IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DATA MINING FOR INTERNET OF THINGS

Received April 23, 2020, accepted May 14, 2020, date of publication May 20, 2020, date of current version June 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2996001

Deep Learning Techniques for Community Detection in Social Networks

LING WU[1], QISHAN ZHANG[2], CHI-HUA CHEN[1], (Senior Member, IEEE), KUN GUO[1], AND DEQIN WANG[1]

[1]College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China
[2]School of Economics and Management, Fuzhou University, Fuzhou 350108, China

Corresponding authors: Qishan Zhang (zhangqs@fzu.edu.cn) and Chi-Hua Chen (chihua0826@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61906043, Grant 61877010, Grant 11501114, and Grant 11901100, in part by the Fujian Natural Science Funds under Grant 2019J01243 and Grant 2018J01791, in part by the Funds of Fujian Provincial Department of Education under Grant JAT190026, in part by the Industry-Academy Cooperation Project under Grant 2017H6008, and in part by the Fuzhou University under Grant 510730/XRC-18075, Grant 510809/GXRC-19037, Grant 510649/XRC-18049, and Grant 510650/XRC-18050.

ABSTRACT Graph embedding is an effective yet efficient way to convert graph data into a low dimensional space. In recent years, deep learning has applied on graph embedding and shown outstanding performance. Adjacency matrix is often taken as the storage data structure of graph. However, there are the problems of insufficient spatial proximity information in adjacency matrix. Therefore, this study proposes a deep community detection method which includes (1) matrix reconstruction method, (2) spatial feature extraction method and (3) community detection method. The original adjacency matrix in social network is reconstructed based on the opinion leader and nearer neighbors for obtaining spatial proximity matrix. The spatial proximity matrix can obtain subspace of the graph which can help convolution neural network easily and quickly extract the spatial localization. The spatial eigenvector of reconstructed adjacency matrix can be extracted by an auto-encoder based on convolution neural network for the improvement of modularity. In experiments, four open datasets of practical social networks were selected to evaluate the proposed method, and the experimental results show that the proposed deep community detection method obtained higher modularity than other deep learning methods. Therefore, the proposed deep community detection method can effectively detect high quality communities in social networks.

INDEX TERMS Community detection, social network, convolutional neural network, auto-encoder.

I. INTRODUCTION

Graph is an important data representation of complex networks. Effective community detection provides users a deeper understanding of networks, and it can benefit a lot of useful applications such as node classification, node recommendation, link prediction etc. However, the large scale and high-dimensionality of networks make community detection method suffer from high computation and space cost [1], [2]. Graph embedding is an effective yet efficient way to convert graph data into a low dimensional space, in which the network structural information and network properties are maximally preserved [3]–[5].

Deep learning (DL) has shown outstanding performance in a wide variety of research fields, such as social networks, graph embedding, etc. DL based graph embedding applies DL models on network. At present, DL based graph embedding can be divided into two categories based on whether random walk is adopted to sample paths from a graph [3]. **(1) DL-based graph embedding with random walk**: a graph is represented as a set of random walk paths sampled from it. The deep learning methods are then applied to the sampled paths for graph embedding which preserves graph properties carried by the paths [4], [5]. **(2) DL-based graph embedding without random walk**: the methods applies deep models on a whole graph (or a proximity matrix of a whole graph) directly [6]–[9].

The graph is represented by an adjacency matrix. It stores the information of direct connections between nodes. The adjacency matrix does not store the proximity of indirect connections between nodes, nor can it express spatial proximity between nodes. So there is a problem of insufficient spatial

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

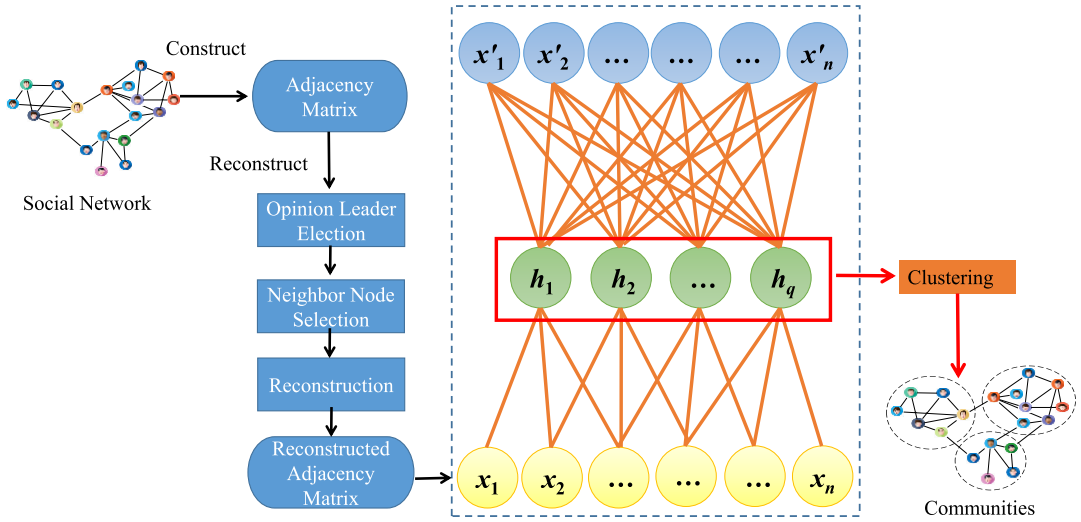


FIGURE 1. The proposed deep community detection method.

proximity information in the adjacency matrix. To solve this problem and improve the accuracy of feature extraction, some auto-encoder (AE) based graph embedding studies [9]–[15] optimized the input vector by performing different preprocessing on the adjacency matrix to improve the extraction accuracy of spatial feature. In [13], the sparse adjacency matrix is preprocessed, and the obtained similarity matrix can not only reflect the similarity between connected nodes in the network topology, but also reflect the similarity between unconnected nodes. What's more, the fully-connected strategy is applied for generating the auto-encoder networks, some noises may be obtained in the fully-connected networks when the unrelated nodes of social network exist. Considering convolutional neural network (CNN) is a powerful tool to extract the spatial localization, CNN is also applied in graph embedding [16]–[23]. It is necessary to simulate the convolution operation on the graph. Therefore, the problem is how to improve the adjacency matrix so that the adjacency matrix storage spatial proximity between nodes.

