Chapter 2

Related Work

2.1 Introduction

Social networks have emerged as a web of interactions, constantly evolving and adapting to the dynamic nature of mutual interactions. Digital advancements and online social interactions have catalyzed a paradigm shift in the study of underlying social structures, necessitating a nuanced understanding of the dynamic interplay within these networks. The thesis seeks to unravel these ever-changing landscapes' underlying patterns and structures.Chapter 1 outlines the concept of community detection and associated issues and challenges. It also provides a brief discussion of thesis objectives and contributions. Now, this chapter presents an exploration of related work on community detection. It is divided into three sections. Firstly, dynamic network modelling is explained, followed by a description of related literature, and the last section is dedicated to explanations of various evaluation metrics. The key focus of this chapter are:

- To present a formal definition of dynamic networks and mathematical models.
- To overview the existing community detection literature dynamic network.
- To discuss approach-based taxonomy of the community detection in dynamic networks with in-depth classification of each approach.

- To introduces the evaluation metrics employed for assessing the performance of the proposed algorithm in comparison to state-of-the-art algorithms.

2.2 Dynamic Network

Researchers have shown keen interest in the analyses of complex networks. Along with community detection, link prediction [98] [107], outlier detection [18] [61], frequent pattern identification [109] [46] are some mining problems that are being focused in complex networks. The structure of complex networks is not regular but complex and tends to change over time. It consists of a large number of nodes and their interactions. A real-world network is an example of this. With digital advancements, a large amount of rich data is available for researchers. Here the rich data refers to extra information available along with graphs for example node/edge attribute and associated time stamps.

Earlier static graphs were used by researchers, which only captured structural patterns and cost them the loss of temporal features of networks. Temporal information is an important factor that helps in capturing the evolution of the network over time. The aggregation of temporal information with graph structure coined a new type of network known as dynamic networks.

The next question that arises here is, what is temporal information? The most simple explanation is the time of occurrence of an event in a given graph. However, it may contain time duration of an element of the graph (element refers to node, edge or corresponding attributes). Representation of dynamic graphs is also a challenging task. In this section, we present a formal definition of dynamic graphs and mathematical models.

Definition 2.1 (Dynamic Network)**.** A dynamic network, *DN*, is a complex network that consist of nodes and edges where each element is associated with a temporal information resulting in an evolving network structure.

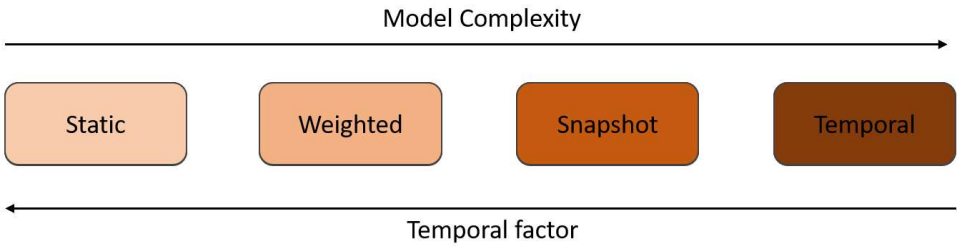

FIGURE 2.1: Dynamic network models in increasing order of temporal granularity

2.2.1 Network Model

Time is an important factor in a dynamic network. The evolution of structural aspects of networks is associated with time stamps. Time provides ordering for various events in a network. Mathematical formulation of such networks is a crucial task in order to study and analyze them. Rossetti and Cazabet [137] have provided a representation of increasing temporal granularity in network models along with model complexities. The leftmost tile represents static graphs with no temporal information, followed by attributed graphs, which contain node/edge weights. Snapshots and temporal tiles show graphs with temporal information. Figure 2.1 shows network tiles with an increase in complexity and temporal granularity. These two models are widely used by authors working with dynamic networks. The following section defines and discusses these two models in detail.

2.2.1.1 Temporal Dynamic Network

This model considers *DN* as a stream of events observed over time where each event can add or remove an edge or a node. Dynamics of network allow structure of network to change or evolve. These changes can be further categorized into three groups depending on the amount of temporal information associated with events. Before describing these groups in more detail, we present a mathematical definition of the temporal network model.

Definition 2.2 (Temporal Dynamic Network)**.** A temporal network model, TNM, is a series of events $TNM = \{E_1, E_2, ...\}$ where each event E_i is represented as a set of triplet $\{< u_i, v_i, T_i >\}$. u_i and v_i are nodes on which event has occurred and T_i is the respective temporal information of that event.

