Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.067783

REVIEW

A Comprehensive Review of Dynamic Community Detection: Taxonomy, Challenges, and Future Directions

Hiba Sameer Saeed[#] **and Amenah Dahim Abbood**[*,#]

Department of Computer Science, College of Science, University of Baghdad, Al-Jadriya Area, Baghdad, 10070, Iraq

*Corresponding Author: Amenah Dahim Abbood. Email: amenah.abbood@sc.uobaghdad.edu.iq

[#]These authors contributed equally to this work

Received: 12 May 2025; Accepted: 30 July 2025; Published: 23 October 2025

ABSTRACT: In recent years, the evolution of the community structure in social networks has gained significant attention. Due to the rapid and continuous evolution of real-world networks over time. This makes the process of identifying communities and tracking their topology changes challenging. To tackle these challenges, it is necessary to find efficient methodologies for analyzing the behavior patterns of dynamic communities. Several previous reviews have introduced algorithms and models for community detection. However, these methods have not been very accurate in identifying communities. Moreover, none of the reviewed papers made an apparent effort to link algorithms that can accurately detect dynamic communities. This review aims to present a taxonomy that shows several algorithms and methodologies for detecting dynamic communities. These algorithms are divided into four categories (heuristic- and modularity-based, metaheuristic, deep learning, and hybrid deep learning). It encompasses the past five years and examines the advantages and disadvantages of conventional and recent methods. Currently, many efforts are utilizing deep learning to improve dynamic networks; however, the instability of the network during the training phase affects the model's accuracy. However, this direction remains unexplored. This study presents a review that aims to tackle this issue. We discuss a research path that explores the integration of deep learning with heuristic, metaheuristic, and hybrid metaheuristic algorithms to facilitate the identification of communities in dynamic networks. This investigation examines how this mixture surpasses the constraints of singular methodologies, resulting in enhanced detection outcomes and enabling researchers to select the most suitable algorithms for their future research.

KEYWORDS: Dynamic structure; social networks; deep learning; hybrid deep learning

1 Introduction

In the context of community detection, a complex network is defined as a network with specific structural characteristics that make the identification process challenging. Some of its types include social networks, biological networks, technological networks, etc. The use of complex networks has become critical in the analysis of a large amount of unstructured information within complex systems. In recent years, the goal of detecting the community structure of networks has garnered significant attention from many fields [1]. Community detection is the task of identifying the structures of clusters within a network to capture the nature of these clusters (i.e., nodes and edges), where nodes are more densely connected within clusters than to nodes in other clusters. Many researchers have proposed community detection approaches for static networks where the number of nodes in a particular network remains constant [2–5]. In the real world, network data expands dynamically over time. The total number of nodes and edges within the network changes, resulting in modifications to the community structure. Dynamic networks can be shown as a sequence of snapshots at various time steps. However, temporal changes in structure result in the emergence and evolution of communities, such as Facebook, Twitter, and Linked-In, which have evolved quickly over time [6]. Recently, various algorithms have been developed to identify dynamic communities, which have been increasing in number. Identifying community structures in dynamic networks can provide insight into how communities evolve over time, leading researchers to devote more attention to analyzing this type of network. Additional details about the dynamic network can be obtained in [7–9]. The topology of a dynamic network can suffer significant structural changes as nodes and edges join or leave their communities over time. This is clearly outlined in Fig. 1.

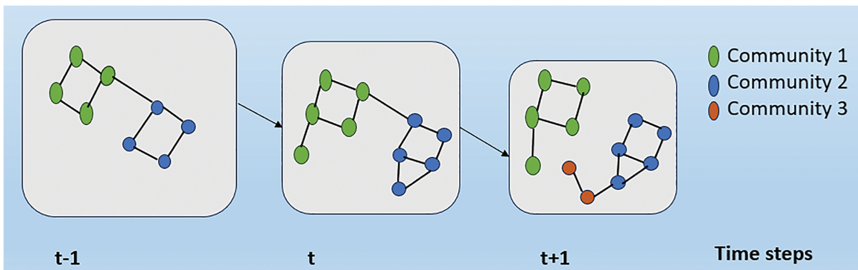

Figure 1: Community structures in dynamic networks over three time steps, time step $(t-1)$ contains two communities, time step (t) also contains two communities, and time step $(t+1)$ contains three communities

A social network is a digital platform that enables connections and interactions between individuals. In terms of social network analysis, many studies have been developed to capture the evolution of community structures [10–12]. A first, official identification of the changes affecting communities was introduced in [13], where six of them were listed (birth, death, growth, contraction, merge, and split). Occasionally, the seventh operation (continue) is added to them. In [14], an eighth operation (resurgence) was presented. These operations are shown in Fig. 2. The three algorithm categories (heuristic, metaheuristic, and hybrid) outlined in this paper are state -of-the-art approaches, which pose numerous challenges. Heuristic algorithms are prone to getting stuck in local minima. Furthermore, these algorithms become inefficient when addressing complex or multilayer networks. Metaheuristic multi-objective optimization algorithms often exhibit longer execution times. The reason is that the topology of dynamic networks changes all the time. In addition, the computational complexity of metaheuristic algorithms arises from their high demand for processing resources. Hybrid metaheuristic algorithms face implementation complications. Although there have been significant improvements made in deep learning for dynamic community detection in recent years, there are several difficulties that require improved solutions and challenges that remain unaddressed [15]. This review concentrates on conventional dynamic community detection algorithms to identify new trends in deep learning for dynamic community detection. Our investigation is divided into five key components:

1- We review dynamic community detection algorithms (heuristic, metaheuristic, and hybrid metaheuristic), explaining their importance and their utilization in detecting dynamic communities.

2- In this paper, we describe the two dynamic approaches that capture the community structure of evolving networks, depending on their nature.

3- We introduce different deep learning methods that operate on temporal graph networks. These methods improve the interactions between nodes and enhance their features in dynamic networks.

4- We discuss the latest advancements in the use of hybrid techniques, combining heuristic algorithms with deep learning categories.

5- We discuss the new direction that integrates deep learning and metaheuristics, as well as hybrid metaheuristic algorithms. We emphasize their importance in capturing evolving networks and propose promising directions for future research.

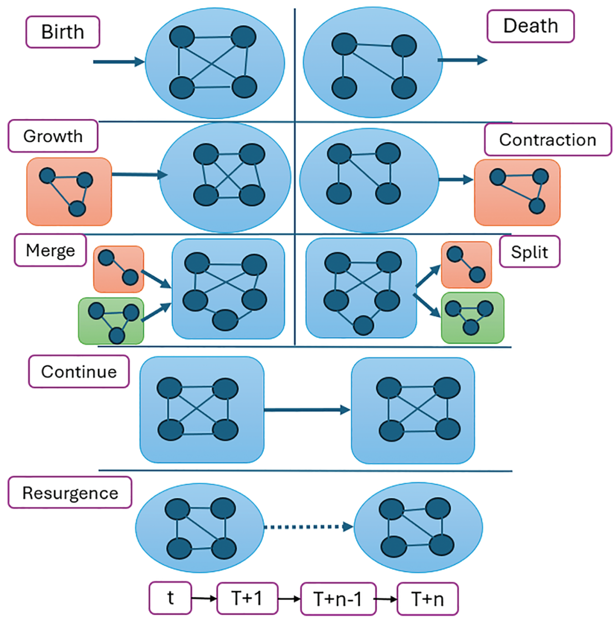

Figure 2: Community structures in dynamic networks [14]

The rest of this paper is organized as follows: Section 2 introduces the preliminary concepts of dynamic community detection algorithms. Section 3 describes the evaluation measures used to evaluate the algorithms and methods employed for detecting dynamic communities. Section 4 shows a taxonomy of approaches for dynamic community detection. Section 5 presents a taxonomy for community detection algorithms in dynamic networks. Section 6 presents an open research area that discusses the hybridization of (heuristic, metaheuristic, and hybrid metaheuristic) with deep learning. Finally, Section 7 summarizes the conclusion of this paper.

2 Preliminary Concepts

Dynamic community detection is the process of identifying the structure of the community in a dynamically evolving network over time. Dynamic networks typically consist of nodes and edges that change over time. For example, it is feasible to add or remove some nodes and edges from the network. A dynamic network is often represented as a sequence of $G = [G^1, G^2, \ldots, G^t, \ldots, G^T]$, where each graph is considered undirected and unweighted for simplicity. These networks are observed in T as a sequence of time steps, where $t \in$ [1, 2,. . .,T] explains the t-th time step. The result of a network's final partition with T time steps is denoted as $P = P_1, P_2, \ldots, P_T$. Each $P_i(1 < i \leq T)$ represents the partition result of a network G_i at time i, and is formed of m communities, $Pi = C_1, C_2, \ldots, C_m$. Where G^1 describes the initial network, and G^t is a snapshot of nodes and their connections at a specific time step t, indicates to $G^t = [V^t, E^t]$, where $V^t = [v_1^t, v_2^t, \ldots, v_n^t]$ refers to a collection of network nodes that, for the sake of simplicity, we will assume to be a fixed number (even if not all of them may have been observed). We will assign a zero to any nodes that have not been observed in the community, as well as $E^t = [(v_i^t, v_j^t) | v_i^t \in V^t, and\ i \neq j]$ where E^t describes a set of connections in G^t linking nodes at time step t. The number of connections in the graph at time t is shown by Lt; $Lt = |E(G^t)|$. Typically, G^t topological knowledge can be expressed as an N * N adjacency matrix A^t, so that $A^t_{ij} = 1$ if there is a connection between v_i and v_j, while $A^t_{ij} = 0$ otherwise. $\sum_{J=1}^{N} A^t_{ij} = 0$ for the unnoticed node i at any t, and we assumed that the community of this node is 0. The graph G^t can be represented by partitioning it into K_t communities $C_i^{t^{kt}} i = 1$. So, every node is to be a part of exactly one community.

3 Evaluation Measures

Several measures are available to assess the degree of similarity between the network-detected partition and the ground truth partition. These measurements are presented in this section.

A- **Normalized Mutual Information score (NMI)** introduced by [16], used when the ground truth partition for the network is known. It is a measure employed to examine the performance of community detection algorithms. The NMI was calculated to determine how closely the discovered partition matches the ground truth partition. When two partitions are more similar to each other, the NMI value gradually increases and vice versa.

$$NMI(A, B) = \frac{-2\sum_{i=1}^{CA}\sum_{j=1}^{CB} C_{ij} \log(C_{ij}N / C_{i.}C_{.j})}{\sum_{i=1}^{CA} C_{i.} \log(C_{i.}/N) + \sum_{j=1}^{CB} C_{.j} \log(C_{.j}/N)} \tag{1}$$

Here, the number of communities in partitions A and B is denoted by CA and CB. The elements of C in the row i are added together, and the elements of C in the column j are added together, and N is the number of nodes in the network. When two partitions A and B are identical, NMI (A, B) = 1, NMI(A, B) = 0 if A and B are different; otherwise, NMI(A, B) ∈ (0, 1).

B- **Modularity (Q)** was introduced by [17,18]. Modularity is frequently used as an internal density metric to assess network partitions when the ground truth partition is unknown. When the modularity values are near 1, they show partitions with a strong community structure, whereas values close to 0 show that the partition does not correspond to a community structure.

$$Q = \sum_{s=1}^{k} \left[\frac{L_s}{m} - \left(\frac{d_s}{2m}\right)^2 \right] \tag{2}$$

Here, L_s represents the number of connections within each cluster, d_s represents the number of connections in and out of each community, and m represents the number of connections in the network. Fortunato and Barthelemy [19] show that ideal modularity partitions are not necessarily consistent with the characteristics of strong communities. The "resolution limit" issue could make it challenging to find small communities within large networks and vice versa.

