

## Research Article

# A Dynamic Community Detection Method for Complex Networks Based on Deep Self-Coding Network

Yusha Zhang <sup>1</sup> and Xiongliang Xiao <sup>2</sup>

<sup>1</sup>*School of Computer Science and Engineering, Hunan University of Information Technology, Changsha 410151, Hunan, China*

<sup>2</sup>*School of Electronic Science and Engineering, Hunan University of Information Technology, Changsha 410151, Hunan, China*

Correspondence should be addressed to Xiongliang Xiao; [xxl1978@hnuit.edu.cn](mailto:xxl1978@hnuit.edu.cn)

Received 11 April 2022; Revised 27 May 2022; Accepted 4 July 2022; Published 31 July 2022

Academic Editor: Kapil Sharma

Copyright © 2022 Yusha Zhang and Xiongliang Xiao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the problem of community detection in complex dynamic networks, a dynamic community detection method based on graph convolution neural network is proposed. An encoding-decoding mechanism is designed to reconstruct the feature information of each node in the graph. A stack of multiple graph convolutional layers is considered as an encoder that encodes the node feature information into the potential vector space, while the decoder employs a simple two-layer perceptron to reconstruct the initial node features from the encoded vector information. The encoding-decoding mechanism achieves a re-evaluation of the initial node features. Subsequently, an additional local feature reconstruction loss is added after the decoder to aid the goal of graph classification. Further, stochastic gradient descent is applied to solve the problem in the loss function. Finally, the proposed model is experimentally validated based on the Karate Club and Football datasets. The experimental results show that the proposed model improves the NMI metric by an average of 7.65% and effectively mitigates the node oversmoothing problem. The proposed model is proved to have good detection accuracy.

## 1. Introduction

With the development of information technology and the increasing popularity of network media, the network community carries the self-information of massive data objects and the relationship between them [1]. How to divide the subnetworks of complex networks in biology, communication, finance, and other fields according to the network structure and node characteristics, to realize accurate community detection and further complete the effective mathematical modeling, evaluation, and analysis of dynamic communities, has important academic research significance and socio-economic value.

Community detection is to identify the existing community structure from the complex network. As a collection of specific objects, the community has a significant node aggregation structure and a node distribution form with statistical characteristics [2], which provides the possibility for accurate dynamic community detection. At present,

community detection algorithms can be roughly divided into graph segmentation-based method [3], random walk-based method, modularity-based method [4, 5], label propagation-based method [6], node centrality-based method [7], and neural network-based method. The existing researches on community detection are often based on the impact of static network structure on community division. For example, Bouguessa et al. used the covariance of links between nodes and interclass inertia to perform the initial division of the network and determined the maximum modular community structure by merging the initial groups in the iterative process, but this method was limited by the impact of different resolution parameters on the correctness of community division [8]. Cai et al. carried out multi-objective optimization of network model based on the evolutionary algorithm, but multiple iterations occupied a lot of computing resources [9]. Ni et al. proposed a detection structure for local overlapping communities, which improved the efficiency of local community detection [10].

However, there was a problem of large deviation of representative nodes, which often led to only the local optimal solution. The above community detection methods based on static network structure have a certain dependence on priori data, and due to the lack of real-time computing ability, it is limited by the influence of parameters on the detection results.

With the continuous expansion of the dataset scale, how to continuously reconstruct the node features according to the increase of data dimension and realize the dynamic detection of complex network communities is still a problem to be solved. For example, Moscato et al. proposed an online social network community detection algorithm based on game theory, but the mathematical statistical model of dynamic game based on individual behavior put forward high requirements for memory space, and there was the possibility of dimension disaster when solving [11]. In addition, with the development of graph neural network in recent years, it has more advantages in extracting node features and preserving graph structure, which has attracted some researchers to use the graph neural network to solve the problem of community detection. For example, Chen et al. regarded community detection as an extended node classification task, proposed a line graph neural network (LGNN) model, and designed the adjacency matrix based on polynomial function and nonbacktracking operators for community detection [12]. However, the graph storage method using an adjacency matrix needs to be traversed when counting elements, which has high time complexity. Luo et al. constructed a context path-based graph neural network (CP-GNN) by using the high-order relationship between nodes, which recursively embedded the high-order relationship between nodes into the node with an attention mechanism to distinguish the importance of different relationships [13]. Based on graph convolutional neural network (GCN) and label propagation technology, Wang et al. constructed a balanced label set to reveal the underlying community structure with topology and attribute information [14]. However, as weak supervised learning algorithms, the above two methods are relatively fragile to abnormal data and are difficult to define relevant parameters. The existing dynamic community detection methods based on the graph neural network overemphasize the ability to distinguish different graph structures and ignore the local expression ability of nodes, resulting in the oversmoothing problem [15, 16]. That is, the node feature representation in the same connected component tends to converge to the same value after multiple convolutions, making it difficult to distinguish nodes.