Time stamp	Event	Set	Remarks
0	E_0	$\{< A, null, T_0 >\}$	Addition of node A
1	E_1	$\{< B, null, T_1 >, < A, B, T_1 >\}$	Addition of node B and edge AB
2	E_2	$\{< C, null, T_2 >, < B, C, T_2 >\}$	Addition of node C and edge BC
3	E_3	$\{< D, null, T_3 >, < B, D, T_3 >\}$	Addition of node D and edge BD
4	E_4	$\{< C, D, T_4 >\}$	Addition of edge CD

TABLE 2.1: Event table for temporal network model

Based on the temporal information T mentioned in definition 2.2, TNM can be further categorized in three different types as mentioned before. These three groups are discussed below:

- For a given pair of nodes, an edge might be added or deleted between them. Consider symbols $+$ and $-$ represent addition and deletion operations respectively. For a given event E_i, T_i is a pair $(+, t_i)$ or $(-, t_i)$ denoting an action occurring at time t_i.
- There are networks which grows in an incremental fashion i.e. only new edges are introduced in current network. For such network model T_i will signify the time stamp of occurring of event.
- A much more complicated representation will incorporate the information of time stamp as well as duration of that event. T_i will be a pair $(t_i, \delta t_i)$, where former is time stamp for E_i and latter is the time duration for which that event will hold in the network.

Consider a DN, G_t, corresponding $TNM = \{E_0, E_1, E_2, E_3, E_4\}$ consisting of five events occurred at respective time steps is shown in figure 2.2. Events associated with each time stamps is tabulated in table 2.1.

2.2.1.2 Snapshot Dynamic Network

Instead of observing all events, this network model, SNM, captures network snapshots. Given a time period Δt, it maintains a series of graphs observed periodically over time.

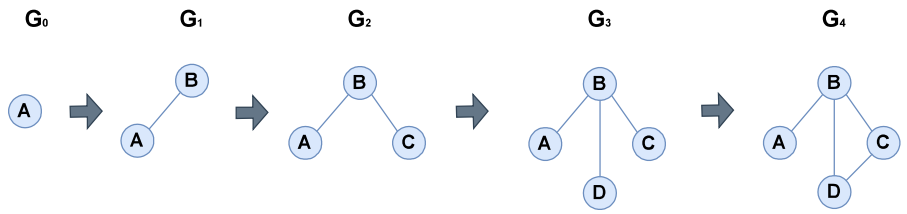

FIGURE 2.2: Temporal network model for a dynamic network

Each graph G_i represents state of network at time t_i and $\Delta t = t_{i+1} – t_i$. One major challenge for *SNM* is to map elements of one snapshot to the next. We need to consider some mapping techniques in order to observe the overall evolution of the network.

Definition 2.3. Given a *DN*, *SNM* is a set of graph snapshots observed over time for a Δt time period difference. $SNM = \{G_1, G_2, ...\}$ where the time difference between two consecutive snapshots is Δt.

Consider a DN, G_t, corresponding $SNM = \{G_0, G_1, ..., G_t\}$ consisting of t snapshots at respective time stamps is shown in figure 2.3.

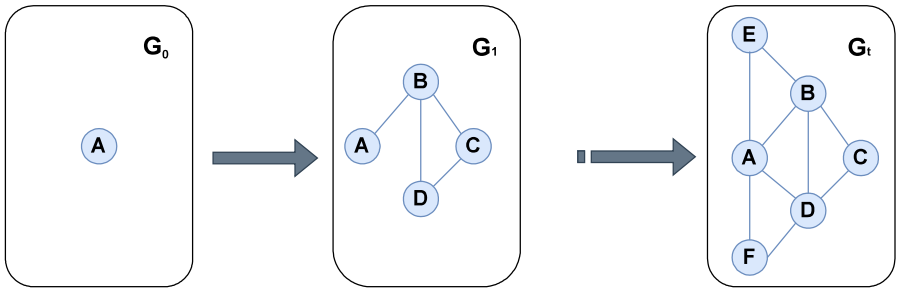

FIGURE 2.3: Snapshot network model for a dynamic network

2.3 Related Literature

Complex networks are often represented as structures that change over time, giving them power to not only the structural but also temporal patterns within networks. Community detection aims to extract meaningful substructures from these networks. This section presents a concise discussion of available literature on community detection. As the thesis focuses on dynamic networks, the discussion pivots around it. Based on approaches to solve

the community detection problem, existing literature can be broadly categorized into three groups: Graph representation-based, Objective-based and Technique-based. Figure 2.4 presents the proposed taxonomy of community detection algorithms in dynamic networks.

