IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 April 2024, accepted 7 May 2024, date of publication 13 May 2024, date of current version 24 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3399883

RESEARCH ARTICLE

Robust Overlapping Community Detection in Complex Networks With Graph Convolutional Networks and Fuzzy C-Means

MOHAMMED NASSER AL-ANDOLI [1], IRIANTO [2], JAMIL ABEDALRAHIM ALSAYAYDEH [3], (Member, IEEE), IBRAHIM M. ALWAYLE [4], CHE KU NURAINI CHE KU MOHD [1], AND FAHD ABUHOUREYAH [5]

[1]Pervasive Computing and Educational Technology, Department of Media Interactive, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Melaka 76100, Malaysia
[2]Department-General Education, Faculty of Resilience, Rabdan Academy, Abu Dhabi, United Arab Emirates
[3]Department of Engineering Technology, Fakulti Teknologi and Kejuruteraan Elektronik and Komputer (FTKEK), Universiti Teknikal Malaysia Melaka (UTeM), Melaka 76100, Malaysia
[4]Department of Computer Science, College of Science and Arts in Sharurah, Najran University, Najran 61441, Saudi Arabia
[5]Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer (FKEKK), Universiti Teknikal Malaysia Melaka (UTeM), Durian Tunggal, Melaka 76100, Malaysia

Corresponding authors: Mohammed Nasser Al-Andoli (nasser@utem.edu.my) and Jamil Abedalrahim Alsayaydeh (jamil@utem.edu.my)

ABSTRACT Community detection is an important task in complex network analysis. A community is a set of cohesive vertices that have more connections within the set than outside. In many real Complex Networks (CNs), these communities naturally overlap, meaning an individual node can belong to more than one community. This overlapping structure is crucial for many real applications, such as social influence detection, cyberattack detection, and recommendation systems. Existing methods often struggle to capture both network topology and node features, leading to suboptimal overlapping community detection. In this paper, we propose an efficient method called GCNFCM, which utilizes Graph Convolutional Networks (GCNs), Fuzzy C-means (FCM), and the modularity Q algorithm for overlapping community detection. The key idea is to achieve robust feature learning for nodes and then identify the best structure for overlapping community detection. GCNFCM extracts node embeddings from CNs, considering both topology and attributes through a dual-decoder design (inner product and GCN), while FCM is employed for optimal overlapping community detection. Furthermore, FCM is guided by the modularity Q algorithm for accurate community identification without requiring prior knowledge of the community count. Experimental results on ten real-world CNs of varying sizes demonstrate that our proposed method outperforms other state-of-the-art overlapping community detection methods in terms of producing cohesive communities and identifying ground-truth communities. Additionally, the results indicate that the developed method effectively identifies good overlapping communities in real-world networks.

INDEX TERMS Graph convolutional networks, fuzzy c-means, complex networks, overlapping structure, community detection.

I. INTRODUCTION

Nowadays, with the rapid expansion of information technology, things in the real world are more connected than ever because and usually represented as complex networks [1]. Complex Networks (CN) are distinguished by intricate interconnections, which serve as a powerful analytical tool for understanding a variety of interconnected systems in the real world. These systems encompass diverse domains such as the internet, biological neural networks, transportation systems,

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

and social networks [1], [2]. Complex networks like social media, protein interactions, and city routes are difficult to understand because they have complicated structures and keep changing. In real-world networks, connected groups are called ''communities'', which share common things. Identifying these communities is crucial for understanding network structures, and is valuable in different areas like biology, sociology, and computer science. For example, in computer science, finding communities helps in online marketing, predicting user behaviour, and understanding complex systems [3].

Community detection was introduced by Girvan and Newman in 2002, which involves discovering the structure of a CN by identifying densely connected subgraphs (communities) with sparser inter-community links. It helps to identify distinct communities or subgraphs within a network, which can represent social groupings, functional groupings in biological networks, or other related topics on the web, among other possibilities [4], [5]. By detecting and identifying these communities, we can gain valuable insights into the organization, function, dynamics, and evolution of CNs. Community detection has become a hot topic in recent years, particularly in the field of social networks analysis [6]. This technique finds broad applications, which ranges from constructing models for detecting cyber-attacks in social networks [7], designing recommendation systems [8], to analyzing social influence [9]. In neurobiology, community detection sheds light on the functional dynamics of neuronal networks. Similarly, data center networks leverage community detection for the efficient placement of online service function chains [10]. Therefore, community detection has garnered massive attention in the research community in the first research work two decades ago.

Various community detection methods have been proposed from different perspectives [4], [10], [11], [12], using various techniques such as autoencoder models [13], [14], [15], nonnegative matrix factorization [16], pointwise mutual information [17]. Learning low-dimensional representations enhances member similarity and preserves the network structure. Traditional methods like generative models and spectral mapping, as well as local expansion methods suffer from high complexity and local optima issues [1], [18]. The rapid evolution of machine learning has transformed community detection algorithms from statistical methods to machine learning-based approaches. Neural network models, including Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Generative Adversarial Network (GAN), have demonstrated remarkable effectiveness in community detection [2], [19], [20]. Machine learning-based methods that leverage the potent ability of neural networks to identify node relationships within graphs outperform traditional approaches. In recent years, Graph Convolutional Network (GCN) has emerged as a robust tool for handling graph-structured data with deep learning algorithms [21], [22]. GCNs has the ability to reveal higher-order structural information through non-linear feature aggregation and information propagation across the network. This lead to find wide applications in various network analysis tasks, such as link prediction, node classification, and community detection [21], [22], [23].

The recognition that most real-world networks exhibit overlapping communities underscores the nuanced nature of social structures. In social networks, users commonly participate in various communities dictated by diverse affiliations, such as school, family, and friends [23], [24]. Discovering the overlapping community structure within these networks holds greater practical significance as it mirrors the multifaceted relationships individuals maintain across different domains [11], [25]. Considering the vast scale and intricate structures of networks, which often encompass thousands to millions of user nodes, the exploration of overlapping community detection has become a prevailing research focus. This trend aligns with the imperative to develop sophisticated methods capable of discerning intricate patterns of connectivity in complex networks.

Several community detection methods have been proposed in the literatures [1], [3], and [19]; however, a prevailing limitation lies in their emphasis on non-overlapping community structures and neglect of the inherent overlap between communities [24]. In real-world scenarios, communities frequently exhibit membership overlap with nodes belonging to multiple communities concurrently. However, a few studies have introduced overlapping community detection methods [24], [26]. This methods often used only topology information of CNs [26], [27], and contend with limitations rooted in their fundamental principles, resulting in suboptimal accuracy in detecting overlapping communities. Moreover, these methods typically rely on prior information about the underlying community network, such as the number of communities, posing a constraint on their applicability. For instance, in [11], a community detection method based on Markov and GCN, requires prior knowledge of the number of clusters and exhibits sensitivity to graph structure. This introduces challenges in generalization, computational complexity for large graphs, assumptions related to Markov stability, and the need for the effective selection of the optimal Markov time. In [25], researchers propose an overlapping community detection method with GCN and first-order similarity optimization. However, the complex structures of community networks, often laden with noisy edges and present a challenge in accurately assigning nodes to their respective communities.

In this paper, we propose a new approach for CN nodes feature extraction and overlapping community detection in complex networks. GCNFCM comprises two key components: (1) a Graph auto-encoder with GCN for learning robust features and embedding nodes into a low-dimensional space, and (2) a combined approach of Fuzzy C-means (FCM) and the modularity Q algorithm for accurately identifying overlapping communities. For effective feature learning, the developed method employs a graph autoencoder architecture based on a dual-decoder design for the decoding process.

This design diverges from conventional graph autoencoders by incorporating two decoders: The Inner Product Decoder (IPD) reconstructs the CN adjacency matrix to effectively capture the intricate relationships between nodes, and the Graph Convolutional Decoder (GCD) reconstructs the node feature matrix X to preserve the unique characteristics and attributes of each node. If the CN lacks a node feature matrix (X), we generate the feature matrix using the similarity of nodes based on shared relations and neighbor nodes. For robust overlapping community detection, FCM algorithm is incorporated with the modularity algorithm version that works for overlapping, in which the quality index (Q) guides FCM in identifying the optimal overlapping community structure.

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

- Dual auto-encoder for feature learning. GCNFCM approach uses a dual auto-encoder architecture for to learn low dimensional embedding for CN nodes. This addresses the challenges of CNs by capturing both network topology information and node attribute information (if available), leading to more robust feature learning.
- FCM for overlapping community detection. GCNFCM integrates the FCM algorithm for handling overlapping communities, which allows for a more accurate and realistic representation of community structures compared to traditional methods.
- Improved overlapping community detection with modularity Q. We present a new integration of the modularity Q algorithm with FCM. This combined approach leverages the strengths of both techniques to enhance the effectiveness of FCM in identifying overlapping communities by optimizing the community structure based on modularity Q.
- Automatic community number detection: GCNFCM eliminates the need for prior knowledge about the number of communities. This is a significant contribution as it automates a crucial step in community detection, making the process more efficient and adaptable to various network structures.

The rest of the paper proceeds as follows. Section II reviews and discusses related works. Section III presents the proposed method. It then presents the experimental results, analysis, and discussion, which are then compared with state-of-the-art methods. Finally, Section V concludes the research work and lists future work.

II. RELATED WORK

Recently, community detection in CNs has secured high attention from researchers, which results to develop diverse community detection methods. These methods can be broadly categorized into two main groups: traditional and machine learning-based techniques. In this section, we first provide an overview of traditional overlapping community detection methods and then delve into machine learning methods. The key traditional methodologies include label propagation, clique percolation, modularity optimization, edge betweenness, and game theory.

The label propagation algorithm [28] stands out as one of the common community detection methods. It is a straightforward yet efficient approach with nearly linear time complexity. The algorithm exhibits a drawback of producing unstable results. To address this limitation for overlapping community detection, researchers have extended label propagation, which leads to develop several algorithms [29]. These extensions introduce new label expressions that enable a node to belong to more than one community. Another method, the Clique Percolation Method (CPM) [30] has been proposed for overlapping community detection. CPM posits that communities are composed of overlapping complete subgraphs. However, it is noted that CPM is most effective in networks with densely connected subgroups, which presents limitations in uncovering community structures within large-scale social networks.

Modularity optimization (Q) is the most popular of community detection in CNs, which was introduced by Newman [4]. The method serves as a metric for evaluating the quality of community structures. Shen et al. proposed an extension of modularity for overlapping community detection to leverage both extended modularity and maximal cliques. This method initializes communities through maximal clique identification and expands them by merging similar communities to maximize extended modularity. On the other side, edge betweenness-based methods rely on edge betweenness centrality, which is defined as the number of shortest paths passing through an edge in a network for community detection. The GN method [10] proposes removing edges with high betweenness to identify communities. Gregory [29] extended this approach for overlapping community detection by introducing the concept of split betweenness to refine the identification of inter-community edges. Alayoub et al. [31] utilized spectral mapping methods to extract latent information and employed the FCM algorithm for overlapping community detection. The authors improved FCM and utilized parallel computing to handle large complex networks.