With the growing importance of network dynamic data detection under the development of information technology, traditional detection methods for dynamic communities in complex networks have limited the performance of dynamic community detection due to the emergence of node oversmoothing problems and the influence of a priori model parameters. To address these problems, a method for detecting dynamic communities in complex networks based on deep self-coding networks is proposed. Compared with the existing community detection methods based on the

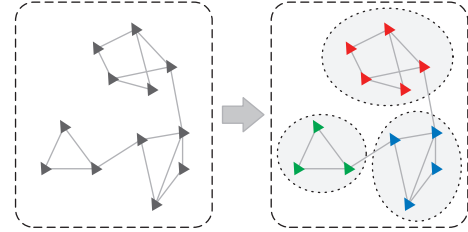


FIGURE 1: Community structures in networks.

graph, the innovations of the proposed method are as follows:

- (1) An algorithm framework of encoding-decoding mechanism based on deep self-coding network is proposed, which realizes the re-evaluation of the features of the initial nodes, effectively solves the problem of node oversmoothing, and improves the accuracy of graph classification.
- (2) From the perspective of community detection task, the loss function related to community detection is defined. It ensures the update of parameters in the network model and improves the efficiency of community detection algorithm.

The following sections are arranged as follows: the second section defines the problems studied; the third section introduces the proposed dynamic community detection method based on deep self-coding network; in Section 4, experiments are designed to verify the performance of the proposed community detection method; the fifth section is the conclusion.

## 2. Problem Definition

Community detection refers to the discovery of densely connected subnetworks in the graph structure. Liu et al. defined the community as follows: (1) the nodes in the community are densely connected; (2) the nodes in different communities are sparsely connected [17]. The community division under a basic network is shown in Figure 1.

In a dynamic network, each timestamp  $i$  corresponds to the corresponding network structure  $G_i = (V_i, E_i)$ , where  $V_i$  and  $E_i$  represent the node set and edge set corresponding to timestamp  $i$ , respectively. It is necessary to obtain the basic community structure under each timestamp and sort it according to the time series  $1, \dots, t$ . In order to extract and learn the network representation information comprehensively, the snapshot iteration method will be used to reflect the information of the dynamic network. Due to the large amount of data of the snapshot information, the time consumption of predicting the community based on the global snapshot iteration method is too high. Although it is reasonable and comprehensive, it is not efficient, so this method is abandoned; in addition, community prediction based on independent snapshot iteration has the same effect as dynamic network community detection. Although it is efficient, the dynamic time correlation is not strong, so it is impossible to predict the community. Therefore, the second-

order snapshot iteration can ensure the rationality of dynamic information and reduce the time overhead to a certain extent. The community structure contains many key points with potential characteristics. Different matrices are used to define different parts of the data, such as links and contents. This method will re-embed the shared matrix composed of multiple information in the network at different times into the shared feature matrix, so as to improve the effect of community detection.

*Definition 1* (dynamic community prediction). Given a series of snapshot sequences of graphs  $G_1, \dots, G_i, \dots, G_t$  with  $n \times n$  adjacency matrix  $P_1, \dots, P_i, \dots, P_t$  and  $n \times d$  content matrix  $Q_1, \dots, Q_i, \dots, Q_t$ , a shared matrix  $W_1, \dots, W_i, \dots, W_t$  is formed to predict the network structure under future time  $t + \Delta t$ .

For the community prediction of dynamic network, the network snapshot information needs to be iterated. When predicting the community structure under the time  $t + \Delta t$ , the sharing matrices  $W_1, \dots, W_i, \dots, W_{t+1}, W_t$  of previous snapshots need to be used to enhance the accuracy of prediction results.

### 3. Dynamic Community Detection Based on Deep Self-Coding Network

*3.1. Node Feature Reconstruction Based on Encoding-Decoding Mechanism.* In the graph convolution neural network, the node representation that captures the local structure and feature information of the graph will be fed to the global readout module to refine for graph-level representation learning. However, the graph-level representation ability and discrimination ability of traditional GCN are limited by over-refining and globalization, ignoring the preservation of local features, which will lead to the problem of oversmoothing. In order to alleviate these problems, an encoding-decoding mechanism is designed in the model to reconstruct the feature information of each node in the graph. The stack of multiple graph convolution layers is regarded as an encoder that encodes the node feature information into the potential vector space, and the decoder uses a simple two-layer perceptron to reconstruct the initial node feature from the encoded vector information. The encoding-decoding mechanism realizes the re-evaluation of the features of the initial node. Then, an additional local feature reconstruction loss is added after the decoder to assist the goal of graph classification and improve the accuracy of graph classification.

According to the model diagram shown in Figure 2, the encoder composed of  $K$  graph convolution layers aggregates local adjacent information at different structural levels. After the  $K$ -iteration of neighborhood aggregation, the output of the encoder extracts the local structure information and node feature information in the  $K$ -hop neighborhood. The encoder encodes the node feature of each node into an intermediate representation, and the decoder aims to map the encoded node representation to the reconstructed feature vector  $\hat{Z} \in R$ . From Figure 2, given the graph-level representation  $k_{(K)G} \in R^{d_0}$  of the readout module of the

