

DETECTION OF MEANINGFUL COMMUNITIES IN SOCIAL LEARNING NETWORKS

HICHAM SADIKI¹, RAJAE ZRIAA², AYOUB ENNASSIRI³, MAHMOUD LHAM⁴, SAID AMALI⁵
^{1,2,3,4}Informatics and Applications Laboratory (IA), Faculty of Sciences, Moulay Ismail university, Morocco
⁵Data Analytics and Intelligent Systems (DAIS), FSJES, Moulay Ismail university, Morocco

E-mail: ¹h.sadiki@umi.ac.ma, ²r.zriaa@edu.umi.ac.ma, ³ayoub.ennassiri@gmail.com,
⁴m.lham@umi.ac.ma, ⁵s_amali@yahoo.com

ABSTRACT

Community detection and analysis in social learning networks is a critical challenge for understanding collaborative dynamics. These networks primarily rely on two distinct structures: pedagogical interests and social interactions. A promising approach to uncover meaningful learning communities is to combine these two structures. In this context, we propose a hybrid and adaptive two-stage approach that integrates both structural and contextual information to efficiently detect and update communities in dynamic educational social networks. This approach proceeds in two stages. First, we introduce a static algorithm called Learning-Enhanced Method for Community Detection (LEMCD). This hybrid algorithm leverages statistical and semantic measures to analyze both the pedagogical content associated with learners and their social interactions. Second, we present Adaptive LEMCD (LEMACD), an adaptive algorithm for detecting and updating community structures in dynamic social networks. A comparative study was conducted using real academic networks to evaluate the performance of LEMACD against other content- and behavior-based community detection algorithms. The results demonstrate that our approach is capable of identifying coherent communities in social networks. Moreover, it adapts effectively to changes in learning networks while reducing response time.

Keywords: *Dynamic Social Networks, Modularity, Content Information, Topic Detection, Meaningful Community Detection, Social Learning Networks.*

1. INTRODUCTION

Social learning networks are now widely used by learners and represent a significant source of data. These platforms allow students to share ideas, collaborate on projects, and access educational resources in fields such as computer science, mathematics, and literature. As a result, they generate large volumes of data that can be harnessed for research purposes. One prominent research area that benefits from such data is the detection of learning communities. These communities are defined as groups of learners who are strongly connected to one another and who share common interests. Learning communities are essential for understanding social and socio-educational dynamics [1].

To analyze learning networks, three main types of variables are generally considered [2]: structural variables, which refer to interactions and relationships between learners; compositional

variables, which describe learners' characteristics; and affiliative variables, which relate to learners' interests and their participation in specific topics.

A fundamental feature of these networks is the presence of learning communities, which are densely connected internally and loosely connected to the rest of the network [3]. However, most traditional community detection approaches rely on the assumption that learning networks are static, an assumption that is often unrealistic in social learning networks where new learners and interactions continuously emerge while others disappear [4][5].

1.1 Problem Statement and Limitations of Existing Approaches

Community detection is a highly challenging task in dynamic social learning networks. Early approaches were static in nature [6][7]. While these methods have contributed to significant advancements, they suffer from major limitations. For instance, they are computationally

expensive and very slow when applied to large-scale networks, as they recalculate community structures from scratch. A more efficient strategy would be to focus only on the portions of the network that have changed and to leverage previously detected community structures for incremental updates [8].

Moreover, most existing algorithms focus exclusively on structural interactions, overlooking contextual information such as learners' interests. This omission can result in heterogeneous communities composed of learners with divergent interests, thereby reducing both their pedagogical relevance and effectiveness [9].

1.2 Proposed Framework: LEMACD

In this work, we propose LEMACD, a hybrid and adaptive two-stage framework which incorporates both structural and contextual information to effectively detect and update communities in social learning networks with dynamical properties

✓ Step 1: Initial community detection with LEMCD

LEMCD (Learning-Enriched Method for Community Detection) uses statistical and semantic characteristics to determine learner's main topics of interest and connect them to the clusters in the network. The algorithm analyzes the structural links among the nodes in the resulting clusters and identifies communities by looking for sets of nodes that like similar concepts.

✓ Step 2: Adaptive update with LEMACD

LEMACD is adaptive in that it takes into account newly updated information in the network (such as new learners that participate or new interactions that occur) and thus can reassign learners to the most desired community. This reassignment is made by only considering new information namely it does association analysis only on the new data, thus reducing computation.

1.3 Illustration Example

As an example of this approach, imagine a social learning network with 13 learners each characterized by a particular topic of interest (fig. 1). If only using structure information, the discovered communities may cluster learners with different interests (see Fig. 1(b)). By contrast, the fusion strategy of structural as well as content-based information (Fig. 1(c)) allows to build communities of learners with similar preferences that are

structurally well connected, without causing homogenization within communities.

This result is also consistent with the primary goal of our LEMACD method, which targets for detecting meaningful communities.

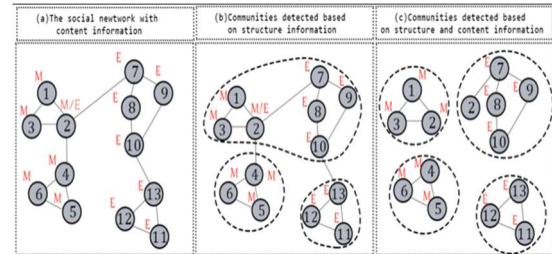


Figure 1: An Example Of Community Detection With Different Aspects.

1.4 Objective and Vision of Our Work

In the field of social network analysis, and more specifically in dynamic community detection, two fundamental questions arise:

1. How can communities be detected that reflect both the density of social interactions and the thematic interests of learners?
2. How can the community structure be updated to identify new communities without recalculating all communities at each network snapshot?

These questions form the foundation of our work. The primary objective is to classify learners engaged in social learning networks into distinct homogeneous communities, where members share similar interests and are strongly connected to one another. In the event of network evolution, community detection should focus exclusively on the modified parts of the network, while previously identified communities should be updated in an adaptive manner.

To formalize this problem, we consider a dynamic social learning network represented as a sequence of static graphs evolving over time, denoted by $G = \{G^0, G^1, \dots, G^T\}$. Each snapshot of the network $G^t = (E^t, V^t)$ is an undirected and unweighted graph, where V^t represents the set of learners (nodes) and E^t their interactions (edges) at time t . The communities detected at time t are denoted by $C_t = \{C_0^t, C_1^t, \dots, C_n^t\}$ where $n > 1$ and some communities may overlap, allowing a learner to belong to multiple groups due to their diverse interests.

The main objective is to partition the graph G^t into different communities C^t at any given time t , such that the members of each community C_i^t have similar interests and are densely connected, while being weakly linked to other communities $C_{j \neq i}^t \in C^t$

. Moreover, as the network evolves, the community structure C^t must be updated based on changes in the network to generate C^{t+1} , while preserving the properties of previously detected communities.

1.5 Impact and Future Perspectives

Our approach also raises several promising avenues by proposing a method that combines both the social learning networks dynamics and the context of the learners:

- ✓ **Analysis of learning dynamics:** Identify collective learning trends, learner behaviors and emerging learning themes.
- ✓ **Personalization of learning paths:** Suggesting content directly to learners according to their learning specifics, such as their social behavior and predictive success models.
- ✓ **Practical applications:** This method can be applied to real-world cases such as online learning and personalized recommendation systems.