Another category of traditional community detection method is based on the game theory. These methods serve as powerful mathematical tools for analyzing situations where the choices of one entity influence others in generating communities. Game methods model the community detection process as a game, with each user represented as a player aiming to maximize individual benefits through strategic choices. Zhou et al. [32] proposed a method based on coalition formation games that emphasized cooperation of the player to enhance the overall score of the community. Avrachenkov et al. [33] introduced two cooperative game-theoretic frameworks for community detection.

The emergence of machine learning has led to the exploration of various neural network methods for community detection, including GANs [34], CNNs [34], and GCNs [25], [35]. Additionally, deep learning approaches that employ graph embedding methods have been leveraged for mining community structures [14], [23].

Cai et al. [36] proposed an algorithm that converts edges into images containing contextual information. The Edge2Image method is based on the concept of transforming community detection into an edge classification problem. To implement this idea, the challenge lies in defining and locating inter-community edges. A pre-trained CNN model is then employed to classify these images, and effectively identify community detection. Similarly, Jia et al. introduced an approach for overlapping community detection using graph representation learning-based, called GAN [34]. In this approach, a generator is designed to produce node sets simulating cliques, while a discriminator aims to distinguish between the generated and actual node sets. The output of GAN and the graph embedding are served as the community affiliation weight matrix representing the community partition. However, it is worth noting that GAN method relies on supplementary methods for pretraining the embedding. GAN also faces challenges in convergence once the graph embedding is not adequately pre-trained.

Graph embedding algorithms play a crucial role in learning low-dimensional vectors for each node, a valuable resource for various network analysis tasks, including community detection. Noteworthy among these algorithms is DeepWalk [37], which uses the Skip-Gram model to learn node embeddings through the generation of node sequences via random walks. Differing from DeepWalk, Node2Vec employs a biased random walk to generate node sequences. Furthermore, the utilization of a deep autoencoder aids in extracting latent features, followed by the application of machine learning clustering algorithms to facilitate community detection [23]. Moreover, innovations extend to the integration of stack autoencoders with metaheuristic algorithms, such as Particle Swarm Optimization, improve the capabilities for community detection [12], [14], [15]. These advancements underscore the versatility of graph embedding techniques in contributing to the broader landscape of CNs analysis tasks.

The GCN stands out as the most popular DNN model for graph and CN analysis. Tsitsulin et al. utilized a single-layer GCN to generate the community affiliation weight matrix, aligning the loss function with the reformulated modularity [38]. It is crucial to recognize that optimizing modularity leads to over- or under-partitioning, which results in a failure to capture the inherent network structure [39]. similarly, a GCN-based community detection method is introduced based on optimizing first-order similarity [25]. The community affiliation weight matrix is obtained through the GCN to emphasize the optimization of first-order similarity. However, relying solely on first-order similarity is inadequate for accurately extracting the community structure, as it exhibits poor performance in networks with intricate structures involving numerous noisy edges connecting different communities. Yuan et al. [11] developed an overlapping community detection method based on GCN to maximize the Markov stability of the community structure. However, markov models face challenges with limited memory, static assumptions, and an inability to capture long-term dependencies or external influences in large complex networks.

III. METHODOLOGY

This section introduces a new approach for graph feature extraction and overlapping community detection in complex networks, called GCNFCM. GCNFCM consists of two key components:

1. *Graph Auto-encoder with GCN (Fig 1(a)):* This part learns robust features and embeds nodes into a low-dimensional space. The feature learning process starts by receiving the adjacency matrix ($\boldsymbol{A}$) of the network and the node attribute matrix ($\boldsymbol{X}$) if available. Otherwise, a similarity matrix ($\boldsymbol{S}$) will be created as the attribute matrix for the given network (CN). Then, $\boldsymbol{A}$ and $\boldsymbol{X}$ are sent to the GCN encoder to generate embedding nodes as a low-dimensional latent representation space ($\boldsymbol{Z}$) that captures essential network information. $\boldsymbol{Z}$ is then transposed ($\boldsymbol{Z}^{\boldsymbol{T}}$) is then created to reconstruct the $\hat{A}$ matrix through an inner product decoder between $\boldsymbol{Z}$ and $\boldsymbol{Z}^{\boldsymbol{T}}$. Additionally, $\hat{X}$ is reconstructed from the GCN decoder.
2. *Combined Approach of Fuzzy C-means and Modularity Q (Fig 1(b)):* This approach combines the strengths of both FCM and Q algorithms. FCM partitions nodes into fuzzy memberships across multiple communities, while a modularity algorithm ($\tilde{Q}$) is adjusted to optimize the overlapping community structure based on network connectivity. The initial number of communities (c) is set to 2, and the algorithm continues searching for the best overlapping community structure between k communities and the maximum number of communities (max_c).

Table 1 provides a concise list of the main symbols used in the paper, along with their respective meanings, to help readers understand the notation used in this work.

A. DEFINITION OF COMPLEX NETWORKS

A CN is represented and defined as a graph $G(V, E)$, where V *refres to* nodes and E edges, which represent the relationship between nodes, i.e. $V = \{v_1, v_2, v_3, \ldots, v_n\}$,where n is number of nodes, $E \in (V, V)_e$,where e is number of edges. The common network representations indicate the similarity between network member is the adjacency matrix $\boldsymbol{A} = \left[a_{ij}\right] \in R^{n\times n}$. Each element in $\boldsymbol{A}$ denotes to the relationship between two nodes, i.e., $e(v_i, v_j)$ that represents as a_{ij} and can be defined as $a_{ij} = w_{ij}, if\ a_{ij} = 0$ refers to a non−relation between nodes and $a_{ij} = 1$ refers to a relation between i and j. The degree matrix $\boldsymbol{D}$ is a diagonal matrix, where d_{ii} is the degree of vertex i, $D_{ii} = \sum_{j=1}^{n} A_{ij}$. The normalized Laplacian matrix of the graph is $L = I_N - D^{-1/2} - AD^{-1/2}$.

The CN can be defined in another form of graph, which called attributed graph, which *is defined as a* $G = (V, E, X)$, where V *and* E are the nodes and the edges similar to

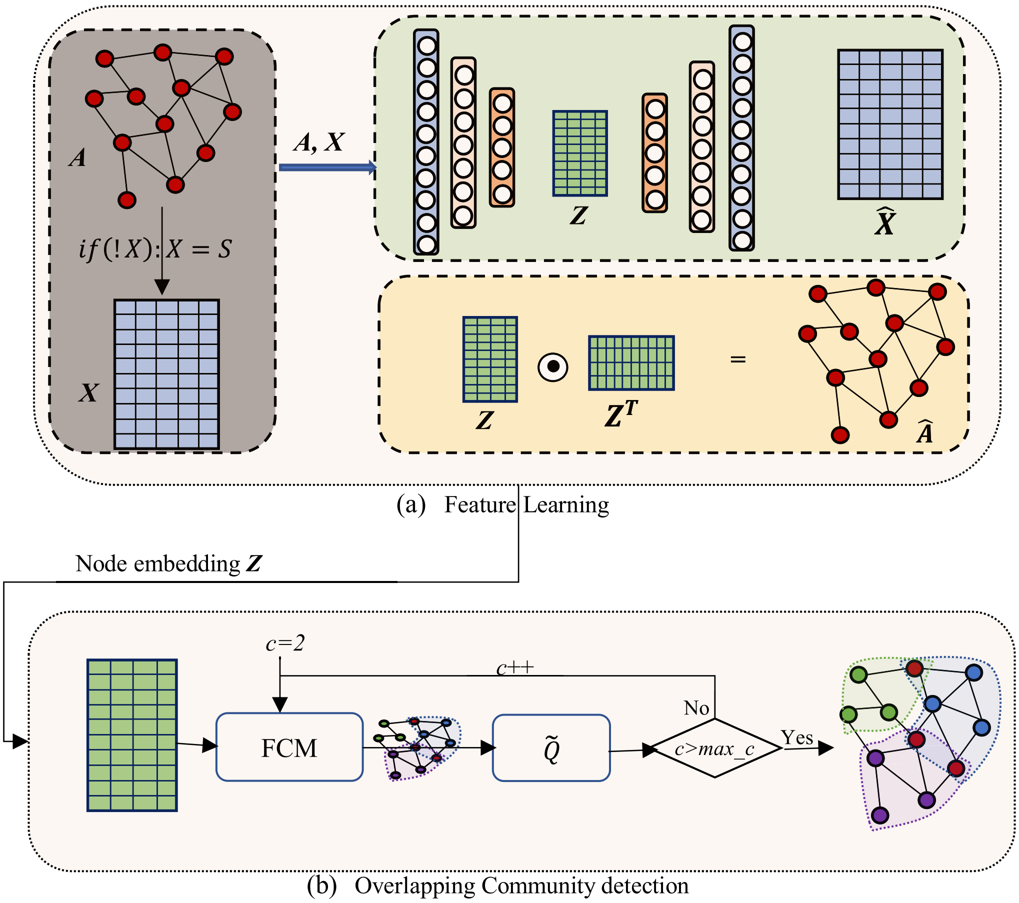


FIGURE 1. Overview architecture of the proposed GCNFCM method.

the normal in addition to the node attribute matrix $X = [X1, X2, X, X1 \ldots, Xn] \in Fn \times d$, *where* $Xi \in Fn \times d$ represents the attributes of node v_i with dimension of node attributes.

In this work, we generate the feature matrix based on the Eq. (1) if the CN lacks a node feature matrix (X).

$$X = S = \frac{2SN(v_i, v_j)}{d\,(v_i) + d\,(v_j)} \tag{1}$$

where $SN(v_i, v_j)$ is a common node between adjacent two nodes v_i and v_j, $d\,(v_i)$ is the degree of node i and $d\,(v_j)$ is the degree of node *j*, and $d\,(v_i) + d\,(v_j)$ is the sum of the degree of both nodes.

Community detection is a process that aims to identify the structure of a given CN/G by assigning nodes to K subgraphs, i.e. communities, $K \in \{k_1, k_2, k_3, \ldots, k_c\}$, where c is the number of clusters. The outcome of community detection can be represented as a community affiliation matrix $H \in [0, 1]_{N\times K}$, where H_{vk} indicates whether node v is affiliated with community k or not. Consequently, community detection can be reduced to finding the community affiliation matrix $\boldsymbol{H}$. Some nodes can be affiliated with more than one community.

B. NODE FEATURE EXTRACTION AND LEARNING

The GCN-Autoencoder (GCNAE) is used as an approach for embedding CN structures and extracting meaningful features. Its objective is to embed the given CN *[V, E, X]*, capturing both CN structures, i.e. topology and node attributes, and represent them in a low-dimensional space. It uses dual auto-encoder for feature extraction. This addresses the challenges of complex networks by capturing both network topology information and node attribute information (if available), leading to more robust feature learning. The workflow of GCNAE depicted in Fig. 1, which comprises two key components: an informative CN encoder and two distinct CN decoders. To achieve this, GCNAE utilizes the adjacency matrix (***A***) and the node feature matrix (***X***) to learn a latent representation (***Z***) through a GCN encoder. The subsequent step involves two separate decoders: one to reconstruct node features and the other to reconstruct CN structures. These decoders leverage the latent information ***Z***, ensuring a comprehensive utilization of the learned representation. The final outcome of GCNAE is a low-dimensional representation ***Z***, which serves as the embedding for the subsequent task—overlapping community detection using the FCM algorithm. This embedding captures essential aspects

TABLE 1. List of symbols used in the paper.