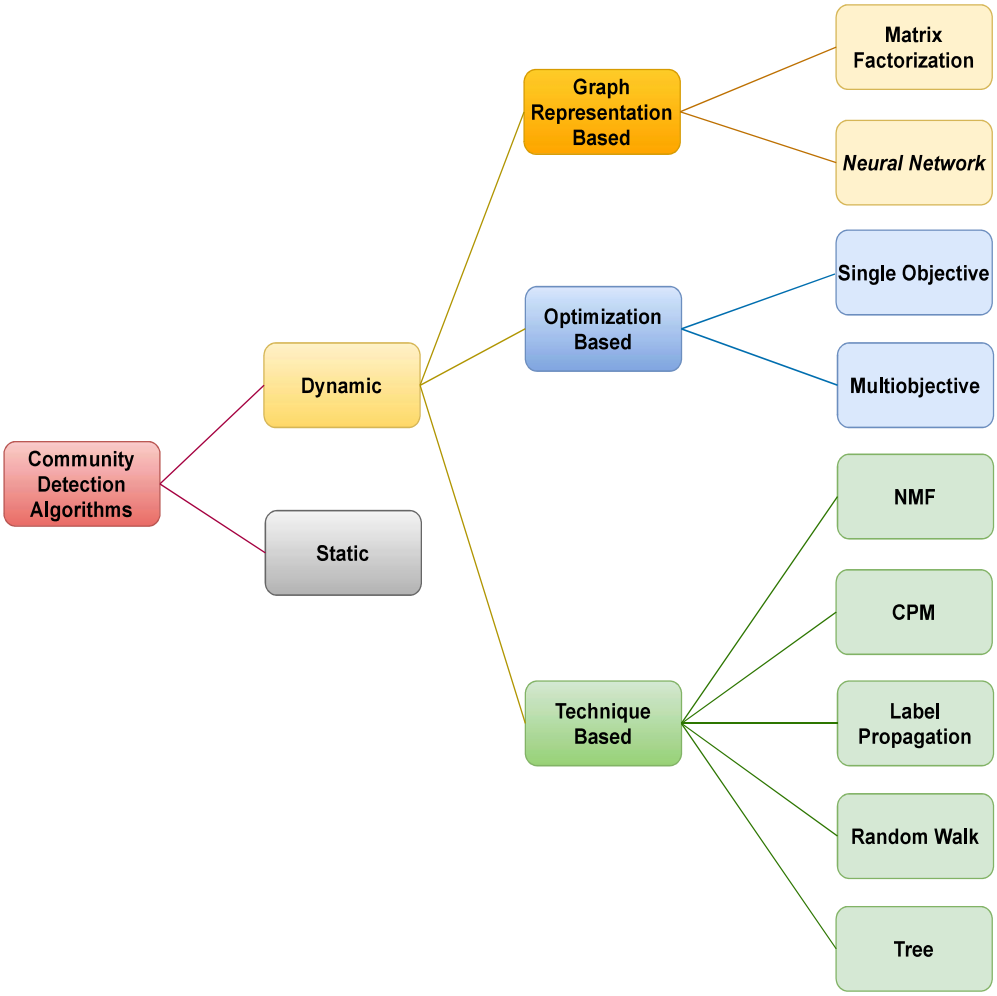

FIGURE 2.4: Dynamic network models in increasing order of temporal granularity

2.3.1 Graph Representation Based

Network graphs often arise from complex real-world systems and are, hence, represented as high-dimensional data. The identification of communities depends significantly on the representation of graphs, especially when considering evolving networks.

High-dimensional data computation increases the space and time complexity of algorithms, resulting in the need for techniques to embed it in lower dimensions. This embedding is used to encode topological and semantic information of a network. It aims to learn the embedding of a network in a d-dimensional space such that the representation exhibits crucial information and can be used to reconstruct the network or to be used as an input for other network analysis tasks like node classification or clustering. Many approaches are found in the literature for generating dynamic graphs embedding, which learn network representation over time, facing challenges related to variation over time, the granularity of temporal information and learning temporal dependencies. Literature shows numerous approaches for generating embeddings of dynamic graphs, which can be broadly classified into three categories: matrix factorization-based, neural network and random sampling-based.

2.3.1.1 Matrix Factorization Based

This approach uses matrices to represents network evolution and hence utilizes the structural correlation among nodes which appears with time. Similarity matrix of a given graph is decomposed into multiple matrices in order to extract meaningful information with lower computational cost. Various node similarity measures are used to generate initial similarity matrix. This approach is also associated with a loss function which aims to minimize the temporal cost between consecutive embeddings at different time stamps also referred as temporal smoothening. Authors in [88] proposed DANE uses Laplacian matrix and it's eigen decomposition to regenerate time bound adjacency matrix of graphs. A popular approach to capture node's proximity is found in literature which takes an inner product function among embeddings for it [43] [183] [140] [176]. Incorporating temporal information into representation is the prime concern when handling dynamic graphs because it is important for any embedding to amalgamate structure as well as time-dependent information in order to represent a dynamic system. DynamicTriad [191] attempts to minimize the temporal shift by minimizing the loss function. Loss function represent the dissimilarity between two embeddings, hence lower the value higher the embedding similarity at consecutive time steps. Ferreira et al. [43] uses similar approach in a study of a political network. Matrix perturbation theory is also used by authors to update

the embedding instead of recalculating it at each time stamp. [183], [88] and [154] are some examples of this approach.