Therefore, analogous to image-based convolutional networks that operate on locally connected regions of the input, the study presents a reconstruction approach of adjacency matrix to storage spatial proximity between nodes. The spatial proximity can make the nodes that are spatially close to each other stored in the matrix close to each other.

Considering AE-based graph embedding and CNN-based graph embedding has obtained robustness and effectiveness, the study incorporates AE and CNN to improve the quality of feature extraction. Therefore, this study proposes a deep community detection method which includes (1) matrix reconstruction method, (2) AE and CNN-based spatial feature extraction method and (3) community detection method (shown in Figure 1). Due to limited cyberspace information in original adjacency matrix to present the structure of network topology, the proposed matrix reconstruction method based on a novel cyberspace structure reconstruction strategy is applied to find the opinion leader in the social network and find the nearer neighbors for reconstructing adjacency matrix with the spatial proximity matrix. Furthermore, this study also proposes a spatial feature extraction method based on auto-encoder and convolutional neural network to extract the spatial features of the reconstructed adjacency matrix. Finally, the extracted spatial features are adopted into the proposed community detection method for clustering the nodes of social network into k groups.

The contributions of this study are summarized as follows.

(1) A matrix reconstruction method based on a novel cyberspace structure reconstruction strategy is proposed to obtain the spatial proximity matrices of social networks, which can obtain subspace of the graph and help convolution neural network easily and quickly extract the spatial localization.

(2) A spatial feature extraction method based on auto-encoder and convolutional neural network is proposed to learn spatial eigenvector and effectively extract the spatial features of social networks.

(3) The matrix reconstruction method and spatial feature extraction method are applied to analyze the topology of social network for the improvement of social community detection.

The remainder of the paper is organized as follows. Section II discusses the literature reviews of deep neural network based network embedding methods. Section III presents the processes of the proposed deep community detection method to analyze the structure of social networks for extracting features of social networks and detecting social communities. The proposed methods will be evaluated, and practical experimental results will be analyzed and discussed in Section IV. The conclusions and future research directions are summarized in Section V.

II. RELATED WORK

It can be seen from above introduction that deep neural network based network embedding methods include two categories: (1) deep learning models with random walk and (2) deep learning models without random walk [3].

Deep neural network based network embedding methods with random walk are applied to the sampled paths for graph embedding which preserves graph properties carried by the paths. The most famous deep learning model with random walk is DeepWalk [4]. The success of DeepWalk motivates many subsequent studies which apply deep learning models, Word2Vec [5] follow the idea of DeepWalk but change the settings of random walk sampling methods.

Deep neural network based network embedding without random walk applies deep models on a whole graph. Two popular deep learning models without random walk used in network embedding are AE and CNN.

A. AE-BASED NETWORK EMBEDDING METHODS

AE is an effective data dimension compression algorithm. AE based network embedding algorithms often change the settings of either input vectors [9]–[15] or loss function [9], [15]. Structural deep network embedding (SDNE) [9] constructed a semi-supervised deep model, which improved the input vectors by exploiting the first-order and second-order proximity jointly to preserve the network structure. Also, the model added first-order proximity to the loss function as supervisory information to capture the local network structure. Deep network representations with adversarially regularized autoencoders (NetRA) [15] improved the loss function by jointly minimizing AE loss and locality-preserving loss. The model learned smoothly regularized vertex representations that well capture the network structure through jointly considering both locality-preserving and global reconstruction constraints.

B. CNN-BASED NETWORK EMBEDDING METHODS

CNN and its variants have been widely adopted in network embedding. CNN based network embedding directly use the original CNN model designed for Euclidean domains [16]–[18] and non-Euclidean domains [19]–[21]. Xu *et al.* re-formatted the complex network topology adjacent matrix into an image and design a CNN model to extract relevant features and classify such features [16]. PATCHY-SAN [18] selected a fixed-length node sequence from a graph and then assembles, normalized and labelled nodes' neighborhood to learn a neighborhood representation with the CNN model. Hanocka *et al.* [19] utilized MeshCNN for a direct analysis of 3D shapes to leverage their intrinsic geodesic connections.

CNN-based graph embedding can also be divided into spectrum-based convolution [23] and space-based convolution [24], [25] according to the different operations of simulated convolution on the graph [22]. Defferrard *et al.* [23] designed fast k-localized (k means the shortest distance between kernel and neighbors) convolutional filters on graphs and presents a formulation of CNNs in the context of spectral graph theory. Dai *et al.* [24] proposed Stochastic Steady-state Embedding (SSE). The model took the sum of neighbor nodes as a potential expression of node degree. The model updated the node's potential random representation in the asynchronous network, and recursively estimates the node's potential representation and updates the parameters of the batch sampling data. Sperli [25] introduced a CNN bassed community detection method for sparse matrices. The convolutional layer and the largest pool layer processed the non-zero values in the adjacency matrix to reduce the dimension of the data. Inspired by AE-based graph embedding and CNN-based graph embedding mentioned above, a CNN+AE based deep community detection method, which reconstructs adjacency matrix with the spatial proximity, is put forward in this paper to extract higher spatial feature with lower dimensions.

III. DEEP COMMUNITY DETECTION METHOD

This section presents the concepts of the proposed matrix reconstruction method, spatial feature extraction method and community detection method.

A. MATRIX RECONSTRUCTION METHOD

Because the original order of nodes in an adjacency matrix is nonsense, a matrix reconstruction method is proposed to adjust the order of nodes in the adjacency matrix for arranging the neighbor nodes with high correlation. Therefore, some spatial features among nodes may be formatted after reconstructing the adjacency matrix for extracting spatial features and detecting social community. The proposed matrix reconstruction method consists of three sub-methods which include (1) opinion leader election method, (2) neighbor node selection method and (3) reconstruction method described in the following subsections.

1) OPINION LEADER ELECTION METHOD

The opinion leader who is an important member in a group (i.e. an important node in an adjacency matrix) may affect the attitudes of other members. In practical social networks, several members may follow and be a friend of the opinion leader in a community. Therefore, the proposed opinion leader election method will analyze the influence of each node to find the most important node (i.e. the most influential opinion leader) in the adjacency matrix as an initial node for matrix reconstruction.