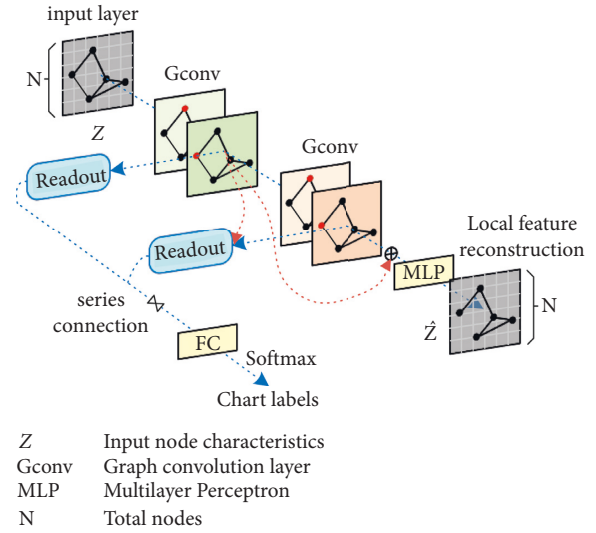


FIGURE 2: A three-layer model for graph classification.

upper layer, the reconstructed features  $\hat{Z}$  can be calculated according to the following formula:

$$\hat{Z} = \text{Dec} \left( \sum_{i=1}^{t-1} k_v^{(i)} \right). \quad (1)$$

A two-layer perceptron is used as the decoder. It is worth mentioning that the encoding-decoding mechanism is different from the graph self-encoder model [18–20]. The decoder is to reconstruct the initial node features, rather than a task-specific classifier to recover the adjacency information of the graph.

*3.2. Community Detection Loss Function.* Referring to the LGNN model [21], this study defines the loss function for the community detection task. In the community detection task, the node label should follow the equivalent replacement invariant property of the community [22]; that is, the divided community is not affected by the specific meaning of a specific label. From the output of Section 3.1, the predicted probability  $\hat{a}_i$  of the community to which each node belongs is obtained through Softmax. Let  $A$  represent the set of labels for all communities, and  $A_i$  represent the real probability of the community to which each node belongs. Therefore, the loss function is defined as follows:

$$S = \text{INF}_{\pi \in S_A} - \sum_{i \in R} \pi(a_i) \log(\hat{a}_i), \quad (2)$$

where  $\text{INF}$  represents the infimum of the function,  $S_A$  is the permutation and combination set of all communities  $A$ , and  $\pi$  is a permutation in  $S_A$ . Assuming that  $F_1: E \rightarrow A$  is the function mapping from the initial feature matrix  $E$  of the node to the real community  $A$ ,  $F_2: E \rightarrow \hat{A}$  is the function mapping from the initial feature matrix  $E$  of the node to the model predicted community  $\hat{A}$ , and  $t_i$  represents the feature vector of each node in the matrix, then the final loss function formula can be obtained from the following formula:

$$S = \text{INF}_{\pi \in S_A} - \sum_{i \in R} F_1(\pi(x_i)) \log(F_2(\pi(x_i))). \quad (3)$$

The ultimate goal of the loss function is to take the minimum cross-entropy loss on all possible permutations, so as to update the parameters in the network model.

**3.3. Self-Coding Learning Process.** Stochastic gradient descent is used to optimize the loss function. Specifically, the parameters  $\gamma = \{M, M^*, u, u^*\}$  of the algorithm framework are updated in each iteration according to the following contents:

$$\begin{cases} M_{ij} = M_{ij} - \beta \frac{\partial}{\partial M_{ij}} S_\gamma(\hat{T}, \hat{T}^*), \\ M_{ij}^* = M_{ij}^* - \beta \frac{\partial}{\partial M_{ij}^*} S_\gamma(\hat{T}, \hat{T}^*), \\ u_{ij} = u_{ij} - \beta \frac{\partial}{\partial u_{ij}} S_\gamma(\hat{T}, \hat{T}^*), \\ u_{ij}^* = u_{ij}^* - \beta \frac{\partial}{\partial u_{ij}^*} S_\gamma(\hat{T}, \hat{T}^*), \end{cases} \quad (4)$$

where  $\beta$  is the learning rate and  $(\cdot)^*$  indicates reconstruction. Due to the similarity of the inference processing between  $M, u$  and  $M^*, u^*$ , only the inference of the update rule of parameter  $M, u$  is shown as follows:

$$\begin{aligned} \frac{\partial}{\partial M_{ij}} S_\gamma(\hat{T}, \hat{T}^*) &= \sum_{i=1}^N \frac{\partial}{\partial M_{ij}} S_\gamma(t_i, g(\phi(t_i))) \\ &= \sum_{i=1}^N \frac{\partial}{\partial k_i^n} S_\gamma(t_i, g(\phi(t_i))) \frac{\partial}{\partial M_{ij}} k_i^n \\ &= \sum_{i=1}^N \alpha_i T_i^\tau, \end{aligned} \quad (5)$$

where  $k_i = M \cdot z_i + u$ , and  $\alpha_i = (\partial/\partial k_i^n) S_\gamma(t_i, g(\phi(t_i)))$  represents the characterization term of each node for the overall error. In order to measure the difference between the input data  $\hat{T}$  and  $\hat{T}^*$ , the  $\alpha_i$  in the encoder is written as follows:

$$\begin{aligned} \alpha_i &= \frac{\partial}{\partial k_i^n} S_\gamma(t_i, g(\phi(t_i))) \\ &= - \sum_{i=1}^N \frac{\partial}{\partial k_i^n} S_\gamma(\hat{T}_{ij}, \hat{T}_{ij}^*) \cdot \text{Relu}'(k_i^n), \end{aligned} \quad (6)$$

where  $\text{Relu}(\cdot)$  is the linear rectification function and  $\text{Relu}'(\cdot)$  is the derivative of  $\text{Relu}(\cdot)$ . In the decoder,  $v_i = M^* \cdot k + u$ , and the characterization item  $\alpha_i^*$  is as follows:

$$\alpha_i^* = \left( \sum_{i=1}^r M_{ij}^* \cdot \alpha_i \right) \cdot \text{Tanh}'(k_i^n), \quad (7)$$

