Knowledge and Information Systems
https://doi.org/10.1007/s10115-024-02201-8

REVIEW

Community detection in social networks using machine learning: a systematic mapping study

Mahsa Nooribakhsh[1] · Marta Fernández-Diego[1] · Fernando González-Ladrón-De-Guevara[1] · Mahdi Mollamotalebi[2]

Received: 8 June 2023 / Revised: 19 July 2024 / Accepted: 30 July 2024

Abstract

One of the important issues in social networks is the social communities which are formed by interactions between its members. Three types of community including overlapping, non-overlapping, and hidden are detected by different approaches. Regarding the importance of community detection in social networks, this paper provides a systematic mapping of machine learning-based community detection approaches. The study aimed to show the type of communities in social networks along with the algorithms of machine learning that have been used for community detection. After carrying out the steps of mapping and removing useless references, 246 papers were selected to answer the questions of this research. The results of the research indicated that unsupervised machine learning-based algorithms with 41.46% (such as k means) are the most used categories to detect communities in social networks due to their low processing overheads. On the other hand, there has been a significant increase in the use of deep learning since 2020 which has sufficient performance for community detection in large-volume data. With regard to the ability of NMI to measure the correlation or similarity between communities, with 53.25%, it is the most frequently used metric to evaluate the performance of community identifications. Furthermore, considering availability, low in size, and lack of multiple edge and loops, dataset Zachary's Karate Club with 26.42% is the most used dataset for community detection research in social networks.

✉ Mahsa Nooribakhsh
mnoorib@doctor.upv.es

Marta Fernández-Diego
marferdi@omp.upv.es

Fernando González-Ladrón-De-Guevara
fgonzal@omp.upv.es

Mahdi Mollamotalebi
motalebi@qiau.ac.ir

1 Instituto Universitario Mixto de Tecnología Informática, Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Spain

2 Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

Published online: 12 August 2024

Keywords Community detection · Social network · Systematic mapping study · Machine learning

1 Introduction

The term social network has emerged in the recent decade, and now it is highly prevalent around the world. In fact, social networks use computer networks (especially the Internet) as their infrastructure. Many different aspects (political affairs, technical experiments, art, etc.) may be discussed on social networks. One of the important issues in such networks that this research focuses on is the social communities that are formed by users. Such communities are similar to real-world communities, but they exist in different environments and use special tools. A community is created by interactions between the members of the social network [1].

Analyzing social networks allows us to discover the relationships between members by exploiting network and graph theories for many useful purposes. One important characteristic of social networks is the community structure, i.e., individuals with high-density interactions in the same group and comparatively low-density interactions among individuals of different groups [2]. The importance of social communities is that they can indicate the relation between users. Such relations subsequently can be useful to analyze and predict user behavior. Predictions are beneficial in different aspects such as business, education, politics, knowledge discovery, crime detection, and so on. As an example, a search engine can narrow down its searching source by considering only topically related subsets of web pages. This reduces the processing load and response time of the search.

The social network can be represented as a graph, $G = (V, E)$, where V is a set of nodes and E is a set of edges. This graph is made of individuals (or organizations) as nodes, which are connected by one or more types of interdependencies (such as friendship, financial exchange, beliefs, etc.) as edges (links) [3]. A community in the social network can be stated as a subgraph such that the edge density within the subgraph is greater than the edge density between its nodes and nodes outside [4, 5]. Social networks are categorized as static and dynamic. Although detecting communities in static networks are simpler than in dynamic ones, the latter has gained popularity in recent years. Different technologies used on the Internet and the growth of social networks attract many people to join them for business, education, politics, etc. [6, 7].

Our study on systematic mapping of machine learning-based community detection methods in social networks mapped 246 papers and conducted a thorough analysis of machine learning algorithms for community detection. This comprehensive evaluation guarantees a wide-ranging and all-encompassing comprehension of the subject. In contrast to other works that classify community detection algorithms based on factors like statistical methods, semantics, or computational characteristics, our research directly centers on machine learning methodologies. This specific focus enables a more thorough examination of how machine learning and its sub-categories improve community detection. Our study specifically looks into detecting overlapping, non-overlapping, and hidden communities. By taking into account these community categories, we provide a detailed perspective on the obstacles and solutions in community detection.

Furthermore, we point out the growing adoption of deep learning starting from 2020, underscoring its benefits in processing extensive datasets. This trend analysis offers important insights into the future of community detection research. Our research shows that normalized

mutual information (NMI) is the most commonly used metric, with the Zachary's Karate Club dataset being the most frequently utilized dataset. The thorough assessment assists researchers in selecting suitable tools and benchmarks for their studies. Our study on the practical uses of machine learning in community detection offers valuable insights for both academic and industry professionals looking to apply these techniques in real-world situations.

The remainder of this paper is organized as follows: Sect. 2 presents related work, Sect. 3 details the research methodology, and Sect. 4 provides the answer to the research question, and the primary studies are discussed in Sect. 5. Finally, Sect. 6 concludes the research and presents future works.

2 Related work

Several survey papers have been published for community detection on social networks. Fortunato et al. [8] classified community detection algorithms and explained the most methods developed focusing on statistical techniques and evaluation approaches. Coscia et al. [9] classified them according to the adopted definition of the community and they provided a manual for the community discovery problem based on features such as the size of the network, the direction of edges, and multidimensionality. Plantié et al. [10] classified community detection algorithms based on both semantics and type of output. Semantics opens up new perspectives and allows the interpretation of high-order social relations. Malliaros et al. [11] handled community detection by directed graphs and presented a methodology-based taxonomy of different approaches. They presented the relevant work along two orthogonal classifications; the first one concerned with the methodological principles of the clustering algorithms and the second one the methods from the viewpoint regarding the properties of a good cluster in a directed network. They presented methods and metrics for evaluating graph clustering results.

Azaouzi et al. [12] provided a taxonomy of community detection methods based on the computational nature (either centralized or distributed) of static and dynamic social networks. Vieira et al. [13] presented a comparative study of representative methods for overlapping community detection from the perspective of the structural properties of the communities. Yassine et al. [14] provided a systematic literature review to assess the use of community detection techniques in social networks. Their analysis covered 65 studies to analyze users' interaction patterns. They highlighted the need to include automated community discovery techniques in online learning environments to facilitate and enhance their use.

Community detection in social networks is a challenging task due to its NP-hard nature [12]. In the context of community detection, NP-hard problems refer to the difficulty of finding the optimal partitioning of nodes into communities that maximizes certain objective functions [15]. The classification of community detection as an NP-hard problem underscores the need for developing scalable and efficient algorithms to handle real-world networks. It also highlights the challenges faced by researchers in finding the best possible solutions or approximations within a reasonable time frame [12, 16]. There are several NP-hard problems that are related to community detection in social networks such as extracting maximum modularity, balanced graph partitioning, maximum clique optimization, and multi-objective community detection.

There are some challenges related to static, dynamic, attributed, edge-weighted, spatial, and temporal networks [8]. Static community detection refers to the process of identifying communities within a network that is assumed to be static, meaning that the relationships

between nodes and edges do not change over time. In static community detection, algorithms aim to partition the network into groups or clusters of nodes that have a higher density of connections within the group compared to connections outside of the group. These algorithms typically analyze the network's topology, such as node degrees and edge connectivity, to identify cohesive groups. Static community detection is widely used in various fields, including social network analysis, biology, and computer science, to understand the structure and organization of complex networks. The challenge of static community detection in networks lies in accurately identifying communities that have stable structures and compositions, where nodes are assigned to specific communities based solely on their static interactions [8, 17].

Dynamic community detection in networks is the study of identifying and tracking evolving communities over time. Unlike static community detection, where the network is treated as a single snapshot, dynamic community detection takes into account the temporal aspect of the network, allowing for the detection of communities that change and evolve over time. This presents a unique set of challenges, including the determination of appropriate time scales for community detection, handling the dynamic nature of communities, and capturing the evolution and transitions between communities. Additionally, the scalability of dynamic community detection algorithms becomes even more critical, as the size and complexity of the network can grow exponentially with time. Overall, dynamic community detection provides a deeper understanding of how communities form, dissolve, and transform in networks, enabling insights into the underlying dynamics and processes at play [17].

