set a node as the initial infected one and others as susceptible ones, and then use the final scale of the outbreak as the influential capability of the node. The final scale of the outbreak denoting the final number of infected and recovered nodes means the number of nodes influenced in SIR simulation experiments, and reflects the influential capability of the initially infected node. A node with a larger outbreak is more influential. For a network, we conduct SIR experiments on each of the nodes to get their influential capabilities. In our experiments, we set the recovery rate λ to be 1. The top 5% influential nodes are considered as the most influential nodes in a network.

Problem 1. Discrimination In SIR simulation experiments, different values of infection rate β have an important impact on measuring nodes' influential capabilities. The epidemic threshold of a real-world network tends to be too small. However, an too small or too large infection rate cannot discriminate the influential capabilities of nodes well, which means most of their final scales of the outbreaks are all too small or too large. Therefore we do not select the epidemic threshold as the infection rate and introduce the concept of discrimination [42] to get the most appropriate infection rate. Discrimination is often used to measure the discrimination capability of psychological assessment. The larger value, the better discrimination capability. The common index of discrimination is D, and the formulation of D is given as follows:

$$D = \frac{XH - XL}{N(H - L)} \tag{3}$$

where XH and XL denote the total influential capabilities of the high influence group and the low influence group respectively, H and L denote the highest influential capability and the lowest influential capability respectively, and N denotes the percentage of the high influence group which is always set to 27%. Given infection rate β, through SIR simulation experiments, we can get influential capabilities of all the nodes in a network and calculate the value of D. When the value is the largest, the infection rate β is the most appropriate to discriminate against the influential capabilities of nodes in a network.

5. Experiment setup

We introduce five real-world networks as experimental networks and selecte some evaluation metrics to quantitatively evaluate our proposed framework. To evaluate our proposed framework better, we use classic machine learning based methods and centrality based methods as comparisons. Relevant implement details are also shown in this section.

5.1. Experimental networks

To verify the effectiveness of our model, we select 5 real-world networks of different types and different sizes: (1) Hamsterster Friendships [43], which is a network containing friendships between users of the website Hamsterster.com; (2) Human protein (Vidal) [44,45], which represents an initial version of a proteome-scale map of Human binary protein–protein interactions; (3) CA-GrQc [46], which is a collaboration network of Arxiv General Relativity; (4) CA-HepTh [46], which is a collaboration network of Arxiv High Energy Physics; (5) CA-CondMat [46], which is a collaboration network of Arxiv Condensed Matter. These networks all are undirected and unweighted, and the basic topological properties of the networks are shown in Table 2.

Different networks have different topological structures. Therefore, when selecting the most appropriate infection rate to discriminate influential nodes, the values of D of different infection rates of different networks may be different [47]. To get the most appropriate infection rate, we set the infection rate between 0.04 and 0.2, and calculate the values of D of different infection rates. In particular, for the power-law property of real-world networks and to better focus on more influential nodes, we do not consider the nodes whose degree are less than 3 when calculating the values of D. The results are shown in Fig. 2. From Fig. 2, we can get the most appropriate infection rates of different networks to discriminate influential nodes, and the values are presented in Table 3. Meanwhile, a problem comes from label imbalance. In this paper, the top 5% nodes are considered the most influential nodes in a network while 95% of nodes are not, leading to serious imbalance. To address this issue, we sample the less influential nodes, and construct a more balanced dataset with the ratio between negative and positive to be 2:1. Especially, to simplify our experiments, we only consider nodes whose degree is lager than 2.

5.2. Evaluation metrics

To evaluate our framework quantitatively, we use two metrics as follows: **Prediction Performance**: We evaluate the predictive performance of our framework in terms of Area Under Curve (AUC) [48], Precision, Recall and F1-Measure (F1).

Table 2
The basic topological properties of the experimental networks.