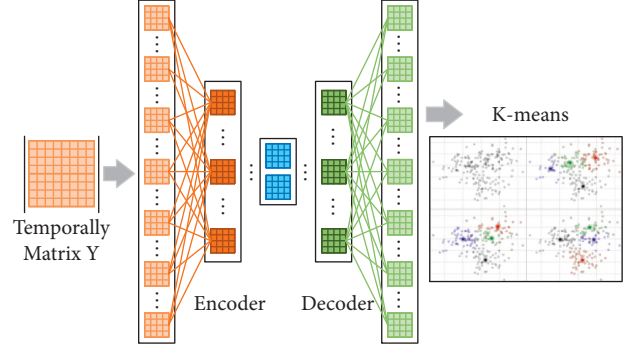


FIGURE 3: Algorithm framework.

where  $\text{Tanh}(\cdot)$  is activation function and  $\text{Tanh}'(\cdot)$  is the derivative of  $\text{Tanh}(\cdot)$ .

**3.4. Algorithm Framework Description.** Deep learning has excellent data learning ability and representation ability. Through training multilayer neural network, the representation information learned in each layer is used as input data and input into the next neural layer. Therefore, a deep automatic encoder with a multilayer automatic encoder is constructed and trained in the first layer of automatic encoder by reconstructing the input matrix  $\hat{Y}$ . Then, the best potential representation  $K^1 \in R^{r \times N}$  is obtained. Finally, the next automatic encoding layer is trained by reconstructing  $K^1$  from the above automatic encoder, and a new potential representation  $K^2 \in R^{r \times N}$  is obtained. Then, the potential feature matrix output by the hidden layer is reinput to the decoding layer with the same layer number as the encoding layer. The encoding and decoding layers have a symmetrical structure.

By continuously optimizing the learning parameters, remapping the compressed matrix to the size of the original matrix, learning the community structure information under multiple timestamps, and analyzing it, the evolution trend of the future community can be obtained, so as to predict the community network structure under timestamps  $t + \Delta t$ . The workflow of this algorithm framework is shown in Figure 3. By fusing the node topology and text information under multiple timestamps, combined with time analysis, the feature information containing community evolution trend is formed, and the potential information is extracted through representation learning.

Because the extracted potential representation contains both the structure information of the current community and the evolution information of the community, it can be used to predict the evolution trend of the future community. The algorithm framework is shown in Algorithm 1. Steps 1 to 4 are to fuse the adjacency matrix  $P$  and content matrix  $Q$  under multiple time snapshots, initialize the network, and set the number of iterations  $\zeta$  and the depth  $L$  of the self-coding network. Steps 5 to 12 are to extract the dynamic network information by using the deep self-coding network. After the two parts of encoding and decoding in the self-coding network, steps 13 and 14 are to reconstruct the network information through the learned potential

**INPUT:** Graph  $G_1, \dots, G_i, \dots, G_t$  corresponding to continuous timestamp  $t_1, \dots, t_i, \dots, t_t$ ;

**OUTPUT:** Community structure  $A_{t+\Delta t} = \{A_1, \dots, A_i, \dots, A_b\}$  under timestamp  $t + \Delta t$ ;

- (1) By extracting a variety of information from the dynamic graph  $G_1, \dots, G_i, \dots, G_t$ , the information is fused into a time matrix  $Y_1, \dots, Y_i, \dots, Y_t$ ;
  - (2)  $\tilde{Y} \leftarrow Y_1, \dots, Y_i, \dots, Y_t$ ;
  - (3) Set time matrix  $\tilde{Y} = [P, Q]$ ,  $L$  is the number of neural network layers and  $\zeta$  is the number of iterations;
  - (4) set  $O^{(0)} = \tilde{Y}$ ;
  - (5) for  $i = 1$  do  $\zeta$ ;
  - (6) for  $l = 1$  do  $L$ ;
  - (7) The stack self-coding network is constructed, and the encoder in the stack self-coding network is used to extract the feature of the graph information;
  - (8) Obtain potential characterization information  $K^{(l)}$ ;
  - (9) The parameters  $\gamma = \{M, M^*, u, u^*\}$ ,  $\hat{\gamma} \leftarrow \arg \min_{\gamma} \sum_{i=1}^N \|t_i - t_i^*\|^2$  are optimized according to iterative update formula (4);
  - (10) set  $O^{(l)} = K^{(l)}$ ;
  - (11) End for
  - (12) End for
  - (13) Decode the extracted potential representation and output the predicted community structure  $K^{(L)}$ ;
  - (14) Cluster  $K^{(L)}$  and output the community detection results  $A_1, \dots, A_i, \dots, A_b$  under the current network structure;
- End**

ALGORITHM 1: Community detection algorithm.