Attributed community detection in networks is the study of identifying and tracking communities based on both the network structure and additional attributes associated with the nodes or edges. Unlike traditional community detection, which solely relies on connectivity patterns, attributed community detection takes into account the additional information available, such as node attributes or edge weights. This approach allows for a more comprehensive understanding of the communities by considering not only the network topology but also the characteristics of the nodes or edges within the communities. By incorporating attributes, attributed community detection algorithms can uncover communities that are not solely based on connectivity but also on shared attributes or behaviors, providing insights into the underlying factors driving community formation and dynamics in complex networks. One of the main challenges in attributed community detection in networks is the integration and interpretation of multiple types of attributes. Networks often contain diverse types of attributes, such as categorical, numerical, or textual information, which may have different levels of relevance and importance in determining community structure [18, 19].

Edge-weighted community detection in networks is a challenging task due to the need to consider the varying weights assigned to edges. These weights reflect the significance of connections between nodes and have a profound impact on community formation and structure. To address this, specialized algorithms and techniques are required to accurately identify community boundaries and optimize community quality measures. However, dealing with these weights requires specialized algorithms and techniques, as they introduce complexities in determining community boundaries and optimizing community quality measures [8].

Spatial community detection in networks involves identifying communities based on the spatial location of nodes. This approach takes into account the physical proximity of nodes and considers the spatial relationships between them to determine community boundaries. By incorporating spatial information, this method can reveal communities that are geographically close and have a higher likelihood of interacting, providing insights into the spatial organization and structure of networks. One challenge of spatial community detection in networks is the presence of overlapping communities, where nodes can belong to multiple communities simultaneously. This makes it difficult to define clear boundaries and accurately

assign nodes to specific communities based solely on their spatial proximity. Additionally, the spatial distribution of nodes may not always align with the underlying community structure, leading to potential misinterpretation of community boundaries and relationships [20].

Temporal community detection in networks involves identifying communities that evolve over time. This presents a challenge as the structure and composition of communities can change dynamically, making it harder to define consistent boundaries. Additionally, the presence of overlapping communities in the temporal dimension further complicates the accurate assignment of nodes to specific communities based solely on their temporal interactions. The challenge of temporal community detection in networks lies in accurately identifying evolving communities that have dynamically changing structures and compositions. Defining consistent boundaries becomes difficult due to these dynamic changes, and the presence of overlapping communities in the temporal dimension further complicates the accurate assignment of nodes to specific communities based solely on their temporal interactions [21–23].

Different methods of community detection have been proposed in the literature and classification of them can be based on various criteria, including the underlying approach, the objective or criteria for partitioning, and the nature of the network being analyzed. Therefore, there is no universally accepted classification of community detection algorithms. A classification of community detection methods for community detection in social networks includes graph partitioning algorithms, density-based algorithms, hierarchical clustering algorithms, spectral clustering, optimization algorithms, game-theoretic algorithms, and deep learning algorithms [8, 24–27].

Community detection in social networks using graph partitioning divides the network into distinct groups (communities) of nodes, aiming to maximize intra-community connections and minimize inter-community connections. Even though such algorithms are scalable, they suffer from the resolution limit, where small communities within larger ones may not be accurately identified due to limitations in the modularity optimization. Moreover, the choice of initial conditions can impact the final community structure, leading to potential variations in results and the need for multiple runs or refined initialization strategies [8].

Optimization-based algorithms use optimizing parameters or decisions made during the process. These algorithms utilize optimization techniques to find the best possible partitioning of nodes into communities. Even though such algorithms improve accuracy they suffer from computational complexity and parameter sensitivity challenges [26].

Game-theoretic algorithms model the interactions between nodes as a game, where each node aims to maximize its own utility or payoff. A game-theoretic model is composed of three elements: players, strategies, and utilities. Even though such algorithms can handle noisy or incomplete network data, they assume that nodes act rationally and strategically, which may not always hold true in social networks. Moreover, their performance relies on the accuracy of the underlying assumptions made about node behavior and interactions [19, 27, 28].

The remaining algorithms including density-based, hierarchical clustering, spectral clustering, and deep learning are different types of machine learning-based algorithms and they are described in Sect. 4 along with their advantages and disadvantages. Taking into account that most of the algorithms for detecting the communities in social networks are based on machine learning and considering the frequency of research related to this approach, machine learning is the most popular approach for community detection in social networks based on the literature.

Some of the reasons for the benefits of machine learning are its significant advantages in terms of scalability, automation, accuracy, and adaptability. Machine learning algorithms enable the analysis of large-scale networks, which is crucial as the size and complexity

of real-world networks continue to grow. Traditional community detection methods often struggle to handle such large networks efficiently. It also allows for the automation of the community detection process, reducing the need for manual intervention and human bias. This enables faster and more objective analysis of networks. These algorithms can leverage complex patterns and relationships in network data to improve the accuracy of community detection. They can identify subtle community structures that may be missed by traditional methods. Moreover, they can adapt and learn from new data, making them suitable for dynamic networks where communities evolve over time. They can continuously update and refine community detection models based on changing network characteristics [29, 30].

In the following, conventional and new machine learning techniques/methods are introduced.

With regard to deep learning as one of the machine learning approaches, the classic methods of community detection such as statistical inference, and traditional machine learning are falling by the wayside as deep learning techniques demonstrate an increased capacity to handle high-dimensional graph data with impressive performance. Liu et al. [31] summarized the contributions of the various frameworks, models, and algorithms in streams of deep neural networks, deep graph embedding, and graph neural networks, along with their unsolved challenges.

In conventional machine learning methods, community detection was considered a problem of forming, extracting, and verifying the accuracy of clusters. Such representation of a low-dimensional space was linear while real-world networks include nonlinear structures and reduce the use of conventional strategies [32]. On large scales, more efficient techniques are required to achieve high performance [33, 34]. Souravlas et al. [35] reviewed recent deep learning-based approaches for community detection in social networks. They have selected to present papers the majority of which have been published between 2019 and 2020 and focused on big data networks.

Su et al. [24] reviewed community detection based on deep learning as a time-lined that introduced community detection applications in the real world. It collected open resources, including benchmark datasets, evaluation metrics, and technique implementations, and covered data mining, artificial intelligence, machine learning, and knowledge discovery. Alotaibi et al. [2] presented a survey that highlights the characteristics and challenges of community detection in dynamic social networks.

The above-introduced papers present the common research trends in community detection. It is noteworthy that there is no single algorithm highly appropriate for all varieties of data, circumstances, and applications. Regarding the importance of community detection in social networks and presenting a vast amount of recent research, there is a need to gather, synthesize, and validate evidence from existing studies to build a corpus of knowledge aiming to present the direction of researchers and practitioners. This paper provides a systematic mapping of community detection approaches that are based on machine learning.

3 Methodology

The systematic literature review provides a clear understanding of a given subject as unbiased. Such research synthesizes multiple studies in one [36]. However, literature review papers address the efficiency of different methods, a systematic mapping study (SMS) classifies primary studies as statistical [37]. It maps out a research area by different classifications of studies and identifies research trends by analyzing the statistics of publications. Hence, it

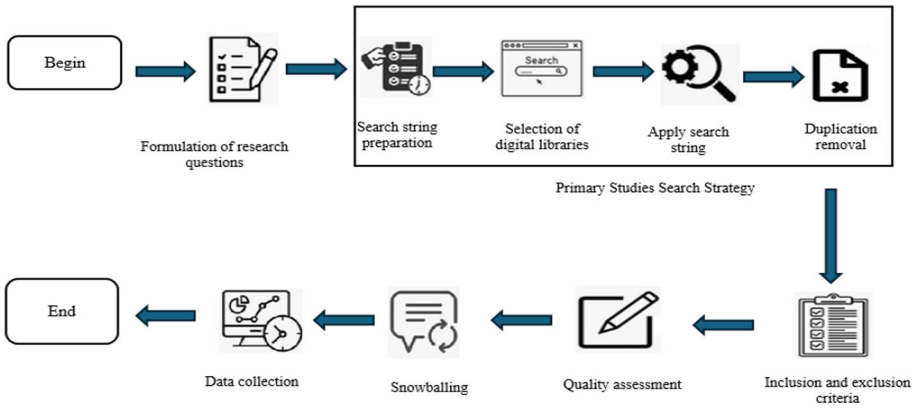


Fig. 1 Steps involved in the systematic mapping

helps to identify gaps in current studies to make a roadmap for future studies [38]. Some guidelines to produce SMSs are available in the literature [37, 39].