In this study, an adjacency matrix $\mathbf{A}$ is constructed according to the edges between each two nodes. The parameter V (i.e. $\{v_1, v_2, \ldots, v_n\}$) denotes as a node set, and the parameter E denotes as an edge set. If the edge between node v_i and node v_j exists, the value of $a_{i,j}$ is 1; otherwise, the value of $a_{i,j}$ is 0 (shown in Equation (1)).

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix}, \tag{1}$$

where

$$a_{i,j} = \begin{cases} 1, & e_{i,j} \in E \\ 0, & e_{i,j} \notin E \end{cases} \tag{1}$$

A transition probability matrix $\mathbf{C}$ (shown in Equation (2)) can constructed according to the adjacency matrix $\mathbf{A}$. When a node connects to several nodes, the weight of each edge is lower; when a node connects to less nodes, the weight of each edge is higher. For instance, if node v_i only connects to node v_j, node v_j is an important node for node v_i, and the transition probability $c_{i,j}$ is 1.

$$\mathbf{C} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{bmatrix}, \text{ where } c_{i,j} = \frac{a_{i,j}}{\sum_{k=1}^{n} a_{i,k}}. \tag{2}$$

In initial stage, the weight of each node is 1, and a node weight matrix $\mathbf{S}$ is constructed in in Equation (3). The limiting weight matrix $\mathbf{S}^*$ is calculated based on the transition probability matrix $\mathbf{C}$ (shown in Equation (4)). Finally, the node *i_leader* with the highest weight (i.e. opinion leader) is found according to Equation (5).

$$\begin{aligned} \mathbf{S} &= \begin{bmatrix} s_1 & s_2 & \cdots & s_n \end{bmatrix}, \\ \text{where } s_i &= 1 \text{ in initial stage.} \end{aligned} \tag{3}$$

$$\begin{aligned} \mathbf{S}^* &= \lim_{m\to\infty} \mathbf{S} \times \mathbf{C}^m \\ &= \lim_{m\to\infty} \begin{bmatrix} s_1 & s_2 & \cdots & s_n \end{bmatrix} \\ &\quad \times \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{bmatrix}^m \\ &= \begin{bmatrix} s_1^* & s_2^* & \cdots & s_n^* \end{bmatrix} \end{aligned} \tag{4}$$

$$i\_leader = \arg\max_{1\le i\le n} s_i^*. \tag{5}$$

The pseudo codes of the proposed opinion leader election method are presented in **Algorithm 1**.

Algorithm 1 Opinion Leader Election Method

Inputs: an adjacency matrix $\mathbf{A}_{n\times n}$ with n nodes and the number of iterations $I_{\max}$
Outputs: the node ID of opinion leader

(1) Construct the adjacency matrix $\mathbf{A}_{n\times n}$.
(2) **for** $i = 1$ to n
(3) **for** $j = 1$ to n
(4) Determine the weight of edge from node v_i to node v_j based on the amount of connections of node v_i.
(5) **end for**
(6) **end for**
(7) **for** $i = 1$ to n
(8) Set the weight of node v_i in the initial stage as 1.
(9) **end for**
(10) **while (the number of iterations is less than $I_{\max}$)**
(11) **for** $i = 1$ to n
(12) Set the weight of node v_i in the next iteration as 0.
(13) **for** $j = 1$ to n
(14) Calculate the weight of node v_j multiplied by the weight edge from node v_j to node v_i as the value of temp.
(15) Set the weight of node v_i in the next iteration plus the value of temp.
(16) **end for**
(17) **end for**
(18) **end while**
(19) Return the ID of the node with a highest weight.

2) NEIGHBOR NODE SELECTION METHOD

After finding the opinion leader (i.e. node *i_leader*), a neighbor node selection method is proposed to find the neighbor node *j_neighbor* which is highest relevant to node *i_leader*.

The node *i_leader* is presented as node v_i, and an Euclidean distance $r(i, j)$ is applied to measure the distance between node v_i and node v_j based on Equation (6). The node *j_neighbor* with the shortest distance (i.e. the nearest neighbor) is found according to Equation (7).

$$\begin{aligned} r(i,j) &= \sqrt{\sum_{k=1}^{n} d(i,j,k)^2}, \\ \text{where } d(i,j,k) &= a_{i,k} - a_{j,k}. \end{aligned} \tag{6}$$

$$j\_neighbor = \arg\min_{1\le j\le n} r(i,j), \text{ where } j \ne i. \tag{7}$$

3) RECONSTRUCTION METHOD

The order of nodes in the adjacency matrix $\mathbf{A}$ can be determined based on the proposed opinion leader election method and neighbor node selection method. The opinion leader node can be elected as the first node, and the nearest neighbor of the opinion leader node can be elected as the second node; then the nearest neighbor of the second node can also be elected by the neighbor node selection method in accordance with Equations (6) and (7), and so on. The adjacency matrix $\mathbf{A}$ can be reconstructed as matrix $\mathbf{X}$ (shown in Equations (8)) by **Algorithm 2**.

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,n} \end{bmatrix},$$

$$\text{where} \quad x_{i,j} = \begin{cases} 1, & e_{i,j} \in E \\ 0, & e_{i,j} \notin E. \end{cases} \tag{8}$$

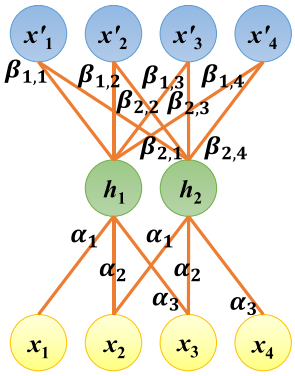


FIGURE 2. A simple case of auto-encoder based on convolutional neural network.

B. SPATIAL FEATURE EXTRACTION METHOD

Due to the convolutional neural network as an excellent deep learning model for spatial analyses, this study proposes an auto-encoder based on convolutional neural network to automatically extract the spatial features of the reconstructed adjacency matrix for social networks. For building the structure of the proposed auto-encoder based on convolutional neural network, the number of neurons in the input layer is same with the number of neurons in the output layer. The spatial features in social networks can be extracted by the convolutional layer, so this study adopts the values of neurons after convolutional computation as spatial feature vectors.