Network	Nodes	Edges	diameter	Maximum degree	Power law exponent
Hamsterster Friendships	1858	12534	14	272	1.82
Human protein (Vidal)	3133	6726	13	129	1.82
CA-GrQc	5242	14496	17	81	1.33
CA-HepTh	9877	25998	17	65	1.13
CA-CondMat	23133	93497	14	279	2.00

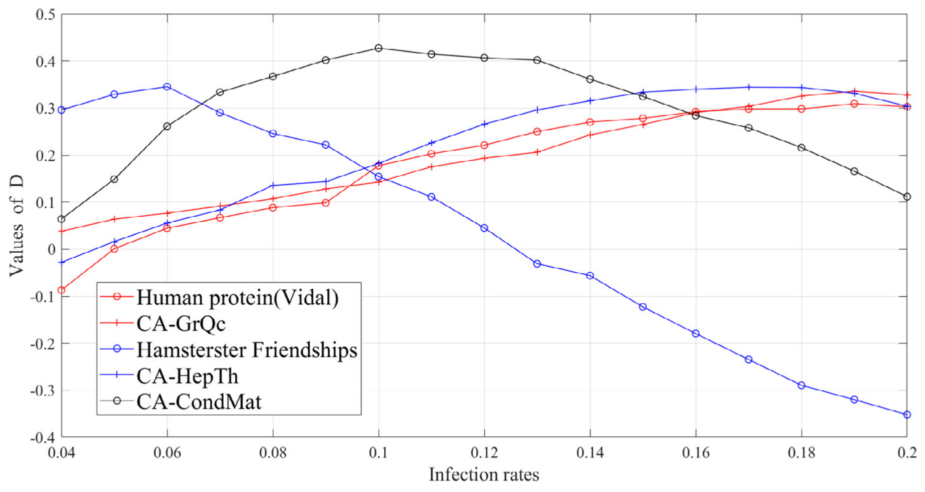


Fig. 2. The values of D with different infection rates.

Table 3
The most appropriate infection rates of different networks.

Networks	Largest values of D	Infection rates
Hamsterster Friendships	0.345	0.06
Human protein (Vidal)	0.309	0.19
CA-GrQc	0.336	0.19
CA-HepTh	0.344	0.17
CA-CondMat	0.417	0.10

Parameter sensitively: We analyze some important hyper-parameters and investigate how different hyper-parameters can influence prediction performance.

5.3. Comparison methods and implement details

5.3.1. Comparison methods

We compare the proposed framework with two classical machine learning based methods: Logistic Regression (LR) and Support Vector Machine (SVM) [49]. Machine learning based methods consider node features. Besides, this paper chooses some classical centrality based methods as a comparison: (1) degree centrality; (2) betweenness centrality; (3) closeness centrality; (4) PageRank centrality. Centrality based methods consider the structures of nodes in networks. For centrality based methods, the top 5% influential nodes are classified as the most influential nodes according to the nodes' values of the centrality based methods. Therefore, according to the definitions of Precision, Recall, and F1, we can get the performance of the centrality based methods. In particular, the AUC metric cannot be used to evaluate centrality based methods according to the definition of AUC.

5.3.2. Implement details

In our framework, we firstly sample fixed-size neighbor networks, and the initial size of the sub-network is set to be 50. There is one GCN layer in our model which contains 8 hidden units. As for the fully connected (FC) layers, three FC layers contain 16, 8, 2 units respectively. All parameters are trained using the Adam [50] optimizer with learning rate 0.0001 and weight decay $1e^{-4}$. The mini-batch size and dropout rate are set to be 32 and 0.2 respectively. In addition, to simplify the experiments, we remove the nodes whose degree is less than 3 and are always the less influential nodes in a network. Finally, we run the model at most 200 epochs, and the best model was selected by early stopping. These implement details are the same across all the data sets.

6. Experimental results and analyses

In this section, we conduct a series of experiments to verify the effectiveness of our proposed framework. For each network, we randomly divide nodes into 70 percent as the training data and 30 percent as the testing data. Among the experimental networks, CA-GrQc, Human protein (Vidal) and Hamsterster Friendships are small networks. Therefore, we use transfer learning technology in these networks, and pre-train the model in the largest network CA-CondMat before training and testing the model in three small networks.

6.1. Prediction performance

We compare the prediction performance of all the methods across the five datasets in Table 4. From the results, we have several interesting observations and insights:

(1) As shown in Table 4, InfGCN model outperforms the baselines in terms of both AUC and F1 in general, demonstrating the effectiveness of our proposed framework. The machine learning based methods are the second-best methods, and the centrality based methods perform the worst in general. Note that among the centrality based methods, degree performs very well. Even in Human protein (Vidal), the performance of degree centrality is the best in terms of F1. It demonstrates that the simplest method also can perform well.
(2) In the experiments, we still rely on several vertex features, such as degree and clustering coefficient. In fact, we want to avoid using any hand-crafted features and just select a few classic structure features of nodes. Quite surprisingly, compared with the machine learning based methods, the InfGCN model can still achieve comparable performance. It suggests that the InfGCN model which considers both structures and node features performs better than the machine learning based methods which only consider node features.
(3) For centrality based methods, except degree centrality, other centrality based methods perform very badly. As a contrast, the machine learning based methods considering 4 structure features perform better. The reason is that centrality based methods measure the importance of nodes from the perspective of network structures, so the importance of nodes based on single structure information cannot fully reflect the functional importance in the influence scenario.