Symbole	Meaning
CN	Complex Network
V	All nodes in a CN
e	number of edges
n	Number of CN nodes
A	Adjacency matrix of a CN
X	Nodes attribute matrix
S	Similarity matrix
θ	Spectral convolution parameters of GCN
$\hat{A}$	Reconstructed adjacency matrix
$\hat{X}$	Reconstructed attribute matrix
Z	Latent representation of CN nodes
Z^T	Transpose matrix of Z
Q	Modularity algorithm
$\tilde{Q}$	Adjusted modularity of overlapping communities
D	Diagonal matrix
L	Laplacian matrix
U	Eigenvector of L
I_n	Identity matrix
W	Trainable parameters of GCD decoder
WE	The weights of edges
λ	A threshold of FCM algorithm
m	fuzziness parameter of FCM
c	Number of communities
max_c	Maximum number of communities
k	*Initial number of communities*
V_c	Nodes in cluster c, $c \in [2, \max\_c]$
U_c	Assignment matrix
ε	Termination threshold of FCM
U_{best}	Final overlapping community detection

of the original CN, facilitating more efficient community detection in the reduced-dimensional space. The proposed GCNAE thus offers a robust framework for embedding and extracting features from complex network data, with applications extending to tasks like community detection.

1) GCN-ENCODER

The spectral convolution on a CN or a graph is a fundamental operation that is defined as the product of a signal $x \in Rn$ (a scalar for each node) with a filter $g_\theta = diag(\theta)$ parameterized by $\theta \in$ R n in the Fourier domain:

$$g_\theta * x = Ug\theta UTx \tag{2}$$

Here, g_θ is a function of the eigenvalues of the CN Laplacian matrix $\mathrm{L} = \mathrm{I}_N - \mathrm{D}^{-1/2} - \mathrm{AD}^{-1/2} = \mathrm{U} \wedge \mathrm{U}^T$, where $\boldsymbol{U}$ is the eigenvector of $\boldsymbol{L}$,and $\boldsymbol{D}$ is the diagonal matrix. To approximate the spectral filter g_θ, R^{th} order Chebyshev polynomials [40] were employed. The resulting spectral convolution on the graph can be expressed as:

$$g_\theta * x \approx \theta\left(I_n + D^{-\frac{1}{2}} - AD^{-\frac{1}{2}}\right)x \tag{3}$$

This equation can be normalized to:

$$I_n + D^{-\frac{1}{2}} - AD^{-\frac{1}{2}} \to \tilde{D}^{-\frac{1}{2}} - \tilde{A}\tilde{D}^{-\frac{1}{2}}, \tag{4}$$

where I_n, A, D, are identity adjacency, and diagonal matrices, respectively, and $\tilde{A} = A + I_n$ *and* $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $\tilde{A} = A + I_n$ denotes to the identity matrix.

After that, the forward path of the graph convolutional layer can be expressed as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{5}$$

where H^l represents the features that extracted from l_{th} layer, the input node feature matrix $\boldsymbol{X}$ *is* received by l_0 and represented by H_0. a nonlinear activation function (e.g., $ReLU(\cdot)$ is represented by $\sigma(\cdot)$.W^l is a trainable parameters of the layer l, which is called the layer weight matrix

The graph encoder used in this study comprises two convolutional layers. Before that, a preprocessing step is conducted to prepare the data for the convolution layers, the preprocessing step implemented using this formulation $\bar{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$. Then, the convolutional encoder model is performed using Eq. (6):

$$Z = f(X, A) = \bar{A}ReLU(\bar{A}XW^0)W^1 \tag{6}$$

where $\boldsymbol{X}$ is the attribute matrix, $\boldsymbol{A}$ is the adjacency matrix, and (W^0 *and* W^1) are the trainable parameters of the convolutional encoder, and $\bar{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$. Nonlinear $ReLU(\cdot)$ linear activation function is employed for the first layer, while the linear activation function is used for the second layer. The encoder of GCNAE model is constructed with two convolutional layers, effectively embedding both the CN topological structure $\boldsymbol{A}$ and the node feature $\boldsymbol{X}$ into the latent representation $\boldsymbol{Z}$.

2) DUAL-DECODER

To enhance the representation of both the CN structure and node features, GCNAE employs a graph autoencoder architecture based on a dual-decoder design for the decoding process. The GCNAE model diverges from conventional graph autoencoders by incorporating two decoders: the IPD, which reconstructs the CN adjacency matrix $\boldsymbol{A}$ to effectively capture the intricate relationships between nodes; and the GCD reconstructs the node feature matrix $\boldsymbol{X}$ to preserve the unique characteristics and attributes of each node.

a: IPD DECODER

The inner product decoder is employed due to its adeptness in reconstructing the CN/graph structure matrix $\boldsymbol{A}$ efficiency and effectively. Its advantage lies in the simplicity of matrix multiplication (ZZ^T), making it computationally efficient, especially for large CNs. Furthermore, the reliance on the latent representation $\boldsymbol{Z}$ grants inherent flexibility in capturing diverse CN structures whether sparse or dense without the need for intricate model modifications. Eq. (7) is used to capture this reconstruction process:

$$\hat{A} = \sigma(ZZ^T) \tag{7}$$

where $\hat{A}$ is the reconstructed matrix, $\boldsymbol{Z}$ is the latent representation, $\boldsymbol{Z^T}$ is the transpose matrix of $\boldsymbol{Z}$ and σ is the sigmoid function ensuring edge probabilities reside within the [1, 0] range.

b: GCD DECODER

The GCD decoder is employed to reconstruct the feature node matrix of a given CN, denoted as $\boldsymbol{X}$. The encoding process utilizes $\boldsymbol{A}$ and $\boldsymbol{X}$ to generate the latent representation Z; consequently, the decoder segment should create $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{X}}$. $\hat{\boldsymbol{A}}$ is reconstructed using the IPD decoder, and then the GCD utilizes the graph convolutional decoder to generate $\hat{X}$. This process is expressed as:

$$\hat{X} = f(Z, A) = \bar{A}ReLU(\bar{A}XW^0)W^1 \tag{8}$$

where $\boldsymbol{Z}$ is the latent representation extracted using the encoder part of the GCNAE model, $\boldsymbol{A}$ is the adjacency matrix, $\bar{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, W represents the trainable parameters of the GCD decoder, and *Relu* is the nonlinear activation function.

3) LOSS FUNCTION OF GCNAE

The loss function (L) of the GCNAE model consists of two parts: one (L_1) for the original adjacency matrix $\boldsymbol{A}$ and the reconstructed adjacency matrix $\hat{\boldsymbol{A}}$, and the other (L_2) for the original node feature matrix $\boldsymbol{X}$ and the reconstructed node feature matrix $\hat{\boldsymbol{X}}$.

L_1 loss function employs the posterior inference, which is commonly known as an encoder. L_1 can be defined as follows:

$$\begin{aligned} L1 &= E_{q(Z|(X,A))}\left[logp\left(A \mid Z\right)\right], p\left(A \mid Z\right) \\ &= \prod_{i=1}^{n}\prod_{j=1}^{n} p(A_{ij}|z_i, z_j) \; with \; p(A_{ij} = 1|z_i, z_j) = \sigma(z_i^T z_j) \end{aligned} \tag{9}$$

where *q(Z|(X, A))* is posterior inference, $\boldsymbol{Z}$ is the extracted latent representation, $\boldsymbol{Z^T}$ is the transpose matrix of $\boldsymbol{Z}$, $\boldsymbol{A}$ is the adjacency matrix, and $\boldsymbol{X}$ is the attribute matrix. This component can be parameterized by encoder neural networks, which facilitates posterior inference across all data points in the dataset. In this context, it can be conceptualized as the CN/graph encoding process, wherein the graph convolutional encoder integrates the graph structure $\boldsymbol{A}$ and attribute matrix $\boldsymbol{X}$ to produce the embedding $\boldsymbol{Z}$ [41].

Regarding the reconstruction of the node feature matrix using the GCD. Loss function L_2 is defined as:

$$L_2 = \frac{1}{2}\left\|X - \hat{X}\right\|^2, \tag{10}$$

where X and $\hat{X}$ are the attribute and the reconstructed attribute matrices, respectively. The whole GCNAE model optimized by L_1 and L_2 to minimize the loss function L that is expressed as:

$$L = L_1 + \lambda L_2 \tag{11}$$

where L_1, L_2, λ are the first loss function, the second loss function, and a non-negative regularization parameter, respectively. The GCNAE can be updated with its stochastic gradient by minimizing L in Eq. (11).

The GCNAE model steps are summarized in Algorithm 1, where V denotes to nodes set, $\boldsymbol{A}$ is the adjacency matrix, and $\boldsymbol{X}$ is the attribute matrix.

Algorithm 1 GCNAE

Input: *V*,$\boldsymbol{A}$, and $\boldsymbol{X}$.
Output: Latent representation, i.e. $\boldsymbol{Z}$.
(a) Pre-processing:
  Compute the degree matrix ($\boldsymbol{D}$) of $\boldsymbol{A}$, $D_{ii} = \sum_{j=1}^{n} A_{ij}$
  Compute the normalized Laplacian ($\boldsymbol{L}$) matrix of the CN,
  $L = I_N - D^{-\frac{1}{2}} - AD^{-\frac{1}{2}}$
(b) GCN-Encoder:
  Obtain the latent representation of CN nodes z using Eq, (6)
(c) Decoders:
  1)*IPD decoder*
    Regenerate the adjacency matrix$\hat{A}$ using Eq. (7)
  2)*GCD decoder*:
    Regenerate the attribute matrix $\hat{X}$ of CN nodes using Eq. (8)
(D) Compute loss function:
  Update the trainable parameters ($\boldsymbol{W}$) of the GCNAE model by minimizing Eq. (11)
Return Latent representation of CN nodes $\boldsymbol{Z}$.

C. FCMQ BASED ON OVERLAPPING COMMUNITY DETECTION

FCMQ is an incorporation between FCM and Q modularity algorithms. FCM for the overlapping community detection and modularity algorithm for measuring the quality.