TABLE 1: Dynamic community detection algorithms.

Algorithm	Description
Line graph neural network (LGNN) [12]	The high-level graph algorithm is realized by sparse matrix multiplication and message passing
Context path-based graph neural network (CP-GNN) [13]	The nodes of an input graph are classified by weak supervision
Graph convolution network (GCN) algorithm [14]	Local parameters are shared and the receptive field is proportional to the number of layers
The proposed model	The temporal information is effectively combined to transition oversmooth process

representation. These steps can predict the future community structure and carry out clustering operation, so as to obtain the community structure of dynamic network.

## 4. Experiment and Analysis

The hardware configuration of the experiment is as follows: 8 GB memory, i5-4200H dual-core processor, and Windows 10 operating system. The compiler used for algorithm development is VisualStudio Code, the code is written in Python language, its version is 3.6.2, and the Pytorch framework is used. In order to measure the convergence and detection accuracy of the proposed algorithm, the line graph neural network (LGNN) algorithm proposed in reference [12], the context path-based graph neural network (CP-GNN) algorithm proposed in reference [13], and the algorithm based on graph convolution network (GCN) and label propagation technology proposed in reference [14] are selected as a comparison to verify the performance of the proposed model. The description of each algorithm is listed in Table 1.

*4.1. Experimental Dataset.* Two classical complex networks such as Karate Club network (Karate) [23] and American Football League network (Football) [24] are used for

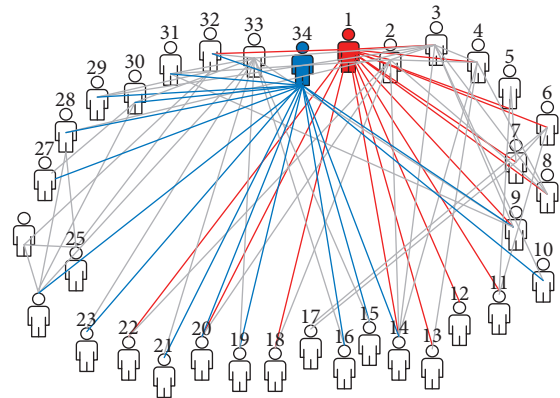


FIGURE 4: Karate Club network.

experiments to verify and analyze the community division performance of different algorithms. The following is a brief description of each dataset:

- (1) *Karate Club.* This network depicts the interpersonal relationship between members of a karate club in an American University. The members of the club are divided into two factions due to differences on a certain issue, as shown in Figure 4.

- (2) *Football*. This dataset is the game relationship between football teams of different universities in the United States in the 2000 season. The node represents the participating football teams, while the side indicates that the two teams have played this season.

The statistics of each dataset are listed in Table 2.

#### 4.2. Experimental Evaluation Index and Convergence Analysis

4.2.1. *Experimental Evaluation Index*. The evaluation indexes used in this experiment are accuracy, F1-score, NMI, and modularity.

- (1) *Accuracy*. It is a common evaluation index for node classification. It evaluates the correctly predicted sample ratio.
- (2) *F1-score*. It is applicable to the evaluation index of classification task. It takes into account the proportion of samples predicted to be true that are actually true, and the proportion of samples actually true that are predicted true. The macro-F1 index is used in the experiment. First, the F1-scores of precision and recall of each data sample are calculated respectively, and then all results are averaged to obtain the final result.
- (3) *NMI*. It refers to normalized mutual information. It measures the similarity between two clusters, and it is an important evaluation index for community detection. The higher the NMI score, the better the detection effect. Supposing that  $Z_{\text{true}}$  is the real cluster and  $Z_{\text{pred}}$  is the cluster prediction result, then the formula is defined as follows:

$$\left\{ \begin{array}{l} \text{NMI}(Z_{\text{pred}}; Z_{\text{true}}) = 2 \frac{I(Z_{\text{pred}}; Z_{\text{true}})}{H(Z_{\text{pred}}) + H(Z_{\text{true}})}, \\ H(Z_{\text{pred}}) = - \sum_{\tilde{z}} p(\tilde{z}) \log(p(\tilde{z})), \\ I(Z_{\text{pred}}; Z_{\text{true}}) = \sum_{\tilde{z}} \sum_z p(\tilde{z}, z) \log\left(\frac{p(\tilde{z}, z)}{p(\tilde{z})p(z)}\right). \end{array} \right. \quad (8)$$

- (4) *Modularity*. It is used to evaluate the quality of the detected community. It is defined as follows:

$$M_{\text{odu}} = \frac{1}{2} \sum_{i \neq j} \left( P_{ij} - \frac{p_i p_j}{2y} I(b_i, b_j) \right), \quad (9)$$

where  $P_{ij}$  is the element in the graph adjacency matrix,  $y$  is the number of edges,  $p_i = \sum_j P_{ij}$  is the sum of degrees of the  $i$ -th node, and  $b_i$  is the community of the  $i$ th node. If  $b_i = b_j$ , then  $I(b_i, b_j) = 1$ ; otherwise,  $I(b_i, b_j) = 0$ . The higher the modularity result, the more edges in the community and the fewer edges between communities.