2.3.1.2 Neural Network Based

Neural network is a powerful tool which is proven to be well-liked by researchers of different field including natural language processing, computer vision and others. The exceptional performance of these models also catches attention of researchers working on graph data. Graph neural network (GNN) [145] and graph convolutional network (GCN) [72] are two popularly used deep learning models for graphs. Their applicability is limited to static graphs as they fail to capture the time dependence of networks. To prevail this shortcoming, GNN and GCN are often supported by a memory unit which is a special neural network designed for sequential data. Recurrent neural network (RNN), long short-term memory (LSTM) [60] [29] and gated recurrent unit (GRU) [30] are used as memory unit to handle the temporal information. The information of correlation of different time stamps is manged by these units. Graph attention network (GAT) [160] is also emerged as a model which utilizes attention mechanism [9] for handling dynamic graphs. GNN primarily exploits message passing mechanism and an aggregation function to capture the neighbouhood of given graph whereas GCN relies on convolutional operation and kernels to capture a node's proximity. Inductive learning-based approaches like GraphSAGE [56] and Gated Graph Neural Networks (GGNNs) [92] uses GRU for temporal settings. Some generative models like generative adversarial networks (GAN) [50] and variational autoencoders (VAE) [71] are also found in literature of dynamic graph based neural networks. Models like [73] and [162] target to learn data distribution of given graph

2.3.2 Optimization Based

An optimization problem aims to find best solution from all feasible solutions by either maximizing or minimizing objective function(s). Methods in this category uses well known optimization algorithms to optimize desired objective functions. There are several metrics proposed in literature to reflect the properties of communities. Modularity and NMI are two famous metrics used in literature. Modularity is used to measure the quality of communities and is widely accepted by researchers for optimization problem. As the

TABLE 2.2: Relevant literature for graph representation based community detection in dynamic network

Algorithm	Authors	Base approach	Network Model		No. of Communities		Limitation
			Temporal	Snapshot	Required	Not required	
EvolveGCN [125]	Pareja et al.	GCN, LSTM, RNN	✗	✓	✓	✗	The effectiveness of EvolveGCN in capturing temporal dependencies and evolving structural patterns may vary depending on the specific characteristics of the dynamic graphs and the chosen evolutionary mechanisms.
RgCNN [113]	Narayan and Roe	CNN, RNN	✗	✓	✓	✗	While the paper presents a promising approach for learning graph dynamics using deep neural networks, its applicability and effectiveness may be limited by factors such as the complexity and size of the dynamic graphs, as well as the scalability of the neural network models. Additionally, the performance of the deep neural network approach may be influenced by the availa bility and quality of training data, as well as the chosen network architecture and learning algorithms.
GAT [160]	Velickovic et al.	Attention mechanishm	✗	✓	✓	✗	Algorithm's effectiveness may be limited by factors such as the complexity of graph structures, scalability issues with large graphs, and the need for extensive computational resources for training and inference. Additionally, the performance of GATs may vary depen- ding on the specific characteristics of the graph data, the choice of hyperparameters, and the quality of the attention mechanisms employed
TGN [139]	Rossi et al.	GNN, Attention mechanism	✓	✗	✓	✗	Designing effective attention mechanisms for different types of graphs and tasks can be challenging and may require empirical exploration and it require large amounts of labeled data for training. It may struggle to generalize to unseen graphs or graphs with different structures from those seen during training.
DANE [88]	Li et al.	Laplacian matrix and Eigen decomposition	✗	✓	✗	✓	The performance of the attributed network embedding approach may depend on the availability and quality of attributed network data, the choice of embedding methods and hyperparameters, and the ability to capture dynamic changes in network structures and node attributes over time.
DBMM [140]	Rossi et al.	Fuzzy logic	✗	✓	✓	✗	Modeling dynamic behavior in large evolving graphs, its applicability and effectiveness may be limited by factors such as the scalability of the proposed models, the complexity of dynamic graph structures, and the com- putational resources required for analysis and inference.

Modularity only captures the quality of network division, NMI is used to manage the temporal cost of network. The limitation of this approach is algorithm outputs a set of representative solutions and the decision maker's responsibility to select the final solution using some domain information or based on subjective preference.

2.3.2.1 Single-objective

Optimization theorem is used by many researchers to detect stable communities. Modularity [115] is an important metric to define quality of communities which is maximized by algorithms for solving community detection problem [63] [32]. Quick community adaptation algorithm (QCA) [117] proposes various actions to find communities on addition/deletion of edges in the network. Cluster based approaches [82] [94] are also utilized in literature where clusters are treated as communities. Dependency on network density and limiting resolution capabilities are some concerns associated with this approach. Nature inspired optimization algorithms are used by authors to identify protein complexes in dynamic protein -protein interaction network. Zhao et al in [184] proposed an improvised version of clustering using Cuckoo search optimization. A bi-clustering algorithm is presented by Lakizadeh and Jalili [80] where they find clusters within each cluster to analyze substructures associated with each cluster and corresponding conditions. To reduce the complexity associated with dynamic network Cordeiro et al. [33] proposes local modularity optimization. It optimizes modularity for regions affected by any network changes resulting the remaining network unchanged.