Our approach thus makes it possible to address the challenges of dynamic community detection while ensuring actionable outcomes for educational stakeholders.

2. LITERATURE REVIEW

Community detection in social networks is a rapidly growing research area, with particular attention given to dynamic networks. This section reviews existing work, focusing on methods for community detection in both static and dynamic networks, as well as on the integration of structural and contextual information.

2.1. Community Detection in Static Networks

In the literature, initial community detection methods follow to represent social networks as static graphs with nodes corresponding to users and edges for interactions between users. Among them, the Louvain algorithm [1] is the most well-known for its effectiveness in maximizing modularity, a commonly used quality measure for evaluating the compactness of the discovered communities. Modularity The modularity of a given partition in the network is a property of the partition and is defined as difference between the realized density of links between the vertices of the community and the expected density of links between the same vertices in a random network [2].

A notable variant of this method is the I-Louvain algorithm that takes some node attribute (e.g., personal features, interested regions) into consideration to improve community quality [8]. Nevertheless these methods have one important drawback that they are focused mainly on static networks, what does not make them particularly suitable for dynamic networks in which the network structure can change thus the appearance or disappearance of the nodes and interactions over the time.

2.2. Community Detection In Dynamic Networks

With the continuous evolution of social networks, it is important to note that user interactions are constantly changing, making static approaches often ineffective. Early attempts were based on applying algorithms to initial snapshots, adapting them through successive detections and associating them with communities identified over time [6][9]. However, this strategy faces a computational burden, as change-detecting methods generally require a full recomputation of communities throughout the entire network timeline.

To overcome this issue, two approaches have been developed, DynaMo and Zhao [3][7], which focus only on the modified parts of the network, thereby emphasizing computational efficiency. Finally, the resulting community structures may differ: the first approach performs a gradual adjustment based on maximizing modularity after each node change, while the second efficiently detects communities by adapting them based on the induced subgraph.

2.3. Integration of Structural and Contextual Information

The main principle of the methods presented so far is to detect communities based on the structure of networks. However, this approach is limited by the exclusion of any other type of information about learners. User profile information, such as interests and activity patterns, is thus ignored. According to recent studies, such contextual information can potentially enhance the quality of detected communities. In this regard, two methods have been proposed for dynamic community detection: NEIWalk, which relies on random walks weighted by contextual similarities [5], and I-Louvain, which modifies the traditional modularity function to combine node attributes with their network connections in order to produce higher-quality communities[8].

Nevertheless, even these newer methods, which rely on enriched networks for community detection, overlook the challenges posed by dynamic network environments.

2.4. Applications in Social Learning Networks

In the educational context, social networks are characterized by frequent interactions among learners and diverse learning modalities. These networks offer several opportunities, such as analyzing learner interactions to identify groups of students who share common learning interests [4], personalizing learning paths by recommending resources based on individual needs using contextual information [10], and predicting academic performance by integrating predictive models such as linear regression to anticipate learners’ outcomes based on their interactions and interests [11][12].

2.5. Summary of Limitations

Nevertheless, there are still several limitations of existing approaches whose optimization has not been fully accounted for large-scale and dynamic graphs and which are not as flexible for coping with both structure and context incorporated in a dynamic way. These deficiencies reveal the requirement of a hybrid approach which can combine structural as well as contextual information while adjusting to the changeable behavior of dynamic networks.

3. PROBLEME DEFINITION

In this section, we present the LEMCD algorithm, the first phase of our methodological framework that aims to detect the initial structure of communities. First, we provide an overview of LEMCD, followed by a detailed description of the different steps.

3.1 Overview of the LEMCD Algorithm

Our LEMCD algorithm (see algorithm 2 and figure 3) is based on community discovery by leveraging both the content information shared by learners and the contextual data extracted from their interactions. In the first step, all texts available in the social learning network are extracted and classified as distinct documents. A data preprocessing phase (line 1 of algorithm 2) is then carried out to produce a homogeneous dataset suitable for subsequent analysis. At line 2, learners’ areas of interest are identified using a hybrid approach that combines statistical and semantic measures.

Based on the thematic topics identified in step 2, learners are grouped into several thematic clusters according to their interests (step 3). Finally, in the last step (step 4), a structural link analysis is applied within each thematic cluster to detect the communities.

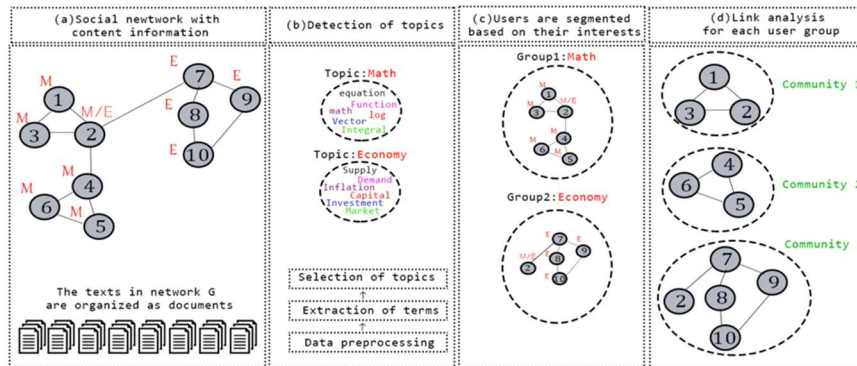


Figure 2: (a) The Content Of G Is Structured Into a Set Of Documents d , Where Each d_i Represents User i 's content. (b) Documents Are Grouped By Shared Topics. (c) Users Are Clustered Based On Their Topics Of Interest, With Multi-topic Users Treated As Overlapping Nodes (e.g., node 2). (d) The Louvain Method Refines Cluster Density, Identifying Dense Groups As Communities.

3.2 Content and Structure Information-Based method For Community Detection: LEMCD Algorithm

In this section, we present the different steps that make up the LEMCD algorithm. These steps are essential for capitalizing on the structural and contextual information related to learners, with the goal of efficiently identifying meaningful communities.

✓ *Step 1: Data preprocessing*

The first step of our approach involves using as input a collection of texts (emails, blogs, comments, messages, etc.) originating from social learning networks. The main objective is to clean and prepare the data so that it becomes homogeneous and usable for the following steps.

✓ *Step 2: Grouping of social subjects and topic detection*

Social learning networks serve as platforms for aggregating information posted by learners through various social objects (texts, images, and

videos). To detect or capture learners’ interests, our approach relies on analyzing the different types of social objects, primarily textual content.

A study conducted by the authors of [13] analyzed the content of emails and blogs to identify themes discussed in social networks. Similarly, other researchers [14], [15] examined several social networks where content was in textual form. In our methodological framework, the topic detection phase is carried out separately from the community analysis phase (structural link analysis). This separation allows us to examine the network content based on its specific type.

Our approach can thus be applied to different types of social objects by slightly adjusting the topic detection phase, using a model capable of grouping the considered objects. This flexibility is one of the major strengths of our methodological framework, making it particularly effective for

communication and information dissemination in real-world networks.

However, it is important to note that the results obtained depend heavily on the model used to group social objects. Therefore, this process must be highly accurate, regardless of content type. Based on the data available in the literature, our work primarily focuses on organized textual social objects. Furthermore, we propose a hybrid model based on statistical and semantic measures to capture learners’ thematic interests.