Table 4
Prediction results on 5 real-world datasets. Best results are in bold.

Networks	Method	AUC (%)	Precision (%)	Recall (%)	F1 (%)
CA-CondMat	Degree	–	89.0	59.8	71.6
	Betweenness	–	13.5	13.7	13.6
	Closeness	–	24.8	7.4	11.4
	PageRank	–	26.8	6.3	10.2
	LR	92.2	88	63.4	73.7
	SVM	91.2	**90.6**	55.6	68.9
	InfGCN	**95.5**	88.8	**75.1**	**81.4**
CA-HepTh	Degree	–	**93.0**	69.7	79.7
	Betweenness	–	16.5	11.8	13.8
	Closeness	–	37.2	10.5	16.4
	PageRank	–	32.5	8.6	13.5
	LR	94.8	91.7	67.8	78
	SVM	95.1	89.1	78.1	83.2
	InfGCN	**98.1**	91	**90.3**	**90.7**
CA-GrQc	Degree	–	83.3	63.4	72.0
	Betweenness	–	29.5	39.4	33.7
	Closeness	–	44	15.5	22.9
	PageRank	–	50	14.1	22
	LR	93.3	84.7	63.3	72.5
	SVM	93.0	85.7	38	52.6
	InfGCN	**97.4**	**85.1**	**84.0**	**85.9**
Human protein (Vidal)	Degree	–	79.2	**73.1**	**76.0**
	Betweenness	–	71.4	19.2	30.3
	Closeness	–	68.8	21.2	32.4
	PageRank	–	76.9	19.2	30.8
	LR	91.7	91.3	44.7	60
	SVM	90.6	**99.0**	38.3	55.4
	InfGCN	**93.9**	77.8	63.7	70
Hamsterster Friendships	Degree	–	88.9	59.3	71.1
	Betweenness	–	58.3	25.9	35.9
	Closeness	–	53.8	25.9	35.0
	PageRank	–	58.3	25.9	35.9
	LR	95.1	86.2	73.5	79.4
	SVM	95.3	88.5	67.6	76.7
	InfGCN	**98.1**	**96.4**	**84.4**	**90.3**

6.2. Parameter analysis

In this section, we select several important hyper-parameters and investigate the effects of different choice of these hyper-parameters on the performance of the deep learning model.

6.2.1. Size of the neighbor network

An important hyper-parameter is the size of the neighbor network. Fig. 3 shows the performance (in terms of AUC and F1) of our model by varying the size from 10 to 90. From the results, it could be seen that the size of the neighbor network has different effects on different networks: (1) For CA-CondMat network, the size of the neighbor network has almost no effect on both AUC and F1. (2) For CA-HepTh, Human protein (Vidal) and Hamsterster Friendships networks, with the size of the neighbor network increasing, both AUC and F1 firstly increase and then decrease in general. (3) For CA-GrQc network, with the size of the neighbor network increasing, both AUC and F1 firstly decrease, and increase, and then decrease, and finally stay stable. Although the performance of the model is different for different networks or different sizes of the neighbor network, the performance is still good in general. The problem is how to adjust the size of the neighbor network