2.3.2.2 Multi-objective

Evolutionary clustering comprises two main goals: firstly, to increase the accuracy of community detection, and secondly, to ensure smooth community structure transitions at subsequent time steps. It is believed that the community structure of any network will not change abruptly in a short period of time. Selecting target weights was a major shortcoming of evolutionary clustering. Multi-objective optimization helps to avoid it and is hence used by many researchers. Multi-objective optimization aims to optimize multiple conflicting objectives simultaneously. Chen et al. [26] proposed a multi-objective optimization-based algorithm, MODTLBO, which uses a decomposition-based approach

for complex networks. To avoid local optimal solutions, it diversifies the population via neighborhood-based mutation. DYNMOGA [45], proposed by Folino and Pizzuti, optimizes modularity and normalized mutual index (NMI) using a genetic algorithm. Modularity represents community quality, while NMI is used to manage the temporal cost of the network. MBBOD is proposed by Zhou et al. in [188] which uses modularity and a normalized mutual index as objective functions, and a biogeography-based optimization is used for community identification. They propose decomposition in their algorithm to optimize two objective functions. Multi-objective genetic optimization is used along with the label propagation approach in [119] for better quality communities. The label propagation algorithm is used for initialization of communities in networks and is also responsible for regulating mutation processes. Bat algorithm-based multi-objective algorithms are used by Zhou et al. in [189] where they have used bat methods in discrete form. The Bat algorithm is a bioinspired algorithm; a modified discrete version of this algorithm is used in this paper. Messaoudi and Kamel in [110] also use the bat algorithm, where they use the mean shift strategy to generate populations. Modularity density and NMI are objective functions that are being optimized by the bat algorithm. A multi-objective optimization algorithm is proposed by Shen et al. in [148], in this paper, a parallel evolutionary optimization approach is applied to a software ecosystem. It is the first algorithm to propose parallel computation in the software ecosystem. The Shen et al. have extracted data from GITHUB to create the network. A swarm optimization-based multi-objective approach is used in [157]. Core nodes are identified by an algorithm using resistant distance, and constant communities are formed with these and their associated nodes, which will be retained over time. Modularity and dynamic community evolution continuity is being optimized by Li et al. in [91] using pareto front. It uses probability fusion at early stage for creating partitions and ensuring fast convergence. Bello et al. in [17] have presented two groups of objective functions where one is internal and other is for external densities. Further they used genetic algorithm to optimize these two objective functions.

The multi-objective optimization approach has the advantage of addressing more than one objective for a problem, and the results obtained from this approach are usually a trade-off among various objectives. Dynamic networks have a time-varying characteristic, which adds an added challenge to the problem. Most of the papers in the literature focus on improving clustering quality and minimizing temporal costs [45][188][119]. Well-known

community quality metric Modularity is widely used by researchers as an objective function for clustering quality. While the temporal cost is being calculated by the normalized mutual index (NMI). Apart from these two, some authors have also proposed novel metrics and used them as objective functions. Ratio cut [148][26], kernel k-mean [148][157], clique number [17], clustering coefficient [17], and others are also used in literature. Various optimization techniques are available in the literature, whereas genetic algorithms [1] [8] and swarm intelligence seem to be preferred by researchers in community detection problems. There are some studies present fuzzy based optimization techniques using different numerical methods [2][3].

2.3.3 Technique Based

The core idea of this category is to exploit graph theory concepts and existing state-of-the-art algorithms to identify communities. Well known methods are adapted and improvised to obtain results in dynamic settings.

2.3.3.1 Clique Percolation Method

The Clique Percolation Method (CPM) operates on the principle of clique formation within subgraphs in which a clique is formed among connected nodes. CPM based approach is introduced by Palla et al. [123] [122] that cliques tend to form internally with high density, while edges connecting different communities do not typically form cliques. A ”K-clique” refers to a complete graph with K vertices. Network is assumed to consists contiguous cliques sharing K-1 nodes among them. Each clique belongs to a specific community, although cliques from different communities may share common nodes. This heuristic aids in identifying overlaps within node clusters. Farkas et al. [41] extended the CPM to analyze weighted, bipartite, and directed graphs. They introduced a concept of threshold and weights of cliques which is computed as a geometric mean of weights of involved edges. This threshold value is selected to be slightly higher than a critical value at which K-clique communities emerge, facilitating the discovery of the most comprehensive clusters. A Sequential Clique Percolation Algorithm (SCP) [78] introduces a rapid implementation of the CPM. SCP initiates from an empty graph and identifies K-clique communities by sequentially adding edges from the graph under examination. Although the complexity of

TABLE 2.3: Relevant literature for objective based community detection in dynamic network