1) FCM

A data clustering technique that assigns data to clusters based on a membership degree within the range of [1, 0]. This indicates the extent to which it belongs to a specific cluster. FCM leverages fuzzy sets to allow data points to have partial memberships in multiple clusters. The technique achieves robust partitioning through iterative optimization of the following objective function.

$$J_m = \sum_{i=1}^{n}\sum_{j=1}^{C} U_{ij}^{m} \left|Z_i - C_j\right|^2 \tag{12}$$

where n is the number of nodes in a CN, C denotes the number of clusters, $\boldsymbol{Z}$ is the latent representation of node i, m is the fuzziness factor, and $\boldsymbol{U_{ij}}$ is the membership degree of node i in cluster j, calculated using Eq. (14).

$$U_{ij} = \frac{1}{\sum_{k=1}^{C}\left[\frac{|Z_i - C_j|}{|Z_i - C_k|}\right]^{\frac{2}{m-1}}} \tag{13}$$

After that, the cluster centers are initially generated randomly and then updated using the following equation:

$$C_j = \frac{\sum_{i=1}^{n} U_{ij}^{m} Z_i}{\sum_{i=1}^{n} U_{ij}^{m}} \tag{14}$$

FCM continues iterations until the specified condition is satisfactorily met to ensure an optimal outcome in the clustering process.

$$\left\{\left|U_{ij}^{t-1} - U_{ij}^{t}\right|\right\} < \varepsilon, \; where \; 0 < \varepsilon < 1, \tag{15}$$

where ε is the termination threshold. Finally, the soft assignments $\boldsymbol{U_c}$, which correspond to n nodes and c clusters, are sent to the modularity computation for measuring the quality of overlapping community detection.

2) MODULARITY COMPUTATION ($\tilde{Q}$)

The Modularity algorithm incorporates FCM to enhance the quality of overlapping community detection and guarantee a significant number of relations within communities compared to the relations between nodes in different communities, especially when there are no overlapping relations between nodes. The original modularity algorithm is designed for crisp community detection, which evaluates the quality of a specific network division. Zhang et al. [27] modified the original Modularity algorithm proposed by Newman [42] to adapt it to overlapping community detections ($\tilde{Q}$) and to integrate with the FCM algorithm [19].

The modularity computation $\tilde{Q}$ receives from the FCM algorithm the membership values, i.e., soft assignment matrix $U_c = [u_1, \ldots, u_c], 0 \le u_{ik} \le 1, and \sum_{k=1}^{c} u_{ic} = 1$.$\tilde{Q}$ is computed as follows

$$\begin{aligned}
\tilde{Q} &= \sum_{k=1}^{c} \left[\frac{A(V_k, V_k)}{A(V, V)} - \left(\frac{A(V_k, V)}{A(V, V)} \right)^2 \right], V_k \\
&= \{ i \,|\, u_{ik} > \lambda, i \in V \}, \\
A(V_k, V_k) &= \sum_{i,j \in V_k}^{c} \left(\frac{U_{ki} + U_{kj}}{2} \right) WE_{ij}, \\
A(V_k, V) &= A(V_k, V_k) \\
&\quad + \sum_{\substack{i \in V_k, \\ j \in (V - V_k)}} \left(\frac{U_{ki} + (1 - U_{kj})}{2} \right) WE_{ij}, \\
A(V, V) &= \sum_{i,j=1}^{n} WE_{ij} \qquad (16)
\end{aligned}$$

where V denotes to all nodes, WE is the weights of edges, V_k represents nodes in cluster k, $k = 1, \ldots, c$. λ is a threshold that can be used to determine the final cluster assignment for each node in a soft assignment matrix.

After obtaining the $\tilde{Q}$ results, the process will back to the FCM to try another clustering with different c clusters, as the main purpose of $\tilde{Q}$ to help FCM to find the best overlapping community detection with the suitable number of clusters.

In summarizing the key steps of the FCMQ algorithm combination, the process begins with the FCM initiating the clustering procedure with c clusters set to 2, and subsequently computing the soft assignment matrix $\boldsymbol{U_c}$. This matrix $\boldsymbol{U_c}$ is then transmitted to the modularity computation $\tilde{Q}$ to assess the quality of overlapping community detection within communities c. The results of the modularity computation are sent back, and the subsequent step involves incrementing the number of communities (c) by 1. The FCM then repeats the clustering process with the updated number of communities (c), calculates the new soft assignment matrix $\boldsymbol{U_c}$, and transmits it once again to $\tilde{Q}$ for evaluating the quality of overlapping community detection. This iterative process continues until it reaches *max_c*, at which point the algorithm selects the best overlapping community detection based on high $\tilde{Q}$ values. Algorithm 2 shows the FCMQ algorithm process. The input data in Algorithm 1 is the CN latent representation $\boldsymbol{Z}$, adjacency matrix $\boldsymbol{A}$, fuzziness parameter m, the termination criterion ε, the threshold for identifying overlapping λ, W is the weight matrix the maximum number of clusters *max_c*,$\boldsymbol{U}_{best}$ is the final overlapping community detection with the best modularity computation $\tilde{Q}$.

Algorithm 2 FCMQ

Input: $\boldsymbol{Z}, \boldsymbol{A}, m, \varepsilon, \lambda$, max_c, WE.
Output: Overlapping community detection.
Set $c = 2$, i.e. initial number of communities
Repeat :
  a)*FCM*
  Initialize the community centers randomly
  Repeat :
    Update the soft assignment matrix U_c with Eq. (14)
    Update the difference of soft assignment matrix
    with $\left| U_{ij}^{t-1} - U_{ij}^{t} \right|$.
    Update the community centre values with Eq. (15)
  Until;
  $\left\{ \left| U_{ij}^{t-1} - U_{ij}^{t} \right| \right\} < \varepsilon$
  b)*Modularity Computation* ($\tilde{Q}$):
  Receive soft assignment matrix $\boldsymbol{U_c}$
  Calculate $\tilde{Q}$ with Eq. (17)
  If $(\tilde{Q}_{c-1} > \tilde{Q}_c)$:
    $U_{best} = U_c$
  Endif
  Increase number of communities, i.e., $c{+}{+}$.
Until;$c > max\_c$
Return the best overlapping community detection $\boldsymbol{U}_{best}$.

IV. IMPLEMENTATION AND RESULTS

This section provides a comprehensive assessment of GCNFCM for overlapping community detection. It begins by introducing the experimental setups. Subsequently, we compare the performance of the proposed method with other comparison methods using datasets from real-world CNs. Finally, we present and analyze the experimental results.

A. EXPERIMENTAL SETUP

1) DATASET DESCRIPTION

To assess the effectiveness of the proposed method (GCNFCM), ten real datasets (CNs) have been employed. These datasets are commonly utilized by researchers in the field of community detection, which offers the diversity in terms of size, sources, and content. This diversity ensures a comprehensive evaluation and provides a fair comparison between the developed method and other approaches through a benchmark study. The datasets are classified into two categories.

The first category comprises six datasets that represent CNs with network topology only. These datasets include:

1. ***Zachary's Karate Club*** network, which consists of two groups and 34 elements.
2. ***High School*** network, which consists of 68 pupils distributed across 6–7 communities (School6 and School7).
3. ***Dolphins*** network, which contains 62 elements in two groups.
4. ***American Football*** network, which comprises 115 nodes distributed in 12 clusters.
5. ***Polbooks*** network, which consists of featuring 105 members across three groups.
6. ***Political blogs (Polblogs)*** network, which involves 1490 blogs assigned into two groups.

For this category, $\boldsymbol{X}$ is generated using Eq. (1).

The second category encompasses three datasets that include node attributes. These datasets are:

1. ***Cora*** network, which involves 2,708 publications with a feature dimension of 1433, categorized into seven clusters.
2. ***Citeseer*** dataset, which includes 3,312 scientific papers with 3703 feature dimensions, assigned to six clusters.
3. ***Pubmed*** network, which composes of 19,717 diabetes-related articles from the Pubmed database. Each article is assigned to one of three classes, and the feature vectors contain term frequency-inverse document frequency (TF/IDF) scores for 500 words.

This comprehensive selection of datasets ensures a robust evaluation of the proposed method for both network topology and node attributes in various contexts.

2) PARAMETERS

The GCNFCM method consists of two phases; GCNAE (feature learning) and FCMQ (overlapping community detection).

During the feature learning phase, the GCN autoencoder in GCNFCM is trained in a supervised manner to extract latent representation of CN node i.e., $\boldsymbol{Z}$. The model utilizes five layers with small-sized CNs. These layers consist of two graph convolutional layers acting as encoders, two acting as decoders, and a middle layer for latent representation. For instance, with the Polblogs dataset, the layer structure and neurons are 1490-256-32-256-1490. For medium-sized datasets, a 7-layer structure is employed. Three layers function as encoders, three as decoders, and one layer serves as the middle layer. An example with the Cora dataset is 3327-256-128-32-128-256-3327. Other parameters such as a learning rate is set to 0.001. The number of iterations is set to 1000 for the Karate, Dolphins, School6, School7, Football, and Polbooks datasets, 200 iterations for Polblogs, Cora, and Citeseer, and 100 iterations for the PubMed dataset.

Following this phase, the latent representation ($\boldsymbol{Z}$) is used in the second phase FCMQ i.e., overlapping community detection. This phase leverages the FCM and Modularity Q algorithms in an unsupervised learning manner.

The GCNFCM method is implemented in *Python* and its libraries such as *NumPy* and *TensorFlow*. The GCNFCM method parameters are carefully selected through iterative experimentation to optimize performance. We have refined these parameters through multiple trials to ensure that GCN-FCM delivers the most effective results. Additionally, some of these parameters, such as the learning rate (e.g., 0.001), have been extensively discussed in the literature and are widely recognized as optimal choices.

3) EVALUATION METRICS

To assess the performance of overlapping community detection methods, we have chosen three community detection metrics: Overlapping Normalized Mutual Information (ONMI) [43], the Omega Index Ω-Index (OMG) [44], and the modularity measurement (Q).

ONMI is a widely utilized metric for evaluating community detection algorithms, particularly in the context of identifying overlapping communities [43]. ONMI employs criteria from information theory to compare the identified communities with the ground truth communities. This metric evaluates the similarity between two partitions, particularly in the context of community detection.

The OMG is a another metric assesses the similarity between two partitions, particularly in the context of community detection [44]. It is an overlapping version of the Adjusted Rand Index, which is designed to handle situations where nodes may belong to multiple communities simultaneously.

The $\tilde{Q}$ modularity is employed to assess the quality of community detection and ensure denser relationships within communities and sparser connections between communities. $\tilde{Q}$ is an updated version of the original modularity algorithm, which is suitable for overlapping communities, as described in Eq(17) [27].

In addition to the above metrics, normalized mutual information (NMI) and adjusted Rand index (ARI) are also used to evaluate community detection regardless of the overlapping process.

B. RESULTS ANALYSIS AND DISCUSSION

To evaluate the effectiveness of GCNFCM, we conduct experiments on ten real-world CNs range from small to medium-scale CNs, as well as includes network topology and network features. ONMI and OMG are used to assess the performance of GCNFCM as all networks have ground-truth node communities and compare it to other baseline methods. In addition, we further evaluate the overlapping community detection using modularity $\tilde{Q}$ to ensure high quality.ENHANCED PERFORMANCE OF GCNFCM

GCNFCM is evaluated across various small and medium-sized CN datasets using a range of evaluation metrics, as summarized in Tables 2 and 3. ONMI, OMG, and $\tilde{Q}$ evaluation metrics are employed for the overlapping community detection. GCNFCM enables also the community detection without overlapping, basically enables the node to belong to only one community, with the highest membership values, so the evaluation of GCNFCM is extended to

TABLE 2. Results of GCNFCM for all metrics on several small CN datasets.