C- **Error rate** Calculates the difference between the true communities and the detected communities [20,21].

$$error(Z, G) = \|ZZ^T - GG^T\| \tag{3}$$

To compute it, we build an $n * k$ indicator matrix Z, where n represents the number of nodes and k represents the number of communities, and a similar matrix G to illustrate the true community. The accuracy of the algorithm increases as the error rate decreases.

D- **F-score** represents the weighted harmonic mean of precision and recall. Where weights are equal for both precision and recall [22].

$$F - score = 2 * \frac{(precision * recall)}{(precision + recall)} \tag{4}$$

E- **Coverage** of clustering *c* is the percentage of intra cluster links to the total number of links [22,23].

$$Coverage = \frac{|eintra|}{|e|} \tag{5}$$

4 A Taxonomy of Approaches for Dynamic Community Detection

Researchers employ various approaches based on network features and algorithm objective functions to detect communities and track changes in their structure. Therefore, we suggest a taxonomy of the key approaches to community detection in dynamic networks, as shown in Fig. 3, which categorizes the currently used approaches into two main groups: Snapshot-based (Independent) and continuous community detection.

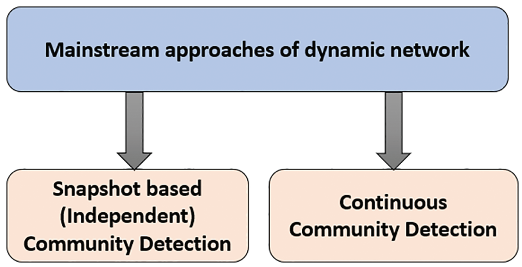

Figure 3: The proposed taxonomy of approaches for dynamic community detection

4.1 Snapshot-Based (Independent) Community Detection

Strategies of this type separate the network into a series of snapshots. Starting by utilizing a static network algorithm to discover communities on each snapshot separately. Next, a comparison is made between the snapshot in the current time step and the previous one. However, the high complexity and instability of conventional community detection strategies can pose challenges to independently detecting communities [24].

4.2 Continuous Community Detection

Strategies of this type also separate the network into a series of snapshots. The strategies used here are based on previously detected communities to identify communities in the new snapshot. However, the inability to directly use conventional community detection strategies may compromise the approaches used here [24].

5 A Taxonomy for Community Detection Algorithms in Dynamic Networks

Community detection (CD) algorithms in dynamic networks

The dynamic community detection algorithms aim to accurately capture the evolving community structure of a network over time. Fig. 4 illustrates the categorization of community detection algorithms. Through our study, we have divided the algorithms for dynamic networks into three distinct classes.

The following taxonomy presents a review of the literature covering research on all classes of dynamic algorithms covered between 2004 and 2025, as shown in Figs. 5 and 6. We annotate this period to provide the reader with a complete understanding of the development and current status of the dynamic community detection field.

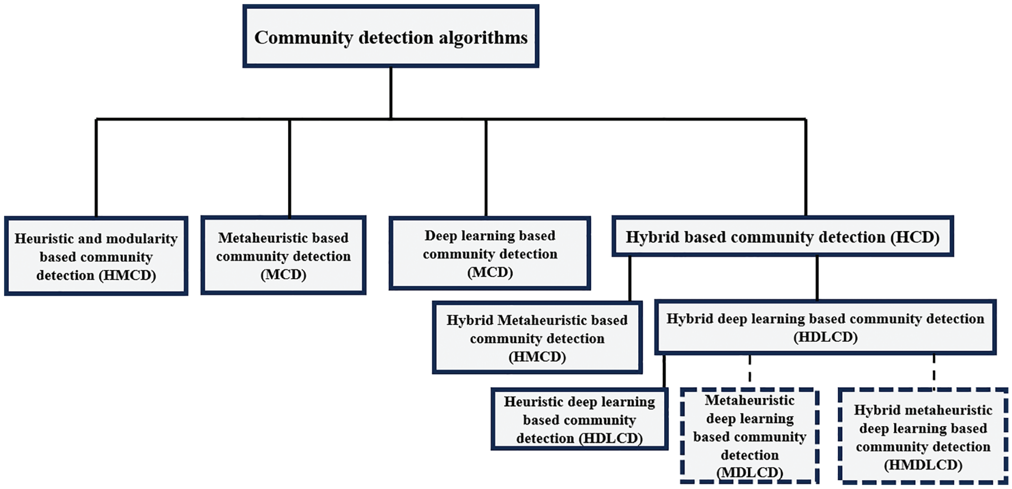

Figure 4: The taxonomy for community detection algorithms in dynamic networks

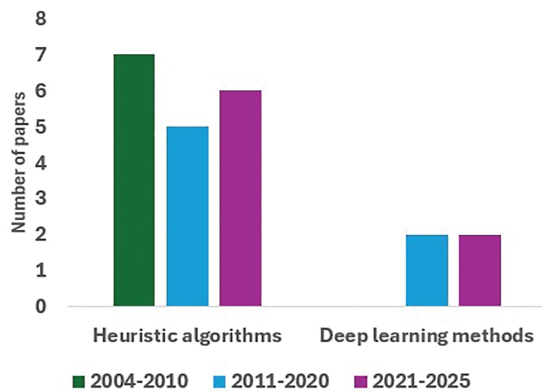

Figure 5: The distribution of published papers from 2004 to 2025. This distribution includes: heuristic algorithms and deep learning methods

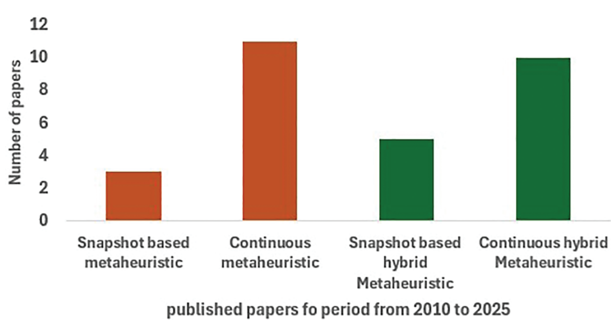

Figure 6: The distribution of published papers from 2010 to 2025. This distribution includes: metaheuristic algorithms and hybrid-metaheuristic algorithms, which utilize snapshot-based and continuous approaches

The classification of the taxonomy in Fig. 4 depends on the type of basic mathematical methodology adopted by the algorithms (heuristic, metaheuristic, deep learning, or hybrid). This taxonomy considers the level of combination between these methodologies in the case of hybrid algorithms. This classification helps researchers select the most suitable algorithm based on the nature of the network, performance requirements, and available mathematical resources. The systematic algorithm contrast, as demonstrated in the taxonomy classification, helps researchers select and accurately identify the most effective approaches to identify communities in networks based on the characteristics of the data, algorithm type, objective function, and performance criteria. This comparison is necessary to obtain the correct and accurate results in the analysis of complex and dynamic networks.

5.1 Heuristic and Modularity-Based Community Detection (HMCD)

Heuristic-based dynamic community detection is the process of identifying and evaluating communities or clusters within a dynamic network using heuristic algorithms, moreover, referred to non-evolutionary algorithms. Heuristics are a collection of deterministic rule-based algorithms. It finds a locally optimum sub-solution while considering the issue domain. In addition, it repeatedly calculates some regional variance for a particular decision, and the solution determines whether the local solution is appropriate [9,25].

Hopcroft et al. [6] presented one of the first studies on monitoring community changes in static network snapshots. This study used a hierarchical agglomerative clustering algorithm to identify communities. The nodes in the dataset were treated individually as clusters. The algorithm joined the two closest clusters after each iteration until all nodes formed a single cluster. A collection of clustering trees was generated using the clustering algorithm, denoted T, which produced $(T_1, T_2, \ldots, T_n)$. T_1 was selected as the base tree. The similarity between the clusters in T_1 and T_2 was assessed by comparing the base tree T_1 with the tree T_2, which was derived from a different clustering run using Eq. (11). Then the two clusters C and C' were matched using Eq. (12).

$$Similarity(p,q) = cost(T_p, T_q) = \frac{T_p T_q}{||T_p||||T_q||} \tag{6}$$

$$Match(c,c') = min(\frac{(|c \cap c'|)}{(|c|)}, \frac{(|c \cap c'|)}{(|c'|)}) \tag{7}$$

Kumar et al. [26] proposed an approach to identify communities independently for each snapshot and observe their evolution over time. Chakrabarti et al. [27] used evolutionary clustering to analyze the temporal development of dynamic networks. The researchers observed that noise can cause abrupt changes in node connections, making it crucial to prevent these changes. A temporal smoothness framework meets this need. This framework enabled a smoother evolution for each community over time by avoiding sudden changes in clustering over short periods. A cost function, as shown in Eq. (8), incorporates two sub-costs for smoothing: the Snapshot Cost (SC), which evaluates how a community structure captures the data at time t, and the Temporal Cost (TC), which evaluates the similarity between the current community structure $C_{R_{(t)}}$ and its previous clustering $C_{R_{(t-1)}}$.

$$Cost = \alpha.SC + (1-\alpha).TC \tag{8}$$

In this framework, the variable α remains constant to control the trade-off between each sub-cost. Several additional efforts have been made, following a similar approach in this field, such as the works in [28–32]. These works were considered within the same framework of temporal smoothness. Chi et al. [28] suggested an evolutionary modification of the spectral clustering algorithm by incorporating the Preserving Cluster Quality (PCQ) and Preserving Cluster Membership (PCM) approaches into the spectral clustering framework. The authors presented a modified cost function optimization that integrates both approaches in Eq. (9).

$$C_{total} = \alpha.C_{temporal} + (1-\alpha)C_{snapshot} \tag{9}$$

In PCQ, $C_{temporal}$ referred to the cost of applying the clustering result at time t to the similarity matrix at time $t-1$. In PCM, $C_{temporal}$ calculates the difference between the clustering results at time t and time $t-1$. Both approaches used $C_{snapshot}$ for the static spectral clustering cost, represented by average association, ratio cut, or normalized cut. Lin et al. [29] introduced the FacetNet framework, a Bayesian technique for dynamic networks. This technique used two models to track community evolution: the stochastic block model for community creation and the Dirichlet distribution for tracking community evolution. The estimated community structure was quantified using the KL-divergence, which specified the SC. After each iteration, the approximation structure was changed to minimize the cost function. In this technique, a monotonically declining cost function leads to an optimal solution. The algorithm focused on soft Q and mutual information as objective functions. The drawback of the previous framework was that the network had a fixed number of clusters; this led to a lack of flexibility for community growth. Asur et al. [30] developed an event-based framework using static snapshots. An efficient and scalable Markov Clustering Algorithm (MCL) was used to identify communities in each snapshot. Kim and Han [31] proposed a particle- and density-based evolutionary clustering method for detecting multiple communities. This method used nano-communities and quasi (l-KK) to describe a dynamic network as a group of particles and identified a variable number of communities. Nano-communities constitute dynamic networks, while (l-KK) includes highly connected communities inside these particles.

$$cost = \alpha . SC(CR_0, CR_t) + (1-\alpha) . TC(CR_{t-1}, CR_t) \tag{10}$$

Kim and Han [31] suggested an effective density-based clustering technique in Eq. (11) using optimal Q that detected high-precision local clusters.

$$Q_S = \sum_{c=1}^{NC} \frac{IS_c}{TS} - \left(\frac{DS_c}{TS}\right)^2 \tag{11}$$

Here, NC denotes the overall number of clusters, TS denotes the entire degree of similarity between each pair of network nodes, IS_c denotes the total degree of similarity between two nodes inside a cluster c, and DS_c denotes the whole degree of similarity between a node in the cluster c and any other node in the network. Maximizing QS leads to the best clustering result. Xu et al. [32] applied the evolutionary clustering concept. Xu's approach automatically incorporates modifications based on the differences between the networks at various time steps, rather than calculating the weight coefficient α as outlined in Eq. (8). Nguyen et al. [33] suggested the QCA (Quick Community Adaptation) framework that began with a basic Blondel-based community structure. It takes into account the structural history and changes in the network that impact the structure of its communities. This framework used Q in Eq. (12) as the objective function and the structural history in its adaptation process.