In the LEMCD algorithm, one or more topics are assigned to each learner according to their areas of interest. To achieve this, we analyze the texts extracted from the network and apply the k-means clustering algorithm to divide the textual data into *k* clusters (or classes).

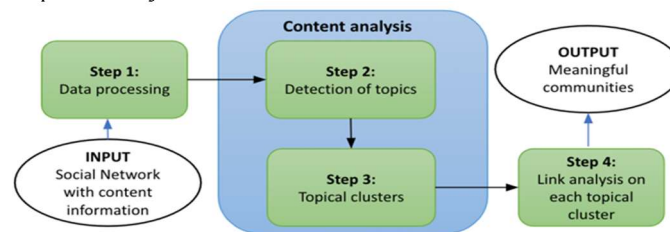


Figure 3: Steps Of LEMCD Algorithm

Algorithm 1: Topic detection

Input: *D*: Set of documents, *k*: Number of clusters, γ : Weighting parameter ($0 \leq \gamma \leq 1$), *L*: Number of selected keywords

Output: Clusters with labels

BEGIN

```

Clusters ← k_means(D, k) // Step 1: Document clustering
Wγ ← Select_Relevant_Terms(D) // Step 2: Selecting relevant terms
for each term w in Wγ do // Step 3: Calculating term scores
    | S(w) ← Compute_Score(w, Wγ) // Based on a similarity equation (7)
end for
Sort Wγ in descending_order based on S(w) // Step 4: Selecting keywords
Labels ← Select_Top_L(Wγ, L)
return Clusters, Labels // Step 5: Return clusters with labels
    
```

END

Algorithm 2: LEMCD algorithm

Input: *G*: Social network with textual content

Output: Overlapping communities

BEGIN

```

G_cleaned ← Clean_Data(G) // Step 1: Data preprocessing
Thematic_Clusters ← Topic_Detection(G_cleaned) // Step 2: Topic detection using Algorithm 1
for each Cluster C IN Thematic_Clusters DO // Step 3: Community construction
    | Apply_Louvain(C) // Detecting dense communities
    
```

```

    end for
    return Overlapping_Communities // Step 4: Return overlapping communities
END
    
```

At this stage, for each cluster, the problem at hand is to find relevant and appropriate terms from a topic. The selection is performed in the light of two kinds of measurements: statistical and semantic. The theme detection is implemented in a series of logical-related steps, which in turn consists of two sub-steps:

- ✓ **Keyword extraction:** The goal is to identify candidate words in the texts that can be used to represent a topic. We do so by relying on a grammar analysis tool which extracts nouns of the text. Then we derive the pre- and post-words of each noun, which will be candidate processing words.
- ✓ **Concept selection:** We choose words from the words extracted in the keyword extraction that are most relevant to a topic. This is achieved by merging two principal techniques: a statistical and a semantic technique. These two methods assist in measuring the relevancy of potential terms to the discovered topics.

Statistical measure: χ^2 improved.

Statistical significance of terms is determined by means of the chi-squared χ^2 statistic, which measures the strength of association between a term w and a topic T it is useful for showing how often a term occurs with respect to a given topic in contrast to the term's occurrence throughout the entire educational social network. Through quantifying this dependence, the χ^2 test allows to differentiate terms, which are both common and discriminative for describing particular thematic clusters.

The χ^2 statistic is defined as follows[1][2]:

$$sim(w_r, w \in W_T) = \frac{1}{2|V|} \sum_i \left(\frac{\min(I(z_i, w_r | w'), I(w, z_i | w'))}{\max(I(z_i, w_r | w'), I(w, z_i | w'))} + \frac{\min(I(w_r, z_i | w'), I(w, z_i | w'))}{\max(I(w_r, z_i | w'), I(w, z_i | w'))} \right) \tag{3}$$

with :

- V is the vocabulary in the theme T ,
- $I(z_i, w_r | w')$ represents the conditional mutual information between the terms z_i, w_r et w' according to their appearance.

The conditional mutual information (CMI) used in this work is formulated in Equation 4:

$$\chi_{w,T}^2 = \sum_{i \in \{w, \bar{w}\}} \sum_{j \in \{T, \bar{T}\}} \frac{(O(i,j) - E(i,j))^2}{E(i,j)} \tag{1}$$

with :

✓ $O(i, j)$ represents the observed frequency of the term w in the theme T ,

✓ $E(i, j)$ is the expected frequency, calculated as:

$$E(w, T) = \frac{1}{d} \times \sum_{i \in \{w, \bar{w}\}} O(i, T) \times \sum_{j \in \{T, \bar{T}\}} O(w, j) \tag{2}$$

with d , the total number of documents.

Term similarity measure based on conditional mutual information

It is evident that in many real-world systems, several words have similar meanings and are used interchangeably (as synonyms) to describe the same concepts. For example, in the domain of community detection, the two terms “graph” and “network” are often used to describe interactions between entities within a system. The main goal of this step is to select, for each topic T , semantically similar words that were identified during the previous step.

To achieve this objective, we calculate the similarity between the words belonging to a topic T and the relevant terms using Conditional Mutual Information (CMI). Considering a set of terms W_T (words belonging exclusively to topic T), W_r (relevant terms), and W' (words not belonging to either W_T or W_r), the conditional mutual information $I(w_r \in W_T, w \in W_r | W')$ measures the average amount of information shared between W_T and W_r given W' .

The similarity measure is defined in equation 3:

$$I(z_i, w_r | w') = \sum_{z_i \in W_T} \sum_{w_r \in W_T} \sum_{w' \in W'} P(z_i, w_r, w') \log \left(\frac{P(w') P(z_i, w_r, w')}{P(w_r) P(z_i, w')} \right) \tag{4}$$

In this approach, $P(w')$ represents the probability of occurrence of the word w' . The terms $P(z_i, w')$ and $P(w_r, w')$ respectively denote the co-occurrence probabilities of the word pairs z_i and w' , as well as terms w_r et w' . More generally, $P(z_i, w_r, w')$ refers to the probability of co-occurrence of the terms z_i, w_r et w' within topic T . This probability is estimated by taking the ratio between the number of occurrences where z_i is followed by the terms w_r et w' within topic T , and the vocabulary size of T (see equation 5):

$$P(z_i, w_r, w') = \frac{f(z_i, w_r, w')}{|V|} \tag{5}$$

with $f(z_i, w_r, w')$ denotes the number of times the word z_i is followed by the terms w_r et w' .

The main objective of the topic detection phase is to identify a set of representative words for a given subject discussed by users in social networks. To achieve this, we propose a hybrid model that identifies the most informative terms shared across the network W_r , and captures the semantic relationships between these terms and the words associated with a given topic T .

In the proposed approach, for each topic T_j , the words that exhibit both strong dependence on T_j and high similarity with the relevant terms W_r are selected as representatives of T_j . Accordingly, our criterion for evaluating the relevance of terms to a given topic consists of two components: a dependence measure and a similarity metric, defined as follows:

$$S(w) = \gamma \times r_{x^2}(w) + (1 - \gamma) \times sim(w_r, w) \tag{6}$$

where:

- ✓ $r_{x^2}(w)$ measures the dependence of the word w on the theme T ,
- ✓ $sim(w_r, w \in W_T)$ evaluate the similarity between w and the relevant terms W_r in the network.