Metrics	Karate	Dolphins	School6	School7	Football	Polbooks
ONMI	1.0	0.889	0.8066	0.8587	0.9083	0.6668
OMG	1.0	0.9348	0.9052	0.9197	0.9379	0.8375
$\tilde{Q}$	0.4335	0.3974	0.6259	0.6251	0.6149	0.514
NMI	1.0	0.889	0.9083	0.9308	0.9568	0.8072
ARI	1.0	0.9348	0.9052	0.9197	0.9379	0.8375
Q	0.371	0.3787	0.5886	0.589	0.5995	0.4253

TABLE 3. Results of GCNFCM for all metrics on several medium-sized CN datasets.

Metrics	Polblogs	Cora	Citeseer	Pubmed
ONMI	0.6821	0.5882	0.5423	0.3858
OMG	0.6915	0.6097	0.5786	0.3946
$\tilde{Q}$	0.5547	0.6811	0.6679	0.4231
NMI	0.6748	0.5824	0.5315	0.3937
ARI	0.6803	0.5923	0.5586	0.4063
Q	0.5480	0.6612	0.6467	0.4472

include NMI, ARI, and Q, where these metrics suitable with non-overlapping community detection.

Table 2 shows the results of GCNFCM on the small-sized CNs. In the *Karate* dataset, the proposed method demonstrates excellent performance and achieves NMI score of 1. This emphasizes the ability to accurately capture community structures. Similarly, the *Dolphins* dataset exhibits a high OMG of 0.9348, which indicates that the effectiveness of GCNFCM in identifying cohesive subgroups. Notably, in the *School6* and *School7* datasets, GCNFCM showcased high quality of overlapping community detection and scored 0.6259 and 0.6251 in $\tilde{Q}$, respectively. This highlights the method ability to maintain community structures even in intricate network configurations. Furthermore, GCNFCM excelled in the *Football* dataset and harvested an OMG of 0.9379, which signifies its aptitude for discerning intricate patterns within the network. In the *Polbooks* dataset, GCNFCM demonstrated moderate performance, with an ONMI of 0.6668 and an ARI of 0.8375, suggesting its proficiency in identifying community structures in diverse network scenarios. Overall, the GCNFCM shows consistent high scores across diverse metrics, demonstrating its effectiveness in community detection within small CNs.

Table 3 presents the comprehensive evaluation results of the GCNFCM method on various medium-sized CN datasets. In the *Polblogs* dataset, GCNFCM demonstrated a commendable performance across multiple metrics, achieving an ONMI of 0.6821 and an OMG of 0.6915, indicating its ability to capture community structures and overlap modularity effectively. Similarly, in the *Cora* dataset, the method exhibited notable results with a $\tilde{Q}$ of 0.6811, showcasing its proficiency in maintaining community structures even though some nodes belong to more than one community. The *Citeseer* dataset further emphasized GCNFCM's robustness, with consistently high scores across all metrics, particularly an $\tilde{Q}$ of 0.6679, ONMI of 0.5786, and ARI of 0.5586, suggesting its effectiveness in identifying community structures amidst complex citation networks. In the *Pubmed* dataset, the method showcased its capability in discerning intricate patterns, achieving a high OMG of 0.3946 and a Q of 0.4472. These results collectively underline GCNFCM's versatility and effectiveness in community detection (overlapping and nonoverlapping) across diverse medium-sized complex network scenarios, providing valuable insights for its application in real-world network analysis.

1) COMPARATIVE ANALYSIS: GCNFCM VS. BASELINE METHODS

a: EVALUATION ON SMALL-SCALE CNS

In evaluating the performance of GCNFCM and other baseline methods on networks with ground-truth community memberships, we employ ONMI and OMG. Tables 4 and 5 present the comprehensive results of GCNFCM and various baseline methods on these small CNs, measured in terms of ONMI and OMG, respectively.

In Table 4, the ONMI scores illustrate the comparative performance of GCNFCM against various baseline methods across different datasets. GCNFCM consistently achieves high scores, indicating its robustness and effectiveness in capturing community structures. For example, with *Karate dataset,* the ONMI score for GCNFCM in the Karate dataset is 1.0, which means an exact match with ground-truth community memberships. GCNFCM outperforms SNMF, NSED, LCEN, CPM, Node2Vec, NOCD, and OCDG and showcases unparalleled accuracy in detecting overlapping communities. Using *Dolphins,* GCNFCM maintains robust performance with ONMI score of 0.8889 and surpasses several

TABLE 4. ONMI score of GCNFCM and various methods on the small CN.

Dataset	SNMF [45]	NSED [46]	LCEN [47]	CPM [30]	Node2Vec [48]	NOCD [25]	OCDG [11]	GCNFCM
Karate	0.1603	0.2206	0.5508	0.2001	1.0	0.8072	0.9166	1.0
Dolphins	0.0897	0.1433	0.8005	0.1916	0.7557	0.5857	0.8072	0.8889
School6	0.0125	0.1582	0.6832	0.2563	0.7043	0.659	0.7384	0.8066
School7	0.1462	0.164	0.691	0.2744	0.7165	0.6647	0.764	0.8587
Football	0.3244	0.3576	0.7582	0.5492	0.8081	0.7731	0.8275	0.9083
Polbooks	0.1018	0.1434	0.4393	0.3354	0.4407	0.4252	0.4623	0.6668

TABLE 5. OMG score of GCNFCM and various methods on the small CNs.

Dataset	SNMF [45]	NSED [46]	LCEN [47]	UOCS [30]	Node2Vec [48]	NOCD [25]	OCDG [11]	GCNFCM
Karate	0.1871	0.3323	0.8822	0.095	1	0.7025	0.8823	1
Dolphins	0.066	0.16	0.8117	0.077	0.7536	0.5798	0.8318	0.9348
School6	0.0117	0.1935	0.704	0.3182	0.734	0.668	0.745	0.9052
School7	0.1321	0.2146	0.731	0.345	0.7562	0.675	0.7734	0.9197
Football	0.347	0.4128	0.81	0.591	0.8245	0.8064	0.8419	0.9379
Polbooks	0.1519	0.2837	0.5683	0.5413	0.6043	0.5881	0.6531	0.8375

TABLE 6. NMI score of GCNFCM and various methods on the medium CNs.

Dataset	Spectral [49]	DeepWalk [37]	GVAE [50]	ADV-GAE [51]	GASN [22]	GAEAS [21]	CDDLHP [12]	GCNFCM
Polblogs	0.1101	0.42	0.397	0.45	0.4360	0.46	0.53	0.6748
Cora	0.126	0.327	0.429	0.505	0.484	0.518	0.32	0.5824
Citeseer	0.055	0.087	0.176	0.365	0.386	0.392	0.20	0.5315
PubMed	0.097	0.279	0.277	0.332	0.313	0.345	0.15	0.3937

baseline methods. This underscores the effectiveness of GCNFCM in identifying overlapping community structures within the Dolphins dataset. GCNFCM consistently achieves competitive ONMI scores on *School6, School7, Football, and Polbooks datasets*. Notably, in the *Football* dataset, GCNFCM outperforms several baseline methods, demonstrating its versatility in handling different network structures.

In Table 5, the OMG scores also emphasize the superior performance of GCNFCM compared to various baseline methods. Particularly noteworthy is its consistently high OMG across diverse datasets, highlighting the ability of GCNFCM to accurately identify community structures with nodes overlapping.

GCNFCM achieves an OMG score of 1.0 with the *Karate* dataset, denoting a perfect match with ground-truth community memberships. It also maintains a high OMG score of 0.9348 with the *Dolphins* dataset. This demonstrates the robustness in detecting overlapping communities. This score positions GCNFCM as a superior method compared to several baseline approaches.

Using *School6, School7, Football,* and *Polbooks* datasets, GCNFCM consistently achieves competitive OMG scores. In the *Football* dataset, GCNFCM outperforms multiple baseline methods, illustrating its versatility and accuracy in handling different network structures. This outstanding performance establishes GCNFCM as a leader in accurately identifying overlapping communities across these datasets. GCNFCM surpasses other methods, including SNMF, NSED, LCEN, UOCS, Node2Vec, NOCD, and OCDG, and this also displays the effectiveness of GCNFCM in capturing complex community structures with overlapping memberships.

The results in Tables 4 and 5 suggest that GCNFCM outperforms the baseline methods in capturing community structures within small CNs, as evidenced by both ONMI and OMG metrics. The robustness and effectiveness of GCNFCM make it a promising approach for overlapping community detection in complex networks.

The superior performance of GCNFCM is due to its design, specifically the use of GCN to generate node embeddings based on both node topology and attributes. Additionally, the method employs a two-step process: first, GCN extracts node embeddings, and then a FCM algorithm is applied for overlapping community detection. Finally, the quality index ($\tilde{Q}$) is used as a metric to guide FCM in identifying the optimal overlapping community structure.

b: EVALUATION ON MEDIUM-SCALE CNS

Table 6 presents the NMI scores for GCNFCM in comparison to various methods on medium-scale CNs. This evaluation is essential for understanding how well GCNFCM performs in capturing community structures in larger networks. The discussion below provides insights into the results and their implications.

TABLE 7. ARI score of GCNFCM and various methods on the medium CNs.

Dataset	Spectral [49]	DeepWalk [37]	GVAE [50]	ADV-GAE [51]	GASN [22]	GAEAS [21]	CDDLHP [12]	GCNFCM
Polblogs	0.1165	0.4147	0.4035	0.4682	0.4214	0.4668	0.5182	0.6803
Cora	0.31	0.242	0.347	0.443	0.392	0.513	0.327	0.5923
Citeseer	0.1	0.092	0.124	0.347	0.371	0.402	0.2267	0.5586
PubMed	0.062	0.299	0.279	0.325	0.310	0.352	0.1483	0.4063