1) THE DESCRIPTION OF A SIMPLE CASE

In this case, both of input layer and output layer have four neurons (i.e. four nodes in a social network), respectively. The size of filter in the convolutional layer (i.e. the first hidden layer) is 1×3, so two neurons are trained in the hidden layer. The structure of auto-encoder based on convolutional neural network is showed in Figure 2.

Each parameter in the neural network is described as follows.

- The weight set of filter in the convolutional layer is $\{\alpha_1, \alpha_2, \alpha_3\}$.
- The adjustment variables of the two neurons in the convolutional layer are $\{b_{1,1}, b_{1,2}\}$.
- The weight set between the convolutional layer and the output layer is $\{\beta_{1,1}, \beta_{1,2}, \beta_{1,3}, \beta_{1,4}, \beta_{2,1}, \beta_{2,2}, \beta_{2,3}, \beta_{2,4}\}$.
- The adjustment variables of the four neurons in the output layer are $\{b_{2,1}, b_{2,2}, b_{2,3}, b_{2,4}\}$.
- The two neurons in the convolutional layer which are $\{h_1, h_2\}$ can be calculated by Equations (9) and (10).
- The four neurons in the output layer which are $\{x'_1, x'_2, x'_3, x'_4\}$ can be calculated by Equations (11), (12), (13) and (14).
- The loss function analyzes the mean squared errors by Equation (15) to optimize the neural network.

$$h_1 = \alpha_1 \times x_1 \alpha_2 \times x_2 + \alpha_3 \times x_3 + b_{1,1}. \quad (9)$$

Algorithm 2 Reconstruction Method

Inputs: an adjacency matrix $\mathbf{A}_{n\times n}$ with n nodes
Outputs: the reconstructed adjacency matrix $\mathbf{X}_{n\times n}$

(1) Construct the adjacency matrix $\mathbf{A}_{n\times n}$.
(2) Create the list $\mathbf{L}_n$.
(3) Create the list $\mathbf{O}_n$.
(4) **for** $i = 1$ to n
(5) Push node v_i into the list $\mathbf{L}_n$.
(6) **end for**
(7) Set the node *i_leader* from the opinion leader election method as node v_i.
(8) **while(the length of the list $\mathbf{L}_n$ is larger than 0)**
(9) Pull node v_i from the list $\mathbf{L}_n$.
(10) **for** $j = 1$ to n
(11) Calculate the Euclidean distance between node v_i and node v_j.
(12) **end for**
(13) Find and Set node *j_neighbor* from the neighbor node selection method as node v_i.
(14) Push node v_i into the list $\mathbf{O}_n$.
(15) **end while**
(16) Construct the adjacency matrix $\mathbf{X}_{n\times n}$ according to the list $\mathbf{O}_n$.
(17) Return the reconstructed adjacency matrix $\mathbf{X}_{n\times n}$.

$$h_2 = \alpha_1 \times x_2 + \alpha_2 \times x_3 + \alpha_3 \times x_4 + b_{1,2}. \quad (10)$$

$$x'_1 = \sum_{1=1}^{2} \beta_{i,1} \times h_i + b_{2,1}. \quad (11)$$

$$x'_2 = \sum_{1=1}^{2} \beta_{i,2} \times h_i + b_{2,2}. \quad (12)$$

$$x'_3 = \sum_{1=1}^{2} \beta_{i,3} \times h_i + b_{2,3}. \quad (13)$$

$$x'_4 = \sum_{1=1}^{2} \beta_{i,4} \times h_i + b_{2,4}. \quad (14)$$

$$E = \sum_{1=1}^{4} \frac{1}{2}(x'_i - x_i)^2 = \sum_{1=1}^{4} \frac{1}{2}\sigma_i^2. \quad (15)$$

The gradient descent method is adopted for the optimization of auto-encoder based on convolutional neural network. The weights in the neural network are updated by Equations (16), (17), (18), (19), (20) and (21). When the training process is complete, the reconstructed adjacency matrix will be inputted into the trained neural network to retrieve the values of neurons (i.e. $\{h_1, h_2\}$) in the convolutional layer for spatial feature extraction.

$$\begin{aligned} \frac{\partial E}{\partial \beta_{j,i}} &= \frac{\partial E}{\partial \sigma_i} \frac{\partial \sigma_i}{\partial x'_i} \frac{\partial x'_i}{\partial \beta_{j,i}} \\ &= \sigma_i \times 1 \times h_j \\ &= \sigma_i \times h_j \end{aligned} \quad (16)$$

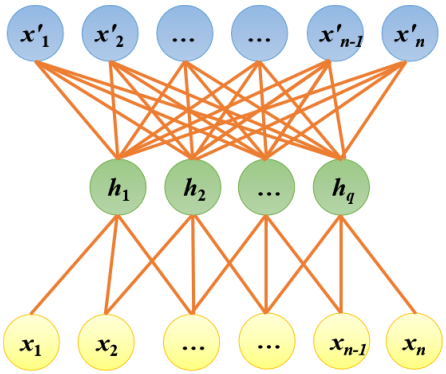


FIGURE 3. A general case of auto-encoder based on convolutional neural network.

$$\begin{aligned}\frac{\partial E}{\partial b_{2,i}} &= \frac{\partial E}{\partial \sigma_i}\frac{\partial \sigma_i}{\partial x_i'}\frac{\partial x_i'}{\partial b_{2,i}} \\ &= \sigma_i \times 1 \times 1 \\ &= \sigma_i \end{aligned} \tag{17}$$

$$\begin{aligned}\frac{\partial E}{\partial \alpha_1} &= \sum_{i=1}^{4}\frac{\partial E}{\partial \sigma_i}\frac{\partial \sigma_i}{\partial x_i'}\sum_{i=1}^{2}\frac{\partial x_i'}{\partial h_j}\frac{\partial h_j}{\partial \alpha_1} \\ &= \sum_{i=1}^{4}\sigma_i \times \sum_{i=1}^{2}\beta_{j,i} \times \frac{\partial h_j}{\partial \alpha_1} \\ &= \sigma_i \end{aligned} \tag{18}$$