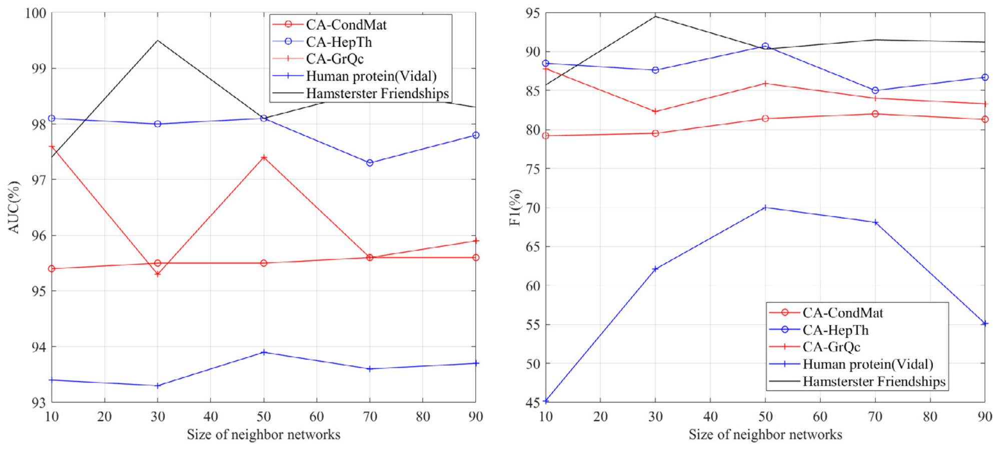


Fig. 3. Effect of the size of the neighbor network on the performance of our proposed model.

to get better performance. The solution is setting the size to a medium value like 50 at first, and then fine-tuned to get better performance. In essence, the size of the neighbor network is a kind of hyper-parameter.

6.2.2. Effect of pre-training

To solve the problem of no enough training data and make it possible to predict nodes in small networks, we pre-trainied the model with the data of the large network and fine-tuned with the training data of the small networks. In this part, we investigate the effect of pre-training in terms of AUC and F1 in CA-GrQc, Human protein (Vidal) and Hamsterster Friendships networks. From Fig. 4, it can be seen that pre-training significantly improve the performance of our model and the speed of convergence. Our work makes an important step toward transfer learning on graphs. There are many interesting avenues for future work. For example, further increasing generalization by improving deep learning architectures as well as pre-training and fine-tuning approaches, is a fruitful direction.

6.2.3. Effect of skip connection

In our model, to make better use of node features, we add Skip Connection (SC). In this part, we investigate the effect of SC on the performance of our model in terms of AUC and F1. As is shown in Table 5, skip connection significantly improve the performance of our model. In previous works, researchers used skip connection mainly to improve gradient dissipation during back propagation. However, our work mainly focuses on the function of improving the representational ability of the model. The results of the experiments also demonstrate that skip connection is worth further study.

6.2.4. Effect of feature normalization

As claimed in Section 3.2, to avoid over-fitting, we use Feature Normalization (FN). Data normalization processing is a basic work of data mining. Different evaluation indices tend to have different dimensions and dimensional units, and this will affect the result of data analysis. In order to eliminate the dimension influence between indicators, data normalization processing is needed and solves the comparability among the index data. After the original data is normalized, each index is in the same order of magnitude, which is suitable for comprehensive comparative evaluation. To investigate the effect of FN, we test the situations of using FN and not using FN. From Table 6, We can see that FN slightly improve the performance of our model in general. **Effect of the percentage of influential nodes** In our model, the top 5% influential nodes are considered as the most influential nodes in a network. To investigate the effects of different percentages of influential nodes selected, we consider top 1% top 5% top 10% and top 15% as the most influential nodes respectively. As shown in Table 7, it can be seen that except the condition of top 1%, the smaller the percentage is, the better our model performs. The result is meaningful and practical. However, under the condition of top 1%, the problem of insufficient data to train the model will arise and may degrade the performance of the model. Therefore, we need to balance these contradictions and select appropriate percentage like top 5% in our work.

Table 5
Effect of skip connection (SC) on the performance of our proposed model.

Networks	Model	AUC (%)	F1 (%)
CA-CondMat	InfGCN with SC	**95.5**	**81.4**
	InfGCN without SC	93.1	76.9
CA-HepTh	InfGCN with SC	**98.1**	**90.7**
	InfGCN without SC	94.1	87.5
CA-GrQc	InfGCN with SC	**97.4**	**85.9**
	InfGCN without SC	78.4	54.1
Human protein (Vidal)	InfGCN with SC	**93.9**	**70.0**
	InfGCN without SC	93.2	65.2
Hamsterster Friendships	InfGCN with SC	**98.1**	**90.3**
	InfGCN without SC	94.6	85.7

Table 6
Effect of feature normalization (FN) on the performance of our proposed model.