All words associated with a topic T are ranked in descending order based on their score $S(w)$, and the L_{top} terms are selected to represent the topic T .

The parameter $\gamma \in [0,1]$ is a weighting factor that controls the balance between a word's dependence on the topic and its semantic similarity to other relevant terms. When $\gamma = 1$, selection is based entirely on topic dependence; when $\gamma = 0$, it relies solely on similarity. Values of γ between 0.5 and 1 favor dependence, while values between 0 and 0.5 prioritize similarity. A value of $\gamma = 0.5$ assigns equal weight to both criteria.

In summary, during the topic detection phase, the texts from the social network are partitioned into k classes (topics). For each topic, words are ranked in descending order based on their score computed using Equation (6), and the L_{top} terms are selected to characterize the topic.

- ✓ *Step 3 : Grouping of social objects and topic detection*

The main objective of this step is to use the identified topics to group users into topic-based clusters. In other words, each group of learners should ideally share a single common interest. However, in our case, a learner may be interested in two topics simultaneously. In such situations, the learner is considered an overlapping node. This is illustrated by the example of a super-node that belongs to two different graph clusters (see figure 4).

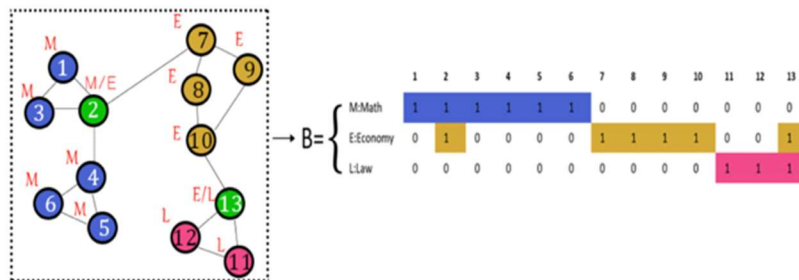


Figure 4: User's Partitioning Process Into Different Topical Clusters

Let $T = \{T_1, T_2, \dots, T_k\}$ be the set of topics and $U = \{u_1, u_2, \dots, u_p\}$ the set of users involved in the social network. We define the notation $Topics(u) = \{T_i\}$ to represent the set of topics associated with a given user u . In order to group users into clusters, we construct a matrix B of dimensions $k \times p$, where:

- ✓ k is the number of topics ($k = |T|$),
- ✓ p is the total number of users in the network ($p = |U|$).

The elements of the matrix, denoted $B = b_{i,j}$, are defined as follows:

$$b_{i,j} = \begin{cases} 1, & \text{if } Topics(u_i) = T_i \quad u \in U, T_i \in T \\ 0, & \text{sinon} \end{cases} \quad (7)$$

Thus, two users u_i and u_l belong to the same group if $B_{j,i} = B_{j,l} = 1$.

For example, consider a social network in which each learner is represented by a node and their interactions are represented by edges. Each node is annotated with the topics corresponding to the learner's interests. In this example, we have three topics

$T = \{M, E, L\}$ and a set of 13 learners $U = \{1, 2, \dots, 13\}$.

By applying this approach, we form three topic-based clusters:

- $T_{Math} = \{1, 2, 3, 4, 5, 6\}$
- $T_{Economy} = \{2, 7, 8, 9, 10, 13\}$
- $T_{Law} = \{11, 12, 13\}$

- ✓ *Step 4: Link analysis within each thematic cluster*

According to some of the above topic clustering of learners, they may have strong interests in common, however have weak interaction between them. This may lead to sparsely populated communities, which is far from what is meant to be a community. On the other hand there are users that tend to cooperatively engage more with others with similar interests, in this case, resulting in learner groups of a higher density. We use a static

community detection method to find such subgroups of students that are densely connected to each other inside them. To have an optimal internal structure for each thematic cluster, We work with Louvain algorithm [1], which maximizes modularity while being relatively fast to compute. The original version of our main technique for detecting isolated communities is presented in Algorithm 2.

3.3 Dynamic Community Detection Via the LEMACD Algorithm

One of the key tasks in community detection for dynamic networks is to identify community structures at each snapshot of the network. Most existing methods recompute the entire network structure from scratch or process the whole network at every update. While theoretically sound, this approach is computationally intensive and results in extremely slow algorithms, even when only minor changes occur in the network. Such a scenario is common in social networks, where connections evolve continuously with the addition or removal of learners and their interactions. In this context, an effective solution is to develop methods that focus only on the modified portions of the network while leveraging prior knowledge of the initial community structure.

In this article, we introduce the LEMACD algorithm (Learning-based Method for Adaptive Community Detection), which is designed to dynamically detect and update community structures in constantly evolving educational networks. This algorithm not only incorporates structural interactions among learners but also contextual information related to their academic interests, while adapting to the network's gradual changes.

LEMACD is based on a detection method that balances accuracy and efficiency while minimizing information costs. It places particular emphasis on learning dynamics. Community evolution is analyzed to identify learner groups with similar engagement patterns and to predict academic performance. By combining historical data on interactions and topic interests with updated network information, LEMACD demonstrates the ability to track community dynamics with high precision and speed.

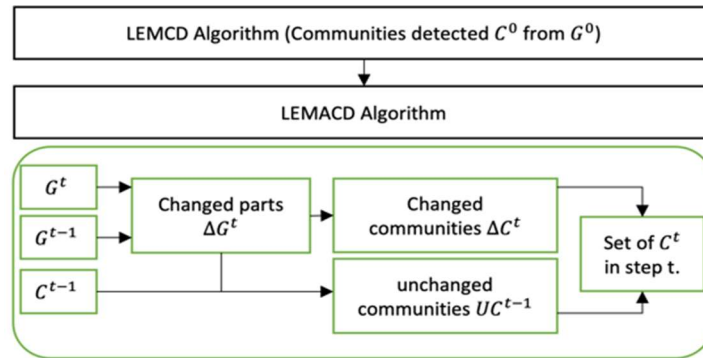


Figure 5: Main Steps Of LEMACD Algorithm

3.3.1. Overview Of The LEMACD Algorithm

In this subsection, we give a complete result one of the key output of our work—the LEMACD algorithm. It is the underlying mechanism that is designed to build dynamically the community structure after each network modification. The main aim of LEMACD is to help in community discovery in dynamic educational networks, by considering the changing behaviours of learners. At each network change iteration, LEMACD tries to adapt the community structure locally rather than doing a full reinitialization. The algorithm focuses on two primary aspects in order to accomplish this:

Altered parts of the network: Reconstruction of the network may require adding/removing nodes (learners) and/or links (interactions), and the algorithm concerns itself only with the partitions that are affected by the change.

The previous community structure: The algorithm uses information from a previous network to optimize the adjustment approach while minimizing the impact on the higher level of network logic.

The reasoning of our technique is illustrated in figure 5, consisting of two primary stages:

- Initial structure of communities

LEMACD begins by identifying the initial community structure using the LEMCD algorithm described earlier. The original network G^0 is divided into several thematic subgraphs $TC^0 = \{TC_1^0, TC_2^0, \dots, TC_k^0\}$, each representing a distinct topic. Then, a link analysis is carried out within each thematic cluster using the Louvain method [1] to detect the final communities. Each community C_i^0 is associated with one of the thematic clusters, allowing each community to be labeled according to the topic in which it was identified.