$$\begin{aligned}\frac{\partial E}{\partial \alpha_2} &= \sum_{i=1}^{4}\frac{\partial E}{\partial \sigma_i}\frac{\partial \sigma_i}{\partial x_i'}\sum_{i=1}^{2}\frac{\partial x_i'}{\partial h_j}\frac{\partial h_j}{\partial \alpha_2} \\ &= \sum_{i=1}^{4}\sigma_i \times \sum_{i=1}^{2}\beta_{j,i} \times \frac{\partial h_j}{\partial \alpha_2} \\ &= \sum_{i=1}^{4}\sigma_i \times (\beta_{1,i} \times x_2 + \beta_{2,i} \times x_3) \end{aligned} \tag{19}$$

$$\begin{aligned}\frac{\partial E}{\partial \alpha_3} &= \sum_{i=1}^{4}\frac{\partial E}{\partial \sigma_i}\frac{\partial \sigma_i}{\partial x_i'}\sum_{i=1}^{2}\frac{\partial x_i'}{\partial h_j}\frac{\partial h_j}{\partial \alpha_3} \\ &= \sum_{i=1}^{4}\sigma_i \times \sum_{i=1}^{2}\beta_{j,i} \times \frac{\partial h_j}{\partial \alpha_3} \\ &= \sum_{i=1}^{4}\sigma_i \times (\beta_{1,i} \times x_3 + \beta_{2,i} \times x_4) \end{aligned} \tag{20}$$

$$\begin{aligned}\frac{\partial E}{\partial b_{1,j}} &= \sum_{i=1}^{4}\frac{\partial E}{\partial \sigma_i}\frac{\partial \sigma_i}{\partial x_i'}\frac{\partial x_i'}{\partial h_j}\frac{\partial h_j}{\partial b_{1,j}} \\ &= \sum_{i=1}^{4}\sigma_i \times 1 \times \beta_{j,i} \times 1 \\ &= \sum_{i=1}^{4}\sigma_i \times \beta_{j,i} \end{aligned} \tag{21}$$

2) THE DESCRIPTION OF A GENERAL CASE

In this case, the reconstructed adjacency matrix can be split n records, and the dimension of each record is expressed as $1 \times n$. Both of input layer and output layer have n neurons (i.e. n nodes in a social network), respectively. There are q neurons in the convolutional layer (i.e. the first hidden layer). The structure of auto-encoder based on convolutional neural network for the general case is showed in Figure 3. The mean squared errors are considered by loss function, and the gradient descent method is adopted to minimize these errors. In testing and performing stage, the spatial features can be extracted according to the values of neurons (i.e. **H** shown in Equation (22)) in the hidden layer.

$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & \cdots & h_q \end{bmatrix}. \tag{22}$$

C. COMMUNITY DETECTION METHOD

The n records from the reconstructed adjacency matrix can be clustered into k groups by the K-means algorithm for detecting communities in a social network. Each record is adopted into the trained neural network, and the spatial features of each record is extracted by the proposed spatial feature extraction method. Therefore, the dimension of each record is expressed as $1 \times q$ (e.g., the vectors of the j-th record shown in Equation (23)) after extracting spatial features and adopted into the K-means algorithm.

$$\mathbf{H}_j = \begin{bmatrix} h_{j,1} & h_{j,2} & \cdots & h_{j,q} \end{bmatrix}. \tag{23}$$

The proposed community detection method includes three steps as follows.

Step (1). The k records are randomly elected as k cluster centers from the n records. The vectors of the i-th cluster center are expressed as Equation (24).

$$\mathbf{Z}_i = \begin{bmatrix} z_{i,1} & z_{i,2} & \cdots & z_{i,q} \end{bmatrix}. \tag{24}$$

Step (2). The Euclidean distance $\phi\,(i, j)$ is applied to measure the distance between the j-th record and the i-th cluster center based on Equation (25). The n records are distributed into the k groups according to the distance between the record and the cluster center, and the center of each cluster is recalculated based on the records in the cluster.

$$\phi\,(i,j) = \sqrt{\sum_{k=1}^{q} d\,(i,j,k)^2}, \text{ where } d\,(i,j,k) = h_{j,k} - z_{i,k}. \tag{25}$$

Step (3). If each cluster center has no changes, the clustering process is finished. Otherwise, Steps (1) and (2) are performed *repeatedly*.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

In experiments, four open datasets of practical social networks were adopted to evaluate the performance of the proposed deep community detection method.

A. EXPERIMENTAL ENVIRONMENTS

This subsection presents datasets, an evaluation factor and experimental designs.

TABLE 1. Datasets of practical social networks.

Dataset name	The number of nodes	The number of edges	The value of k	Dataset description
network19	19	37	3	The dataset 'network19' presents a simple network; the network includes 19 nodes which can be clustered into 3 groups.
karate	34	78	3	The dataset 'karate' presents the friendships among 34 members of a karate club at an university.
dolphins	62	159	3	The dataset 'dolphins' presents the social network of 62 dolphins in Doubtful Sound in New Zealand.
football	115	613	12	The dataset 'football' presents the American football games of 115 football teams.

1) DATASETS

Four open datasets of practical social networks which include (1) network19, (2) karate, (3) dolphins, and (4) football were collected for the evaluation of the proposed method. Table 1 shows the description of datasets and the value of k for K-means algorithm.

2) EVALUATION FACTOR

The modularity (shown in Equation (26)) is a popular factor for evaluating the performance of community detection method.

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta(C_i, C_j). \tag{26}$$

Each parameter in Equation (26) is defined and described as follows.

- The parameter A denotes an adjacency matrix.
- The parameter m denotes the number of edges.
- The parameter k_i denotes the degrees of the i-th node.
- If the i-th node and the i-th node are clustered into the same group, the value of $\delta(C_i, C_j)$ is 1.
- The range of Q (i.e. modularity) is between 1 and -1. If the value of Q equals to 1, the performance of community detection method is higher.

The NMI (shown in Equation (27)) is another popular factor for evaluating the performance of community detection method.

$$NMI = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij}\log\left(\frac{N_{ij}\times N}{N_{i\cdot}\times N_{\cdot j}}\right)}{\sum_{i=1}^{C_A} N_{i\cdot}\log\left(\frac{N_{i\cdot}}{N}\right) + \sum_{j=1}^{C_B} N_{\cdot j}\log\left(\frac{N_{\cdot j}}{N}\right)}. \tag{27}$$

Each parameter in Equation (27) is defined and described as follows.