$$Maximize \quad Q = \sum_{c \in c} \left(\frac{m_c}{M} - \frac{d_c^2}{4M^2}\right) \tag{12}$$

Xu et al. [34] proposed a Density Peak Clustering Algorithm (DPC). This algorithm employed a distance function that relies on shared nodes to manage social networks. Li et al. [35] suggested a framework that combined three methods: TILES, GenLouvain, and PisCES for detecting communities in dynamic networks. TILES used label propagation to influence neighborhood community membership; GenLouvain developed a multi-slice modularity generalization based on the equivalence between the Q quality function and community stability in Laplacian dynamics; and PisCES enhanced spectral clustering for network evolution. Kilic and Muldoon [36] suggested skeleton coupling as a new algorithm for generating inter-layer links in temporal networks. This method was specifically designed to improve connectivity between communities over time. Xiong et al. [37] proposed a Simple-based Dynamic Decentralized Community Detection Algorithm (S-DCDA) to address the challenge of resource consumption in the network and inaccurate community structure identification in standard detection algorithms. This algorithm promoted dynamic decentralization, letting each community node participate in being a community or network core at any point. Kumar et al. [38] presented a Community-enhanced Link Prediction Algorithm for Dynamic Networks (COMMLP-DYN). This algorithm introduced a link prediction approach that uses parameterized influence groups of nodes. Then, unique features were created utilizing local, global, quasi-local, and community information-based similarity functions. Scoring-based was used to optimize this feature set. Finally, four machine learning-based categorization models were used to predict links. Sahu et al. [39] suggested two methods; the first is the Louvain method, which represents a greedy optimization method to maximize modularity. The second method, the Label Propagation Algorithm (LPA), represents a fast heuristic for community assignment. These two methods are implemented in parallel to take advantage of multi-core architectures and gain computational speed. Djurdjevac et al. [40] suggested the Louvain heuristic algorithm to maximize modularity and compared its performance in various scenarios involving missing edges, nodes, or temporal information. The results showed that data insufficiency significantly impacts the quality of the detected communities.

Table 1 summarizes the heuristic algorithms reviewed, detailing their benefits, drawbacks, the measures applied, the evaluation measures used, and the datasets utilized.

5.2 Metaheuristic Based Community Detection (MCD)

Metaheuristic algorithms, moreover, known as evolutionary algorithms, encouraged researchers to tackle complex global optimization challenges. These algorithms used a fitness function to capture the evolution of the community by evaluating each solution to find optimal or near-optimal solutions, depending on genetic operators [41,42]. The main approaches in Fig. 3 classify dynamic networks in metaheuristic community detection into two groups:

A- **Snapshot Based (Independent) Community Detection**

Guerrero et al. [43] proposed a novel Pareto-based multiple-objectives evolutionary algorithm. They employed three objectives, including Q, CO, and the imbalance for clustering nodes into communities.

$$Minimize \quad CO = \sum_{i=1}^{N} \frac{L_i}{L_i + 2Z_i} \tag{13}$$

Here, L_i represents the external connections in the ith community, Z_i denotes the internal connections of the ith community, while N refers to the total number of communities.

$$Minimize \quad IMB = \sqrt{\frac{\sum_{i=1}^{N}(X_i - \overline{X})^2}{N}} \tag{14}$$

Table 1: Heuristic and modularity based community detection (HCD)

Authors	Name of method or algorithm	Advantages of algorithm	Disadvantages of algorithm	Measure	Evaluation measures	Dataset
[35]	(TILES, GenLouvain and PisCES)	TILES: provides high-resolution, realistic dynamic tracking of communities methods, GenLouvain: enhances modularity for time, accuracy/speed balance, pisCES:Stability of communities over time	TILES: Sensitivity to noise, Computational complexity, Instability, GenLouvain: The challenge of small-scale accuracy, depending on modularity optimization, PisCES: Computational cost, parameters sensitivity.	Modularity (Q)	F1-Score, Accuracy, AUC	1-Social networks with social account users and friends. 2-Cell phone call. 3-Tech-website answering questions.
[36]	Skeleton coupling algorithm	Improved temporal consistency, Better detection of singleton and transient communities	Computational complexity and Parameter sensitivity	Multilayer Modularity Maximization (MMM)	NMI	(78) Synthetic Neuronal time series datasets.
[37]	Simple-based Dynamic Decentralized Community Detection Algorithm (S-DCDA)	Ease of implementation and minimal resources requirements, Improved accuracy and reliability, Adaptive to changes	Limitations in detecting overlapping communities, Results are affected by parameters, Limitations in extensive networks	Familiar set model, Node invitation model, Community maintenance model	Accuracy, Stability, Execution time	Socially Aware Networks nodes are forwarding messages.
[38]	Community enhanced Link Prediction Algorithm for Dynamic Networks (COMMLP-DYN)	Improving Prediction Accuracy and Improving Performance	Computational Complexity, Data Quality Dependence and Continuous improvement Needed	Modularity (Q)	AUC, AUPR and Average Precision	1-MIT3 (human interaction data of 100 MIT students). 2-Radoslaw-Email(email transmission of company employees). 3-EU-Core5 (email transmission with a European institution). 4-FB-Forum (open interactions in an asocial network). 5-CollegeMsg7 (student engagement on social media platform).
[39]	Louvain method and Label Propagation Algorithm (LPA)	1-Louvain:High quality community identification and Suitable for large networks. 2-Label propagation algorithm: Very high speed and easy to parallelize	1-Louvain: Slower than LPA and Less efficient in parallelization than LPA. 2-LPA: The quality of the communities is lower than Louvain, and the results may be unstable at times.	Modularity (Q)	Q	1-WebGraphs (LAW). 2-Social Networks (SNAP). 3-Road Networks (DIMACS10). 4-Protein k-mer Graphs (GenBank).

(Continued)

Table 1 (continued)

Authors	Name of method or algorithm	Advantages of algorithm	Disadvantages of algorithm	Measure	Evaluation measures	Dataset
[40]	Louvain heuristic algorithm	It is an efficient and relatively fast at finding good community partitions, even in large networks, and it is suitable for real-world applications where the data is extensive or incomplete.	Algorithm Flaws, resolution Limit Problem, Overfitting Problem, and Solutions Are Not Perfect.	Modularity (Q)	LAMI score and	Synthetic and two real-world datasets.

where N is the total number of communities, X_i is the total number of nodes in the ith community, and $\overline{X}$ is the total number of nodes in the network. Abbood et al. [44] proposed the (HMM-MODCD) algorithm, which utilized a Hidden Markov Model (HMM) and a multi-objective optimization algorithm to identify and capture the evolved community structures in complex networks. This algorithm employed multiple objectives to find community structures for each time step. Two objectives, namely, intra-links in Eq. (15) and inter-links in Eq. (18), were optimized simultaneously to identify the set of partitions that best balance a measure of inter-community sparsity with a measure of intra-community density. Subsequently, the Viterbi algorithm was employed to choose the most consistent partitions over time.

$$Minimize \quad f_{Intra}(C) = 2(N-K) - \sum_{K=1}^{K} \frac{1}{|C_K|} \sum_{V \in C_K} \frac{\underline{d}(v, C_k)^2}{d(v)} \tag{15}$$

Let C represents a network partition that is split into K communities, N is the total number of nodes, $d(v)$ is the degree of node v, in addition, $\underline{d}(v, C)$ is the total number of edges connecting node v to other nodes in C in Eq. (16).

$$\underline{d}(v_i, C) = \sum_{v_i \in C, v_j \in C} A_{ij} \tag{16}$$

The greatest average number of external connections among communities is quantified by the second objective function in Eq. (17). Let $l(v, C_j)$ be the ratio of the highest number of connections between node $v \in C_j$ and any other community, and the internal degree of v:

$$I(V, C_j) = \frac{max_{C_i \neq C_j} \sum_{w \in_{c_i}} A_{vw}}{max(\sum_{w \in_{c_j}} A_{vw}, 1)} \tag{17}$$

Here, Inter-Score is defined as:

$$Minimize \quad f_{Inter}(C) = \sum_{j=1}^{K} \frac{1}{|C_j|} \sum_{v \in C_j} L(V, C_j) \tag{18}$$

Sun et al. [45] suggested a Core Node Knowledge-based multi-objective Particle Swarm Optimization Algorithm (CNPSO) to detect dynamic network community structure. This algorithm minimized two conflicting objective functions: intra (KKM) and inter (RC) to detect community structure at each time step. Assigning weights to KKM and RC for optimization would be difficult. To identify an appropriate solution, the Q metric was introduced to assess the quality of the community structure. To measure the similarity between two successive time steps, NMI was employed for this purpose.

Table 2 summarizes the metaheuristic algorithms reviewed using the independent approach, detailing their benefits, drawbacks, the objective functions applied, the evaluation measures used, and the datasets utilized.

Table 2: Metaheuristic based Community Detection (MCD)

Authors	Name of algorithm	Advantages of algorithm	Disadvantages of algorithm	Objective function	Evaluation measure	Dataset
[43]	Pareto-based multiple objectives evolutionary algorithm	This algorithm balances multiple objectives	Performance difficulties when the number of objectives increases, which complicates the computations.	Modularity, Conductance, Imbalance and NMI	Modularity, Conductance, Imbalance, Average Precision	Power grids datasets.
[44]	Multi-Objective Evolutionary Algorithm based Community Detection (MOEA-CD), Hidden Markov Model (HMM) and Viterbi algorithm	The algorithm's ability to effectively detect communities in complex networks, enhancing understanding of social structure.	Despite HMM and Viterbi models providing high accuracy in predicting temporal patterns, their complexity can lead to difficulties in implementation and interpretation.	Intra-links, Inter-links and NMI	Q and NMI	1-Fix-Net, Var-Net. 2-Green dataset. 3-Cell phone call, MP Twitter network, and Enron email networks.
[45]	Core Node Knowledge based multi-objective Particle Swarm Optimization algorithm (CNPSO)	The CNPSO algorithm improved performance in detecting dynamic communities by utilizing background knowledge, which enhances the accuracy of solutions.	It may face some drawbacks, such as computational complexity and increased execution time compared to traditional algorithms.	KKM, RC, Q and NMI	Q, NMI and Error rate	1-Green dataset 2-SYN-FIX and SYN-VAR. 3-Cellphone calls and Enron mail.