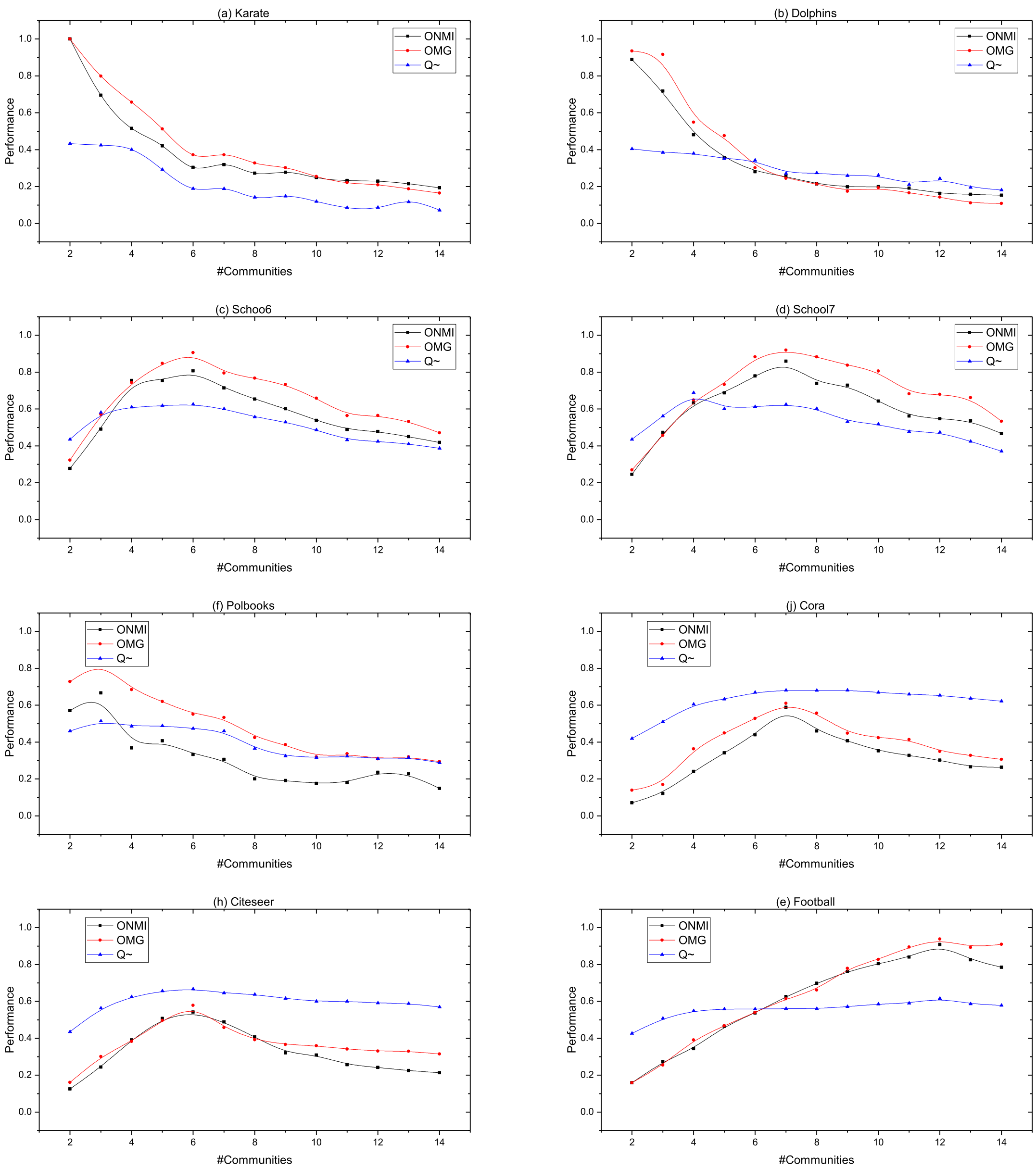


FIGURE 2. Performance of GCNFCM Across Diverse Community Numbers using ONMI, OMG, and $\tilde{Q}$ metrics.

From Table 6, GCNFCM achieves an impressive NMI score of 0.6748 on the *Polblogs* dataset and outperforms various methods. This shows the effectiveness of GCNFCM in identifying communities within this medium-scale network. GCNFCM demonstrates a competitive NMI score of 0.5824 on the *Cora* dataset, positioning it as a reliable option for community detection in larger CNs. While GASN and GAEAS show comparable performance, GCNFCM remains noteworthy. On the *Citeseer* dataset, GCNFCM exhibits proficiency with an NMI score of 0.5315 within medium-scale networks. On the *PubMed* dataset, despite challenges posed by medium-scale CNs, GCNFCM achieves an NMI score of

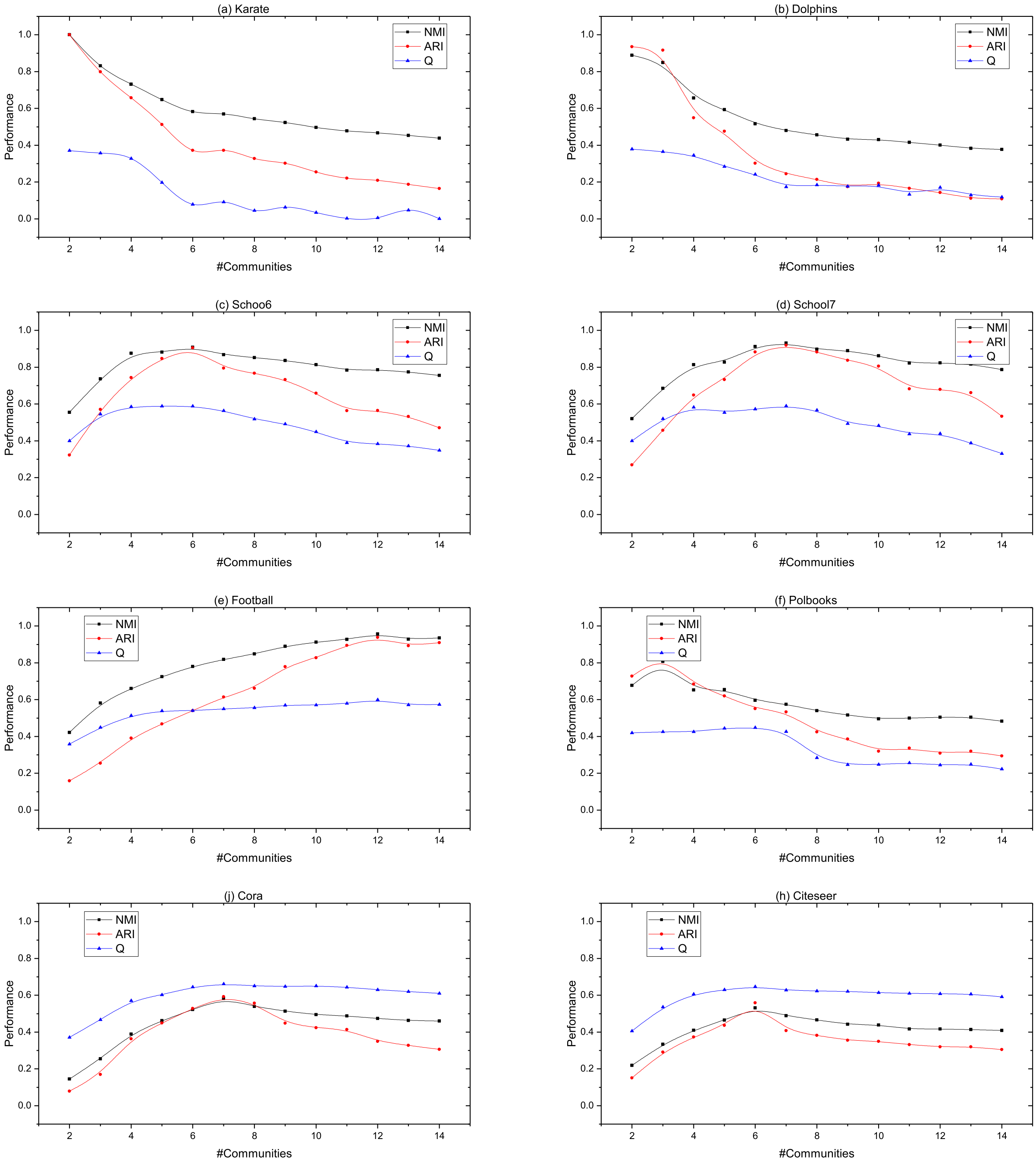


FIGURE 3. Performance of GCNFCM across diverse community numbers using NMI, ARI, and Q metrics.

0.3937 on the *PubMed* dataset. This evidence highlights its potential as a noteworthy option for community detection in diverse network contexts.

Table 7 presents the ARI scores to evaluate the performance of GCNFCM alongside various methods (e.g., Spectral [49], DeepWalk [37], etc.) on medium-scale CNs. GCNFCM demonstrates robustness across datasets and harvests notable ARI scores, such as 0.6803 on *Polblogs*, 0.5923 on *Cora*, 0.5586 on *Citeseer*, and 0.4063 on *PubMed*. These findings showcase the effectiveness of GCNFCM in identifying community structures within larger CNs. This makes GCNFCM as a promising method for community detection in diverse medium-scale CN contexts.

NMI and ARI scores in Tables 6 and 7 affirm the efficacy and reliability of GCNFCM in capturing community structures. This competitive performance highlights its potential as a valuable tool for community detection in various research fields such as recommendation systems, anomaly detection, etc.

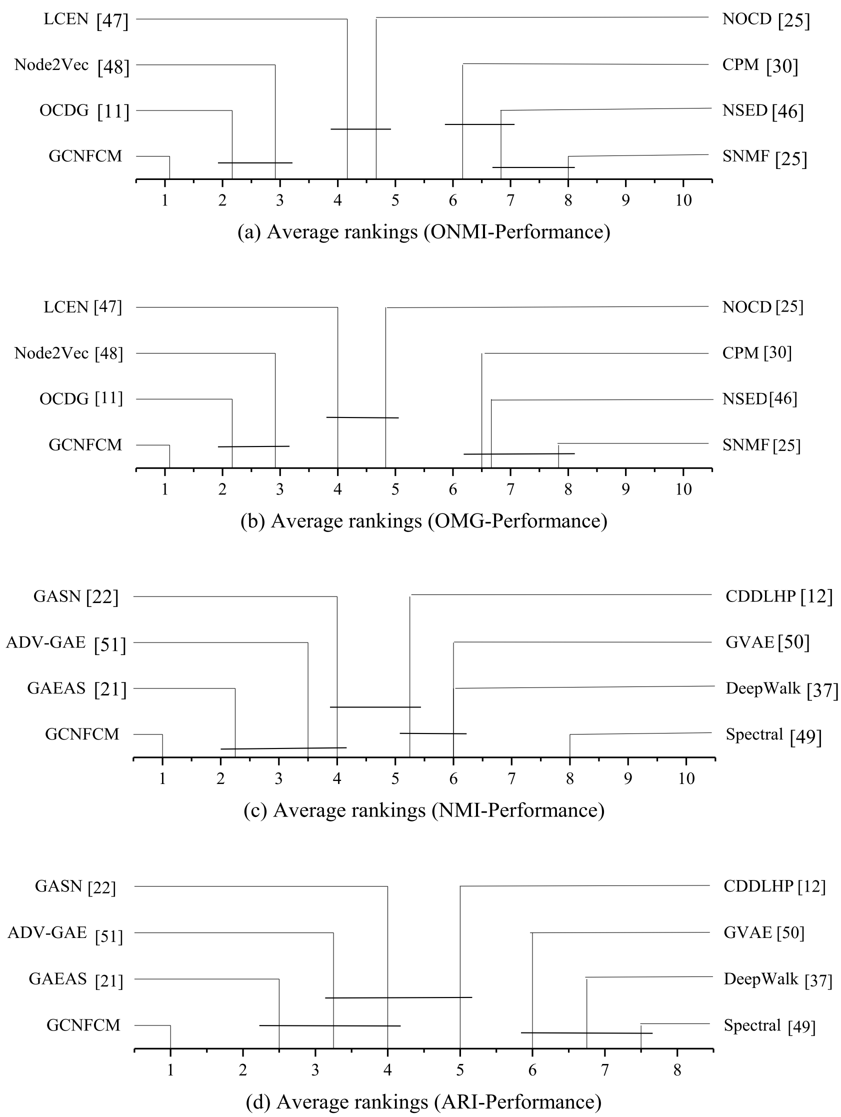


FIGURE 4. Results of GCNFCM from the Nemenyi tests.