- The parameter N denotes community matrix. The rows of the matrix N correspond to the standard community results and the columns correspond to the community detection results obtained by the algorithm.
- The parameter $N_{i\cdot}$ denotes Sum of values in the i-th row of community's matrix.

TABLE 2. The structure of neural network.

Dataset name	The number of nodes in social network	The structure of neural network
network19	19	19-40-80-160-80-40-19
karate	34	34-40-80-160-80-40-34
dolphins	62	62-40-80-160-80-40-62
football	115	115-40-80-160-80-40-115

- The parameter $N_{\cdot j}$ denotes Sum of values in the i-th column of community's matrix.
- The parameter C_A denotes the result of the standard community results.
- The parameter C_B denotes the result of the community results obtained by the algorithm.
- When the algorithm obtains a community division result that is more consistent with the standard division result, the value of NMI is closer to 1, otherwise the value of NMI is closer to 0.

3) EXPERIMENTAL DESIGN

For the evaluation of the proposed method, four cases were designed to compare the modularity of different method combinations as follows.

Case (1): The auto-encoder method was adopted to extract the features of original adjacency matrix in a social network for community detection. The label of Case (1) was expressed as 'AE'.

Case (2): The auto-encoder method was adopted to extract the features of reconstructed adjacency matrix in a social network for community detection. The label of Case (2) was expressed as 'RM+AE'.

Case (3): The auto-encoder method based on convolutional neural network was adopted to extract the features of original adjacency matrix in a social network for community detection. The label of Case (3) was expressed as 'AE+CNN'.

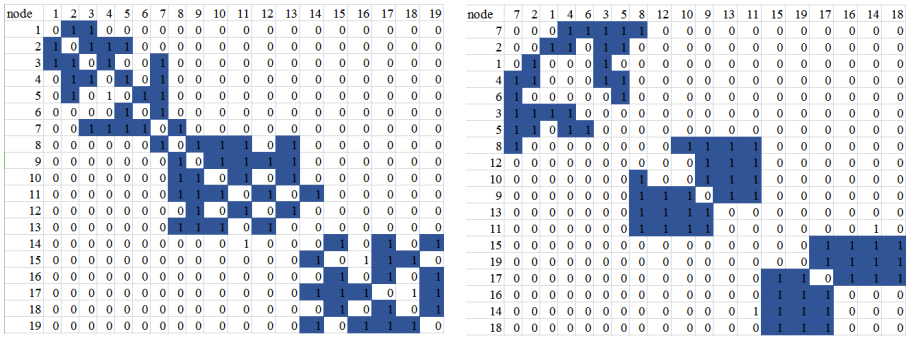
Case (4): The auto-encoder method based on convolutional neural network was adopted to extract the features of reconstructed adjacency matrix in a social network for community detection. The label of Case (4) was expressed as 'RM+AE+CNN'.

In experiments, four cases and four datasets were considered to evaluate the performance of the proposed deep community detection method. The structure of neural network for each dataset is showed in Table 2. Furthermore, for the evaluation of the proposed method, node2vec [3], SDNE [9] and NetRA [15] are taken as baselines for comparison.

B. EXPERIMENTAL RESULTS OF MATRIX RECONSTRUCTION METHOD

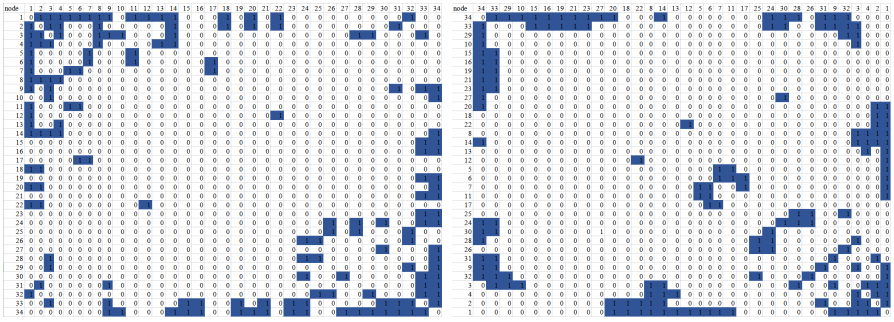
Three steps in the proposed matrix reconstruction method include (1) finding the opinion leader in a social network, (2) finding neighbor nodes, and (3) reconstructing the adjacency matrix of the social network.

The reconstructed adjacency matrices of practical social networks in open datasets are showed in Figures 4, 5, 6 and 7.



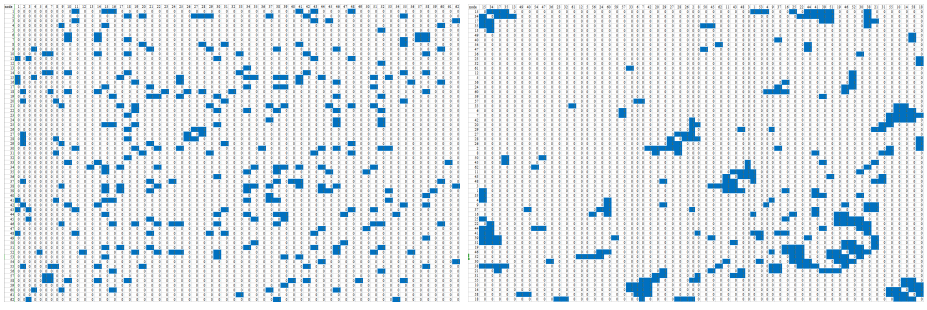
(a) Original adjacency matrix (b) Reconstructed adjacency matrix

FIGURE 4. The matrix reconstruction method for 'network19'.



(a) Original adjacency matrix (b) Reconstructed adjacency matrix

FIGURE 5. The matrix reconstruction method for 'karate'.



(a) Original adjacency matrix (b) Reconstructed adjacency matrix

FIGURE 6. The matrix reconstruction method for 'dolphins'.