B- **Continuous Community Detection**

Folino and Pizzuti [46] introduced the concept of a multi-objective optimization problem, which focused on optimizing two objectives. The primary objective was to maximize the CS by maximizing connections within each community and minimizing connections between communities. Second objective, reduce the similarity *NMI* between the current community structure CR^t and the past community structure (CR^{t-1}). The definition of CS is given in Eq. (20).

$$Maximize \quad CS(CR^t) = \sum_{i=1}^{k} score(C_i^t) \tag{19}$$

where

$$score(C_i^t) = \frac{\sum_{i\epsilon c^t}(\mu_i)^2}{|C^t|} * \sum_{i,j\epsilon c^t} A_{i,j}^t \tag{20}$$

The second equation represents the number of connections linking the vertices in C^t, or the number of values 1 in the adjacency sub-matrix A^t. This refers to C^t. The first term calculates the square mean of :.

$$\mu_i = \frac{1}{|C^t|}\sum_{j\epsilon c^t} A_{i,j}^t \tag{21}$$

where μ_i refers to the fraction of links connecting each node i of C^t to the other nodes in the same community C^t. The dynamic Multi-Objective Genetic Algorithm (DYNMOGA) is presented by [47]. This algorithm is designed to identify evolved communities that employ the temporal smoothness framework suggested by Chakrabarti et al. [27]. The algorithm optimizes four objectives: CS, Q, CO, and Normalized Cut (NC) as snapshot costs and NMI as temporal cost. This represents the optimal trade-off between the precision of the clustering in the current step and the drift from one time step to the next.

$$Maximize \quad CS = \sum_{s=1}^{k} \left(\sum_{u \in C^t} \left(\frac{m_s(v)}{n_s} \right)^2 \right) * \frac{2m_s}{n_s} \tag{22}$$

$$Maximize \quad Q = \sum_{s=1}^{k} \left[\frac{L_s}{m} - \left(\frac{d_s}{2m} \right)^2 \right] \tag{23}$$

$$Minimize \quad CO = \sum_{s=1}^{k} \frac{c_s}{2m_s + c_s} \tag{24}$$

$$Minimize \quad NC = \sum_{s=1}^{k} \frac{c_s}{2m_s + c_s} + \frac{c_s}{2(m - ms) + c_s} \tag{25}$$

In the above equations, n_s represents nodes, m represents edges, and m_s represents the internal connections between nodes within a cluster. On the other hand, c_s represents the external connections between the clusters, l_s is the total number of edges connecting nodes inside the cluster, and d_s is the sum of the degrees of the cluster nodes. Ma et al. [48] suggested a multi-objective evolutionary algorithm called DYN-DMEA for detecting dynamic communities. They used the MOEA/D algorithm [49]. The authors employed Q as the first objective to assess the quality of the community structure and NMI as the second objective to assess the temporal cost. At each time step, the partition that maximized Q was selected as the optimal. Zhou et al. [20] introduced a Multi-Objective Discrete Cuckoo Search Algorithm (MODCS) to identify clusters in dynamic networks. This algorithm simultaneously optimized Q and NMI as objective functions. Gao et al. [50] presented an efficient approach to optimizing Multiple Objective Particle Swarms using a decomposition (DYN-MODPSO) to identify the dynamic community structure. This approach utilized Modularity Density to evaluate the accuracy of the community structure at each time step and NMI to determine the similarity of the structure between two successive time steps. Wang et al. [51] proposed a label-based swarm intelligence approach for dynamic community detection. The fundamental idea incorporated the particle swarm optimization algorithm with both the genetic approach and label propagation. This algorithm maximized Q and NMI simultaneously. Zou et al. [52] suggested a new multi-objective optimization genetic algorithm called Feature Transfer Based Multi-Objective Optimization Genetic Algorithm (TMOGA). This algorithm relied on collecting information from stable characteristics of past community structures to be utilized in later optimization processes. Optimized two objective functions, Q and NMI. Wang et al. [53] suggested a dynamic community detection algorithm based on the Optional Pathway Guide Pity Beetle Algorithm (DYN-OPGPBA). This algorithm optimized Q and NMI simultaneously. Li et al. [54] presented a multi-objective optimization algorithm based on the Characteristics of Dynamic Social Networks for Community Discovery (MOCCD). This algorithm aimed to identify communities in dynamic networks by combining the features of temporal variability, stability, and continuity. The researchers developed a dynamic structural evolution model that considered both the continuity of community evolution and the separation of community structure. To assess the objective function, they used two measures: Q and NMI. When the number of nodes was different between two successive time steps, the dynamic community evolution continuity (DCEC) measure was used instead of NMI.

$$DCEC(A, B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} \frac{C_{ij}}{N_{A\cup B}} \log\left(\frac{C_{ij}N_{A\cap B}}{C_{i.}C_{.j}}\right)}{\sum_{i=1}^{C_A} \frac{C_{i.}}{N_A} \log\left(\frac{C_{i.}}{N_A}\right) + \sum_{j=1}^{C_B} \frac{C_{.j}}{N_B} \log\left(\frac{C_{.j}}{N_B}\right)} \tag{26}$$

Here, N_A and N_B represent the total number of nodes in the partition, $N_{A\cup B}$ is the total number of nodes in the partitions A, and B, $N_{A\cap B}$ is the total number of nodes in partition A or partition B. Ma et al. [55] proposed a multiobjective genetic algorithm called Higher Order Knowledge Transfer (HOKT) that detects first- and higher-order knowledge by analyzing the adjacency matrix of snapshots and integrating higher-order information from prior snapshots to detect dynamic communities with significant changes effectively. This algorithm optimized two objectives, Q and NMI, simultaneously. Mishra et al. [22] introduced two algorithms (Main and Rearrange Check) to identify non-overlapping communities in evolving networks. These algorithms employed three contradictory objective functions: community score, similarity coefficient, and pull. Yu et al. [56] suggested a consensus community-based discrete spider wasp optimization (SWO) algorithm to identify the dynamic community structure of the network. This algorithm aimed to enhance the accuracy of the detected communities over time. This algorithm uses three update strategies, search/follow, nesting, and mating, to detect discrete data communities. These strategies compensate for the absence of random mutation.

However, concerning the snapshot cost function, several other familiar community detection models have not been studied in our literature review. These include Expansion (EX) and Internal Density (ID) [57,58]. The formulation of EX and ID is restated as follows:

$$Minimize \quad EX = \sum_{i=1}^{k} \frac{c_s}{n_i} \tag{27}$$

$$Maximize \quad ID = \sum_{i=1}^{k} \frac{m_s}{n_i * (n_i - 1)/2} \tag{28}$$

In the above equations, c_s represents the external connections between clusters, m_s represents the internal connections between nodes within a cluster, and n_i represents the number of nodes in each cluster. Table 3 summarizes the metaheuristic algorithms reviewed using the continuous approach, detailing their benefits, drawbacks, the objective functions applied, the evaluation measures used, and the datasets utilized.

Table 3: Metaheuristic based Community Detection (MCD)

Authors	Name of algorithm	Advantages of algorithm	Disadvantages of algorithm	Objective function	Evaluation measure	Dataset
[52]	Feature Transfer Based Multi-Objective Optimization Genetic Algorithm (TMOGA)	Improved Performance, Provides a diverse range of solutions, Speeds up the search process	Design complexity, requires sufficient training Data, Increased implementation Time	Modularity (Q) and NMI	Q, NMI, and CS.	SYN-FIX and SYN-VAR.
[53]	Dynamic Community Detection Algorithm based on Optional pathway Guide Pity Beetle Algorithm (DYN-OPGPBA)	The algorithm offers the ability to detect dynamic communities in complex networks, enhancing understanding of interactions between elements.	It may face drawbacks such as complexity in implementation and the need for precise performance evaluation metrics.	Modularity (Q) and NMI	Q and NMI	1-karate, dolphins, football, polbooks. 2-NetScience network, the Euroroad network, and the PGP network.

(Continued)

Table 3 (continued)

Authors	Name of algorithm	Advantages of algorithm	Disadvantages of algorithm	Objective function	Evaluation measure	Dataset
[54]	Multi-objective optimization algorithm based on the Characteristics of Dynamic Social Networks for Community Discovery (MOCCD)	The algorithm offers improved community detection in dynamic social networks by using multiple criteria, which enhances the accuracy of the results.	This may face drawbacks such as design complexity and the need for significant computational resources	Modularity (Q) and (NMI or dynamic community evolution continuity (DCEC))	Error rate, Q and NMI	1-SYN-FIX and SYN-VAR. 2-Green datasets. 3-VAST dataset. 4-DBLP Co-author Network.
[55]	Multi-objective genetic algorithm is called Higher-Order Knowledge Transfer (HOKT)	This algorithm improved inter-generational knowledge transfer, which enhances the efficiency of multi-objective solution searches.	It may face drawbacks such as implementation complexity and the need for fine-tuning parameters.	Modularity (Q) and NMI	F-score and NMI	1-Cell Phone Call and Enron mails. 2-SYN-FIX, SYN-VAR and Four Events (Green dataset).
[22]	Two algorithms (Main and RearrangeCheck)	Balanced optimization of multi-objectives, Unbiased community identification, and Adaptability to dynamic networks	Computational complexity, Parameter sensitivity, Interpretability challenges, and Limited real-world validation	Community Score (CS), similarity coefficient, ratio of intra and inter community (pull)	NMI, Conductance and Adjusted Mutual Information (AMI).	1-Karate-Football-Dolphin. 2-emailEU-AstroPh.
[56]	Consensus community-based discrete Spider Wasp Optimization (SWO) algorithm	Improving the accuracy of community detection, Adapting to dynamic networks, Using the concept of a consensus community, and Achieving a balance between competing objectives.	Implementation complexity, Need to fine-tune parameters, Increased implementation time, Possibility of falling into local solutions	Kernel k-means (KKM), ratio cut (RC) and NMI	NMI	1-Cell Phone Call and Enron mails. 2-SYN-FIX, SYN-VAR and Four Events (Green dataset).

5.3 Deep Learning Based Community Detection (MCD)

Deep learning can generate more robust representations of node properties and dynamic community structures. Additionally, discover the patterns of nodes, neighbors, and sub-graphs. The deep learning framework generates lower-dimensional vectors from high-dimensional data, which reflect dynamic structural connections within the network [59]. Sankar et al. [60] introduced the Dynamic Self-Attention Network (DySAT), which is an innovative neural architecture that accumulates node representations to capture the evolution of dynamic graph structures. DySAT specifically calculated node representations by combining self-attention across the domains of structural neighborhood and temporal dynamics. Compared to other advanced approaches for simulating graph evolution, dynamic self-attention proved to be economical and consistently delivered higher performance. This study performed link prediction on two types of graphs: communication networks and bipartite rating networks. Rossi et al. [61] introduced Temporal Graph Networks (TGNs), a versatile and practical framework for deep learning on dynamic networks, which are represented as a series of temporal events. TGNs are significantly more effective than older methods because they integrate multiple components (memory modules, memory updater, embedding module, message function) and graph-based operators in a new way, making them more efficient. Additionally, provided an innovative training method that allowed the model to learn from the sequential nature of the input, ensuring effective parallel processing. Costa [62] presented a graph embedding method that used the Node2vec algorithm to transform an adjacency matrix of a dynamic network into a low-dimensional space. Next, they used deep reinforcement learning to improve the density of aggregation in community identification. To capture the structure of communities in dynamic online networks, researchers used an actor-critic reinforcement learning algorithm to maximize the modularity density of the community. This was achieved by combining the efficiency of GNN-based networks, specifically Graph Attention Network (GAT) and Graph Convolution Network (GCN).

This study presents a comparative table to illustrate why deep learning methods are preferred over traditional methods, as shown in Table 4.

Table 4: The comparative study between deep learning methods and traditional methods

Criteria	Deep learning methods	Traditional methods
Large-scale evolving networks	Best scalability like (GNNs) [63]	Suffering from a scalability struggle
Networks with extensive node/connection features	Utilize attributes to improve accuracy [64]	Disregard attribute data
Noisy networks	Powerful representation of network [65]	Sensitive to noise
Real time detection	High computational complexity [66]	Faster than deep learning methods

5.4 Hybrid Based Community Detection (HCD)

This section presents a brief description of two types of hybrid metaheuristic-based community detection algorithms.