4.2.2. *Convergence Analysis*. The convergence of the proposed model will be analyzed on the Karate Club and Football

TABLE 2: Description of dataset.

Network	Node	Edge	Community
Karate Club	34	78	2
Football	115	613	12

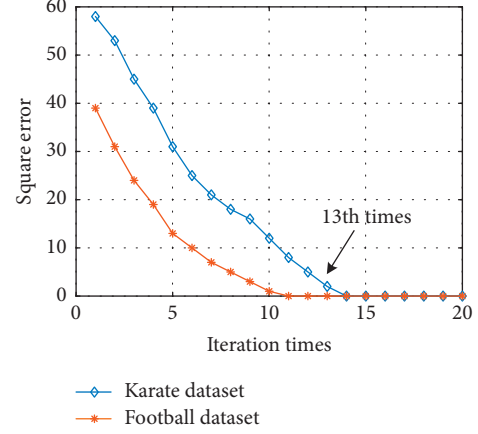


FIGURE 5: Convergence analysis.

datasets. Since each iteration of the algorithm mainly updates the relationship tightness between nodes, the convergence of the algorithm is analyzed by comparing the difference of the relationship matrix before and after updating with the number of iterations. Based on the trusted nearest neighbor graph, the indirect tightness of non-nearest neighbor nodes is obtained through similarity propagation. In addition, considering the inheritance and update of information in the iterative process of the algorithm, the similarity update rule between two sections  $i$  and  $j$  is defined as follows:

$$X_{s-i,j}^t = \lambda \tilde{X}_{s-i} (\tilde{X}_{s-j})^T + (1 - \lambda) X_{s-i,j}^{t-1}. \quad (10)$$

The first item is the nonzero element in the indirect tightness  $\tilde{X}_{s-i}$  of the two nodes, that is, the highly trusted nearest neighbor node of node  $i$ . In the above iterative process, not only the connection information before the update of the two nodes is retained, but also the indirect tightness of the node pair is introduced by its common nearest neighbor, and the parameter  $\lambda$  controls the proportion of the two in the similarity update process. After several iterations, the tightness between nodes tends to be stable, and then, the square error  $\|X_s^{t+1} - X_s^t\|_F$  of the similarity matrix before and after the iteration can be recorded based on the updated node similarity matrix. Figure 5 shows the change curves of the square error with the number of iterations in the two datasets.

It can be seen from the figure that after about 13 iterative updates, the tightness between nodes tends to be stable, which verifies the convergence of the algorithm.

#### 4.3. Experimental Results

4.3.1. *Comparison and Analysis of Test Results*. The community detection results are listed in Tables 3 and 4. On the

TABLE 3: Community detection results for different algorithms on the Karate Club dataset (%).

Algorithms	Accuracy	F1-score	NMI	Modularity
LGNN [12]	89.3	92.3	85.2	63.7
CP-GNN [13]	94.1	92.1	87.9	62.3
GCN [14]	95.4	94.8	88.1	64.9
The proposed method	96.6	95.2	93.7	65.2

TABLE 4: Community detection results for different algorithms on the Football dataset (%).

Algorithms	Accuracy	F1-score	NMI	Modularity
LGNN [12]	60.1	60.9	49.7	31.8
CP-GNN [13]	58.9	61.2	43.6	33.7
GCN [14]	61.4	64.8	44.7	34.9
The proposed method	64.2	67.0	50.3	40.2

experimental results of Karate Club dataset (Table 3), the performance of the proposed model is significantly better than all other methods in all evaluation indexes.

Because the LGNN algorithm [12] adopts the graph storage method of adjacency matrix, it needs to traverse when counting elements, which has high time complexity and further affects the detection accuracy and performance. The CP-GNN algorithm [13] and GCN algorithm [14] due to the relative vulnerability of weak supervised learning algorithm to abnormal data and the oversmoothing problem of the algorithm itself, the accuracy of clustering detection results has been affected to a certain extent. It can be seen from Table 3 that the proposed model is relatively optimal in the experimental results of Karate Club dataset, especially in the NMI index. Compared with the LGNN algorithm [12] model, the NMI is increased by 9.98%. Compared with the CP-GNN algorithm [13] model, the NMI is improved by 6.60%. Compared with the GCN algorithm [14] model, the NMI is improved by 6.36%. In addition, the accuracy and F1-score of the proposed model show that the proposed method can detect the community more accurately. Compared with the methods with oversmoothing problem (such as the GCN algorithm [14]), the proposed model uses an encoding-decoding mechanism to reconstruct the feature information of each node in the graph, which can explain its superior performance. It can be seen from the results of Karate Club dataset that it is very important to consider the reconstruction of node features in the process of algorithm execution.