SMSs require searching the related documents and performing an accurate inclusion and exclusion process. The data extraction process of such studies is broader than usual surveys or review papers as it includes a classification or categorization stage. The analysis process in mapping studies summarizes the data through statistical presentations [40]. This section provides the steps of our systematic mapping study, following the Petersen method [37] including formulation of research questions, primary studies search strategy, inclusion and exclusion criteria, quality assessment, snowballing and data collection [37]. The primary studies search strategy consists of a) search string preparation b) selection of digital libraries c) apply search string d) duplication removal. Figure 1 shows the steps involved in the systematic mapping.

3.1 Formulation of research questions

The most popular social networks (Facebook, Twitter, and LinkedIn) began after 2000 [41]; therefore, this research reviewed papers that were published between 2000 until July 2024. The review aimed to attain the answer to the following research questions:

RQ1. What algorithms of machine learning have been used for community detection in social networks?
RQ2. Which metrics/parameters are used to distinguish communities in social networks? (E.g., accuracy, running time).
RQ3. Which datasets are used for community detection in social networks?
RQ4. What types of communities are on social networks?

3.2 Search strategy for primary studies

3.2.1 Search string preparation

To find the least number of documents that are most relevant to the subject of our research, the search phrase has been formed as limited to the specific metadata (Title, Abstract, and Keywords) of the publications in the search process. The prepared search phrase is as follows:

> ("social network" OR "social networks" OR "social networking") AND ("community detection" OR "community discovery" OR "community recognition" OR "community search" OR "community identification") AND ("machine" OR "supervised" OR "unsupervised" OR "semi-supervised" OR "reinforcement" OR "ensemble" OR "instance-based" OR "learning" OR "clustering" OR "deep" OR "regularization").

3.3 Selection of digital library

Selecting the most related peer-reviewed and reliable publications is highly effective in enriching the knowledge of research aspects and issues. There are some well-known scientific databases (e.g., IEEE and ACM) to index published papers and books. We investigated the capabilities of several search databases and chose five of them (ACM, Scopus, Web of Science, IEEE Xplore, and Springer) for our research regarding their usability and validity. Further, the selected search databases are the most related ones to the computer science research resources [42, 43].

3.4 Apply search string

Based on the results of applying the search phrase, the extracted documents are as follows: 165 from Springer, 107 from IEEE, 52 from ACM, 154 from Scopus, and 111 from WoS. It can be noted that Springer is not capable of limiting the search based on Title, Keyword, and Abstract; thus, first, the documents were searched as full text, and 7731 results were obtained. Subsequently, the results were filtered based on Title, Keyword, and Abstract using the Endnote software capabilities.

The search database ACM has two databases to search; thus, the search has been done within both of its databases and then, the results were merged. After merging the results, some duplicated documents appeared that have been subsequently removed in the next steps. The search databases IEEE and ACM were not capable of searching within Title, Keyword, and Abstract simultaneously; therefore, the documents were searched separately on the above fields, and then, the results were merged. After merging the results, some duplicated documents appeared that were removed. On the other hand, employing EndNote as an analyzing tool resulted in some difficulties as it could not extract Keywords from the imported references. Therefore, we handled the Keyword imports manually.

It should be emphasized that wildcards are not supported in all the search databases; therefore, to keep uniformity for all the search databases, the search phrase was prepared consisting of derivations of Keywords (e.g., social network, social networks, social networking). The documents found during the search process were of different types including journal papers, conference papers, conference proceedings, books, and book chapters. After completion of the search process, 589 references were obtained before deleting duplicated records. Table 1 shows the statistics of documents before duplication removal.

Table 1 The statistics before duplication removal

Nos.	Search database	Book	Journal paper	Book chapter/section	Conference paper[a]	Sum of found documents
1	Scopus	0	56	3	95	154
2	IEEE	1	23	1	82	107
3	ACM	13	21	1	17	52
4	WoS	0	58	3	50	111
5	Springer	3	62	11	89	165
Sum		17	220	19	333	589

[a]Including conference papers, conference proceedings, and serials (The serial contains all the papers that are published in a conference. So, it is equivalent to conference proceeding)

Table 2 The statistics after duplication removal

Nos.	Search database	Book	Journal	Book chapter/section	Conference paper*[a]	Sum of found documents
1	Scopus	0	29	1	74	104
2	IEEE	0	16	0	68	84
3	ACM	12	15	0	11	38
4	WoS	0	41	1	34	76
5	Springer	1	55	9	72	137
Sum		13	156	11	259	439

[a]Including conference papers, conference proceedings, and serials

3.5 Duplication removal

As remarked above, the result of each search database contains some duplications. We sorted the documents based on Title to find the duplications. Some reasons for duplications were as follows: a) the same documents with different publication years; we removed the older ones for such cases b) the same documents with differences in Keyword order, and format of authors' naming; we removed the duplications manually in these cases c) the same documents in the result of different search databases; we found and removed the repeating ones.

After eliminating all the above-mentioned duplications from the results, 439 documents remained. Table 2 presents the statistics of the results after the duplication removal process.

At this point, we have the non-duplicated results of the primary study search phase. In the next step, the inclusion and exclusion criteria would be applied to the results.

3.6 Inclusion and exclusion criteria

Inclusion and exclusion criteria are used to assess the gathered documents. In the phase of the primary study, 439 documents remained after duplication removal. In this phase, in order to choose the documents most related to our scope, we used some filters for inclusion and

Table 3 Results after full-text screening

Reason for exclusion	Count of papers
Total from primary studies	439
a1) Not published in a conference or journal	99
a2) Not written in English	3
b1) Not full-text available	3
b2) Not related to community detection in social networks based on machine learning	20
Subtotal papers eliminated	125
Remaining total	314

exclusion. The documents are assessed accurately by the authors to remove the irrelevant documents.

The inclusion process had two constraints as the following: a1) peer-reviewed conference or journal papers: sometimes scientific search databases construe the documents differently; for instance, some documents in the results are considered as serial or conference proceeding while they do not contain a full paper. Thus, in Table 3, documents that are not published in a conference or journal are excluded. a2) only the papers that are written in English. Thus, documents that are not written in English are excluded.

In addition, the exclusion process had two constraints as the following: b1) to eliminate studies not accessible as full text. Three documents had not contained the full text of the paper. b2) papers not related to community detection in social networks based on machine learning. At this step, all the documents have been read to exclude cases that are not exactly related to community detection in social networks based on machine learning. At the end of this step, 314 documents have remained. Table 3 presents a summary of the results of this stage.

3.6.1 Quality assessment

The quality assessment rules were applied to the results of the inclusion and exclusion phase in order to evaluate the papers in accordance with the research questions. Based on Kitchenham et al. [40], eleven rules were identified. In addition, we considered two rules (12 and 13) for a more detailed evaluation that each one is worth a mark out of 1. The quality assessment rules 1 to 13 are scored as follows: "fully answered" = 1, "above average" = 0.75, "average" = 0.5, "below average" = 0.25, "not answered" = 0. The scoring of quality assessment rules 12 and 13 is identified separately as mentioned in the following. The overall score of each paper is the summation of the marks obtained for each rule. Papers with an overall score of less than 6.5 are considered low quality and they are removed from the final set of results. The final set of the results after completion of the quality assessment is presented in Table 7 of Online Appendix A. The rules that were used to assess the quality of papers are as follows:

1. Is the problem of the research clearly stated?
2. Has the literature been reviewed appropriately?
3. Are the aims of the research clearly specified?
4. Is the scope of the research clearly defined?
5. Is the contribution/novelty of the research specified?

6. Have the proposed technique's details been clearly described (shown by algorithm/flowchart/…)?
7. Have the results been compared with multiple recent research?
8. Are the comparisons based on different metrics?
9. Is the data collection process adequately described?
10. Are the results appropriately shown and interpreted/justified?
11. Has the experimental setup (HW/SW) been defined clearly?
12. Has the publication reached enough citations so far (1: at least four per year, 0.75: at most three per year, 0.5: at most two per year, 0.25: at most one per year)?
13. What is the quality of the publisher? (1: Journal ISI-IF-Quartile $< = 3$, 0.75: Journal ISI-IF, 0.5: International Journal, 0.25: valid conferences, 0: non-Indexed publications)

3.7 Snowballing

In the snowballing phase, the references found in the selected papers are used to identify additional documents matched our search criteria [36]. This backward snowballing was performed after the quality assessment. The process is done recursively such that some related references are chosen to apply the above steps to them. Then, a snowballing is carried out on the remaining documents until no new references were identified.