5.4.1 Hybrid Metaheuristic Based Community Detection (HMCD)

High-level propagated hybrid metaheuristic algorithms (HR-HMDC) are used to identify communities. This type of hybridization displays a clear collaboration between heuristic algorithms (HCD) and meta-heuristic algorithms (MCD) [9,67,68]. Mainstream approaches in Fig. 3 classify dynamic networks in the hybrid metaheuristic community detection into two groups:

A- **Snapshot Based (Independent) Community Detection**

Zeng et al. [69] presented the concept of a consensus cluster, maintaining the stability of these clusters throughout the evolution process. This concept has been previously discussed in several studies by [70,71]. This concept addresses the previous problem of the DYNMOGA algorithm, as presented by [47]. It utilized Particle Swarm Optimization (PSO), which combined the CCPSO algorithm to detect communities with the label propagation method for initialization. This algorithm used multiple objectives: KKM, RC, and NMI.

$$KKM = 2(n-k) - \sum_{i=1}^{k}\left(\frac{L(v_i, v_i)}{|v_i|}\right) \tag{29}$$

$$RC = \sum_{i=1}^{k}\left(\frac{L(v_i, \overline{v}_i)}{|v_i|}\right) \tag{30}$$

$$X^* = arg_x min_{x \in PS}\left(\frac{KKM(x) - KKM_{min}}{KKM_{max} - KKM_{min}} + \frac{RC(x) - RC_{min}}{RC_{max} - RC_{min}}\right) \tag{31}$$

where X^* is defined as the knee solution, KKM and RC are the two objective functions, PS refers to the Set of Pareto-optimal solutions, and x is the solution related to PS. $KKM(x)$ and $RC(x)$ are the two objective functions, while KKM_{max} and KKM_{min} represent the maximum and minimum values of the objective function KKM in PS. Similarly, RC_{max} and RC_{min} represent the maximum and minimum values of the objective function RC in PS. Besharatnia et al. [72] proposed a new framework called (IGWO-LP) that combined the improved multi-objective Gray Wolf Optimizer algorithm and the Label Propagation algorithm, resulting in improved efficiency and better precision of the result. The primary objective is to prevent overlap between communities and ensure that all nodes within each community share the same values. This framework employed benchmark functions (F1-F23) for each wolf and optimized two objectives: Q and NMI. Ranjkesh et al. [73] introduced the Robust Dynamic Memetic Algorithm (RDMA-NET), which aimed to enhance the identification of dynamic communities in complex networks. Table 5 summarizes the hybrid metaheuristic algorithms reviewed using the independent approach, detailing their advantages, drawbacks, the heuristic algorithm employed, the objective functions applied, the evaluation measures used, and the datasets utilized.

Table 5: Hybrid Metaheuristic based community detection (HR-MCD)

Authors	Name of algorithm	Advantages of algorithm	Disadvantages of algorithm	Name of heuristic algorithm	Objective function	Evaluation measure	Dataset
[72]	Improved multi-objective Gray Wolf Optimizer algorithm	Improve performance, Balancing competing objectives, Reducing the need for local solutions, Efficient processing of big data, and the algorithms' ability to adapt to changes in the data or environment.	Implementation complexity, Need to fine-tune parameters, Increased implementation time, and the effectiveness of the algorithm depends on the quality of the input data	Label propagation algorithm	Modularity (Q), and NMI	NMI	1-SYN-VAR, SYN-FIX. 2-Cell phone calls and Enron mail. 3-Football dataset.
[73]	Robust Dynamic Memetic Algorithm (RDMA-NET)	Robust Optimization, Scalability and Stability in Dynamic Environments	Computational Complexity, Parameter Sensitivity and Overfitting Risk	local search Strategy	community robustness (Rc)	NMI	1-(LFR benchmark). 2-Enron Email, Cellphone Calls.

B- **Continuous Community Detection**

Gong et al. [74] proposed a multi-objective immunity algorithm that enhanced precision and efficiency through genetic operators and local search. This algorithm optimized Modularity and NMI simultaneously to enhance community accuracy and stability. Chen et al. [75] proposed a new multi-objective evolutionary algorithm for the identification of dynamic communities. This algorithm is based on the framework of a non-dominated sorting genetic algorithm. The MD function is utilized as a snapshot cost to address the limitations of the Q function. Additionally, NMI was utilized to estimate the temporal cost. Attea and Khoder [8] introduced a new framework for managing the dynamic nature of social networks. This framework integrated the evolutionary algorithm with the migration mutation operator. The algorithm optimized two objective functions: the (intra-neighbor score) in Eq. (32) and the (inter- and intra-temporal cost) in Eq. (34).

$$min \quad \phi_1 = n^2 - \sum_{i=1}^{K} \frac{(m_i + nstrong_i)}{n_i} \tag{32}$$

Here, ϕ_1 indicates the intra-community structure (intra-snapshot cost), and $nstrong_i$ is the overall number of strongly neighbored nodes in the community i. A node v is treated as a highly neighborhood node in the partitioning solution C if and only if there is no other community j in which node v can establish further intra-connections.

$$nstrong_i = |\forall_{v \in C_i} I \nexists C_j \in C \implies in_j(v) > in_i(v)| \tag{33}$$

$$min \quad \phi_2 = \frac{2((1 - NMI) * (\sum_{i=1}^{K} inter_i))}{(1 - NMI) + (\sum_{i=1}^{K} inter_i)} \tag{34}$$

where

$$inter_i = \frac{\sum_{v \in C_i} out_i(v)}{nstrong_i} \tag{35}$$

where ϕ_2 is described as the mixture of inter-temporal and temporal cost that takes into account the function of inter-community structure and normalized mutual information (NMI). The ratio of the total number of closely adjacent nodes in the community i to all nodes in the community that have connections outside the community is shown by the term $inter_i$. Niu et al. [76] presented the integration of a Multi-objective Genetic algorithm with a Label Propagation method (L-DMGA), which enhanced the initial individual clusters and produced better cluster quality. Furthermore, the combination of the label propagation method and the mutation operator improved the quality of clustering and increased the speed of convergence. This algorithm optimized two objectives simultaneously: NMI and Q. (1) and the Error rate in the following equation.

$$||ZZ^T - GG^T|| \tag{36}$$

An $n * k$ indicator matrix, Z, was developed to calculate the error rate, where n represents the total number of nodes and k represents the total number of communities. Additionally, a similar indicator matrix, G, is used to indicate the true community. Eq. (36) calculates the distance between the community matrix Z and the true community matrix G. Messaoudi and Kamel [21] presented a novel multi-objective Bat Algorithm for community discovery. This algorithm is a swarm intelligence-based bio-inspired algorithm, that utilizes two objectives, MD and NMI. Furthermore, it utilized the Mean Shift algorithm during the initialization phase to achieve enhanced clustering. In addition, the primary mechanism of the mean shift is used to characterize a novel mutation operator.

$$MD = \sum_{i=1}^{N} \frac{L(V_i, V_i) - L(V_i, \overline{V_i})}{|V_i|} \tag{37}$$

$L(V_i, V_i)$ denotes the quantity of connections inside the community V_i, while $L(V_i; \overline{V_i})$ represents the number of connections that connect a node in the community V_i to nodes outside the community. The quantity of nodes in the community is given by $|V_i|$. Panizo et al. [77] employed a new multi-objective evolutionary algorithm to identify dynamic communities. The authors utilized an immigrant scheme that adds new individuals to populations to enhance diversity. In addition, three local search methods (label propagation, random walks, and hybrid approaches) were used to achieve high-quality solutions and speed computations. The local search methods aimed to transform invalid solutions arising from network modifications into valid ones, while maintaining the highest level of quality. The algorithm optimized three objectives: the "Inverse Community Score" in Eq.(38), the "Average ODF" in Eq. (39), and NMI.

$$Minimize \quad C - SCORE^{-1} = \frac{1}{\sum_{S \in C} \left(\frac{2ms}{ns}\right)^2} \tag{38}$$

The inverse community score indicates the connectivity of the nodes within a community. Where *ms* refers to the number of connections in a community *S*, while *ns* represents the number of nodes within it.

$$Minimize \quad AVG - ODF = \sum_{S \in C} \left(\frac{1}{ns} \sum_{u \in S} \left(\frac{|u, v| v \notin S|}{d(u)} \right) \right) \tag{39}$$

Here, "Average ODF" measures the number of edges outside a community; *C* represents the collection of communities encoded by an individual, *S* represents the community, *ns* represents the number of nodes, and $d(u)$ represents the degree of the node.

Liu et al. [78] presented a framework that combined a multi-objective evolutionary algorithm called "Detecting Evolving Community Structure" (DECS), a label propagation method in initialization, and a migration operator in mutation. This framework employed the temporal smoothness proposed by Chakrabarti et al. [27]. The DECS algorithm maximized Q and NMI simultaneously in every subsequent time step.

$$C_t^* = \begin{cases} argmax_{C_t}[Q(C_t), \quad NMI(C_t, C_t - 1)], t \geq 2 \\ argmax_{C_t}[Q(C_t)], \quad t = 1 \end{cases} \tag{40}$$

At the initial time step, that is, $t = 1$, without a comparable previous community structure, C_t^* represents a network partition that maximizes modularity and represents a community structure.

Besharatnia et al. [79] proposed a new approach (MOGWO-LP) that combines a Multi-Objective Metaheuristic Gray Wolf algorithm with the Label Propagation method to progressively identify communities. This algorithm optimized two objectives: Q and NMI. Yin et al. [80] proposed the Multi-Objective Discrete Particle Swarm Optimization for Dynamic Network algorithm (DYN-MODPSO). Typically, a random walk method was employed to alter the initial snapshot population deviation in an evolving network. The algorithm optimized two objectives: CS and NMI. Cai et al. [81] presented the NOME algorithm, which is based on the combination of multiple objective evolutionary clustering with a node occupancy assignment mutation operator. NOME utilized the MOEA/D framework, which optimized both Q and NMI simultaneously to detect evolving communities. Wang et al. [82] suggested a community identification approach that combines a multi-objective evolutionary algorithm with decomposition and an evolutionary clustering algorithm. This approach used the adaptive mutation evolutionary strategy. This strategy was employed in conjunction with a corresponding change in the number of subproblems in MOEA/D as a means of addressing environmental influences. Furthermore, we dynamically adjust the mutation rate according to population changes. Table 6 summarizes the hybrid metaheuristic algorithms reviewed using the continuous approach, detailing their advantages, drawbacks, the heuristic algorithm employed, the objective functions applied, the evaluation measures used, and the datasets utilized.

Table 6: Hybrid Metaheuristic based community detection (HR-MCD)

Authors	Name of algorithm	Advantages of algorithm	Disadvantages of algorithm	Name of heuristic method or algorithm	Objective function	Evaluation measure	Dataset
[77]	A new multi-objective evolutionary algorithm	Improved performance in detecting community structures	Implementation complexity	Immigrant scheme,label propagation, random walks, and hybrid approaches	Inverse Community Score, Average ODF and NMI	Mean square error	Synthetic dynamic attributed networks (DANCER) using micro and macro operations
[78]	Detecting Evolving Community Structure algorithm (DECS)	Improving community detection accuracy, Flexibility and Balancing objectives	Implementation complexity, Need to fine-tune parameters, Increased implementation time and Possibility of falling into local solutions	Label propagation in initialization, and a migration operator in mutation	Modularity (Q) and NMI	NMI	1-SYN-FIX, SYN-VAR and SYN-EVENT. 2-The Cellphone Calls and Enron Mail.
[79]	Multi-objective gray wolf algorithm	Achieving a balance between objectives, High flexibility: Adapts to a variety of optimization problems	Implementation complexity, Possibility of falling into local solutions, Complexity of parameters and Increased computation time	Label propagation method	Modularity (Q) and (NMI)	NMI and Error rate.	1-SYN-VAR,SYN-FIX and LFR model. 2-Cell phone calls and Enron mail.
[80]	Multi-Objective Discrete Particle Swarm Optimization for Dynamic Network algorithm (DYN-MODPSO)	Balancing Conflicting Objectives, Efficiency in Large-Scale Networks, Dynamic adaptation and avoiding structural bias	Computational complexity and Parameter sensitivity	Random walk algorithm	Community score (CS) and NMI	NMI	Datasets z3, z4, z5 and z6 YouTube, LiveJournal, DBLP and Flickr.
[81]	NOME algorithm (Node occupancy assignment and multi-objective evolutionary clustering)	Improved efficiency: Helps optimize resource allocation effectively. Flexibility in applications: Can be used in multiple fields, such as networks and big data.	Implementation complexity: It may require advanced technical knowledge to implement correctly, and Possibility of falling into local solutions.	Boundary node occupancy assignment method	Modularity (Q) and NMI	Q and NMI	1-SYN-VAR and SYN-FIX. 2-Cellphone calls and Enron mail.
[82]	Dynamic community detection algorithm based on Decomposition based Evolutionary Algorithm (DYN_BDEA)	Improved detection of dynamic communities and Supports optimization of multiple objectives simultaneously, helping to achieve a balance between different objectives.	Implementation complexity and The need to adjust parameters	Adaptive Evolutionary Strategy	Q and NMI	Modularity Density (MD) and NMI	1-Artificial synthetic network dataset. 2-Football dataset.