2) PERFORMANCE OF GCNFCM WITH DIFFERENT COMMUNITIES

GCNFCM has the ability to identify community structures, both overlapping and non-overlapping without relying on prior knowledge about the number of communities. $\tilde{Q}$ and Q modularity metrics guide GCNFCM to detect high-quality communities that have denser intra-relations nodes and sparser inter-relations nodes.

Figure 2 visually represents the performance of GCNFCM across 8 CNs, employing metrics such as ONMI, OMG, and $\tilde{Q}$. The evaluation encompasses a range of community numbers 2 - 14 communities, revealing a compelling correlation between $\tilde{Q}$ (unconstrained by real labels) and ONMI/OMG (dependent on real labels and communities). High $\tilde{Q}$ values signify a community structure aligning with real-world principles, where intra-relations nodes are denser and inter-relations nodes are sparser.

Examining specific datasets in Fig. 2 reinforces the efficacy GCNFCM. For example, in Karate and Dolphins datasets, GCNFCM achieves high performance ($\tilde{Q}$, ONMI, OMG) with 2 communities, which corresponds the actual community structure. Similarly, across School6, School7, Football, Polbooks, Cora, and Citeseer datasets, GCNFCM attains optimal performance with varying community numbers. This emphasizes the adaptability to diverse network contexts. In addition, the obtained results underscore GCNFCM as a potent and versatile approach for community detection in CNs, offering valuable insights without relying on explicit information about the ground-truth communities.

Figure 3 focuses on the performance of GCNFCM in non-overlapping community detection on eight CNs using NMI, ARI, and Q metrics alongside a range of community sizes (2 to 14). Similar to Fig 2, GCNFCM shines in

Karate and Dolphins with high performance (Q, NMI, ARI) at two communities, reflecting the actual structure. It then adapts to different community sizes in *School6, School7, Football*, *Polbooks, Cora*, and *Citesee* and achieves optimal performance across diverse contexts. These findings further establish GCNFCM as a potent and versatile approach for community detection, shedding light on networks even without relying on ground-truth community information

3) PERFORMANCE ANALYSIS USING NON-PARAMETRIC STATISTICAL TESTS

In this section, we employed two non-parametric statistical tests, namely the Friedman and Nemenyi post hoc tests [52], to compare the performance of the proposed GCNFCM method with SNMF, NSED, LCEN, UOCS, Node2Vec, NOCD, OCDG, Spectral, DeepWalk, GVAE, ADV-GAE, GASN, GAEAS, and CDDLHP methods. Specifically, we evaluated the ONMI, OMG, NMI, and ARI performances.

The Friedman test statistic was applied based on the average ranked performances of overlapping community detection techniques on each CN, and it was performed at an α-level of 0.05. The obtained p-values for ONMI and OMG were 11.095e-6 and 1.332e-6, respectively. For community detection without overlapping, the p-values for NMI and ARI were 1.178e-3 and 1.648e-3, respectively.

Consequently, the null hypothesis claiming no difference between the GCNFCM method and the comparison methods was rejected. Following the rejection of the null hypothesis, a post-hoc analysis was conducted using the Nemenyi test, given the comparison of multiple methods across multiple datasets.

Figure 4 illustrates the results of the Nemenyi post-hoc analysis, indicating that GCNFCM ranks among the top four methods in terms of ONMI, OMG, NMI, and ARI performances. Methods that are not significantly different in terms of these performance metrics from a statistical viewpoint are connected in Figs. 4(a) to 4(d).

Figure 4(a) reveals that GCNFCM exhibits the best ONMI performance, while SNMF performs the worst. The ONMI performance of GCNFCM is found to be significantly different from all comparison methods. Similarly, Figure 4(b) indicates that the OMG performance of GCNFCM significantly differs from all comparison methods. Moreover, GCNFCM demonstrates significant differences from baseline methods in terms of NMI and ARI, as shown in Figures 4(c) and 4(d), respectively.

The superior performance of GCNFCM can be attributed to the use of GCN in generating the node embeddings based on both node topology and attributes. Additionally, the method employs a two-step process: first, GCN extracts node embeddings, and then a FCM algorithm is applied for overlapping community detection. Finally, the quality index ($\tilde{Q}$) is utilized as a metric to guide FCM in identifying the optimal overlapping community structure.

V. CONCLUSION

This paper introduces a robust method to address the challenges of identifying overlapping communities within CNs, called GCNFCM. GCNFCM tackles the challenge of identifying overlapping communities in CNs by leveraging the strengths of graph auto-encoders and FCM clustering. FCM is guided by the modularity Q algorithm for accurate overlapping community identification. Modularity Q algorithm enables GCNFCM to eliminate the need for prior knowledge about the number of communities. Additionally, it utilizes a dual-decoder architecture for efficient node embedding extraction, considering both node topology and attributes within CNs. This dual-decoder incorporates both inner product and GCN decoders. Extensive experiments on diverse real-world CNs validate the effectiveness of GCNFCM. Experimental results show that GCNFCM can identify overlapping communities effectively and efficiently. Additionally, the results demonstrate that GCNFCM outperforms several state-of-the-art methods in terms of producing cohesive overlapping communities and accurately identifying ground-truth communities. These findings establish GCNFCM as a promising solution for accurate overlapping community detection in CNs.

Although GCNFCM has shown its abilities in identifying overlapping community detection effectively, GCNFCM is limited by scalability issues and sensitivity to hyperparameter selection. Future work will focus on addressing these limitations. Potential extensions include: (1) Developing scalable algorithms for large-scale CNs, this could involve leveraging parallel processing techniques to enhance efficiency. (2) Implementing hyperparameter auto-tuning, which would automate hyperparameter selection to improve generalizability and user-friendliness. (3) Enhancing interpretability, unveiling the rationale behind community detection will further advance the understanding and application of GCNFCM.

ACKNOWLEDGMENT

The authors express their gratitude to the Centre for Research and Innovation Management (CRIM), Universiti Teknikal Malaysia Melaka (UTeM), for their valuable support in this research.

REFERENCES

[1] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, Q. Z. Sheng, and P. S. Yu, "A comprehensive survey on community detection with deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4682–4702, Apr. 2024, doi: 10.1109/TNNLS.2021.3137396.

[2] B. A. Attea, A. D. Abbood, A. A. Hasan, C. Pizzuti, M. Al-Ani, S. Özdemir, and R. D. Al-Dabbagh, "A review of heuristics and metaheuristics for community detection in complex networks: Current usage, emerging development and future directions," *Swarm Evol. Comput.*, vol. 63, Jun. 2021, Art. no. 100885, doi: 10.1016/j.swevo.2021.100885.

[3] M. N. Al-Andoli, S. C. Tan, W. P. Cheah, and S. Y. Tan, "A review on community detection in large complex networks from conventional to deep learning methods: A call for the use of parallel meta-heuristic algorithms," *IEEE Access*, vol. 9, pp. 96501–96527, 2021, doi: 10.1109/ACCESS.2021.3095335.