- Incremental Community Detection

After the layered structure is developed, LEMACD uses a step-by-step method to

incrementally update communities with the evolution of the network at each step during the time evolution. There are four essential steps to this process:

- a. *Identification of modified parts ΔG^t :* The structure of social learning networks is continuously changing, in contrary to static graphs. To find out dynamic changes, we use the definition of the algorithm to identify the modified components. Changes can be node adding or removing (new learners appear or disappear) and edge adding and removing (new or interrupted interactions).
- b. *Node addition :* The addition of new learners is detected by comparing the sets of nodes in the graph at time t with those at time $t - 1$. If these new nodes interact with existing members of a community, the structure remains stable. Otherwise, new communities may emerge in the relevant topic clusters.
- c. *Deleting nodes :* When a learner drops out of the network, the local connectivity of communities can change. If removing a node splits a community, LEMACD restructures the community accordingly.
- d. *Adding and deleting edges :* New interactions (intra-community or inter-community) influence community density differently. Adding edges within a community strengthens cohesion without altering the overall structure, while interactions between different communities can lead to mergers or structural adjustments.

The adapted parts of the graph are considered with the academic content of the learners so that community updates reflect both the interests and the learning behavior of the members.

3.3.2. How Does The LEMACD Model Handle Network Evolution Events?

In this subsection, we explain how the LEMACD model handles the four types of network evolution events mentioned earlier.

a) Handling the addition of a new learner

Let us examine the first scenario, in which a new learner u and their interactions are added to the graph G^t . To insert u into the appropriate community, we identify their topics of interest (thematic groups) using Algorithm 1. Based on the local information associated with u ($TC(u)$), two situations can occur:

- ✓ If $TC(u)$ does not exist in the set of thematic clusters TC^{t-1} , a new thematic cluster is created containing only u , and the existing communities remain unchanged.
- ✓ If $TC(u)$ exists in TC^{t-1} , u is inserted into the corresponding thematic cluster. We then check whether u is an isolated node, i.e., $d(u) = 0$. If so, a new community is created containing only u , and the existing community structure is preserved. The event becomes more significant and commonly observed when u is introduced along with links that connect them to one or more existing communities. In this case, we check whether all neighbors of u belong to the same community.

b) Handling the addition of a new interaction

When a new interaction appears between two nodes (u, v), a new edge $e_{u,v}$ is added to the graph G^t . Depending on the type of edge, this event falls into one of two categories:

- ✓ $e_{u,v}$ is considered an intra-link if both endpoints u and v belong to the same community.
- ✓ $e_{u,v}$ is considered an inter-link if it connects nodes from different communities.

Cas 1: If both endpoints of $e_{u,v}$ are located within the same subgraph, we first check whether $e_{u,v}$ is an intra-link. If it is, the current community structure remains unchanged. If not, we update the community structure within the corresponding subgraph using the Louvain method, while leaving the other subgraphs unaffected.

Cas 2: If the endpoints of $e_{u,v}$ belong to different subgraphs, we update the current community structure by inserting each endpoint into its appropriate community.

c) Deletion of an existing learner

In order to refine the community structure when the current learner u is removed in the network, we discuss two sub-cases:

- ✓ **If u has a degree equal to 1 ($d(u) = 1$),** the existing community structure is not affected. In this case we just delete u and the edge adjacent to it on the subgraph.
- ✓ **If u has more than 1 ($d(u) > 1$),** interactions in the affected subgraph may decrease, which may generate isolated communities or small communities that may be merged. In this case, we apply the Louvain method to discover the new community structure in the modified subgraph $TC(u)$, while leaving the other thematic clusters unchanged.

d) Deleting an existing edge

The last event concerns the deletion of an existing interaction $e_{u,v}$ between two nodes u and v of the network. Depending on the type of edge $e_{u,v}$, we distinguish two cases similar to those of the addition of a new interaction:

- ✓ If $e_{u,v}$ is an intra-link, we recompute the community structure only on the corresponding subgraph $TC(u)$ (with $TC(u) = TC(v)$) using the Louvain method, while keeping the other subgraphs unchanged.
- ✓ If $e_{u,v}$ is an interlink, the community structure remains unchanged.

e) Analyse de la complexité

In this subsection, we discuss the theoretical complexity of our proposed method in the time and space domains. We first study the applicability of our approach for large-scale networks, since at least some of the existing approaches are not very applicable in such contexts. The computational effects, and particularly the temporal and spatial complexity of the algorithms, are some of the driving motivations for studying dynamic community detection in social networks that are considered in this paper. It follows, that we evaluate the theoretical performance of our method on the basis of combined time and space their complexity. The total complexity of our scheme has two terms: the complexity of LEMCD algorithm and the complexity of LEMACD.

Before detailing these aspects, we specify that our framework uses the Louvain and K-means algorithms, whose time complexity is linear as a function of $|E|$ (number of edges) and $|d|$ (average degree of nodes), respectively.

✓ Time complexity of the LEMCD algorithm

To calculate the time complexity of our approach, we analyze each of its phases separately. We start with the LEMCD algorithm, evaluating its

time complexity through two major steps: topic detection (step 2) and link analysis (step 4) In the topic detection phase, we first divide the initial set of documents into k clusters using the k-means algorithm, whose time complexity is $O(d)$. Each cluster i contains a number of words w_i , and the total sum of words equals w . To name these clusters, we select the relevant words according to Equation (7), with a complexity of $O(w \times w) = O(w^2)$. Then, we sort the words in each cluster, which adds a complexity of $O(w \log w)$. Thus, the time complexity of the topic detection (TD) step is:

$$T_{temp}(TD) = O(\max\{d, w^2, w \log w\}) = O(w^2)$$

For the link analysis (LA) step, we apply the Louvain method on each thematic cluster, with a complexity of $O(|E|)$. Thus, the total time complexity of the LEMCD algorithm is:

$$T_{temp}(LEMCD) = T_{temp}(TD) + T_{temp}(LA) = O(w^2 + |E|).$$

✓ Time Complexity of the LEMACD Algorithm

The time complexity of the LEMACD algorithm depends largely on the changes occurring in the network. To evaluate it, we analyze the complexity of each type of event: node or edge removal, addition of new edges, and addition of new users.

For the first three types of events, the update only involves link analysis using the Louvain method, which has a time complexity of $O(|E_c|)$, where $|E_c|$ is the number of edges in the affected subgraph.

The most computationally expensive case involves the addition of new users, which requires both topic detection with a time complexity of $O(w_u^2)$, where w_u is the number of words associated with the new user and link analysis with complexity $O(|E_c|)$. Therefore, the time complexity for adding a new user is $O(w_u^2 + |E_c|)$.

Consequently, the overall time complexity of the LEMACD algorithm at each time snapshot is:

$$T_{temp}(LEMCD) = O(\Delta(w_u^2 + |E_c|)) \text{ where } \Delta \text{ denotes the number of changes occurring at each snapshot.}$$

✓ Space complexity of the LEMCD algorithm

In our approach, we use a vector to store all the words extracted from social networks, with a space complexity of $O(w)$, where w is the size of the vector. We also use k additional vectors to store words related to each topic, which adds a complexity of $O(w)$. The matrix B , of dimensions $|U| \times k$, represents users' topic interests and has a complexity of $O(k \cdot |U|)$. For link analysis using the Louvain method, the required space is $O(|E|)$, where $|E|$ is the number of edges in the network.