The superiority of the proposed model can also be observed in the experimental results of Football dataset (Table 4). However, the NMI of LGNN algorithm [12] is similar to that of the proposed model. One possible reason is that the LGNN algorithm [12] uses a so-called “nonbacktracking” operator, which can well preserve the spatial node information across multiple steps. In contrast, the proposed model may have slight data loss in the reconstruction of data nodes by using the encoding-decoding mechanism, so it has no significant advantage in the NMI index. Although the other two algorithms (CP-GNN algorithm [13] and GCN algorithm [14]) can better retain the long-term time features,

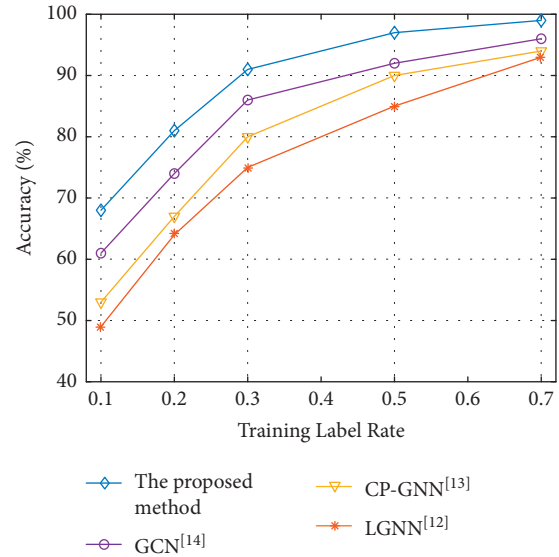


FIGURE 6: Experimental results of accuracy index with different training label rates.

if more time map data are convoluted, then the embedded features may become too smooth, so that the distance between all nodes becomes close. Therefore, it cannot accurately detect the community, resulting in a slight weakness in the evaluation index. In the modularity index, the proposed model performs best. Compared with the LGNN algorithm [12] model, the modularity is improved by 26.42%. Compared with the CP-GNN algorithm [13] model, the modularity is improved by 19.29%. Compared with the GCN algorithm [14] model, the modularity is improved by 15.19%.

**4.3.2. Results and Analysis of Label Rate Impact.** In order to observe the influence of the training label rate on the experimental results, the experiment selects accuracy and NMI indexes to evaluate all methods on the Football dataset. In this experiment, the training label rates are 10%, 20%, 30%, 50%, and 70%, respectively. The experimental results are shown in Figures 6 and 7.

As shown in Figure 6, in terms of accuracy index, with the increase of training label rate, the model performance of all methods is gradually improving. However, the accuracy index of LGNN algorithm [12] is not ideal. The possible reason is that the LGNN model affects the detection accuracy of the model in the process of traversing elements through the graph storage method of adjacency matrix. When the training label rate is 70%, the accuracy index of the proposed model is improved by 6.45% compared with the LGNN algorithm [12]. Compared with the CP-GNN algorithm [13], accuracy is improved by 5.32%. Compared with the GCN algorithm [14], accuracy is improved by 3.13%. Experimental results show that the proposed algorithm has higher execution accuracy. Based on the graph convolution neural network algorithm, the inherent oversmoothing problem of the algorithm is overcome through the encoding-decoding mechanism, so as to improve the detection efficiency and accuracy.

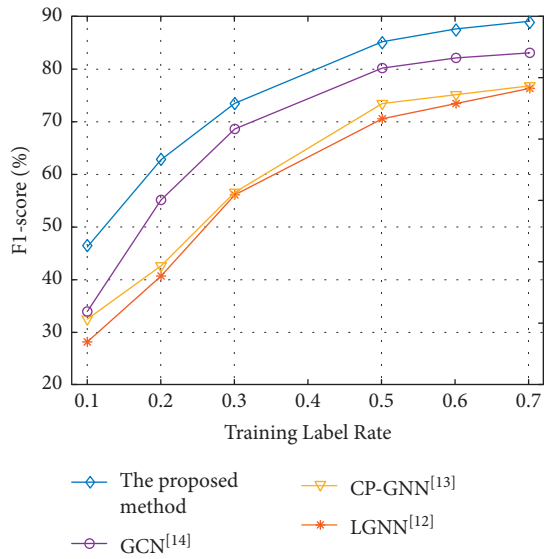


FIGURE 7: Experimental results of F1-score with different training label rates.

Figure 7 shows that when the training label rate increases from 10% to 70%, the F1-score index of all algorithms gradually increases. Because the typical graph convolution neural network method belongs to the weak supervised learning algorithm, it is relatively fragile to abnormal data and difficult to define relevant parameters. Therefore, with the expansion of the scale of training label rate, the F1-score index continues to slow down. Due to the setting of community detection loss function, the proposed model will update the parameters in the network model during the detection process, so it has better performance. When the training label rate is 70%, the F1-score index of the proposed model is 17.33% higher than that of LGNN algorithm [12]. Compared with the CP-GNN algorithm [13], F1-score is increased by 16.56%. Compared with the GCN algorithm [14], F1-score is improved by 7.32%.