Networks	Model	AUC (%)	F1 (%)
CA-CondMat	InfGCN with FN	**95.5**	**81.4**
	InfGCN without FN	94.8	80.9
CA-HepTh	InfGCN with FN	**98.1**	**90.7**
	InfGCN without FN	98.0	89.2
CA-GrQc	InfGCN with FN	**97.4**	**85.9**
	InfGCN without FN	96.9	84.6
Human protein (Vidal)	InfGCN with FN	**93.9**	**70.0**
	InfGCN without FN	95.5	66.7
Hamsterster Friendships	InfGCN with FN	**98.1**	**90.3**
	InfGCN without FN	98.0	84.7

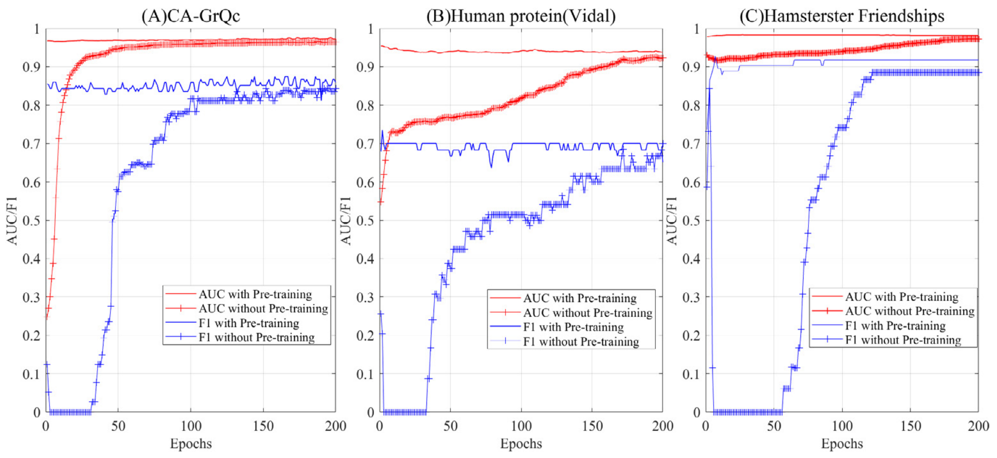


Fig. 4. Effect of pre-training on the performance of our proposed model.

Table 7
Effect of different Percentages of influential nodes selected on the performance of our proposed model.

Networks	Percentages of influential nodes selected	AUC (%)	F1 (%)
CA-CondMat	Top 1% Top 5% Top 10% Top 15%	**96.7** 95.5 94.3 92.0	**85.5** 81.4 79.5 74.8
CA-HepTh	Top 1% Top 5% Top 10% Top 15%	96.4 **98.1** 97.8 96.3	77.8 **90.7** 86.4 83.9
CA-GrQc	Top 1% Top 5% Top 10% Top 15%	**99.8** 97.4 95.3 94.8	**93.8** 85.9 80.8 76.9
Human protein (Vidal)	Top 1% Top 5% Top 10% Top 15%	91.0 **93.9** 90.5 84.4	66.7 **70.0** 52.1 35.6
Hamsterster Friendships	Top 1% Top 5% Top 10%Top 15%	81.8 **98.1** 96.1 96.0	40.0 **90.3** 50.0 40.0

7. Conclusion

Previous papers reveal that there are only a small number of nodes which are the most influential in complex networks. To identify these nodes is of great significance for its wide applications. In this work, we investigated the problem of identifying the most influential nodes in complex networks. Traditional methods, such as centrality based methods or machine learning based methods, consider either network structures or node features, which limits the performance of these methods. To solve this problem, we transformed the most influential nodes identification problem into a classification problem and proposed a deep learning model, named InfGCN, for identifying the most influential nodes. A method to construct datasets is proposed which conducts SIR simulation experiments with quantitative infection rates. To evaluate the effectiveness of our proposed model, this paper conducted a series of experiments in five real-world networks. The experimental results show our model significantly outperforms baselines in identifying the most influential nodes in complex networks.

Declaration of Competing 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.

CRediT authorship contribution statement

Gouheng Zhao: Conceptualization, Methodology, Software, Writing - original draft. **Peng Jia:** Visualization, Investigation, Writing - review & editing, Supervision. **Anmin Zhou:** Software, Validation. **Bing Zhang:** Project administration.