For the analyses of the dataset 'network19', Figure 4(a) shows the original adjacency matrix, and Figure 4(b) shows the reconstructed adjacency matrix. The 7-th node in the original adjacency matrix is elected as the opinion leader. The boundaries of some communities in the original adjacency matrix are overlapped. After performing the proposed matrix reconstruction method, the nodes in the same community are closer, and the boundaries of communities are easier to be detected in the reconstructed adjacency matrix. Therefore, the reconstructed adjacency matrix is suitable for spatial feature extraction and community detection.

For the analyses of the dataset 'karate', Figure 5(a) shows the original adjacency matrix, and Figure 5(b) shows the reconstructed adjacency matrix. The 34-th node in the original adjacency matrix is elected as the opinion leader. The boundaries of communities in the original adjacency matrix are difficult to be detected.

Although the reconstructed adjacency matrix in this case may have a little of improvement for community detection, the number of isolated points in the reconstructed adjacency matrix is less.

For the analyses of the dataset 'dolphins', Figure 6(a) shows the original adjacency matrix, and Figure 6(b) shows the reconstructed adjacency matrix. The 15-th node in the original adjacency matrix is elected as the opinion leader. The edges in the original adjacency matrix are scattered, and it is difficult to detect social communities. After performing the proposed matrix reconstruction method, the boundaries of communities are clearer for detection, and the number of isolated points is less. The cyberspace

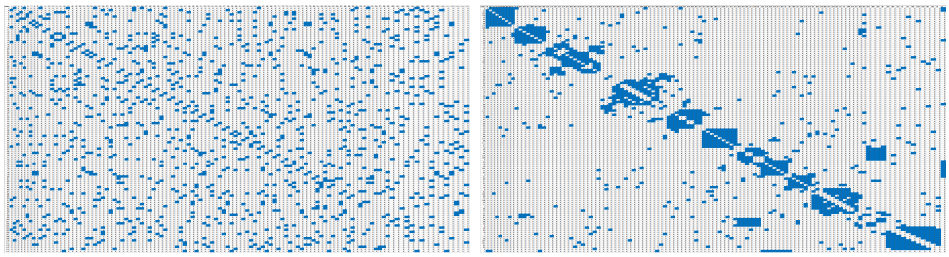


FIGURE 7. The matrix reconstruction method for 'football'.

structure of communities is clearer for spatial feature extraction.

For the analyses of the dataset 'football', Figure 7(a) shows the original adjacency matrix, and Figure 7(b) shows the reconstructed adjacency matrix. The 2-nd node in the original adjacency matrix is elected as the opinion leader. The cyberspace structure of communities in the original adjacency is difficult to be detected owing to the scattered distribution. After performing the proposed matrix reconstruction method, 12 communities in the reconstructed adjacency matrix are easier to be detected, so the proposed method can support to extract the spatial features of community in this case.

In accordance with the experimental results in Figures 4, 5, 6 and 7, the nodes in the same community are closer, and the boundaries of communities are clearer. Therefore, the reconstructed adjacency matrices are suitable for spatial feature extraction and community detection.

C. EXPERIMENTAL RESULTS OF SPATIAL FEATURE EXTRACTION METHOD AND COMMUNITY DETECTION METHOD

This subsection uses the modularity to evaluate the proposed spatial feature extraction and community detection methods, and experimental results of each case in Subsection IV.A.3 are illustrated in Table 3.

For the evaluation of matrix reconstruction method, the modularity in Case (2) (i.e. RM+AE) is higher than the modularity in Case (1) (i.e. AE) for each dataset, and the modularity in Case (4) (RM+AE+CNN) is higher than the modularity in Case (3) (AE+CNN) for each dataset. Therefore, the proposed matrix reconstruction method can support to reconstruct the cyberspace structure of communities for community detection.

For the evaluation of spatial feature extraction method, the modularity in Case (3) (i.e. AE+CNN) is higher than the modularity in Case (1) (i.e. AE) for each dataset, and the modularity in Case (4) (RM+AE+CNN) is higher than the modularity in Case (2) (RM+AE) for each dataset. Therefore, the proposed spatial feature extraction method based on convolutional neural network can extract the spatial features in social networks for the improvement of community detection.

TABLE 3. The modularity of each case.

Dataset name	The value of k	Case (1): AE	Case (2): RM+AE	Case (3): AE+CNN	Case (4): RM+AE+CNN
network19	3	**0.667**	**0.667**	**0.667**	**0.667**
karate	3	0.051	0.217	0.216	**0.370**
dophlins	3	0.258	0.307	0.362	**0.393**
football	12	0.540	0.540	0.586	**0.589**

TABLE 4. The modularity of comparison algorithms.

Dataset name	The real community	node2vec +Kmeans	NetRA +Kmeans	SDNE +Kmeans	Case (4): RM+AE+CNN
network19	0.61	0.61	0.61	0.61	**0.67**
karate	0.37	0.36	0.32	**0.37**	**0.37**
dophlins	0.37	0.38	0.38	0.38	**0.39**
football	0.55	0.59	0.54	**0.60**	0.59

TABLE 5. The NMI of comparison algorithms.

Dataset name	node2vec +Kmeans	NetRA +Kmeans	SDNE +Kmeans	Case (4): RM+AE+CNN
network19	**1.00**	**1.00**	**1.00**	**1.00**
karate	0.84	0.57	**1.00**	**1.00**
dophlins	**0.89**	0.70	0.81	0.81
football	0.91	0.86	0.91	**0.92**

D. COMPARISON RESULTS WITH BASELINES

The comparison results are showed in Tables 4 and 5. The modularity in Case (4) (RM+AE+CNN) is higher than the modularity in node2vec and NetRA for each dataset. In network19 and dophlins, the modularity in Case (4) (RM+AE+CNN) is higher than the modularity of SDNE+Kmeans. In karate, the modularity in Case (4) (RM+AE+CNN) and the modularity of SDNE+Kmeans are equal. In football the modularity in Case (4) (RM+AE+CNN) is slightly lower than the modularity of SDNE+Kmeans in Table 4. Furthermore, The NMI in Case (4) (RM+AE+CNN) is best for three datasets in Table 5.