## 5. Conclusion

Aiming at the problems that the traditional community detection methods are greatly affected by priori parameters, a complex network dynamic community detection method based on deep self-coding network is proposed. The basic ideas are as follows: ① algorithm framework based on encoding-decoding mechanism; ② local feature reconstruction based on decoding results; ③ establishment of community detection loss function. Experimental results show that compared with the original method, the proposed algorithm has detection accuracy and realizes the effective detection of complex network community structure. At present, the proposed model can only be tested on graphs of moderate size, and it has been proved that the proposed model is not enough for large-scale graphs in terms of architecture and training algorithm. The main research direction in the next step is how to achieve the correct compromise between performance, computational complexity, memory consumption, training, and reasoning time, and finally improve the performance.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Natural Science Foundation Project of Hunan Province (no. 2020JJ5397) and the Educational Science Research of Hunan Province (nos. 20B409 and 20A347).

## References

- [1] W. Liu, M. Gong, Z. Tang, A. Qin, K. Sheng, and M. Xu, "Locality preserving dense graph convolutional networks with graph context-aware node representations," *Neural Networks*, vol. 143, no. 143, pp. 108–120, 2021.
- [2] Y. Kichikawa, H. Iyetomi, T. Iino, and H. Inoue, "Community structure based on circular flow in a large-scale transaction network," *Applied Network Science*, vol. 4, no. 1, pp. 1–23, 2019.
- [3] J. Qiu, J. Peng, and Y. Zhai, "Network community detection based on spectral clustering," in *Proceedings of the 2014 International Conference on Machine Learning and Cybernetics*, pp. 648–652, IEEE, Lanzhou, China, July 2014.
- [4] L. Tang, X. Wang, and H. Liu, "Community detection via heterogeneous interaction analysis," *Data Mining and Knowledge Discovery*, vol. 25, no. 1, pp. 1–33, 2012.
- [5] L. Tang, X. Wang, and H. Liu, "Uncovering groups via heterogeneous interaction analysis," in *Proceedings of the IEEE International Conference on Data Mining*, pp. 503–512, IEEE, Miami Beach, FL, USA, December 2009.
- [6] J. Chen, M. Liu, and X. Liu, "Research on of overlapping community detection algorithm based on tag influence," *Cluster Computing*, vol. 22, no. S3, pp. 6669–6679, 2019.
- [7] T. Ma, Q. Liu, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, "LGIEM: global and local node influence based community detection," *Future Generation Computer Systems*, vol. 105, pp. 533–546, 2020.
- [8] M. Bouguessa, R. Missaoui, and M. Talbi, "A novel approach for detecting community structure in network," in *Proceedings of the 2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, pp. 469–477, IEEE, Limassol, Cyprus, November 2014.
- [9] Q. Cai, L. Ma, M. Gong, and D. Tian, "A survey on network community detection based on evolutionary computation," *International Journal of Bio-Inspired Computation*, vol. 8, no. 2, p. 84, 2016.
- [10] L. Ni, W. Luo, W. Zhu, and B. Hua, "Local overlapping community detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 14, no. 1, pp. 1–25, 2020.
- [11] V. Moscato, A. Picariello, and G. Sperli, "Community detection based on game theory," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 773–782, 2019.
- [12] Z. Chen, X. Li, and J. Bruna, "Supervised community detection with line graph NeuralNetworks," in *Proceedings of the International Conference on Learning Representations*, pp. 1–24, Open Review, New Orleans, LA, USA, May, 2019.



- [13] L. Luo, Y. Fang, X. Cao, X. Zhang, and W. Zhang, "Detecting communities from heterogeneous graphs: a context path-based graph neural network model," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 1170–1180, ACM, Queensland Australia, November 1-5 2021.
- [14] X. Wang, J. Li, L. Yang, H. Mi, and J. Y. Yu, "Weakly-supervised learning for community detection based on graph convolution in attributed networks," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 12, pp. 3529–3539, 2021.
- [15] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3438–3445, AAAI Press, New York, NY, USA, February 7-12 2020.
- [16] E. Protas, J. D. Bratti, J. F. O. Gaya, P. Drews, and S. S. C. Botelho, "Visualization methods for image transformation convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2231–2243, 2019.
- [17] F. Liu, S. Xue, J. Wu et al., "Deep learning for community detection: progress, challenges and opportunities," in *Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp. 4981–4987, International Joint Conferences on Artificial Intelligence, Yokohama, Japan, July 2020.
- [18] N. Vercheval, H. De Bie, and A. Pižurica, "Variational auto-encoders without graph coarsening for fine mesh learning," in *Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2681–2685, IEEE, Abu Dhabi, United Arab Emirates, October 2020.
- [19] Z. Wang, L. Wang, S. Liu, and G. Wei, "Encoding-Decoding-Based control and filtering of networked systems: insights, developments and opportunities," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 3–18, 2018.
- [20] T. Li and L. Xie, "Distributed coordination of multi-agent systems with quantized-observer based encoding-decoding," *IEEE Transactions on Automatic Control*, vol. 57, no. 12, pp. 3023–3037, 2012.
- [21] X. Xiong, K. Ozbay, L. Jin, and C. Feng, "Dynamic origin-destination matrix prediction with line graph neural networks and kalman filter," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 8, pp. 491–503, 2020.
- [22] K. Guo, X. Huang, L. Wu, and Y. Chen, "Local community detection algorithm based on local modularity density," *Applied Intelligence*, vol. 52, no. 2, pp. 1238–1253, 2022.
- [23] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.
- [24] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.