Therefore, the total space complexity of the LEMCD algorithm is:

$$T_{spatial}(LEMCD) = O(w + |E| + k \cdot |U|)$$

✓ Space complexity of the LEMACD algorithm

For the LEMACD algorithm, we do not repeat the user partitioning step; instead, we focus on the number of shared words (w_u) and link analysis within the modified clusters. Therefore, the space complexity at each snapshot is: $T_{spatial}(LEMCD) = O(w_u + |E_c|)$

The computational complexity of our approach was compared with several static and dynamic community detection algorithms. Table 1 presents the time and space complexities of these methods.

For dynamic algorithms such as NEIWalk and FacetNet, time complexity is heavily influenced by the number of network changes (i.e., $|\Delta U_t|$ and $|\Delta E_t|$). In contrast, our approach, LEMACD, stands out with a time complexity of $O(\Delta(w_u^2 + |E_c|))$, where Δ represents the number of changes per snapshot, and a space complexity of $O(w_u + |E_c|)$, demonstrating its efficiency in handling dynamic network evolution.

Table 1: Computational complexity of some community detection algorithms. $|U_t|$, $|E_t|$ represent respectively the number of nodes and edges of the network at time step t . d_t is the average degree at snapshot t . L denotes the number of iterations. $|\Delta U_t|$, $|\Delta E_t|$ denote respectively the number of changed nodes and edges at snapshot t

Approaches	Methods	Time complexity	Space complexity
Static community detection algorithms	I-Louvain[8]	$O(U ^2)$	$O(U)$
	Edge betweenness[16]	$O(E ^2 \cdot U)$	$O(E \cdot U)$
	LPA[17]	$O(L \cdot E)$	$O(U)$
	LEMCD	$O(w^2 + E)$	$O(w + E + k \cdot U)$
Dynamic community detection algorithms	NEIWalk[5]	$O(U_t (1 + U_t \cdot \log U_t))$	$O(U_t ^2)$
	FacetNet[18]	$O(L \cdot E_t \cdot U_t)$	$O(U_t ^2)$
	MIEN[19]	$O(U_t + E_t)$	$O(U_t)$
	LEMCD	$O(\Delta(w^2 u + E_c))$	$O(w_u + E_c)$

4. RESULTS AND DISCUSSION

In this section, we evaluate the performance of the proposed algorithm by conducting in-depth experiments on three real-world dynamic social networks. As mentioned earlier, the LEMCD algorithm is executed only on the initial network G_0 to identify the initial community structure. In contrast, the LEMACD method is responsible for handling all subsequent modifications in the network over time. Therefore, this experimental phase focuses primarily on assessing the performance of the LEMACD method.

Before analyzing and discussing the experimental results, we first describe the algorithms used for comparison, the datasets employed, and the evaluation metrics adopted to benchmark our approach against existing methods.

4.1. Experimental Context And Reference Algorithms

In this study, we compare the LEMACD algorithm with seven other community detection algorithms, including both static and dynamic approaches. Two static methods were selected: the Louvain algorithm, which relies solely on structural information, and I-Louvain, which combines content-based and structural data. In addition, we include five dynamic algorithms, notably NEIWalk, which was specifically designed for content-based networks.

4.2. Dataset Description

For our experiments, we used a real-world dynamic social network dataset: DBLP, a digital library focused on computer science publications, containing co-authorship relations among researchers. The dataset includes more than 3 079 007 authors and 25 166 964 co-publication relationships. In our experiments, each node represents an author and each edge represents a co-authorship relationship. The DBLP dataset spans over 50 years of scholarly collaborations. For our analysis, we selected citation data from 1993 to 2017, using annual snapshots. The first five years (from 1993 to 1997) were used to establish the initial community structure. We focused on three types of publications: journal articles, book chapters, and conference proceedings. During the preprocessing phase, more than 2.2 million articles and 1 million authors were processed, and 299 993 unique words were extracted from publication titles. For this study, we used version 10 of the DBLP dataset, available at: <http://www.arnetminer.org/citation>.

4.3. Performance Metrics

The primary objective of our approach is based on the idea that the detected communities should group users who share similar interests and

are strongly connected to each other. Therefore, a rigorous evaluation must be conducted along two dimensions: the purity of the communities and their density. In addition, another important aspect is to assess the stability and smooth evolution of dynamic communities over time.

For these reasons, we selected two essential evaluation metrics modularity and purity which respectively measure the density and quality of the detected communities.

Modularity: Introduced by Newman, modularity is a widely recognized metric used to assess the quality of a network partition into communities. It is based on the principle that the higher the modularity score, the better the separation between communities. The modularity of a thematic cluster TC is calculated as follows:

$$Q^{TC} = \frac{1}{2\alpha} \sum_{i,j} \left(A_{i,j} - \frac{d(i)d(j)}{2\alpha} \right) s(C_i, C_j) \quad (10)$$

where:

- ✓ i and j represent vertices in the thematic cluster TC .
- ✓ $A_{i,j}$ is the adjacency matrix associated with the cluster TC .
- ✓ $d(i)$ is the degree of the node i .
- ✓ C_i is the community containing the node i .
- ✓ $s(C_i, C_j) = 1$ if $s(C_i) = s(C_j)$ and 0 otherwise.
- ✓ $\alpha = \frac{1}{2} \sum_{i,j} A_{i,j}$ is a normalizing factor.

The overall modularity is obtained by taking the average of the modularities of the different clusters:

$$Q = \frac{1}{k} \sum_{i=1}^k Q^{TC_i} \quad (11)$$

with k is the number of topics.

Purity: To evaluate the quality of the detected communities, we use a metric known as purity, which is defined by the following equation:

$$Purity = \frac{1}{m} \sum_{i=1}^m \max_{1 \leq j \leq k} \left(\frac{n_{i,j}}{n_i} \right) \quad (12)$$

where :

- ✓ m represents the total number of communities detected.
- ✓ $n_{i,j}$ is the number of nodes belonging to the community i and topic j .
- ✓ n_i denotes the total number of nodes in the community i .
- ✓ k represents the total number of subjects detected.

The partition obtaining the highest purity value is considered the best, from the point of view of the themes associated with the communities.

a) Modularity

✓ *CORA*: The figure 6 shows the evolution of modularity scores obtained by different community detection methods on the CORA dataset. Overall, LEMACD demonstrates stable and competitive performance, achieving the highest modularity values in several cases (notably from iteration 7 onward), with a peak of

0.4598. In comparison, the FacetNet method yields the lowest results, showing significant fluctuations and a modularity score below 0.44 across multiple iterations. MIEN and I_Louvain exhibit relatively consistent behavior, with similar performance levels, although MIEN slightly outperforms I_Louvain in some iterations, peaking at 0.4583. Finally, NEWalk also demonstrates stable performance but remains slightly below that of LEMACD and MIEN.