V. CONCLUSIONS AND FUTURE WORK

An AE and CNN-based deep community detection method for social networks is proposed in this paper. A matrix reconstruction method based on a novel cyberspace structure reconstruction strategy is proposed to acquire spatial proximity matrices and generate reconstructed adjacency matrices. The matrix adds spatial proximity to traditional adjacency matrix, obtains clear subspace characteristics, makes convolutional operations easy and quick to extract graph spatial localization. Meanwhile, the AE+CNN-based model is constructed to learn spatial eigenvector that automatically extract the spatial features of graph. The spatial eigenvector provides the basis and support for the analysis and application on the network graph. Applying the combined model of AE and CNN into the community detection, and experimental results show that the proposed deep community detection method can effectively detect high quality communities for each practical dataset, especially for the graph with obscure subspace structure. The combined model of AE+CNN based community detection method also becomes the foundation of dynamic network community detection.

Although the combined model of AE and CNN is a meaningful exploration of graph embedding model, the number of neurons in input and output layers is changeless after the deep learning model is constructed. For dynamic community detection, the relationships among nodes may be dynamically changing, so the time-series models are required to extract the spatio-temporal features for arbitrary social networks.

REFERENCES

[1] D. Yang, X. Liao, H. Shen, X. Cheng, and G. Chen, "Modeling the reemergence of information diffusion in social network," *Phys. A, Stat. Mech. Appl.*, vol. 490, pp. 1493–1500, Jan. 2018.

[2] D. Yang, X. Liao, J. Wei, and X. Cheng, "Modeling information diffusion with the external environment in social networks," *J. Internet Technol.*, vol. 20, no. 2, pp. 369–377, 2019.

[3] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sep. 2018.

[4] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, 2014, pp. 701–710.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781

[6] Y. Li, C. Sha, X. Huang, and Y. Zhang, "Community detection in attributed graphs: An embedding approach," in *Proc. 22nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, Feb. 2018, pp. 338–345.

[7] Z. Tao, H. Liu, J. Li, Z. Wang, and Y. Fu, "Adversarial graph embedding for ensemble clustering," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3562–3568.

[8] C. Wang, C. Wang, Z. Wang, X. Ye, J. X. Yu, and B. Wang, "DeepDirect: Learning directions of social ties with edge-based network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2277–2291, Dec. 2019.

[9] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1225–1234.

[10] J. Cao, D. Jin, L. Yang, and J. Dang, "Incorporating network structure with node contents for community detection on large networks using deep learning," *Neurocomputing*, vol. 297, pp. 71–81, Jul. 2018.

[11] V. Bhatia and R. Rani, "A distributed overlapping community detection model for large graphs using autoencoder," *Future Gener. Comput. Syst.*, vol. 94, pp. 16–26, May 2019.

[12] L. Yang, X. Cao, and D. He, "Modularity based community detection with deep learning," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, New York, USA, Jul. 2016, pp. 2252–2258.

[13] J. Shang, C. Wang, X. Xin, and X. Ying, "Community detection algorithm based on deep sparse autoencoder," *J. Softw.*, vol. 28, no. 3, pp. 648–662, 2017.

[14] S. Cavallari, V. W. Zheng, H. Cai, K. C.-C. Chang, and E. Cambria, "Learning community embedding with community detection and node embedding on graphs," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 377–386.

[15] W. Yu, C. Zheng, W. Cheng, C. C. Aggarwal, D. Song, B. Zong, H. Chen, and W. Wang, "Learning deep network representations with adversarially regularized autoencoders," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2663–2671.

[16] Y. Xu, Y. Chi, and Y. Tian, "Deep convolutional neural networks for feature extraction of images generated from complex networks topologies," *Wireless Pers. Commun.*, vol. 103, no. 1, pp. 327–338, Nov. 2018.

[17] J. Xu, M. Li, J. Jiang, B. Ge, and M. Cai, "Early prediction of scientific impact based on multi-bibliographic features and convolutional neural network," *IEEE Access*, vol. 7, pp. 92248–92258, 2019.

[18] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proc. 33rd Int. Conf. Mach. Learn.*, New York, NY, USA, Jun. 2016, pp. 2014–2023.

[19] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, "MeshCNN: A network with an edge," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, Jul. 2019.

[20] W. Gao and P. Zhou, "Customized high performance and energy efficient communication networks for AI chips," *IEEE Access*, vol. 7, pp. 69434–69446, 2019.

[21] I. Sosnovik and I. Oseledets, "Neural networks for topology optimization," *Russian J. Numer. Anal. Math. Model.*, vol. 34, no. 4, pp. 215–223, Aug. 2019.

[22] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[23] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 3844–3852.

[24] H. Dai, Z. Kozareva, and B. Dai, "Learning steady-states of iterative algorithms over graphs," in *Proc. 35th Int.Conf. Mach. Learn.*, Stockholm, Sweden, Jul. 2018, pp. 1106–1114.

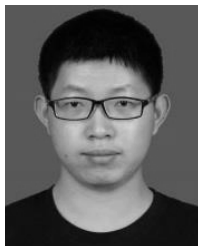
[25] G. Sperlí, "A deep learning based community detection approach," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 1107–1110.

LING WU received the Ph.D. degree from the School of Economics and Management, Fuzhou University. She is currently a Lecturer with the College of Mathematics and Computer Science, Fuzhou University. Her recent research interests include social networks, grey systems, and machine learning.

QISHAN ZHANG received the Ph.D. degree from the Huazhong University of Science and Technology. He is currently a Professor with the School of Economics and Management, Fuzhou University. His recent research interests include grey systems, logistics systems, and business intelligence.

CHI-HUA CHEN (Senior Member, IEEE) is currently a Distinguished Professor with Fuzhou University. He is also a Chair Professor with Dalian Maritime University, China. He has published over 270 academic articles and patents. His contributions have been published in the IEEE INTERNET OF THINGS JOURNAL, IEEE ACCESS, *IEICE Transactions*, and so on. His recent research interests include the Internet of Things, machine learning, deep learning, big data, cellular networks, and intelligent transportation systems. He is an Associate Editor of IEEE ACCESS.

KUN GUO is currently an Associate Professor with the College of Mathematics and Computer Science, Fuzhou University. His research interests include data mining, grey system theory, and distributed parallel computation. He is a member of the China Computer Federation (CCF) and the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing.



DEQIN WANG is currently pursuing the master's degree with the College of Mathematics and Computer Science, Fuzhou University. His research interests include social networks and machine learning.