[4] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113, doi: 10.1103/physreve.69.026113.
[5] M. Azaouzi, D. Rhouma, and L. Ben Romdhane, "Community detection in large-scale social networks: State-of-the-art and future directions," *Social Netw. Anal. Mining*, vol. 9, no. 1, p. 23, Dec. 2019, doi: 10.1007/s13278-019-0566-x.
[6] G. Li, D. Zhang, K. Xie, T. Huang, and Y. Li, "A GPU based fast community detection implementation for social network," in *Proc. Int. Conf. Algorithms Architectures Parallel Process.* (Lecture Notes in Computer Science), vol. 9528. Cham, Switzerland: Springer, 2015, pp. 688–701, doi: 10.1007/978-3-319-27119-4_48.
[7] J. Zhang, L. Pan, Q.-L. Han, C. Chen, S. Wen, and Y. Xiang, "Deep learning based attack detection for cyber-physical system cybersecurity: A survey," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 3, pp. 377–391, Mar. 2022, doi: 10.1109/JAS.2021.1004261.
[8] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang, "Friendbook: A semantic-based friend recommendation system for social networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 3, pp. 538–551, Mar. 2015, doi: 10.1109/TMC.2014.2322373.
[9] M. Dhilber and S. D. Bhavani, "Community detection in social networks using deep learning," in *Distributed Computing and Internet Technology* (Lecture Notes in Computer Science), vol. 11969, D. Hung and M. D'Souza, Eds. Cham, Switzerland: Springer, 2020, pp. 241–250, doi: 10.1007/978-3-030-36987-3_15.
[10] J. Zu, G. Hu, J. Yan, and S. Tang, "A community detection based approach for service function chain online placement in data center network," *Comput. Commun.*, vol. 169, pp. 168–178, Mar. 2021, doi: 10.1016/j.comcom.2021.01.014.
[11] S. Yuan, H. Zeng, Z. Zuo, and C. Wang, "Overlapping community detection on complex networks with graph convolutional networks," *Comput. Commun.*, vol. 199, pp. 62–71, Feb. 2023, doi: 10.1016/j.comcom.2022.12.008.
[12] M. N. Al-Andoli, S. C. Tan, and W. P. Cheah, "Distributed parallel deep learning with a hybrid backpropagation-particle swarm optimization for community detection in large complex networks," *Inf. Sci.*, vol. 600, pp. 94–117, Jul. 2022, doi: 10.1016/j.ins.2022.03.053.
[13] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, "Autoencoders and their applications in machine learning: A survey," *Artif. Intell. Rev.*, vol. 57, no. 2, p. 28, Feb. 2024, doi: 10.1007/s10462-023-10662-6.
[14] M. Al-Andoli, W. P. Cheah, and S. C. Tan, "Deep autoencoder-based community detection in complex networks with particle swarm optimization and continuation algorithms," *J. Intell. Fuzzy Syst.*, vol. 40, no. 3, pp. 4517–4533, Mar. 2021, doi: 10.3233/jifs-201342.
[15] M. Al-Andoli, S. C. Tan, and W. P. Cheah, "Parallel stacked autoencoder with particle swarm optimization for community detection in complex networks," *Int. J. Speech Technol.*, vol. 52, no. 3, pp. 3366–3386, Feb. 2022, doi: 10.1007/s10489-021-02589-8.
[16] K. Berahmand, M. Mohammadi, R. Sheikhpour, Y. Li, and Y. Xu, "WSNMF: Weighted symmetric nonnegative matrix factorization for attributed graph clustering," *Neurocomputing*, vol. 566, Jan. 2024, Art. no. 127041, doi: 10.1016/j.neucom.2023.127041.
[17] K. Berahmand, Y. Li, and Y. Xu, "A deep semi-supervised community detection based on point-wise mutual information," *IEEE Trans. Computat. Social Syst*, vol. 10, pp. 1–13, Nov. 2023, doi: 10.1109/TCSS.2023.3327810.
[18] A. De Santo, A. Galli, V. Moscato, and G. Sperlì, "A deep learning approach for semi-supervised community detection in online social networks," *Knowl.-Based Syst.*, vol. 229, Oct. 2021, Art. no. 107345, doi: 10.1016/j.knosys.2021.107345.
[19] D. Jin, Z. Yu, P. Jiao, S. Pan, D. He, J. Wu, P. S. Yu, and W. Zhang, "A survey of community detection approaches: From statistical modeling to deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1149–1170, Feb. 2023, doi: 10.1109/TKDE.2021.3104155.
[20] J. Sun, W. Zheng, Q. Zhang, and Z. Xu, "Graph neural network encoding for community detection in attribute networks," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7791–7804, Aug. 2022, doi: 10.1109/TCYB.2021.3051021.
[21] M. Lin, K. Wen, X. Zhu, H. Zhao, and X. Sun, "Graph autoencoder with preserving node attribute similarity," *Entropy*, vol. 25, no. 4, p. 567, Mar. 2023, doi: 10.3390/e25040567.
[22] J. Wang, J. Liang, K. Yao, J. Liang, and D. Wang, "Graph convolutional autoencoders with co-learning of graph structure and node attributes," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108215, doi: 10.1016/j.patcog.2021.108215.
[23] M. Al-Andoli, W. P. Cheah, and S. C. Tan, "Deep learning-based community detection in complex networks with network partitioning and reduction of trainable parameters," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 2, pp. 2527–2545, Feb. 2021, doi: 10.1007/s12652-020-02389-x.
[24] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Comput. Surv.*, vol. 45, no. 4, pp. 1–35, Aug. 2013, doi: 10.1145/2501654.2501657.
[25] O. Shchur and S. Günnemann, "Overlapping community detection with graph neural networks," 2019, *arXiv:1909.12201*.
[26] M. Alandoli, M. Shehab, M. Al-Ayyoub, Y. Jararweh, and M. Al-Smadi, "Using GPUs to speed-up FCM-based community detection in social networks," in *Proc. 7th Int. Conf. Comput. Sci. Inf. Technol. (CSIT)*, Jul. 2016, pp. 1–6, doi: 10.1109/CSIT.2016.7549467.
[27] S. Zhang, R.-S. Wang, and X.-S. Zhang, "Identification of overlapping community structure in complex networks using fuzzy -means clustering," *Phys. A, Stat. Mech. Appl.*, vol. 374, no. 1, pp. 483–490, Jan. 2007, doi: 10.1016/j.physa.2006.07.023.
[28] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, Sep. 2007, Art. no. 036106, doi: 10.1103/physreve.76.036106.
[29] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, no. 10, Oct. 2010, Art. no. 103018, doi: 10.1088/1367-2630/12/10/103018.
[30] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005, doi: 10.1038/nature03607.
[31] M. Al-Ayyoub, M. Al-andoli, Y. Jararweh, M. Smadi, and B. Gupta, "Improving fuzzy C-mean-based community detection in social networks using dynamic parallelism," *Comput. Electr. Eng.*, vol. 74, pp. 533–546, Mar. 2019, doi: 10.1016/j.compeleceng.2018.01.003.
[32] L. Zhou, K. Lü, P. Yang, L. Wang, and B. Kong, "An approach for overlapping and hierarchical community detection in social networks based on coalition formation game theory," *Expert Syst. Appl.*, vol. 42, no. 24, pp. 9634–9646, Dec. 2015, doi: 10.1016/j.eswa.2015.07.023.
[33] K. E. Avrachenkov, A. Y. Kondratev, and V. V. Mazalov, "Cooperative game theory approaches for network partitioning," in *Computing and Combinatorics* (Lecture Notes in Computer Science), vol. 10392, Y. Cao and J. Chen, Eds. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-62389-4_49.
[34] D. Huang, J. Song, and Y. He, "Community detection algorithm for social network based on node intimacy and graph embedding model," *Eng. Appl. Artif. Intell.*, vol. 132, Jun. 2024, Art. no. 107947, doi: 10.1016/j.engappai.2024.107947.
[35] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 257–266, doi: 10.1145/3292500.3330925.
[36] B. Cai, Y. Wang, L. Zeng, Y. Hu, and H. Li, "Edge classification based on convolutional neural networks for community detection in complex network," *Phys. A, Stat. Mech. Appl.*, vol. 556, Oct. 2020, Art. no. 124826, doi: 10.1016/j.physa.2020.124826.
[37] 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*, Aug. 2014, pp. 701–710, doi: 10.1145/2623330.2623732.
[38] A. Tsitsulin, J. Palowitch, B. Perozzi, and E. Müller, "Graph clustering with graph neural networks," *J. Mach. Learn. Res.*, vol. 24, no. 127, pp. 1–21, 2023.
[39] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 1, pp. 36–41, Jan. 2007, doi: 10.1073/pnas.0605965104.
[40] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011, doi: 10.1016/j.acha.2010.04.005.
[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[42] M. Madiajagan and S. S. Raj, ''Parallel computing, graphics processing unit (GPU) and new hardware for deep learning in computational intelligence research,'' in *Deep Learning and Parallel Computing Environment for Bioengineering Systems*. Amsterdam, The Netherlands: Elsevier, 2019, pp. 1–15, 2019, doi: 10.1016/B978-0-12-816718-2.00008-7.

[43] A. F. McDaid, D. Greene, and N. Hurley, ''Normalized mutual information to evaluate overlapping community finding algorithms,'' 2011, *arXiv:1110.2515*.

[44] L. M. Collins and C. W. Dent, ''Omega: A general formulation of the Rand index of cluster recovery suitable for non-disjoint solutions,'' *Multivariate Behav. Res.*, vol. 23, no. 2, pp. 231–242, Apr. 1988, doi: 10.1207/s15327906mbr2302_6.

[45] D. Kuang, C. Ding, and H. Park, ''Symmetric nonnegative matrix factorization for graph clustering,'' in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2012, pp. 106–117, doi: 10.1137/1.9781611972825.10.

[46] B.-J. Sun, H. Shen, J. Gao, W. Ouyang, and X. Cheng, ''A non-negative symmetric encoder–decoder approach for community detection,'' in *Proc. ACM Conf. Inf. Knowl. Manag.*, 2017, pp. 597–606, doi: 10.1145/3132847.3132902.

[47] 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. Manag.*, Nov. 2017, pp. 377–386, doi: 10.1145/3132847.3132925.

[48] A. Grover and J. Leskovec, ''node2vec: Scalable feature learning for networks,'' in *Proc. 22nd ACM SIGKDD KDD*, 2016, pp. 855–864, doi: 10.1145/2939672.2939754.

[49] L. Tang and H. Liu, ''Leveraging social media networks for classification,'' *Data Mining Knowl. Discovery*, vol. 23, no. 3, pp. 447–478, Nov. 2011, doi: 10.1007/s10618-010-0210-x.

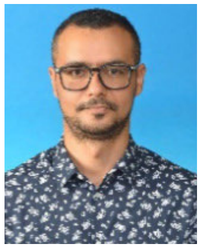
[50] T. N. Kipf and M. Welling, ''Variational graph auto-encoders,'' 2016, *arXiv:1611.07308*.

[51] T. Huang, Y. Pei, V. Menkovski, and M. Pechenizkiy, ''On generalization of graph autoencoders with adversarial training,'' in *Machine Learning and Knowledge Discovery in Databases*. Bilbao, Spain: Springer, 2021, pp. 367–382, doi: 10.1007/978-3-030-86520-7_23.

[52] J. Demšar, ''Statistical comparisons of classifiers over multiple data sets,'' *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.

MOHAMMED NASSER AL-ANDOLI received the B.Sc. degree in computer information systems from Mutah University, Jordan, in 2011, the M.Sc. degree in computer science from Jordan University of Science and Technology, Jordan, in 2016, and the Ph.D. degree in information technology from Multimedia University, Malaysia, in 2022. He is currently a Senior Lecturer with the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM). His research interests include malware analysis, complex network analysis, interactive media, machine learning, high-performance computing, deep learning, and parallel computing.

IRIANTO received the master's degree, in 2009, and the Ph.D. degree in applied and computational statistics from Universiti Putra Malaysia, in April 2021. After his master's degree, he started to teach at several universities in Indonesia. From 2012 to 2021, he was a Lecturer with the Faculty of Engineering Technology, Universiti Teknikal Malaysia Melaka. In 2021, he was an Assistant Professor with American International University, Kuwait. Since 2022, he has been an Assistant professor with the Rabdan Academy and Zayed Military University, United Arab Emirates. His research interests include applied mathematics and multivariate statistical process control.



JAMIL ABEDALRAHIM ALSAYAYDEH (Member, IEEE) received the degree in computer engineering and the M.S. degree in computer systems and networks from Zaporizhzhia National Technical University, Ukraine, in 2009 and 2010, respectively, and the Ph.D. degree in engineering sciences (automation of control processes) from National Mining University, Ukraine, in 2014. Since 2015, he has been a Senior Lecturer with the Department of Electronics and Computer Engineering Technology, Faculty of Electrical and Electronic Engineering Technology, Universiti Teknikal Malaysia Melaka (UTeM). He is also a Research Member with the Center for Advanced Computing Technology. He has supervised bachelor's and master's students. His research interests include formal methods, simulation, the Internet of Things, computing technology, artificial intelligence, and machine learning: computer architecture, algorithms, and applications. He is the author/coauthor of more than 43 research publications in his research, which were cited by more than 95 documents. He actively publishes research articles and receives grants from the government and private sectors, universities, and international collaboration. He is a member of the Board of Engineers Malaysia (BEM). He is also a reviewing member of various reputed journals.

IBRAHIM M. ALWAYLE received the B.Sc. degree in mathematics from San'aa University, Yemen, in 1999, and the M.Sc. and Ph.D. degrees in computer science from Al Mansurah University, Egypt, in 2004 and 2008, respectively. He is currently an Assistant Professor with the Department of Computer Science, Najran University, Saudi Arabia. He is permanently an Assistant Professor with Amran University, Yemen. His research interests include pattern recognition, image processing, and AI.

CHE KU NURAINI CHE KU MOHD received the degree in computer science (interactive media), the Master of Science degree in information and communication technology, and the Ph.D. degree in information and communication technology from Universiti Teknikal Malaysia Melaka, in 2009, 2011, and 2016, respectively. She is currently a Senior Lecturer with the Department of Media Interactive, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka. She has experience in teaching for several private colleges and universities primarily on multimedia interactive, computer applications, game-based learning, and assistive technology for special needs. Her professional competence in this field through a standing collaboration with a research group during her postdoctoral years (2018–2021), which has delivered presentations, contributed to research proposals and journal articles, and supervised students.

FAHD ABUHOUREYAH received the B.Sc. degree in electrical and electronics engineering from the University of Bradford, U.K., in 2013, and the master's degree in control system engineering from Universiti Teknikal Malaysia Melaka (UTeM), in 2020, where he is currently pursuing the Ph.D. degree in signal processing with AI. His current research interests include computer systems and their intersection with innovation studies, science and technology, and electronic devices.