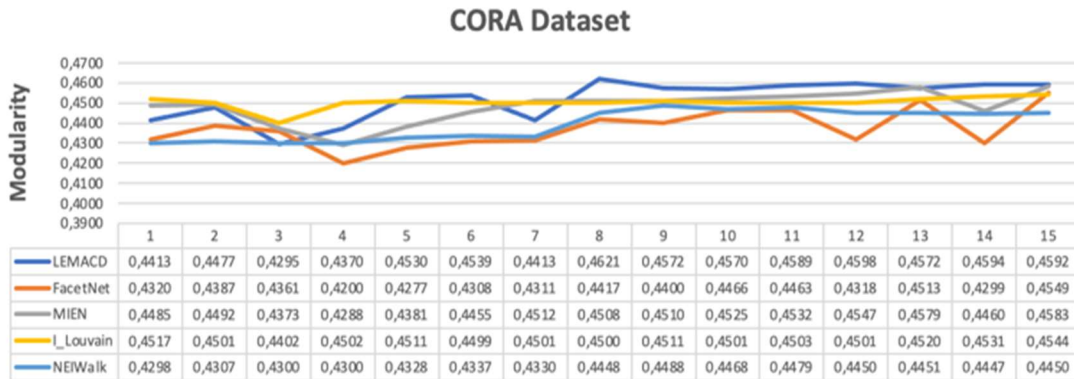


Figure 6: Experimental Results On CORA Dataset Using Modularity

✓ *DBLP*: The modularity curve for the DBLP dataset (Figure 7) shows that LEMACD outperforms the other methods in the majority of iterations, reaching a peak modularity score of 0.4978 at iteration 20. Despite some slight drops in the middle iterations, LEMACD consistently maintains higher modularity than both FacetNet and NEWalk. MIEN, which initially shows lower performance, gradually improves from iteration 15 onward, ultimately reaching a modularity score of 0.4877. I_Louvain remains stable

throughout all iterations, with scores around 0.467, showing no significant improvement. In contrast, FacetNet remains the least effective method, with values fluctuating around 0.46 and exhibiting considerable variability. NEWalk also lags slightly behind LEMACD and MIEN. This evaluation highlights not only the superior partitioning quality achieved by LEMACD on the DBLP dataset, but also its ability to consistently maintain high performance throughout the experiments.

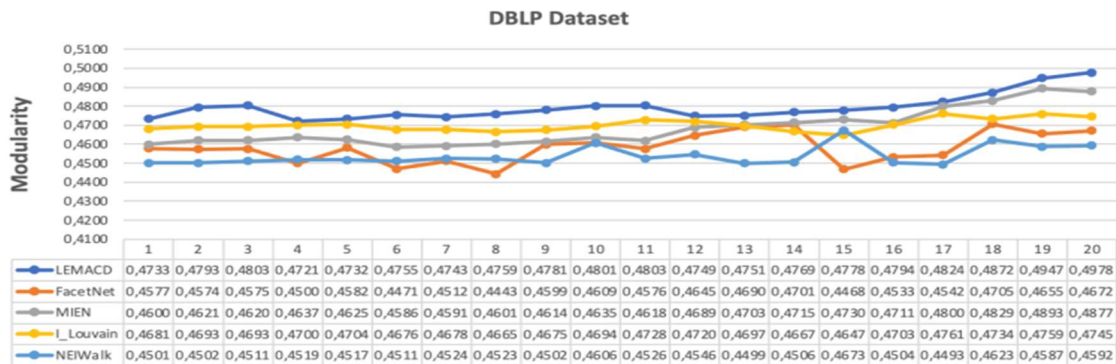


Figure 7: Experimental Results On DBLP Dataset Using Modularity

b) Purity

✓ CORA : The evolution of purity on the CORA dataset (Figure 8) clearly demonstrates the superiority of LEMACD compared to the other evaluated methods. LEMACD consistently maintains high purity values, fluctuating around 0.90 and reaching a peak of 0.9350 at iteration 15. In comparison, I_Louvain and NEWalk offer decent but slightly lower performance, with results mostly ranging between 0.88 and 0.91. On the other hand, FacetNet and MIEN show

significantly weaker performance, with FacetNet dropping as low as 0.8534 and MIEN reaching a minimum of 0.8333. The observed trend highlights the robustness of LEMACD not only in terms of modularity, but also in terms of grouping quality, providing more accurate community separation than competing approaches. These results confirm the effectiveness of LEMACD on complex academic graphs such as CORA.

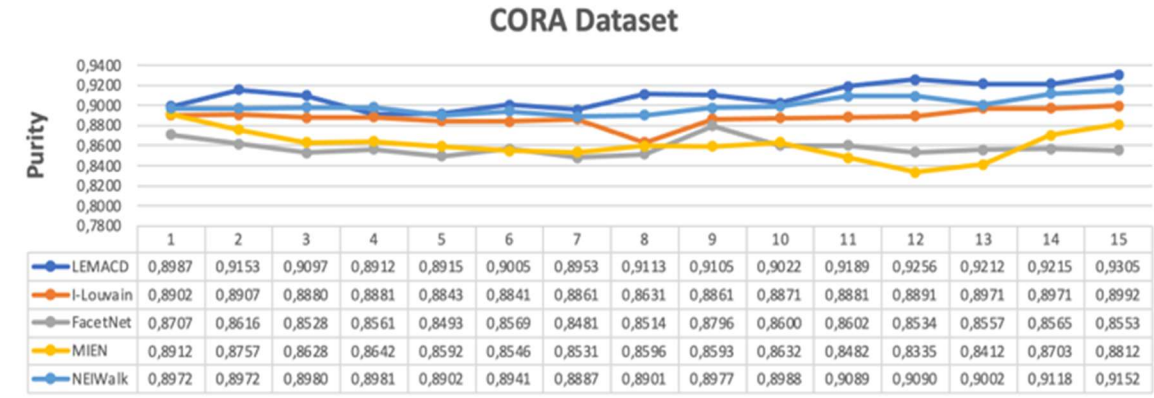


Figure 8: Comparison Results Of Purity Between LEMACD And Other Algorithms (CORA)

✓ DBLP : On the DBLP dataset (Figure 9), the LEMACD algorithm once again confirms its superiority in terms of purity. Its performance ranges between 0.91 and 0.94, with a peak observed around iteration 10. The I_Louvain algorithm ranks second, demonstrating consistent results around 0.89, yet it fails to reach the performance levels achieved by LEMACD. In contrast, the FacetNet and MIEN methods yield lower and less stable outcomes, with purity values dropping below 0.84 for FacetNet and below 0.85

for MIEN on several occasions. The NEWalk approach occupies an intermediate position, delivering results slightly below those of LEMACD but occasionally outperforming I_Louvain depending on the iteration. This overall trend highlights the robustness of LEMACD in accurately identifying communities within large-scale academic graphs, clearly outperforming other evaluated methods in terms of clustering quality.