The first run of 221 documents directed us to find 49 more documents. Twenty-nine out of 49 documents did not pass the quality criteria. Thus, only 20 new documents were added to the results. Five new documents were added to the results at the second iteration of the snowballing. Consequently, a total set of 246 documents were obtained at the end of the snowballing process (see in Online Appendix B).

3.8 Data collection

After filtering, the most relevant information was obtained from each of the 246 remaining studies. This includes both general information and data addressing the five research questions. After reading each paper, the data was extracted and stored in a spreadsheet using the data extraction form. The set of 246 primary studies is listed in Online Appendix B and analysis of them is presented below in two sections. First, the bibliometric features of the final subset of studies were described focusing on the year of publication and the type of paper. The second section deals with the research questions stated in Sect. 3.1. The final set of the results at each phase is listed in Online Appendix A. Figure 2 shows the distribution of the primary studies according to the type of document (journal papers or conference papers) and the year of publication. One hundred and eight out of the 246 primary studies have been published in conferences, while the remaining 138 have been published in journals.

Regarding geographical distribution, the primary studies have been authored by researchers from a reduced set of countries, 42 in total, mainly in Asia (66.66%) and the USA (13.41%), see Fig. 3. This analysis is restricted to the institutional affiliation of the first author. The country with the most publications is China (96 papers equivalent to 39.02%), followed by the India (papers equivalent to 11.38%) and USA (papers equivalent to 10.56%). Other countries had 96 papers equivalent to 39.02% of publications, see Fig. 4.

Figure 5 shows a summary of the process followed to select the documents along with the results obtained after each phase.

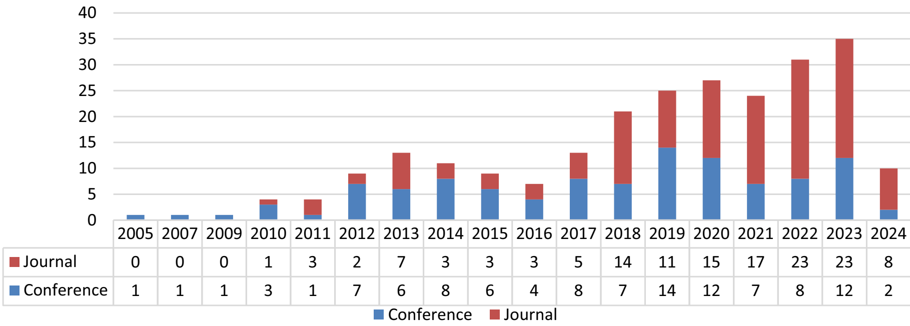


Fig. 2 Number of papers published in journals and conferences per year

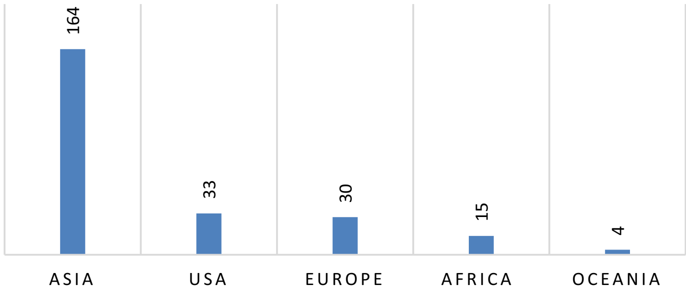


Fig. 3 Number of papers published per continent

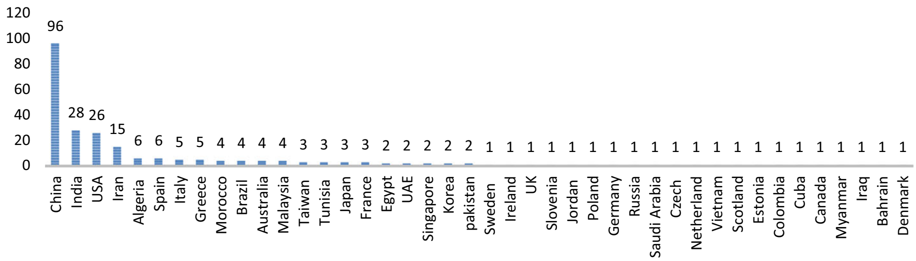


Fig. 4 Number of papers published per country

4 Answer to the research question

This section includes the results derived from the data extracted from the primary studies. These results allowed us to answer the research questions referred to in this study. Data extraction was performed by the same team of four researchers and followed the same procedure as those employed to select the primary studies. The percentages (shown in Tables 4, 5, 6, 7, 8, and 9) were calculated as the number of papers (column 4) divided by the total number of 246 primary papers.

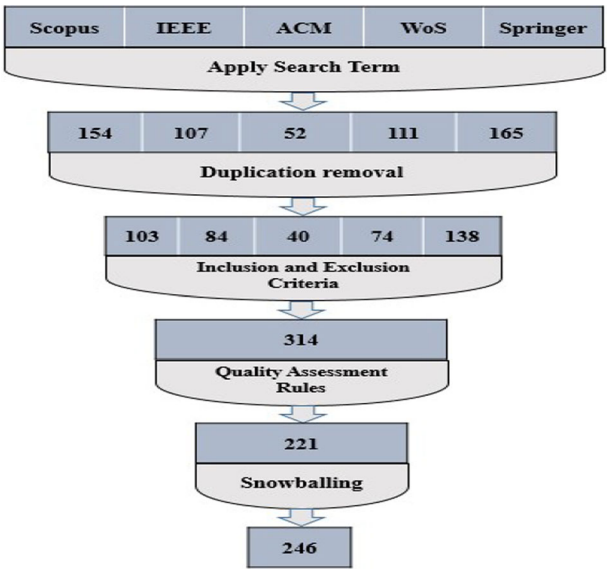


Fig. 5 The selection process of primary study

4.1 RQ1: What algorithms of machine learning have been used for community detection in social networks?

This research question has been addressed by the following three sub-questions:

4.1.1 Types of machine learning in community detection in social networks

The research that handles community detection in social networks concentrates on seven general categories including supervised learning, unsupervised learning, semi-supervised learning, deep learning, and other types (ensemble-based learning, instance-based learning, and reinforcement learning) [44–46]. In the following, these machine learning types are described concisely. Supervised learning is used to predict tasks and it aims to forecast or classify outcomes of interest such as mental disorder classification. In supervised (or Inductive) machine learning, the goal is to learn a function of predicting the values of a class [47, 48]. Unsupervised machine learning is applicable when the training data is not labeled. It does not identify the appropriate output. It analyzes the dataset to discover hidden patterns from unlabeled data [49]. Semi-supervised machine learning acts like supervised and also unsupervised machine learning techniques. It uses labeled and also unlabeled data for training [50]. Deep learning uses supervised and unsupervised techniques and learns multi-level features hierarchically in order to classify data and recognize patterns [51].

Reinforcement learning (RL) refers to methods that reward desired behaviors and punish undesired behaviors. The agents seek long-term and maximum overall rewards to achieve an optimal solution. Long-term goals prevent agents from stalling on lesser goals. The agents learn to avoid the negative and seek the positive gradually [52]. Ensemble learning methods use multiple machine learning algorithms to predict based on features extracted from data and integrate the results with voting to achieve higher performance [53, 54]. Instance-based