Algorithm	Authors	Base approach	Network Model		No. of Communities		Limitation
			Temporal	Snapshot	Required	Not required	
CCPSO [180]	Zeng et al.	PSO	✗	✓	✗	✓	It is sensitive to parameter settings and initialization, potentially leading to suboptimal results.
DYNMOGA [44]	Folino and Pizzuti	MOP	✗	✓	✓	✗	Optimization could be biased towards an objective whose value undermine the other with higher margin.
MBBOD [188]	Zhou et al.	MOP	✗	✓	✗	✓	It may face challenges in effectively handling the dynamic nature of networks, particularly in scenarios where community structures undergo rapid or abrupt changes.
MODPSO [173]	Yin et al.	PSO	✗	✓	✗	✓	performance may be influenced by its ability to adapt to evolving network characteristics, potentially impacting the accuracy of community detection results.
QCA [117]	Nguyen et al.	SOO	✓	✗	✓	✗	With increase in size of network performance decreases.
ICSC [184]	Zhao et al.	SOO	✗	✓	✗	✓	Weighted information may introduce noise or biases into the prediction process, impacting the accuracy.
BiCAMWI [80]	Lakizadeh and Jalili	SOO	✗	✓	✓	✗	Scalability may be a concern for very large protein interaction networks, potentially limiting its applicability to datasets of extensive size
MODTLBO [26]	Chen et al.	MOP	✗	✓	✓	✗	It may face challenges in handling large-scale networks or datasets due to its computational complexity, potentially limiting its scalability and applicability to real-world scenarios with extensive data.
MS-DYN [110]	Messaoudi and Kamel	MOP	✗	✓	✓	✗	Its performance may be influenced by its ability to effectively adapt to dynamic network characteristics and to handle complex interactions between nodes over time.
PMOEO-DCD [148]	Shen et al.	MOP	✗	✓	✗	✓	Its performance may be influenced by its ability to parallelize computation effectively and to adapt to changing software ecosystem dynamics over time

SCP is proportional to the number of cliques, it outperforms the original CPM in terms of speed.

However, traditional CPM algorithms may fail to cover the entire network, leaving some nodes without any assigned clusters despite their connections to other nodes. To address this issue, Maity et al. [104] proposed an innovative algorithm that extends CPM to ensure every node is assigned to some community. Unassigned nodes are associated with belonging coefficient, which quantifies the strength of a node's association with a specific community [118]. For applying CPM in dynamic networks, a base framework is used by majority of researchers where it is performed in two steps, initially communities are identified using CPM followed by community matching step. Lovain [20] and MOSES [51] are example of this approach. Boudebza et al. [21] have proposed OLCM which uses label propagation along with CPM for identifying communities. LSCPM [13] is an extended version of CPM where they consider graph as a stream.

2.3.3.2 Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) model is presented by Lee and Seung [83] which acts as an analyzing tool for matrices with non-negative elements. NMF is preferred over other factorization because of its clustering [38] and generative capabilities [133]. It factorizes the feature matrix, $X_{n,m} = \{X_1, X_2, \dots\}$ of a network into two matrices, $W_{n,d}$ and $H_{m,d}$ having lower dimension (say d) and whose product well approximate the original matrix, see equation 2.1. The dimension d usually represents number of communities in the network.

$$X \approx WH^T \tag{2.1}$$

The value of W and H can be obtained by formulating them in an optimization problem which minimizes the loss between actual matrix and the approximated one. Mathematically it can be expressed as:

$$minL(W,H) = ||X - WH^T||_F^2 \tag{2.2}$$

where right side of the equation is square of the Frobenius norm. Many researchers have applied NMF in dynamic network and it can be classified in two categories: one which uses

information of only previous time step called as online mode and later uses entire history of network evolution called as offline mode. Authors in [101] presented a semi-supervised algorithm sE-NMF which is an example of online mode and also acted as a base method for many authors. Two improvised versions GrENMF [102] and CrENMF [103] is proposed by exploiting graph regularization for enhanced performance. DBNMF [164], DGR-SNMF [163] and ECGNMF [174] are some more variations of sE-NMF. Whereas [106], [175], [68] are example of offline methods.

2.3.3.3 Random Walk

Random walks offer a viable strategy for detecting clusters within a graph by randomly traversing nodes and merging them in groups in bottom-up paradigm. It can also be extended for weighted graph by considering edges weights in path length [65]. Literature of community detection presents numerous papers of random walk based community detection algorithms for community detection. Nodes belonging to same community have less distance between themselves. The concept of distance between nodes through random walk is proposed by Zhou et al. [186], it is the average number of steps involved in a random walk between two vertices. Proximity in terms of edges suggests potential membership in the same community or cluster. Building upon this concept, Zhou and Lipowsky in [187] introduced biased random walkers, which tend to move towards vertices with the highest number of neighbors relative to the starting node in graphs. Employing Brownian movement, they proposed the ”Netwalk” procedure for detecting communities within this biased random walk framework. Netwalk operates as an agglomerative hierarchical clustering method, where the proximity between vertices indicates their similarity. Walktrap proposed in [130] uses a hierarchical clustering and utilizes value of modularity for dendrogram slicing while incorporating random-walk-based similarities among nodes and clusters. An adaptive random walk sampling (ARWS) proposed by Xin et al. [171] considers the concept of close friends and update only those nodes which are impacted from network change, in this way they claim to minimize the computational overhead caused by network perturbations.

2.3.3.4 Tree Based

Tree representation of a complex network provides interesting and simpler opportunities for analysis. For example, calculation of diameter of a tree is easier than that of a graph. Literature on tree-based methods can be broadly classified into two categories: first, where the graph is transformed into a tree followed by a community detection algorithm, and second, the tree is maintained according to identified communities.