References

[1] M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, H.A. Makse, Identification of influential spreaders in complex networks, Nat. Phys. 6 (11) (2010) 888–893.
[2] E. Estrada, Virtual identification of essential proteins within the protein interaction network of yeast, Proteomics 6 (1) (2010) 35–40.
[3] T. Heinze, PhilipShapira, J.D. Rogers, J.M. Senker, Organizational and institutional influences on creativity in scientific research, Res. Policy 38 (4) (2009) 610–623.
[4] M. Gallegati, S. Keen, T. Lux, P. Ormerod, Worrying trends in econophysics, Phys. A Stat. Mech. Appl. 370 (1) (2006) 1–6.
[5] R. Heimler, S. Rosenberg, E. Morote, Predicting career advancement with structural equation modelling, Educ. Train. 54 (2/3) (2012) 85–94.
[6] A. Flammini, V. Colizza, M. Serrano, A. Vespignani, Rich-club ordering in complex networks, Nat. Phys. 2 (2) (2006) 110–115.
[7] R. Pastor-Satorras, A. Vespignani, Epidemic spreading in scale-free networks, Phys. Rev. Lett. 86 (14) (2001) 3200–3203.
[8] S.P. Borgatti, M.G. Everett, Models of core/periphery structures, Social Networks 21 (4) (2000) 375–395.
[9] W.L. Hamilton, R. Ying, J. Leskovec, Representation learning on graphs: methods and applications, IEEE Data(base) Eng. Bull. 40 (2017) 52–74.
[10] N.D. Cao, T. Kipf, Molgan: an implicit generative model for small molecular graphs, ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models, 2018.
[11] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, J. Tang, Deepinf: Social influence prediction with deep learning, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery Data Mining, 2018, pp. 2110–2119.
[12] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, K. Sima'an, Graph convolutional encoders for syntax-aware neural machine translation, empirical methods in natural language processing, (2017) 1957–1967.
[13] W.O. Kermack, A.G. McKendrick, A contribution to the mathematical theory of epidemics, Proc. Roy. Soc. Lond. 115 (772) (1927) 700–721.
[14] C. Guo, L. Yang, X. Chen, D. Chen, H. Gao, J. Ma, Influential nodes identification in complex networks via information entropy, Entropy 22 (2) (2020).
[15] D.-B. Chen, H.-L. Sun, Q. Tang, S.-Z. Tian, M. Xie, Identifying influential spreaders in complex networks by propagation probability dynamics, Chaos Interdisc. J. Nonlinear Sci. 29 (3) (2019).
[16] L. Lü, D. Chen, X. Ren, Q.-M. Zhang, Y.-C. Zhang, T. Zhou, Vital nodes identification in complex networks, Phys. Rep. 650 (07 2016).
[17] Ulrik Brandes, A faster algorithm for betweenness centrality*, J. Math. Sociol. 25 (2) (2004) 163–177.
[18] K. Okamoto, W. Chen, X.Y. Li, Ranking of closeness centrality for large-scale social networks, Int. Workshop Front. Algorithm. (2008).
[19] T. Opsahl, Degree centrality in a weighted network, Phys. Rev. E Stat. Nonlinear Soft Matter Phys. 77 (4 Pt 2) (2008), 046105.
[20] X. Zhao, F. Liu, J. Wang, T. Li, Evaluating influential nodes in social networks by local centrality with a coefficient, Isprs Int. J. Geo Inf. 6 (2) (2017) 35–39.
[21] P. Bonacich, Some unique properties of eigenvector centrality, Social Networks 29 (4) (2007) 555–564.
[22] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, Comput. Networks ISDN Syst. (1998) 107–117.
[23] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: a new perspective, Neurocomputing 300 (jul.26) (2018) 70–79.
[24] G. Ramirez-De-La-Rosa, E. Villatoro-Tello, H. Jiménez-Salazar, C. Sánchez-Sánchez, Towards automatic detection of user influence in twitter by means of stylistic and behavioral features, Mexican International Conference on Artificial Intelligence, 2014.
[25] M. Nouh, J.R. Nurse, Identifying key-players in online activist groups on facebook social network, in: IEEE 15th International Conference on Data Mining (ICDM) Workshop on Intelligence and Security Informatics, 2015, pp. 969–978.
[26] I. Inuwa-Dutse, M. Liptrott, I. Korkontzelos, Detection of spam-posting accounts on twitter, Neurocomputing 315 (2018) 496–511.
[27] X. Zheng, Z. Zeng, Z. Chen, Y. Yu, C. Rong, Detecting spammers on social networks, Neurocomputing 159 (2015) 27–34.
[28] E.-Y. Yu, Y.-P. Wang, Y. Fu, D.-B. Chen, M. Xie, Identifying critical nodes in complex networks via graph convolutional networks, Knowl.-Based Syst. 198 (2020), 105893.
[29] W. Wang, X. Lu, J. Shen, D. Crandall, L. Shao, Zero-shot video object segmentation via attentive graph neural networks, International Conference on Computer Vision, 2019, pp. 9236–9245.
[30] S. Qi, W. Wang, B. Jia, J. Shen, S. Zhu, Learning human-object interactions by graph parsing neural networks, European Conference on Computer Vision (2018).
[31] J. Bruna, W. Zaremba, A. Szlam, Y. Lecun, Spectral networks and locally connected networks on graphs, International Conference on Learning Representations, 2014.
[32] M. Henaff, J. Bruna, Y. LeCun, Deep convolutional networks on graph-structured data, in: Conference and Workshop on Neural Information Processing Systems, 2015.
[33] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, Neural Inf. Process. Syst. (2016) 3844–3852.
[34] W.L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, Neural Inf. Process. Syst. (2017) 1024–1034.
[35] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, M. Sun, Graph neural networks: A review of methods and applications, arXiv: Learning, 2019.
[36] P. Zhang, J. Wang, X. Li, M. Li, Z. Di, Y. Fan, Clustering coefficient and community structure of bipartite networks, Phys. A Stat. Mech. Appl. 387 (27) (2008) 6869–6875.
[37] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, International Conference on Learning Representations, 2017.
[38] T.U. Djork-Arné Clevert, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (elus), International Conference on Learning Representations, 2015.
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, IEEE Conference on Computer Vision and Pattern Recognition (2016).