Table 4 The references using different types of machine learning algorithms

	Types of machine learning	ID	#	%
1	Supervised learning	A16, A17, A25, A34, A39, A42, A44, A50, A54, A73, A76, A88, A101, A106, A107, A109, A111, A128, A135, A137, A158, A185, A186, A196, A197, A206	26	10.56
2	Unsupervised learning	A2, A3, A4, A5, A6, A9, A11, A12, A13, A14, A20, A21, A22, A24, A26, A28, A29, A32, A33, A36, A38, A43, A45, A47, A48, A49, A52, A53, A55, A56, A57, A58, A59, A60, A62, A64, A65, A68, A69, A70, A74, A79, A87, A89, A90, A92, A93, A95, A102, A105, A114, A115, A116, A117, A119, A122, A123, A124, A129, A133, A136, A138, A141, A143, A144, A146, A148, A150, A152, A156, A157, A159, A160, A165, A167, A170, A175, A178, A181, A188, A189, A193, A195, A200, A204, A208, A209, A214, A217, A220, A223, A224, A225, A227, A229, A230, A231, A235, A238, A240, A245, A246	102	41.46
3	Semi-supervised learning	A10, A15, A18, A19, A27, A41, A46, A51, A63, A67, A72, A77, A83, A85, A86, A91, A100, A103, A118, A121, A127, A131, A134, A140, A145, A149, A151, A153, A154, A155, A161, A162, A163, A166, A168, A171, A173, A174, A176, A183, A184, A187, A191, A198, A201, A202, A205, A216, A226, A232, A237, A241	52	21.13
4	Deep learning	A1, A8, A23, A30, A31, A37, A40, A75, A78, A80, A81, A84, A94, A96, A99, A108, A110, A112, A113, A125, A126, A130, A139, A147, A169, A177, A179, A182, A194, A199, A203, A207, A210, A211, A212, A213, A215, A218, A221, A222, A228, A234, A236, A239, A242, A243, A244,	47	19.10
5	Ensemble base learning	A7, A35, A61, A66, A82, A104, A120, A142, A164, A180	10	4.06
6	Reinforcement learning	A98, A132, 172, 190, A233	5	2.03
7	Instance base learning	A71, A97, 192, 219	4	1.62

learning produces a class prediction using the query similarity to the nearest neighbors in the training set. This type of learning stores all the data and then finds an answer from the examination of the query's nearest neighbors [44, 55].

Table 4 shows the references that have used different types of machine learning algorithms for detecting communities in social networks along with the statistical values of their count. The types extracted from the primary studies were identified as supervised learning, unsupervised learning, semi-supervised learning, deep learning, etc. Based on the statistical analysis, most of the researchers used unsupervised learning algorithms for detecting the communities in social networks with 1.46% followed by semi-supervised learning with 21.13%. On the

Table 5 Sub-categories of machine learning used for community detection in social networks

	Type of machine learning		ID	#	%
1	Supervised learning	Classification/probabilistic classifier	A16, A25, A34, A39, A42, A44, A50, A73, A88, A107, A109, A111, A128, A135, A137, A185, A186, A196, A197, A206	20	8.13
		Liner classifier	A17, A54, A76, A101, A106, A158	6	2.43
2	Unsupervised learning	Clustering	A3, A4, A5, A6, A9, A11, A13, A14, A20, A21, A22, A24, A28, A29, A32, A33, A36, A38, A47, A48, A58, A59, A62, A65, A68, A79, A90, A92, A93, A95, A102, A105, A115, A116, A119, A136, A138, A144, A146, A150, A152, A157, A159, A160, A165, A167, A178, A181, A195, A204, A209, A217, A220, A223, A225, A227, A230, A231, A240, A246	60	24.39
		Density-based spatial clustering/(DBSCAN)	A53, A55, A57, A60, A64, A69, A74, A117, A124, A129, A170, A189, A193, 214, A229	15	6.09
		Spectral clustering	A26, A49, A52, A70, A87, A89, A123, A133, A143, A148, A156, A175, A208, A238	14	5.69
		Representation learning clustering: A2, A43, A45, A56, A141, A188, A200, A224, A235, A245(4.06%) Expectation maximization (EM): A12, A114 (0.81%) Incremental learning based on LPA: A122 (0.40%)	A2, A12, A43, A45, A56, A114, A122, A141, A188, A200, A224, A235, A245	13	5.28

Table 5 (continued)

	Type of machine learning		ID	#	%
3	Semi-supervised Learning	Heuristic Label propagation algorithm	A18, A46, A67, A72, A83, A91, A103, A134, A149, A151, A153, A154, A155, A161, A162, A166, A168, A171, A173, A176, A183, A184, A187, A191, A201, A216, A226, A232	28	11.38
		Graph theory	A10, A15, A41, A51, A63, A77, A118, A131, A140, A145, A198, A205	12	4.87
		Semi-supervised representation learning (RL): A19, A85, A237 (1.21%) Nonnegative matrix factorization (NMF): A86, A100, A163, A174, A202, A241(2.43%) Active learning: A121, A127 (0.81%) Co-training algorithm: A27(0.40%)	A19, A27, A85, A86, A100, A121, A127, A163, A174, A202, A237, A241	12	4.87
4	Deep learning	Auto-encoder	A8, A78, A84, A94, A96, A108, A110, A125, A130, A139, A147, A177, A179, A194, A207, A211, A213, A215, A222, A228, A239	21	8.53

Table 5 (continued)

	Type of machine learning		ID	#	%
		Graph convolutional neural network (CNN): A31, A40, A75, A99, A113, A169, A199, A212, A218, A221, A236, A242, A243(5.28%) Graph neural network (GNN): A1, A23, A30, A81, A182, A210, A234, A244 (3.25%) Recurrent neural network (RNN): A37, A80, A112, A203 (1.62%) Restricted Boltzmann machines (RBM): A126 (0.40%)	A1, A23, A30, A31, A37, A40, A75, A80, A81, A99, A112, A113, A126, A169, A182, A199, A203, A210, A212, A218, A221, A234, A236, A242, A243, A244	26	10.56
5	Ensemble-based learning	Similarity matrix: A35 (0.40%) Bagging: A82 (0.40%) Stacking: A61, A164 (0.81%) Boosting: A66 (0.40%) Random forest: A7 (0.40%) Parallel: A104 (0.40%) Ensemble clustering: A120, A142, A180 (1.21%)	A7, A35, A61, A66, A82, A104, A120, A142, A164, A180	10	4.06
6	Reinforcement learning	Policy gradient: A98 (0.40%) Q-Learning: A132, A172, A233 (1.21%) Other: A190(0.40%)	A98, A132, A172, A190, A233	5	2.03
7	Instance base learning	k-nearest neighbor graph (k-NN graph)	A71, A97, A192, A219	4	1.62

Table 6 metrics/parameters are used to distinguish communities in social networks

Metric	ID	#	%
NMI	A1, A6, A8, A10, A11, A12, A18, A21, A23, A29, A30, A33, A35, A41, A43, A45, A46, A47, A48, A53, A54, A56, A58, A63, A64, A65, A68, A69, A72, A73, A74, A75, A78, A83, A84, A86, A89, A91, A93, A94, A95, A96, A100, A103, A106, A108, A110, A114, A115, A116, A117, A119, A120, A121, A123, A125, A128, A129, A132, A134, A138, A139, A140, A141, A142, A147, A152, A154, A157, A161, A162, A163, A165, A166, A168, A169, A171, A174, A175, A176, A178, A179, A181, A183, A184, A186, A188, A189, A191, A192, A193, A194, A199, A200, A201, A202, A204, A207, A208, A209, A211, A212, A213, A214, A215, A216, A217, A218, A221, A222, A223, A224, A225, A226, A227, A228, A229, A231, A232, A233, A234, A235, A236, A237, A238, A240, A241, A242, A243, A244, A245	131	53.25
Modularity	A1, A2, A4, A6, A9, A10, A12, A13, A14, A18, A22, A32, A37, A41, A42, A43, A48, A49, A52, A53, A54, A55, A56, A61, A64, A65, A69, A71, A74, A75, A77, A78, A91, A92, A94, A97, A100, A102, A104, A115, A116, A117, A119, A122, A125, A127, A128, A129, A132, A133, A134, A138, A141, A142, A152, A155, A159, A164, A166, A168, A175, A178, A179, A182, A186, A189, A191, A192, A193, A197, A204, A207, A208, A209, A212, A216, A217, A218, A224, A225, A227, A229, A230, A233, A237, A239, A241, A243, A246	89	36.17
F-measure[a]	A16, A19, A29, A30, A35, A36, A38, A39, A40, A45, A50, A61, A67, A70, A73, A78, A81, A83, A88, A89, A90, A95, A96, A101, A102, A108, A109, A110, A113, A116, A118, A123, A124, A129, A135, A139, A140, A141, A145, A150, A153, A161, A162, A164, A166, A173, A177, A180, A188, A191, A196, A198, A199, A200, A205, A210, A211, A213, A219, A224, A234, A236, A240, A242, A244	65	26.42

Table 6 (continued)