Spanning tree is popularly used in first category. Basuchowdhuri et al.[11] proposed P-SPAT and K-SPAT as a spanning tree with maximum cost based community detection algorithm. Authors' use Prim's and Kruskal algorithms for finding spanning tree, then apply hierarchical clustering. They also extend the work considering modularity maximization. Spanning tree based approach is also considered in [15] using MIN-MAX modularity. Within a community, highly connected nodes are rewarded as oppose of less connected nodes which are penalized in this version of modularity. [54] creates a minimum spanning tree by pruning the tree based on average euclidean distance between nodes of network. Process continues until stable sub-spanning trees are obtained.

Some authors uses tree to save community structure of network. Zappia and Oshlack in [179] presents the importance of tree structure in evaluation of clusters at various levels of resolution. Vehicular Ad-hoc Network(VANET) is a highly dynamic wireless network of vehicles. Probabilistic B-tree based clustering solution for VANET is proposed in [161].

2.3.3.5 Label propagation

Label propagation algorithms (LPA) offer a promising framework for uncovering communities based on local information propagation dynamics. LPA proves to be favorable for community detection problem because they are liner time solvable. Each node is initialized with a label which is further shared among nodes' neighbours. They update their labels to one which is carried by majority neighbours. Process iterates to maximum iterations. Many researchers have extended the basic LPA algorithms to enhance its performance [144][53][181]. LPA based approaches offer versatile and computationally efficient framework for identifying structural patterns in networks. Speaker-listener label propagation (SLAP) [169] is widely used technique for handling

large networks. Many algorithms such as GANXiSw and labelRank [168] uses SLAP for static networks while an improvisation LabelRankT [167] deals with dynamic networks. While presenting promising solutions it exhibit sensitivity to initial conditions [134], scalability issue [126] and biasness towards high degree nodes [141].

2.4 Evaluation Metrics

This section introduces the evaluation metrics employed for assessing the performance of the proposed algorithm in comparison to state-of-the-art algorithms. Evaluation metrics play a crucial role in objectively measuring the effectiveness and efficiency of an algorithm's performance across various tasks or domains. By comparing the proposed algorithm's outcomes against those of established methods, researchers can gain insights into its strengths, weaknesses, and overall competitiveness within the field. These metrics provide a standardized framework for quantitative analysis, enabling rigorous evaluation and meaningful comparisons to guide further algorithmic advancements.

2.4.1 Precision and Recall

Precision and recall are two widely used accuracy metrics in classification tasks. Precision is the fraction of correctly classified results. While recall is the fraction of correctly classified results. They can be written as [131]:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{2.3}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2.4}$$

TABLE 2.4: Relevant literature for technique based community detection in dynamic network

Algorithm	Authors	Base approach	Network Model		No. of Communities		Limitation
			Temporal	Snapshot	Required	Not required	
Tiles [138]	Rossetti et al.	Label Propagation	✓	✗	✗	✓	Online computation might face challenge to massive stream of data and also it could be sensitive to parameter initializations.
OLCPM [21]	Boudebza et al.	Clique percolation and label propagation	✓	✗	✓	✗	OLCPM may struggle with accurately identifying overlapping regions in networks with dense or highly interconnected communities.
BT-LPD [**?**]	Singh et al.	B+ tree	✗	✓	✗	✓	Scalability may be a concern for very large protein interaction networks, potentially limiting its applicability to datasets of extensive size
jLDCE [87]	Li et al.	NMF	✗	✓	✓	✗	The high time complexity prevents the application of algorithms to large-scale temporal networks. It is also sensitive to parameter initialization which can effect the efficiency of algorithm.
ALPA [57]	Han et al.	Label Propagation	✓	✗	✗	✓	Its non-determinism and sensitivity to parameter settings, such as the threshold for label updating or the neighborhood size considered during label propagation might affect the performance of algorithm.
TIB [7]	Alduaiji et al.	Clique percolation method	✗	✓	✗	✓	The approach primarily emphasizes the temporal dynamics and interaction biases in social networks, potentially overlooking other important factors that contribute to community detection.
LSCPM [13]	Baudin et al.	Clique percolation method	✗	✓	✗	✓	The algorithm's efficiency is also influenced by the number of k-cliques in the link stream, which increases with k, particularly in large datasets. Therefore, the complexity of the algorithm depends strongly on k and the number of k-cliques.
sE-NMF [101]	Ma and Dong	NMF	✗	✓	✓	✗	Potential degradation of algorithm's capabilities for large networks due to the various strategies for network sparsity without destroying the community structure is a concern.

2.4.2 F Score

Assuming P and R to be precision and recall of the output respectively, F score is the weighted harmonic mean of P and R:

$$F - Score = 2 * \left(\frac{P * R}{P + R} \right) \tag{2.5}$$

Equation 2.5 considers equal weight for P and R. Therefore, it is also referred as balanced F score.

2.4.3 Normalized Mutual Information

Information theory facilitates the knowledge of mutual dependence of two variables to determine the quality of clustering, given the original labels. This mutual dependence of two variables is known as mutual information. Normalized Mutual Information (NMI) [155] is the normalized value, in range $[0,1]$, of mutual information between two data distribution.