The key limitations and future directions of the community detection algorithms in dynamic networks are clarified in Table 7.

Table 7: The key limitations and future directions of the community detection algorithms in dynamic networks

Algorithms	Key limitations	Future directions
Heuristic and modularity based community detection	Suffered from (resolution limitation [83], suboptimal results [84], temporal conflict [85], sensitivity to noise)	Employing multi-resolution methods to address the resolution limitation, utilizing the incremental algorithms to enhance the dynamic community structure, hybridization between heuristic and metaheuristic or deep learning to improve quality of solutions.
Metaheuristic-based community detection	Premature convergence, sensitivity of parameters, challenge of scalability, absence of combination	Make hybridization between metaheuristic algorithms and heuristic, especially in (initialization and mutation), in addition to metaheuristic with deep learning.
Deep learning based community detection	High computational cost, overfitting issue [86], the problem of interpreting identified communities, the training process requires a large amount of data [86].	Effective architectures, transfer learning
Hybrid Metaheuristic based community detection	The challenges of computational cost and Scalability, Sensitivity of Parameters setting, the challenge of balancing between convergence and diversity	Create self-adaptive or learning-based parameter setting to enhance robustness and decrease the level of manual intervention, Set up standard evaluation measures for hybrid metaheuristic methods to allow a simple way of looking at comparison and reproducibility.

5.4.2 Hybrid Deep Learning Based Community Detection (HDLCD)

Heuristic deep learning based community detection(HDLCD): This type of hybridization uses a deep model to show the network in a low-dimensional embedded space, and then a clustering method in the feature space can show how the communities change over time. This hybridization between them offers a robust framework for improving performance and accuracy in the detection of dynamic networks.

Wang et al. [87] employed an evolutionary clustering framework to capture the evolution of the community structure. The sE-Auto-encoder semi-supervised algorithm was suggested to lessen the effect of nonlinear features on the low-dimensional representation. This was achieved by utilizing three overlapping auto-encoders, taking into account time constraints to help maintain relationships between nodes, as well as ensuring temporal smoothness to prevent abrupt changes between time steps. Finally, after obtaining the results from the previous autoencoders, the k-means algorithm was used to divide the results into final communities. Qu et al. [88] proposed a dynamic community detection framework that integrated dynamic community detection based on the evolutionary deep-walk algorithm (DEDW) with the K-means clustering algorithm. The DEDW algorithm integrates the graph embedding algorithm with the deep-walk algorithm. Graph embedding was used to address the issue of data scarcity, transforming the dynamic network into a low-dimensional representation to facilitate enhanced data processing. After that, the Deep Walk static graph embedding algorithm was implemented in the dynamic network to generate node embedding feature vectors derived from the stable characteristics of community structure modifications. Finally, the K-means clustering algorithm was applied to identify the dynamic structure of communities. Zhang et al. [89] suggested a node representation method that combines node influence with a random walk strategy to improve node representation, thus better reflecting the network structure. Subsequently, they employed a dynamic community detection algorithm. This algorithm consisted of two stages: the first stage involved altering the local community structure to maximize modularity. In contrast, the second stage adapted to changes in the number of communities. Pan et al. [86] introduced an innovative dynamic community identification framework that integrated a deep learning model with an evolutionary clustering algorithm (DLEC). Addressing the insufficiency of the adjacency matrix to represent the similarity relationships among nodes suggested a matrix construction technique to produce a similarity matrix that encoded the structure of communities. Subsequently, the deep auto-encoder was used to maximize the quality of community detection at this step. A graph regularization term is incorporated as a temporal smoothness constraint to reduce the evolution of the community. This framework balanced snapshot cost and temporal cost, enhanced accuracy in community detection outcomes across successive time steps, and reduced abrupt modifications in community structures. Ultimately, the K-means clustering method was utilized within the low-dimensional network space to derive the community structure.

Table 8 summarizes the comparative analysis of community detection categories.

Table 8: Simplified comparative analysis of community detection categories

Category	Advantages	Limitations	Computational complexity	Scalability	Practical applicability
Heuristic and Modularity-based (HMCD)	Simple design; fast for small networks	May get stuck in local optima; limited resolution	Fast or Moderate: seconds to minutes for small networks	Limited to small/moderate size	Suitable for static small networks
Metaheuristic-based (MCD)	Balanced weighted exploration; supports multi-objective approaches	Requires parameter tuning; sometimes converges slowly	Moderate: minutes to hours depending on generations	Can be tuned for better scalability	Efficient for medium to large networks
Hybrid-based (HCD) [90]	Integrates strengths; robust results	Complex to integrate; computationally expensive	Slower: hours to days depending on design	Moderate to high scalability	Useful when robustness is priority
Deep Learning-based (DCD) [91]	Learns complex patterns; scalable after training	Requires large data; training with high cost	High: hours to days for training; fast inference	High scalability but resource intensive	Effective for large, rich networks

6 Open Research Area

The development of network representation learning provides an efficient alternative for mining and analyzing network structures. Network representation learning seeks to derive a low-dimensional representation of the original networks, which can be achieved using deep learning techniques; hence, it facilitates the successful execution of various network research tasks. The importance of deep learning in discovering dynamic communities has emerged due to its high ability to analyze complex patterns of nodes, understand their dynamic interactions, predict the behavior of nodes within communities, and analyze extensive, and complex data. Recently, numerous research studies and review articles have made significant strides in utilizing deep learning methods to effectively address the problem of CD. Although there is a growing interest in this study area, the design of such methods remains open to further development. In general, the utilization of deep learning in the detection of dynamic community structures encounters the following challenges:

1. Deep learning models are dependent on complicated architectural designs, making them challenging to understand and analyze. This complexity affects the evaluation and fixing of errors, thereby affecting the precision of community detection.

2. Instability challenge for dynamic network during deep learning training: In the training of dynamic communities, the problem in learning a constant and steady representation of the network that changes over time is known as network instability. This instability has appeared due to the change of topological dynamics and node interaction in the network; this poses several issues:

A- Temporal variability and non-stationarity: The structure of the network changes over time, resulting in a repeated shift in the distribution of training data. This lack of stationarity is problematic for deep models, as it is hard to converge to a stable representation or community affiliation. Models that are trained on previous images can become inaccurate quickly because new connections or nodes are added or removed.

B- Noise and incomplete data: Dynamic networks often suffer from noise, insufficient communication, or faulty connections. These connections may distort the intended meaning of deep learning during training, leading to the acquisition of poor-quality information.

C- Overfitting to transient patterns: Deep models may be unable to adapt to short- term changes or noise in the network, which would lead to a poor grasp of patterns for the future.

D- Computational complexity and scalability: Training deep models on large-scale graphs that evolve is computationally expensive and may necessitate frequent retraining to adapt to new situations, which is challenging in practice.

E- The absence of real ground and the difficulties of evaluation: Dynamic community detection is often performed without supervision, which makes it difficult to assess the consistency of the model's stability and performance of the model over time.

To address this challenge, the researchers [86,91–93] have made additional efforts to enhance the stability of communities in dynamic networks.

3. The issue of overfitting has emerged. Models can become overly dependent on training data, resulting in inadequate performance with new or unfamiliar data. This may negatively influence the model's ability to detect communities effectively.

Consequently, despite the recent adoption of deep learning in the community detection domain, the demand for hybrid deep learning and opportunities for cross-fertilization between deep learning and (heuristic, metaheuristic, and hybrid metaheuristic) algorithms can arise and persistently escalate. The primary motivation stems from the realization that providing comprehensive models, in terms of hybridization, may play an essential role in utilizing the advantages and overcoming the limitations of using a single technique from other classes. The following subsections aim to present researchers with innovative design challenges for hybrid deep learning frameworks, which have still received minimal attention or remain unexplored.

6.1 Heuristic Deep Learning Based Community Detection (HDLCD)

The primary motivation for researchers to employ heuristic deep learning-based community detection (HDLCD) comes from deep learning's abilities to enhance the representation of dynamic networks and extract hidden data patterns. This procedure enhances the precision of identifying dynamic communities using heuristic algorithms. This integration represents a promising direction. However, this integration faces many challenges:

1. Large amounts of high-quality data are necessary for practical model training, and the complex nature of these models requires significant computational resources.

2. Complex models often store training data rather than extracting general patterns, resulting in poor efficiency when processing novel data.

3. Deep learning algorithms operate as black boxes, making it difficult to comprehend their decision-making processes.

4. Data bias can lead to unfair results, necessitating careful control to ensure the fairness of the models constructed.

However, merging heuristics algorithms with methods based on deep learning provides several benefits:

1- Improved feature extraction: Heuristics enable feasible feature optimization and support heuristic feature extraction within unguided, pre-developed deep network learning.

2- Provide better accuracy: Combining heuristics and deep learning enhances model adaptability to various datasets, thereby improving the classification and prediction performance of the model.

3- Create the model straight: The heuristic helps to make optimal adjustments to the parameters and select how the model should be created, halting extended trial and error.

6.2 Metaheuristic Deep Learning Based Community Detection (MDLCD)

The integration of deep learning with metaheuristic algorithms represents a promising direction in optimizing the efficiency of both. Deep learning can facilitate the collection of significant, high-dimensional representations of nodes and edges within a social network, capturing complex interactions that conventional features may miss. Metaheuristic algorithms can subsequently use the extracted features as input to optimize single or multiple objectives. This combination offers several advantages that can improve the detection of dynamic networks in future research.

1- Improved precision: This integration improves the ability of deep learning to accurately identify complex non-linear patterns. After that, metaheuristic optimization is applied to enhance both the precision and the efficiency of community detection.

2- Enhanced efficiency: Metaheuristic algorithms can optimize the deep learning training process, potentially reducing computational running costs and accelerating convergence rates.

3- Flexibility in objective management: Metaheuristic-based optimization facilitates adaptable objective formulations in deep learning-driven community discovery, therefore enhancing the model's alignment with specific requirements and limitations.

Despite the advantages mentioned above, this hybridization faces two challenges:

1- Computational complexity: The integration of metaheuristics with deep learning complicates computation due to dynamic inputs. These algorithms frequently deal with dynamic inputs that change throughout the solution process, requiring continuous updates to both the learning model and metaheuristic parameters, which results in added computing costs.

2- Complexity of model tuning: This requires balancing between multi-objective optimization and deep learning parameter optimization, which necessitates complex balancing methodologies in both metaheuristics and deep learning.

However, there are several challenges related to merging metaheuristic algorithms with deep learning methods:

1- Computational Complexity: Metaheuristic algorithms can be very computationally expensive when applied to deep learning on large-scale data.

2- Parameter Setting Difficulty: Setting of parameters for both deep learning and metaheuristics is challenging to achieve for good performance.

3- Issues in Convergence and Stability: There may also be issues with the stability of the training process and convergence to optimal solutions, as metaheuristics are stochastic.

The combination of deep learning with metaheuristic algorithms represents a growing approach that holds significant promise in improving the detection of dynamic networks. As academics continue to explore this field, we can expect creative solutions that enhance the performance of both. In addition, the applicability of these mixtures expanded to encompass more complex networks.