Metric	ID	#	%
Accuracy	A7, A19, A20, A25, A27, A28, A30, A34, A40, A42, A43, A59, A61, A63, A64, A67, A68, A80, A82, A85, A91, A94, A96, A99, A100, A101, A110, A111, A112, A113, A114, A118, A121, A131, A135, A140, A145, A149, A158, A164, A165, A168, A169, A188, A199, A206, A211, A215, A218, A219, A221, A222, A224, A234, A235, A238, A242, A244, A245	59	23.98
Running time	A10, A12, A22, A26, A29, A33, A41, A48, A52, A61, A63, A64, A69, A74, A83, A87, A91, A98, A99, A102, A104, A105, A120, A122, A124, A125, A128, A138, A140, A144, A145, A148, A151, A152, A154, A157, A163, A165, A168, A170, A172, A176, A187, A189, A190, A191, A192, A203, A210, A215, A217, A218, A223, A224, A241, A242	56	22.76
Precision	A16, A19, A25, A30, A34, A35, A36, A38, A39, A50, A56, A61, A67, A79, A82, A84, A90, A108, A113, A135, A136, A138, A145, A153, A161, A164, A173, A177, A180, A191, A205, A207, A210, A219	34	13.82
Recall	A16, A19, A24, A30, A34, A35, A36, A38, A50, A56, A61, A82, A90, A108, A111, A113, A135, A145, A153, A161, A164, A173, A177, A180, A191, A205, A207, A210, A219	29	11.78
Adjusted randomized index (ARI)	A1, A58, A65, A98, A116, A125, A141, A161, A162, A175, A189, A193, A199, A200, A202, A209, A211, A212, A213, A215, A216, A226, A228, A229, A232, A243, A244	27	10.97
AUCROC	A7, A19, A25, A34, A39, A51, A67, A81, A93, A106, A137, A139, A146, A185, A203, A244	16	6.50
Jaccard	A7, A33, A34, A46, A47, A96, A98, A110, A130, A153, A154, A165, A198, A205	14	5.69
Conductance	A55, A70, A78, A89, A94, A116, A118, A123, A150, A154, A166, A231	12	4.87
Density[b]	A20, A22, A32, A55, A57, A59, A70, A76, A124, A154	10	4.06

[a]Similar metrics: F-score, Pairwise F1-score [A35, A140], Macro F1 measure [A42, A45, A101, A123, A139, A141], Micro F1 measure [A42, A81, A101, A123, A139, A141]
[b]Similar metric: Density Without Isolates [A32]

Table 7 datasets used for community detection of social networks in literature

Dataset	ID	Network format	Number of node	Number of edge	#	%
Zachary's Karate Club	A1, A2, A6, A9, A12, A28, A33, A41, A47, A49, A52, A53, A54, A56, A57, A64, A72, A75, A82, A83, A86, A94, A97, A100, A115, A117, A119, A125, A127, A129, A132, A133, A134, A138, A142, A148, A162, A176, A178, A179, A182, A183, A184, A189, A191, A192, A193, A196, A204, A207, A208, A209, A215, A216, A224, A226, A227, A229, A231, A233, A237, A239, A240, A241, A243	Undirected	34	78	65	26.42

Table 7 (continued)

Dataset	ID	Network format	Number of node	Number of edge	#	%
Lancichinetti–Fortunato–Radicchi (LFR)	A11, A12, A31, A33, A41, A56, A63, A64, A65, A72, A74, A78, A83, A84, A89, A91, A93, A94, A95, A103, A105, A116, A117, A118, A119, A121, A127, A129, A134, A139, A140, A152, A163, A165, A166, A175, A176, A178, A181, A182, A183, A184, A189, A191, A192, A193, A204, A207, A208, A209, A212, A213, A214, A215, A216, A217, A218, A237, A240	Undirected	–	–	59	23.98

Table 7 (continued)

Dataset	ID	Network format	Number of node	Number of edge	#	%
American College Football	A6, A8, A12, A21, A30, A33, A41, A47, A52, A53, A54, A56, A64, A75, A83, A86, A89, A115, A117, A125, A128, A132, A134, A138, A142, A148, A162, A174, A176, A178, A179, A182, A183, A184, A189, A191, A192, A193, A204, A207, A208, A209, A215, A216, A225, A227, A229, A231, A232, A233, A237, A239, A240, A241, A243	Undirected	115	613	55	22.35

Table 7 (continued)

Dataset	ID	Network format	Number of node	Number of edge	#	%
Bottlenose Dolphin	A1, A6, A8, A12, A33, A41, A47, A49, A53, A54, A57, A64, A75, A82, A83, A86, A100, A117, A119, A121, A125, A129, A132, A133, A134, A142, A152, A162, A176, A178, A179, A182, A183, A184, A189, A191, A192, A196, A204, A208, A209, A215, A216, A225, A226, A227, A229, A237, A240, A241, A243	Undirected	62	159	51	20.73

Table 7 (continued)

Dataset	ID	Network format	Number of node	Number of edge	#	%
Facebook	A3, A12, A16, A19, A23, A25, A39, A45, A46, A55, A63, A64, A66, A80, A82, A89, A92, A96, A98, A102, A106, A114, A117, A119, A120, A122, A125, A128, A135, A136, A145, A147, A152, A163, A164, A172, A176, A177, A205, A207, A208, A216, A223, A224, A238, A239, A246	Undirected	4,039	88,234	47	19.10

Table 7 (continued)

Dataset	ID	Network format	Number of node	Number of edge	#	%
DBLP (Digital Bibliography & Library Project)	A14, A25, A29, A36, A48, A61, A63, A70, A73, A74, A84, A89, A95, A100, A106, A116, A118, A123, A131, A143, A145, A150, A152, A158, A163, A165, A166, A168, A174, A177, A181, A183, A191, A198, A199, A200, A205, A211, A214, A216, A224, A228, A231, A239, A244	Undirected	317,080	1,049,866	45	18.29

Table 7 (continued)

Dataset	ID	Network format	Number of node	Number of edge	#	%
Twitter	A4, A13, A18, A20, A31, A33, A38, A39, A40, A46, A55, A60, A67, A76, A79, A80, A88, A96, A107, A108, A109, A112, A126, A128, A135, A140, A156, A159, A160, A161, A163, A170, A172, A173, A201, A203, A205, A206, A219, A221, A222, A223, A240, A246	Directed	81,306	1,768,149	44	17.88
Political books/pollbooks	A1, A6, A8, A47, A52, A53, A54, A56, A64, A85, A86, A89, A91, A100, A117, A119, A121, A125, A132, A162, A176, A178, A179, A182, A183, A184, A189, A191, A193, A196, A207, A209, A215, A216, A226, A227, A229, A232, A237, A240, A241, A243	Undirected	105	441	42	17.07

Table 7 (continued)

Dataset	ID	Network format	Number of node	Number of edge	#	%
Cora	A14, A19, A23, A85, A94, A96, A110, A113, A118, A121, A125, A139, A141, A169, A175, A185, A188, A194, A202, A205, A207, A210, A211, A213, A217, A221, A224, A234, A235, A236, A237, A238, A242, A243, A245	Directed	2,708	5,429	35	14.22
Amazon	A14, A29, A63, A70, A83, A95, A118, A119, A150, A152, A153, A163, A165, A166, A168, A174, A177, A183, A187, A191, A198, A200, A205, A214, A228, A235, A236, A239	Undirected	334,863	925,872	28	11.38

Table 7 (continued)

Dataset	ID	Network format	Number of node	Number of edge	#	%
Citeseer	A14, A19, A23, A36, A45, A85, A96, A110, A118, A121, A125, A169, A185, A188, A194, A207, A210, A213, A221, A224, A234, A236, A237, A242, A243, A245	Undirected	3,312	4,715	26	10.56
Political Blogs	A8, A47, A56, A64, A73, A86, A89, A94, A124, A125, A174, A179, A182, A184, A185, A193, A207, A215, A237, A241, A243	Directed	1,490	16,718	21	8.53
Email/Enron	A7, A14, A37, A41, A59, A63, A68, A83, A85, A93, A136, A142, A144, A146, A164, A183, A184, A191, A228, A229	Directed	36,692	183,831	20	8.13

Table 7 (continued)