[40] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, Neural Evol. Comput. 3 (4) (2012) 212–223.
[41] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.
[42] R.K. Hambleton, G. Arrasmith, D.S. Sheehan, R.P. Grobe, W. Hathaway, V. Doherty, D. Forbes, R. Houser, G. Ingebo, G. Kingsbury, Standards for educational and psychological testing: six reviews, J. Educ. Measure. 23 (1) (1986) 83–98.
[43] Hamsterster friendships network dataset – KONECT (Apr. 2017). http://konect.uni-koblenz.de/networks/petster friendships hamster.
[44] Human protein (vidal) network dataset – KONECT (Apr. 2017). URL http://konect.uni-koblenz.de/networks/maayan-vidal .
[45] J.-F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G.F. Berriz, F.D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou, Towards a proteome-scale map of the human protein–protein interaction network, Nature 437 (7062) (2005) 1173–1178.
[46] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: densification and shrinking diameters, ACM Trans. Knowl. Discov. Data 1 (1) (2007) 2–43.
[47] J.L. Peng Jia, Modeling and analyzing malware propagation in social networks with heterogeneous infection rates, Phys. A Stat. Mech. Appl. 507 (2018) 240–254.
[48] C. Buckley, E.M. Voorhees, Retrieval evaluation with incomplete information, in: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2004, pp. 25–32.
[49] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, C.J. Lin, Liblinear: a library for large linear classification, J. Mach. Learn. Res. 9 (9) (2008) 1871–1874.
[50] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations, 2014.

Gouheng Zhao received the B.Eng. degree from the College of Computer Science and Technology, Wuhan university of Science and Technology, Wuhan, China, in 2018. He is currently pursing the master's degree with the College of Cybersecurity, Sichuan University, Chengdu, China. His current research interests include network science, and artificial intelligence.

Peng Jia received the B.Eng. degree and Ph.D. degree from Sichuan University, Chengdu, China in 2012 and 2019 respectively. He is currently an assistant research fellow with the College of Cybersecurity, Sichuan University, China. His current research interests include bianry security, network science and malware propagation.

Anmin Zhou received the B.Eng. degree from the Northwest Institute of Telecommunication Engineering, Xi'an, China, in 1984. He is currently a Research Fellow with the College of Cybersecurity, Sichuan University, China. His current research interests include malicious detection, network security, system security, and artificial intelligence.

Bing Zhang received the master degree from Beijing Institute of Technology, in 2010. His current interests include information security.