6.3 Hybrid Metaheuristic Deep Learning Based Community Detection (HMDLCD)

This subsection highlights the expected significance of hybrid metaheuristic deep learning-based community detection (HMDLCD) algorithms as a robust framework to understand the interaction between HCD, MCD algorithms, and deep learning. This is achieved by capturing hidden patterns and relationships in complex dynamic networks using different deep learning methods.

On the other hand, deep learning enhances the performance of heuristic and metaheuristic methods. Combining deep learning with hybrid frameworks facilitates enhanced feature extraction from complex networks. This feature enhances the understanding of the underlying, leading to improved data on the hybrid algorithms' performance in detecting communities over time. We expect that deep learning will have an effective role in improving hybrid algorithms, as shown below:

1- The combination of two heuristic algorithms: Deep learning can enhance heuristic algorithms by producing more efficient features, enabling conventional methods such as K-Means or Louvain to perform effectively on more robust data, resulting in improved community detection and decreased computational complexity.

2- The combination of two metaheuristic algorithms: Combining deep learning with metaheuristic algorithms, such as genetic algorithms or particle swarm optimization, enhances their exploration efficiency. Deep learning methods can facilitate the search process by providing knowledge about the structure of the network, thus improving the convergence rates and the quality of the solutions.

3- Combining heuristic and metaheuristic algorithms: The combination of deep learning with heuristic and metaheuristic approaches has a mutually beneficial effect. Deep learning can preprocess data to identify local structures that are expected, while heuristics effectively improve the speed of metaheuristic algorithms. This blend creates a community that identifies the most comprehensive and efficient solutions within a dynamic network. The collaboration of these three methods will produce robust and scalable solutions that can efficiently tackle the difficulties of detecting the dynamic structure of communities.

This study identifies significant gaps and promising areas for further analysis, synthesizing the limitations of prior approaches with the potential offered by combining hybrid metaheuristics and deep learning. For instance,

1- Dimensionality curse in deep models: The use of metaheuristic dimension reduction methods [94].

2- Design of hybrid frameworks that adapt and combine the strengths of different algorithms.

3- Developing community detection methods that are scalable, interpretable, and robust, and remain dynamic over time and for all community applications.

7 Conclusion

The community detection problem has been determined to be NP-hard and remains an active research topic. The purpose of this paper is to capture the nature of dynamic communities and assess the effectiveness of algorithms used to detect them. This paper focuses on the current gaps to detecting communities that change over time, its challenges, and provides a comprehensive review of dynamic community detection algorithms by dividing it into four categorize: heuristic, metaheuristic, deep learning and hybrid based community detection to explain the differences between them in the efficiency of understanding and identifying the behavior of the complex networks. This study also highlighted community detection approaches that are divided into two categories according to the kind of dynamic network: (snapshot based (independent) and continuous(dependent)), to provide a critical view about the importance of choosing the suitable approach to specifying the nature of the evolution of communities over time and requirements of these networks to analyze it. In the future, this review paper suggests utilizing deep learning in conjunction with metaheuristic and hybrid metaheuristic algorithms to overcome the main challenges mentioned in this study, thereby enhancing the detection of dynamic communities and providing more accurate solutions.

Acknowledgement: Not applicable.

Funding Statement: No specific funding for this research.

Author Contributions: All authors contributed equally to the conception and writing. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Ma L, Gong M, Liu J, Cai Q, Jiao L. Multi-level learning based memetic algorithm for community detection. Appl Soft Comput. 2014;19:121–33. doi:10.1016/j.asoc.2014.02.003.
2. Zhao X, Li Y, Qu Z. A density-based clustering model for community detection in complex networks. AIP Conf Proc. 2018;1955(1):040161. doi:10.1063/1.5033825.
3. Pizzuti C. Ga-net: a genetic algorithm for community detection in social networks. In: International Conference on Parallel Problem Solving from Nature; 2008 Sep 13–17; Dortmund, Germany. p. 1081–90. doi:10.1007/978-3-540-87700-4_107.
4. Abdulrahman MM, Abbood AD, Attea BA. The influence of NMI against modularity in community detection problem: a case study for unsigned and signed networks. Iraqi J Sci. 2021;62(6):2064–81. doi:10.24996/ijs.2021.62.6.32.
5. Ibraheem SF, Al-Sarray B. A review of community detection based on modularity optimization in complex networks. Iraqi J Sci. 2024;65(5):2775–93. doi:10.24996/ijs.2024.65.5.34.
6. Hopcroft J, Khan O, Kulis B, Selman B. Tracking evolving communities in large linked networks. Proc Nat Acad Sci. 2004;101(suppl_1):5249–53. doi:10.1073/pnas.0307750100.
7. Khoder HS, Attea BA. Community tracking in time evolving networks: an evolutionary multi-objective approach. Iraqi J Sci. 2016;57(4A):2539–48.

8. Attea BA, Khoder HS. A new multi-objective evolutionary framework for community mining in dynamic social networks. Swarm Evol Comput. 2016;31:90–109. doi:10.1016/j.swevo.2016.09.001.
9. Attea BA, Abbood AD, Hasan AA, Pizzuti C, Al-Ani M, Özdemir S, et al. A review of heuristics and metaheuristics for community detection in complex networks: current usage, emerging development and future directions. Swarm Evol Comput. 2021;63:100885. doi:10.1016/j.swevo.2021.100885.
10. Rossetti G, Cazabet R. Community discovery in dynamic networks: a survey. ACM Comput Surv (CSUR). 2018;51(2):1–37. doi:10.1145/3172867.
11. Dakiche N, Tayeb FB-S, Slimani Y, Benatchba K. Tracking community evolution in social networks: a survey. Inform Process Manage. 2019;56(3):1084–102.
12. Amin F, Choi J-G, Choi GS. Advanced community identification model for social networks. Comput Mater Contin. 2021;69:1687–707.
13. Palla G, Barabási A-L, Vicsek T. Quantifying social group evolution. Nature. 2007;446(7136):664–7. doi:10.1038/nature05670.
14. Cazabet R, Rossetti G, Amblard F. Dynamic community detection. In: Encyclopedia of social network analysis and mining. New York, NY, USA: Springer; 2017. p. 1–10. doi:10.1007/978-1-4614-7163-9_383-1.
15. Yu G, Ma L, Wang X, Du W, Du W, Jin Y. Towards fairness-aware multi-objective optimization. Complex Intell Syst. 2025;11(1):50.
16. Danon L, Diaz-Guilera A, Duch J, Arenas A. Comparing community structure identification. J Stat Mech Theory Exp. 2005;2005(9):P09008. doi:10.1088/1742-5468/2005/09/p09008.
17. Newman MEJ. Fast algorithm for detecting community structure in networks. Phys Rev E. 2004;69(6):066133. doi:10.1103/physreve.69.066133.
18. Newman MEJ, Girvan M. Finding and evaluating community structure in networks. Phys Rev E. 2004;69(2):026113. doi:10.1103/physreve.69.026113.
19. Fortunato S, Barthelemy M. Resolution limit in community detection. Proc Nat Acad Sci. 2007;104(1):36–41. doi:10.1073/pnas.0605965104.
20. Zhou X, Liu Y, Li B, Li H. A multiobjective discrete cuckoo search algorithm for community detection in dynamic networks. Soft Comput. 2017;21:6641–52. doi:10.1007/s00500-016-2213-z.
21. Messaoudi I, Kamel N. A multi-objective bat algorithm for community detection on dynamic social networks. Appl Intell. 2019;49:2119–36. doi:10.1007/s10489-018-1386-9.
22. Mishra S, Singh SS, Mishra S, Biswas B. Multi-objective based unbiased community identification in dynamic social networks. Comput Commun. 2024;214:18–32. doi:10.1016/j.comcom.2023.11.021.
23. Brandes U, Gaertler M, Wagner D. Experiments on graph clustering algorithms. In: Algorithms—ESA 2003. ESA 2003. Berlin/Heidelberg, Germany: Springer; 2003. p. 568–79. doi:10.1007/978-3-540-39658-1_52.
24. Alotaibi N, Rhouma D. A review on community structures detection in time evolving social networks. J King Saud Univ-Comput Inf Sci. 2022;34(8):5646–62. doi:10.1016/j.jksuci.2021.08.016.
25. Yang B, Huang T, Li X. A time-series approach to measuring node similarity in networks and its application to community detection. Phys Lett A. 2019;383(30):125870. doi:10.1016/j.physleta.2019.125870.
26. Kumar R, Novak J, Tomkins A. Structure and evolution of online social networks. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2006 Aug 20–23; Philadelphia, PA, USA. p. 611–7.
27. Chakrabarti D, Kumar R, Tomkins A. Evolutionary clustering. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2006 Aug 20–23; Philadelphia, PA, USA. p. 554–60.
28. Chi Y, Song X, Zhou D, Hino K, Tseng BL. Evolutionary spectral clustering by incorporating temporal smoothness. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2007 Aug 12–15; San Jose, CA, USA. p. 153–62.
29. Lin Y-R, Chi Y, Zhu S, Sundaram H, Tseng BL. FacetNet: a framework for analyzing communities and their evolutions in dynamic networks. In: Proceedings of the 17th International Conference on World Wide Web; 2008 Apr 21–25; Beijing, China. p. 685–94.