Dataset	ID	Network format	Number of node	Number of edge	#	%
Wikipedia	A17, A21, A43, A49, A114, A123, A139, A140, A141, A145, A168, A172, A187, A200, A213, A231, A234, A240, A242	Undirected	2,277	31,421	19	7.72
YouTube	A29, A34, A84, A89, A95, A98, A116, A118, A152, A163, A165, A168, A174, A181, A183, A191, A214, A235	Undirected	1,134,890	2,987,624	18	7.31
LiveJournal	A29, A34, A70, A89, A95, A100, A104, A118, A152, A165, A168, A174, A181, A191, A198, A205, A214	Directed	3,997,962	34,681,189	17	6.91
PubMed	A19, A23, A94, A96, A110, A118, A169, A210, A213, A217, A221, A234, A236, A242, A243, A245	Undirected	19,717	44,338	16	6.50
Email-Eu-Core	A48, A64, A65, A82, A98, A130, A132, A145, A164, A168, A183, A207, A240	Directed	1005	25,571	13	5.28

Table 7 (continued)

Dataset	ID	Network format	Number of node	Number of edge	#	%
Power grid	A54, A117, A134, A166, A182, A183, A184, A185, A189, A191, A193, A228, A229	Undirected	4,941	6,594	13	5.28
Orkut	A29, A34, A70, A84, A89, A95, A163, A165, A174, A181, A191, A235	Undirected	3,072,441	117,184,899	12	4.87
NetScience	A33, A39, A41, A54, A91, A162, A183, A184, A191, A196, A197, A229	Undirected	379	914	12	4.87
Jazz musicians	A21, A41, A64, A176, A178, A181, A182, A184, A192, A193	Undirected	198	2,742	10	4.06

Table 8 The usage rate of real network and synthetic/artificial datasets in the literature

Type of dataset	Datasets	#	Rate (%)
Real network dataset	136 datasets including Zachary's Karate Club, Facebook, Twitter, Bottlenose Dolphin, American College Football, DBLP (Digital Bibliography & Library Project), Political books, Enron, Cora, Citeseer, Amazon, Wikipedia, Political Blogs, YouTube, LiveJournal, Email-Eu-Core, SINA Weibo, Blog Catalog, PubMed, Les Misérables, Orkut, Epinions, WebKB, etc. (the full list is mentioned in Table 3 of Online Appendix A	167	67.88
Synthetic or artificial benchmark	Lancichinetti–Fortunato–Radicchi (LFR), Girvan–Newman (GN), simulated SBM, simulated environment using heat capacity mapping mission (HCMM),	5	2.03

other side, the instance-based machine learning algorithms are the least used algorithms with less than 2%. It is noteworthy that some papers have proposed hybrid methods containing multiple machine learning-based algorithms which have been categorized based on the focus of the authors of the papers on the main algorithm.

4.1.2 To classify machine learning sub-categories in community detection social networks

According to the general categories mentioned in the above section (a), each category includes different sub-category methods. Based on the literature review, as demonstrated in Table 5, the following sub-categories are more prevalent for community detection in social networks:

Supervised learning (classification, liner classifier): the advantage is accurate predictions and its disadvantage is the dependency on labeled data and inaccurate classification of non-linear relations [56, 57]. Unsupervised learning (Clustering): the advantage is to identify communities without the need for labeled data but they are sensitive to initial conditions and parameters or computationally intensive, especially for large-scale networks [2]. Density-based spatial clustering of applications with noise (DBSCAN) is robust to noise and outliers, but DBSCAN's performance can be sensitive to the parameters [58].

Spectral clustering is effective for nonlinearly separable data; however, the choice of the appropriate affinity matrix and parameter values can be challenging [59]. Representation learning (RL) clustering or feature learning (FL) automatically extracts meaningful and relevant features from the raw data although this method can produce highly complex representations or features [60]. Expectation maximization (EM) is effective in dealing with missing or incomplete data. EM is sensitive to initialization and local optima [6]. Graph theory semi-supervised learning allows the incorporation of the network structure and relation among data points. The performance of graph-based semi-supervised learning is highly dependent on the quality and accuracy of the graph construction, including the choice of neighbors, edge weights, and overall graph topology [61].

Table 9 Types of community are on social networks

Type of community	ID	#	%
Non-overlapping community/disjoint	A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A13, A14, A15, A16, A17, A18, A19, A20, A23, A26, A27, A30, A31, A32, A35, A37, A40, A43, A44, A45, A47, A48, A49, A50, A51, A52, A53, A54, A55, A56, A58, A60, A61, A62, A64, A67, A73, A74, A75, A76, A77, A80, A81, A82, A84, A85, A87, A88, A92, A94, A95, A98, A99, A100, A101, A104, A105, A106, A107, A109, A110, A112, A113, A114, A115, A116, A118, A119, A121, A122, A124, A125, A127, A129, A130, A132, A133, A134, A135, A136, A139, A143, A144, A145, A146, A147, A149, A151, A152, A155, A159, A164, A167, A169, A170, A171, A172, A173, A174, A175, A176, A177, A179, A180, A182, A183, A184, A185, A186, A188, A189, A190, A191, A193, A194, A195, A196, A197, A198, A199, A200, A201, A202, A203, A204, A205, A207, A208, A210, A212, A213, A215, A216, A217, A218, A220, A221, A222, A224, A225, A226, A227, A228, A229, A231, A232, A233, A234, A235, A237, A238, A239, A241, A242, A244, A245, A246	167	67.88
Overlapping communities	A11, A12, A24, A25, A28, A29, A33, A34, A36, A38, A39, A41, A42, A46, A57, A59, A63, A65, A66, A69, A70, A72, A78, A79, A83, A86, A89, A90, A91, A93, A96, A97, A102, A103, A108, A111, A117, A120, A123, A128, A131, A137, A138, A140, A141, A142, A148, A150, A153, A154, A156, A157, A160, A161, A162, A163, A165, A166, A168, A178, A181, A187, A192, A206, A209, A211, A214, A219, A223, A230, A236, A240, A243	73	29.67
Hidden communities	A21, A22, A68, A71, A126, A158	6	2.43

Label propagation algorithm is computationally efficient and can handle large datasets with many unlabeled instances but is dependent on graph structure [25]. Heuristic algorithms are often computationally efficient; however, they rely heavily on predefined heuristics and rules [62]. Semi-supervised representation learning leverages both labeled and unlabeled data, allowing the model to learn more informative and robust representations. Its disadvantage is dependency on quantity and quality of labeled data [63]. Nonnegative matrix factorization (NMF or NNMF) enforces nonnegativity constraints on the factor matrices, leading to a parts-based representation of the data. NMF can identify meaningful and interpretable parts or components, aiding in understanding the underlying structure of the data; however, NMF involves a non-convex optimization problem, and depending on the initialization and algorithm used, it may converge to local minimal [64].

Active learning allows for the efficient use of labeling resources by selecting the most informative and uncertain data points for annotation, but this algorithm heavily depends on the quality of the initial model [65]. Co-training is advantageous in scenarios where multiple views or modalities of data are available. The assumption of conditionally independent views is a disadvantage of training [66]. Auto-encoder deep learning is highly effective for dimensionality reduction. Interpreting the features learned by auto-encoders can be challenging [31, 67]. Graph neural network (GNN) excels in community detection tasks by leveraging the relation and dependencies between nodes in a graph. GNNs might struggle with large graphs or graphs with irregular structures [31, 68]. Recurrent neural networks (RNN) are effective in modeling temporal dependencies and sequential data, but RNNs are prone to the vanishing and exploding gradients problem during training [69].

Graph convolutional neural network (CNN) can learn and leverage the underlying graph topology to extract meaningful features and detect community. GCNNs may face challenges when dealing with complex or irregular graph structures due to computational complexity [70]. Restricted Boltzmann machines (RBM) can automatically learn and extract meaningful features from the input data, but they may face challenges in scalability due to the computational cost of training large RBMs [71]. Bagging helps improve the generalization performance of machine learning models by reducing overfitting. Increased training time and model complexity are its disadvantages [72]. Stacking combines the predictions of multiple diverse base models (meta-model) to achieve better predictive performance; however, complexity and potential overfitting are considered its weaknesses [73].

Boosting focuses on iteratively improving a single model by training multiple models sequentially, and correcting mistakes made by previous models. Its disadvantage is sensitive to noisy data and outliers in the training set. Gradient bootstrap machine (GBM) is a specific implementation of boosting, which uses gradient descent optimization to train new models and combine their predictions [72]. Random forest is an ensemble learning technique that combines multiple decision trees to improve accuracy and reliability in detecting communities within a network, even though understanding the exact decision-making process of a random forest model can be complex, especially when dealing with a large number of trees [74].