Suppose T an C represent true and predicted community structures for a network of N nodes. Entropies, $H(T)$ and $H(C)$, of data distribution can be expressed as:

$$H(T) = -\sum_{i=1}^{|T|} \frac{|T|}{N} \log \left(\frac{|T|}{N} \right) \tag{2.6}$$

$$H(C) = -\sum_{i=1}^{|C|} \frac{|C|}{N} \log \left(\frac{|C|}{N} \right) \tag{2.7}$$

$|.|$ represents the cardinality of a set. Mutual information between T and C, $MI(T,C)$, can be written as:

$$MI(T,C) = \sum_{i=1}^{|T|} \sum_{i=1}^{|C|} \frac{|T_i \cap C_j|}{N} \log \left(\frac{N|T_i \cap C_j|}{|T_i||C_j|} \right) \tag{2.8}$$

Normalized mutual information between T and C can be expressed as:

$$NMI(T,C) = \frac{MI(U,V)}{mean(H(T),H(C))} \tag{2.9}$$

2.4.4 Adjusted Rand Index

Adjusted Rand Index (ARI) is a corrected version of the Rand Index (RI) [135] which computes the similarity between data partitions. ARI overcomes the limitation of RI to handle random partitions. The correction proposed by [64] is formulated in 2.10 where $E[RI]$ represents the expected RI.

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]} \tag{2.10}$$

Considering T and C as the true and predicted community structures of a given network, contingency table 2.5 represents the overlapping of T and C where X_i is i^{th} community of set X and n_{ij} is $T_i \cap C_j$. Using table 2.5 equation 2.10 can be rewritten as:

$T-C$	C_1	C_2	...	C_k	Sum
T_1	n_{11}	n_{12}	...	n_{1k}	a_1
T_2	n_{21}	n_{22}	...	n_{2k}	a_2
...	...	...	...	...	...
T_m	n_{m1}	n_{m2}	...	n_{mk}	a_m
Sum	b_1	b_2	...	b_k	

TABLE 2.5: Contingency table

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \frac{\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}}{\binom{n}{2}}}{\frac{1}{2}\left(\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right) - \frac{\left(\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right)}{\binom{n}{2}}} \tag{2.11}$$

2.4.5 Entropy

Entropy [147][34] is a measure of information associated with a random variable. In context of community detection, entropy measures how many real communities of network are classified among the predicted communities [185]. The entropy of the predicted communities C with respect to the true communities T can be mathematically formulated as:

$$H(C,T) = \sum_{i=1}^{k} \frac{|C_i|}{N} \left(-\frac{1}{\log m} \sum_{j=1}^{m} \frac{|C_i \cap T_j|}{|C_i|} \log \frac{|C_i \cap T_j|}{|C_i|} \right) \tag{2.12}$$

Where, k and m are number of predicted and true communities of network having N nodes. $|C_i|$ is the number of nodes present in i_{th} predicted community C.

2.4.6 Purity

Purity is a measure of how correctly the nodes are being classified to their respective communities by an algorithm. For predicted communities T and true communities C, it can be written as [105]:

$$Purity(T,C) = \frac{1}{n} \sum_{k} \max_{m} (T_m \cap C_k) \tag{2.13}$$

2.4.7 Average Isolability

Isolability measures degree of disconnectedness of communities in a given network [19]. It is a ratio of the internal connectivity among the nodes to the external connections. For community prediction C with k number of communities, the average isolability can be written as:

$$AvgIsolability(C) = \frac{1}{k} \sum_{i=1}^{k} \left(\frac{Connection_{internal}(C_i)}{Connection_{external}(C_i)} \right) \tag{2.14}$$

2.4.8 External Density

It is the ratio of all existing external connections in the community structure of given network to all possible external connections [19]. Let C be the predicted community list with k communities in a network of n nodes, external density can be written as:

$$ExtD(C) = \frac{\{(u,v)|u \in C_i, v \in C_j, i \neq j\}}{n(n-1) - \sum_{i=1}^{k}(|C_i|(|C_i|-1))} \tag{2.15}$$

where, u and v are pair of nodes and C_i is i^{th} community of C.

2.4.9 Coverage

Coverage of a clustering C is the ratio of intra cluster edges to total edges [22].

$$Coverage = \frac{|e_{intra}|}{|e|} \tag{2.16}$$

2.4.10 Modularity

Modularity (Q) [115] is a widely used quality measure for community structure. Consider an algorithm which has predicted k communities in a network. Let $\mathbf{e}$ be a symmetric $k \times k$ matrix where e_{ij} is the fraction of the edges in the network from community i to j. The trace, $Tr(\mathbf{e}) = \sum_i e_{ii}$, of matrix is a measure of the fraction of inter community edges to all edges. The row sum $a_i = \sum_j e_{ij}$ is the fraction of edges that are connected to i^{th} community. Using $Tr(\mathbf{e})$ and a_i, modularity can be formulated as:

$$Q = \sum_i (e_{ii} - a_i^2) = Tr(\mathbf{e}) - ||\mathbf{e}_2|| \tag{2.17}$$

where $||\mathbf{x}||$ is the sum of the elements of $\mathbf{x}$. Hence, Q is the difference of the fraction of inter community connections to expected value of the quantity if these connections are random.