30. Asur S, Parthasarathy S, Ucar D. An event-based framework for characterizing the evolutionary behavior of interaction graphs. ACM Trans Knowl Discov Data (TKDD). 2009;3(4):1–36. doi:10.1145/1631162.1631164.
31. Kim M-S, Han J. A particle-and-density based evolutionary clustering method for dynamic networks. Proc VLDB Endow. 2009;2(1):622–33. doi:10.14778/1687627.1687698.
32. Xu KS, Kliger M, Hero III AO. Adaptive evolutionary clustering. Data Min Knowl Discov. 2014;28(2):304–36. doi:10.1007/s10618-012-0302-x.
33. Nguyen NP, Dinh TN, Shen Y, Thai MT. Dynamic social community detection and its applications. PLoS One. 2014;9(4):e91431. doi:10.1371/journal.pone.0091431.
34. Xu M, Li Y, Li R, Zou F, Gu X. EADP: an extended adaptive density peaks clustering for overlapping community detection in social networks. Neurocomputing. 2019;337:287–302. doi:10.1016/j.neucom.2019.01.074.
35. Li T, Wang W, Wu X, Wu H, Jiao P, Yu Y. Exploring the transition behavior of nodes in temporal networks based on dynamic community detection. Future Gener Comput Syst. 2020;107:458–68. doi:10.1016/j.future.2020.02.006.
36. Kilic BU, Muldoon SF. Skeleton coupling: a novel interlayer mapping of community evolution in temporal networks. arXiv:2301.10860. 2023.
37. Xiong Z, Zeng M, Xu F, Deng M, Zhang X, Zhou B, et al. Simple-based dynamic decentralized community detection algorithm in socially aware networks. Heliyon. 2024;10(5):e26965. doi:10.1016/j.heliyon.2024.e26965.
38. Kumar M, Mishra S, Singh SS, Biswas B. Community-enhanced link prediction in dynamic networks. ACM Trans Web. 2024;18(2):1–32. doi:10.1145/3580513.
39. Sahu S, Kothapalli K, Banerjee DS. Parallel multicore algorithms for community detection in dynamic graphs. Int J Netw Comput. 2025;15(1):2–31. doi:10.15803/ijnc.15.1_2.
40. Djurdjevac Conrad N, Tonello E, Zonker J, Siebert H. Detection of dynamic communities in temporal networks with sparse data. Appl Netw Sci. 2025;10(1):1. doi:10.1007/s41109-024-00687-3.
41. Bartz-Beielstein T, Branke J, Mehnen J, Mersmann O. Evolutionary algorithms. Wiley Interdiscip Rev Data Min Knowl Discov. 2014;4(3):178–95. doi:10.1002/widm.1124.
42. Abbood AD, Hasan AA, Bara'a AA. Pearson coefficient matrix for studying the correlation of community detection scores in multi-objective evolutionary algorithm. Periodicals Eng Nat Sci (PEN). 2021;9(3):796–807. doi:10.21533/pen.v9i3.2284.
43. Guerrero M, Gil C, Montoya FG, Alcayde A, Banos R. Multi-objective evolutionary algorithms to find community structures in large networks. Mathematics. 2020;8(11):2048. doi:10.3390/math8112048.
44. Abbood AD, Attea BA, Hasan AA, Everson RM, Pizzuti C. Community detection model for dynamic networks based on hidden Markov model and evolutionary algorithm. Artif Intell Rev. 2023;56:9665–97. doi:10.1007/s10462-022-10383-2.
45. Sun Y, Sun X, Liu Z, Cao Y, Yang J. Core node knowledge based multi-objective particle swarm optimization for dynamic community detection. Comput Ind Eng. 2023;175:108843. doi:10.1016/j.cie.2022.108843.
46. Folino F, Pizzuti C. A multiobjective and evolutionary clustering method for dynamic networks. In: 2010 International Conference on Advances in Social Networks Analysis and Mining; 2010 Aug 9–11; Odense, Denmark. p. 256–63.
47. Folino F, Pizzuti C. An evolutionary multiobjective approach for community discovery in dynamic networks. IEEE Trans Knowl Data Eng. 2014;26(8):1838–52. doi:10.1109/tkde.2013.131.
48. Ma J, Liu J, Ma W, Gong M, Jiao L. Decomposition-based multiobjective evolutionary algorithm for community detection in dynamic social networks. Sci World J. 2014;2014:402345. doi:10.1155/2014/402345.
49. Zhang Q, Li H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput. 2007;11(6):712–31. doi:10.1109/tevc.2007.892759.
50. Gao C, Chen Z, Li X, Tian Z, Li S, Wang Z. Multiobjective discrete particle swarm optimization for community detection in dynamic networks. Europhys Lett. 2018;122(2):28001. doi:10.1209/0295-5075/122/28001.
51. Wang C, Deng Y, Li X, Chen J, Gao C. Dynamic community detection based on a label-based swarm intelligence. IEEE Access. 2019;7:161641–53. doi:10.1109/access.2019.2951527.
52. Zou J, Lin F, Gao S, Deng G, Zeng W, Alterovitz G. Transfer learning based multi-objective genetic algorithm for dynamic community detection. arXiv:2109.15136. 2021.

53. Wang Y-J, Song J-X, Sun P. Research on dynamic community detection method based on an improved pity beetle algorithm. IEEE Access. 2022;10:43914–33. doi:10.1109/access.2022.3168714.
54. Li W, Zhou X, Yang C, Fan Y, Wang Z, Liu Y. Multi-objective optimization algorithm based on characteristics fusion of dynamic social networks for community discovery. Inf Fusion. 2022;79:110–23. doi:10.1016/j.inffus.2021.10.002.
55. Ma H, Wu K, Wang H, Liu J. Higher-order knowledge transfer for dynamic community detection with great changes. IEEE Trans Evol Comput. 2024;28(1):90–104. doi:10.1109/tevc.2023.3257563.
56. Yu L, Zhao X, Lv M, Zhang J. A consensus community-based spider wasp optimization for dynamic community detection. Mathematics. 2025;13(2):265. doi:10.3390/math13020265.
57. Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D. Defining and identifying communities in networks. Proc Nat Acad Sci. 2004;101(9):2658–63. doi:10.1073/pnas.0400054101.
58. Yang J, Leskovec J. Defining and evaluating network communities based on ground-truth. In: Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics; 2012 Aug 12–16; Beijing, China. p. 1–8.
59. Su X, Xue S, Liu F, Wu J, Yang J, Zhou C, et al. A comprehensive survey on community detection with deep learning. IEEE Trans Neural Netw Learn Syst. 2024;35(4):4682–702. doi:10.1109/tnnls.2021.3137396.
60. Sankar A, Wu Y, Gou L, Zhang W, Yang H. Dysat: deep neural representation learning on dynamic graphs via self-attention networks. In: Proceedings of the 13th International Conference on Web Search and Data Mining; 2020 Feb 3–7; Houston, TX, USA. p. 519–527.
61. Rossi E, Chamberlain B, Frasca F, Eynard D, Monti F, Bronstein M. Temporal graph networks for deep learning on dynamic graphs. arXiv:2006.10637. 2020.
62. Costa AR. Adaptive model to community detection in dynamic social networks. Brasília, Brazil: Universidade de Brasília; 2024.
63. La Malfa E, La Malfa G, Nicosia G, Latora V. Characterizing learning dynamics of deep neural networks via complex networks. In: 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI); 2021 Nov 1–3; Washington, DC, USA. p. 344–51.
64. Tahabi FM, Luo X. Dynamicg2b: dynamic node classification with layered graph neural networks and BiLSTM. Int FLAIRS Conf Proc. 2023;36(1).
65. Zhang S, Xiong Y, Zhang Y, Sun Y, Chen X, Jiao Y, et al. Rdgsl: dynamic graph representation learning with structure learning. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management; 2023 Oct 21–25; Birmingham, UK. p. 3174–83.
66. Dumpis M, Navakauskas D. Accuracy vs complexity: a small scale dynamic neural networks case. Baltic J Mod Comput. 2024;12(3):327–48.
67. Abdulrahman MM, Abood AD, Attea BA. An enhanced multi-objective evolutionary algorithm with decomposition for signed community detection problem. In: 2020 2nd Annual International Conference on Information and Sciences (AiCIS); 2020 Nov 24–25; Fallujah, Iraq. p. 45–50.
68. Saeed HS, Abbood AD. Enhanced evolutionary algorithm for dynamic community detection using a vulnerable node reassignment-based mutation. Int J Intell Eng Syst. 2025;18(7):388–411.
69. Zeng X, Wang W, Chen C, Yen GG. A consensus community-based particle swarm optimization for dynamic community detection. IEEE Trans Cybern. 2019;50(6):2502–13. doi:10.1109/tcyb.2019.2938895.
70. Lancichinetti A, Fortunato S. Consensus clustering in complex networks. Sci Rep. 2012;2(1):336. doi:10.1038/srep00336.
71. Mandaglio D, Amelio A, Tagarelli A. Consensus community detection in multilayer networks using parameter-free graph pruning. In: Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018. Cham, Switzerland: Springer; 2018. p. 193–205.
72. Besharatnia F, Talebpour A, Aliakbary S. An improved grey wolves optimization algorithm for dynamic community detection and data clustering. Appl Artif Intell. 2022;36(1):2012000. doi:10.1080/08839514.2021.2012000.
73. Ranjkesh S, Masoumi B, Hashemi SM. A novel robust memetic algorithm for dynamic community structures detection in complex networks. World Wide Web. 2024;27(1):3. doi:10.1007/s11280-024-01238-7.

74. Gong M-G, Zhang L-J, Ma J-J, Jiao L-C. Community detection in dynamic social networks based on multiobjective immune algorithm. J Comput Sci Technol. 2012;27(3):455–67. doi:10.1007/s11390-012-1235-y.
75. Chen G, Wang Y, Wei J. A new multiobjective evolutionary algorithm for community detection in dynamic complex networks. Math Problems Eng. 2013;2013:161670.
76. Niu X, Si W, Wu CQ. A label-based evolutionary computing approach to dynamic community detection. Comput Commun. 2017;108:110–22. doi:10.1016/j.comcom.2017.04.009.
77. Panizo-LLedot A, Bello-Orgaz G, Camacho D. A multi-objective genetic algorithm for detecting dynamic communities using a local search driven immigrant's scheme. Future Gener Comput Syst. 2020;110:960–75. doi:10.1016/j.future.2019.10.041.
78. Liu F, Wu J, Xue S, Zhou C, Yang J, Sheng Q. Detecting the evolving community structure in dynamic social networks. World Wide Web. 2020;23:715–33. doi:10.1007/s11280-019-00710-z.
79. Besharatnia F, Talebpour AR, Aliakbary S. Metaheuristic multi-objective method to detect communities on dynamic social networks. Int J Comput Intell Syst. 2021;14(1):1356–72.
80. Yin Y, Zhao Y, Li H, Dong X. Multi-objective evolutionary clustering for large-scale dynamic community detection. Inf Sci. 2021;549:269–87. doi:10.1016/j.ins.2020.11.025.
81. Cai L, Zhou J, Wang D. Improving temporal smoothness and snapshot quality in dynamic network community discovery using NOME algorithm. PeerJ Comput Sci. 2023;9:e1477. doi:10.7717/peerj-cs.1477.
82. Wang W, Li Q, Wei W. An adaptive dynamic community detection algorithm based on multi-objective evolutionary clustering. Int J Intell Comput Cybern. 2024;17(1):143–60. doi:10.1108/ijicc-07-2023-0188.
83. Lancichinetti A, Fortunato S. Limits of modularity maximization in community detection. Phys Rev E. 2011;84(6):066122. doi:10.1103/physreve.84.066122.
84. Aref S, Mostajabdaveh M, Chheda H. Heuristic modularity maximization algorithms for community detection rarely return an optimal partition or anything similar. In: International Conference on Computational Science. Cham, Switzerland: Springer; 2023. p. 612–26.
85. Paoletti G, Gioacchini L, Mellia M, Vassio L, Almeida JM. Benchmarking evolutionary community detection algorithms in dynamic networks. arXiv:2312.13784. 2023.
86. Pan Y, Liu X, Yao F, Zhang L, Li W, Wang P. Identification of dynamic networks community by fusing deep learning and evolutionary clustering. Sci Rep. 2024;14(1):23741. doi:10.21203/rs.3.rs-4236904/v1.
87. Wang Z, Wang C, Gao C, Li X, Li X. An evolutionary autoencoder for dynamic community detection. Sci China Inf Sci. 2020;63(11):212205. doi:10.1007/s11432-020-2827-9.
88. Qu S, Du Y, Zhu M, Yuan G, Wang J, Zhang Y, et al. Dynamic community detection based on evolutionary DeepWalk. Appl Sci. 2022;12(22):11464. doi:10.3390/app122211464.
89. Zhang B, Mi Y, Zhang L, Zhang Y, Li M, Zhai Q, et al. Dynamic community detection method of a social network based on node embedding representation. Mathematics. 2022;10(24):4738. doi:10.3390/math10244738.
90. Nassef AM, Abdelkareem MA, Maghrabie HM, Baroutaji A. Hybrid metaheuristic algorithms: a recent comprehensive review with bibliometric analysis. Int J Electr Comput Eng (IJECE). 2024;14(6):7022–35. doi:10.11591/ijece.v14i6.pp7022-7035.
91. Liu F, Xue S, Wu J, Zhou C, Hu W, Paris C, et al. Deep learning for community detection: progress, challenges and opportunities. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 20); 2021 Jan 7–15; Yokohama, Japan. p. 4981–7.
92. Morea F, De Stefano D. Enhancing stability and assessing uncertainty in community detection through a consensus-based approach. arXiv:2408.02959. 2024.
93. Wang H, Zhang Y, Zhao Z, Cai Z, Xia X, Xu X. Less is more: simple yet effective heuristic community detection with graph convolution network. arXiv:2501.12946. 2025.
94. Khosrowshahli, Rahnamayan S, Ombuki-Berman B. Massive dimensions reduction and hybridization with metaheuristics in deep learning. In: 2024 lEEE Canadian Conference on Electrical and Computer Engineering (CCEcE). Kingston, ON, Canada; 2024. p. 469–75.