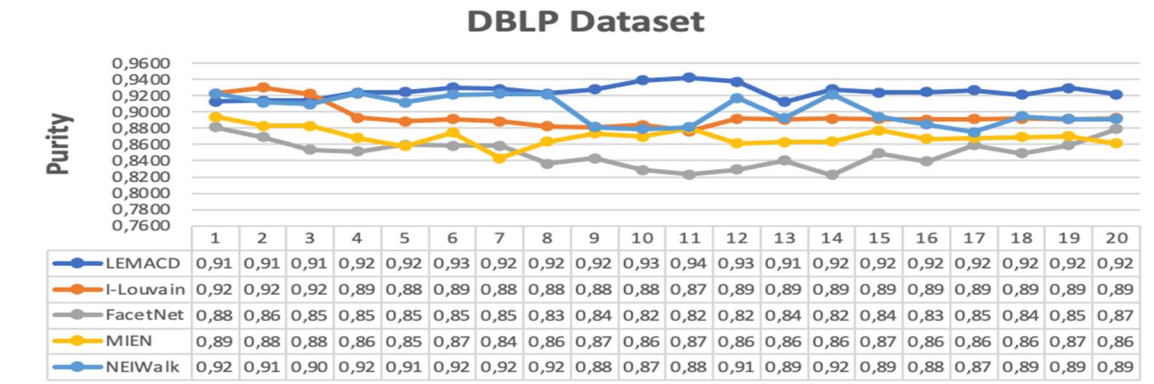


Figure 9: Comparison Results Of Purity Between LEMACD And Other Algorithms (DBLP)

5. CONCLUSION

In this paper, we proposed a two-step methodological model, named LEMACD, for community detection in evolving learning social networks. The first step, based on the LEMCD method, enables the identification of the initial community structure, while the second step ensures incremental and adaptive updates in response to network evolution.

The proposed framework relies on an approach that integrates both the network topology and the semantic content of nodes to identify communities with improved structural and semantic coherence. This approach is particularly relevant to online learning environments, as grouping learners with shared interests helps foster engagement and enhance educational outcomes. Our experiments on real-world networks demonstrate that the LEMACD algorithm is effective in detecting high-quality communities, both in terms of connectivity and user preferences, while ensuring fast and scalable handling of network modifications.

However, certain challenges remain. On online learning platforms, community structures continuously evolve due to learner enrollments and withdrawals, interpersonal interactions, and changes in educational content. As a result, between two time points t_i and t_j (with $j > i$), a learner community may expand, split, shrink, remain stable, or merge with another community. We define these transformations as critical events, which need to be monitored and analyzed to optimize the organization of learning groups. Yet, few studies explicitly address these dynamics in the context of online learning. As a first future research direction, we aim to extend LEMACD to dynamically track the evolution of learner communities and adapt educational pathways accordingly.

REFERENCES:

- [1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, et E. Lefebvre, « Fast unfolding of communities in large networks », *J. Stat. Mech. Theory Exp.*, vol. 2008, n° 10, p. P10008, oct. 2008, doi: 10.1088/1742-5468/2008/10/P10008.
- [2] M. E. J. Newman, « Modularity and community structure in networks », *Proc. Natl. Acad. Sci.*, vol. 103, n° 23, p. 8577-8582, juin 2006, doi: 10.1073/pnas.0601602103.
- [3] D. Zhuang, J. M. Chang, et M. Li, « DynaMo: Dynamic Community Detection by Incrementally Maximizing Modularity », *IEEE Trans. Knowl. Data Eng.*, p. 1-1, 2019, doi: 10.1109/TKDE.2019.2951419.
- [4] G. Rossetti et R. Cazabet, « Community Discovery in Dynamic Networks: A Survey », *ACM Comput. Surv.*, vol. 51, n° 2, p. 1-37, mars 2019, doi: 10.1145/3172867.
- [5] C.-D. Wang, J.-H. Lai, et P. S. Yu, « NEIWalk: Community Discovery in Dynamic Content-Based Networks », *IEEE Trans. Knowl. Data Eng.*, vol. 26, n° 7, p. 1734-1748, juill. 2014, doi: 10.1109/TKDE.2013.153.
- [6] S. Fortunato, « Community detection in graphs », *Phys. Rep.*, vol. 486, n° 3-5, p. 75-174, févr. 2010, doi: 10.1016/j.physrep.2009.11.002.
- [7] X. Zhang, R. Mo, H. Zhao, X. Luo, et Y. Yang, « RETRACTED: Statistical analysis of photodynamic therapy and stent drainage for unresectable cholangiocarcinoma », *Future Gener. Comput. Syst.*, vol. 91, p. 511-517, févr. 2019, doi: 10.1016/j.future.2018.09.028.
- [8] D. Combe, C. Largeron, M. Géry, et E. Egyed-Zsigmond, « I-Louvain: An Attributed Graph Clustering Method », in *Advances in Intelligent Data Analysis XIV*, vol. 9385, E. Fromont, T. De Bie, et M. Van Leeuwen, Éd., in Lecture Notes in Computer Science, vol. 9385, Cham: Springer International Publishing, 2015, p. 181-192. doi: 10.1007/978-3-319-24465-5_16.
- [9] M. Rosvall et C. T. Bergstrom, « Maps of random walks on complex networks reveal community structure », *Proc. Natl. Acad. Sci.*, vol. 105, n° 4, p. 1118-1123, janv. 2008, doi: 10.1073/pnas.0706851105.
- [10] Y. Sun, J. Han, X. Yan, P. S. Yu, et T. Wu, « PathSim: meta path-based top-K similarity search in heterogeneous information networks », *Proc. VLDB Endow.*, vol. 4, n° 11, p. 992-1003, août 2011, doi: 10.14778/3402707.3402736.
- [11] J. McAuley et J. Leskovec, « Discovering social circles in ego networks », *ACM Trans. Knowl. Discov. Data*, vol. 8, n° 1, p. 1-28, févr. 2014, doi: 10.1145/2556612.
- [12] M. E. J. Newman et M. Girvan, « Finding and evaluating community structure in networks », *Phys. Rev. E*, vol. 69, n° 2, p. 026113, févr. 2004, doi: 10.1103/PhysRevE.69.026113.
- [13] Z. Zhao, S. Feng, Q. Wang, J. Z. Huang, G. J. Williams, et J. Fan, « Topic oriented community detection through social objects and link analysis in social networks », *Knowl.-Based Syst.*, vol. 26, p. 164-173, févr. 2012, doi: 10.1016/j.knosys.2011.07.017.
- [14] I. Salhi, H. El, M. Qbadou, et K. Mansouri, « Towards the Identification of Student Learning Communities using Centrality », *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, n° 12, 2019,

- doi: 10.14569/IJACSA.2019.0101247.
- [15] A. Reihanian, B. Minaei-Bidgoli, et H. Alizadeh, « Topic-oriented community detection of rating-based social networks », *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 28, n° 3, p. 303-310, juill. 2016, doi: 10.1016/j.jksuci.2015.07.001.
- [16] M. Girvan et M. E. J. Newman, « Community structure in social and biological networks », *Proc. Natl. Acad. Sci.*, vol. 99, n° 12, p. 7821-7826, juin 2002, doi: 10.1073/pnas.122653799.
- [17] U. N. Raghavan, R. Albert, et S. Kumara, « Near linear time algorithm to detect community structures in large-scale networks », 2007, doi: 10.48550/ARXIV.0709.2938.
- [18] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, et B. L. Tseng, « Facetnet: a framework for analyzing communities and their evolutions in dynamic networks », in *Proceedings of the 17th international conference on World Wide Web*, Beijing China: ACM, avr. 2008, p. 685-694. doi: 10.1145/1367497.1367590.
- [19] T. N. Dinh, Ying Xuan, et M. T. Thai, « Towards social-aware routing in dynamic communication networks », in *2009 IEEE 28th International Performance Computing and Communications Conference*, Scottsdale, AZ, USA: IEEE, déc. 2009, p. 161-168. doi: 10.1109/PCCC.2009.5403845.