Instance base learning (k-nearest neighbor graph (KNN) connects each instance to its k-closest neighbors, capturing local relations efficiently and with simplicity. However, it may be sensitive to noise and the choice of k value [75]. Reinforcement learning (policy gradient) methods directly optimize the policy, allowing them to handle continuous action spaces and complex policies, but they often suffer from high variance in the estimates of the gradients, which can lead to slow convergence and unstable training [76]. Q-learning is a model-free reinforcement learning algorithm that learns optimal actions without needing an environment model. Q-Learning can suffer from the curse of dimensionality, especially in high-dimensional states and action spaces [77].

Sub-categories that had been applied by researchers for community detection in social networks are shown in Table 5.

As noticed before, among the main categories of machine learning algorithms, unsupervised algorithms are the most used ones in literature to detect communities in social networks. The most used sub-category that is identified under the unsupervised category is clustering with 24.39%. After that, heuristic as semi-supervised learning and classification as supervised are the most used sub-categories with 11.38% and 8.13% of usage, respectively.

4.1.3 Machine learning-based algorithms/methods have been used for community detection in social networks

Each identified sub-category of the machine learning-based approach includes some algorithms/methods to handle the task of community detection. It should be mentioned that some papers have proposed hybrid methods containing multiple algorithms/methods. In such cases, we considered the dominant or well-known method to determine the classification of machine learning approaches. Some well-known algorithms/methods for the sub-category of supervised classification using machine learning contain naïve Bayesian [A34], support vector machine (SVM) [A42, A185, A186], and decision tree [A135]. In the sub-category of unsupervised classification using machine learning, hierarchical clustering [A13], [A33], [A38], [A48], [A92], [A93], [A95], [A116], [A136], [A144], [A146], [A152], [A157] and k means [A4], [A5], [A11], [A59], [A68], [A167], [A209], [A220], [A231] are the most frequently used algorithms. Some sub-categories such as unsupervised learning/representation learning clustering [A43], [A56], unsupervised learning/density [A170], semi-supervised learning/graph theory [A77], auto-encoder deep learning [A8], [A147], [A207], [A215], and GNN deep learning [A30], [A210] have used such algorithms (e.g., k means) as supplementary to prepare a hybrid algorithm/method.

Among other commonly used algorithms as semi-supervised, LPA (label propagation algorithm) is used in several papers mentioned in Table 8.3 in Online Appendix A as the primary algorithm, and in some papers [A104], [A122] as the secondary.

Representation learning (feature learning) is another algorithm that is used in supervised, unsupervised, and semi-supervised sub-categories [A2], [A19], [A43], [A45], [A56], [A85], [A141], [A188], [A200], [A235]. Furthermore, nonnegative matrix factorization (NMF) is used mostly in semi-supervised and unsupervised learning sub-categories [A14], [A21], [A68], [A86], [A100], [A121], [A150], [A163], [A174], [A202], [A217], [A241], [A245]. Tables 8-1 to 8-7 in Online Appendix A present different types of machine learning-based algorithms/methods have been used for community detection in social networks.

4.2 RQ2. Which metrics/parameters are used to distinguish communities in social networks?

The methods used for detecting the communities in social networks cannot be evaluated without employing appropriate metrics to measure their efficiency [78]. According to Table 6, the efficiency metrics most frequently used are normalized mutual information (NMI), modularity, F-measure, running time or time of execution, accuracy (AC), precision, recall, adjusted randomized index (ARI), area under the receiver operating characteristic curve (AUCROC), Jaccard similarity coefficient, graph conductance, and density. Among the above metrics, NMI which represents 53.25% is the most employed one with 131 papers out of 246. Furthermore, metrics that are used in less than 10 papers are considered as underused metrics. According to the literature, 57 metrics are rarely used; all of them are used in 133 papers. Such rarely used metrics/parameters are listed in Table 2 of Online Appendix A. For such metrics, we calculated[1] the average amount of their usage as 0.94%.

In the following, the most employed metrics/parameters are described concisely:

[1] The percentage is the average value for each rarely used metric. The average value of percentage is calculated as $\left(\frac{Count\ of\ References\ for\ all\ rarely\ used metric}{Count\ of\ All\ References}\right)/Count\ of\ Metrics * 100$

Normalized mutual information (NMI) is a commonly used metric for measuring the mutual dependence between the possibilities of an alter belonging to the detected communities and the actual communities [79, 80]. It is a normalization of the mutual information (MI) score to scale the results between 0 (no mutual information) and 1 (perfect correlation). NMI is always used to calculate the similarity between two partitions. A higher NMI value represents a greater similarity between partition A and partition B [32, 81, 82]. In other words, NMI is the quality measure value which indicates how much two clusters are correlated. In community detection, it presents a set of communities based on how much they resemble a set manually labeled by experts, where 1 means identical results and 0 completely different results [81]. NMI is calculated by the following equation:

$$\mathrm{NMI}(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]}$$

where Y refers to class labels, C refers to cluster labels, H refers to entropy, and $I(Y;C)$ refers to mutual information b/w Y and C.

In the following, some of the most frequent metrics are briefly introduced:

Modularity) Q) is proposed by Newman and Girvan [83] as a measure of the quality of a particular division of a network and is defined as follows: $Q =$ (number of edges within communities)—(expected number of such edges) [84, 85]. It is always used in estimating the quality of community structure discovered by different solutions if the community structure of a network is unknown [86]. The community partition with larger modularity usually indicates a better solution [78]. Modularity and NMI are two commonly used metrics for evaluating the quality of community structure [87]. Its value shows how pronounced the cluster structure is in the graph and how easily the graph can be divided into subgroups of vertices [88].

F-measure (balanced F-score) in statistical analysis of binary classification is a measure of accuracy of a test. It is calculated from the precision and recall of the test. In other words, it is the harmonic average of the precision (P) and recall (R), $F = 2\mathrm{PR}/P + R$ [89]. The highest possible value of an F-measure is one (1.0), indicating perfect precision and recall, and the lowest possible value is zero (0) if both precision and recall are zero. It signifies the preciseness (how many instances it classifies correctly) and robustness (it does not miss a significant number of instances) of ML models. Less F-measure indicates that the dataset is unbalanced [90, 91]. Over the whole set of communities, the F-measure evaluates the true/false positive/negative node clustering results in detail [24].

Runtime/time of execution time refers to the time required to complete the process of community detection. It is proportional to the size of the input, the complexity of the used algorithm, and the frequency of the processor [92].

According to the above table, *Accuracy* is another metric frequently seen in community detection papers on how close a given set of measurements (observations or readings) are to their true value. More commonly, it is a description of only systematic errors, a measure of statistical bias of a given measure of central tendency; low accuracy causes a difference between a result and a true value; [86, 93] it is used for calculating the percentage of nodes that are delegated to the right communities [65].

Precision is the fraction of relevant instances among the retrieved instances. In other words, precision quantifies the number of correct positive predictions and is calculated as the ratio of correctly predicted positives divided by the total number of positive predicted examples. Therefore, it measures how near the calculated results are to one another, whereas accuracy deals with how close they are to the actual value of the measurement. It is defined as $P = \mathrm{TP}/\mathrm{TP} + \mathrm{FP}$, (true positive/true positive + false positive) [92, 94]. In the community

detection evaluation, precision describes the percentage of clustered nodes in each detected community.

The *Recall* metric is the fraction of vertex pairs labeled with the same labels which are also clustered in the same community. In fact, recall is the portion of the correctly predicted topics from the total topics linked during the testing phase [89]. It is defined as $R = \text{TP/TP} + \text{FN}$, (true positive/true positive + false negative) where FN represents the similar type of participants that are assigned to different communities [89, 95]. The recall for overlapping nodes is defined as the proportion of accurately detected overlapping nodes to the entire number of overlapping nodes [96].

Adjusted randomized index (ARI) is a measure used to assess the quality of clustering results in community detection on social networks. It takes into account the randomness that can occur in clustering algorithms by comparing the detected communities with a random assignment of nodes into communities. The ARI calculates the agreement between the true and detected communities, considering both agreements and disagreements. A higher ARI value suggests a better clustering result, indicating that the algorithm successfully captures the underlying community structure of the network. *RandI index* is defined as $\text{TP} + \text{TN/TP} + \text{FP} + \text{FN} + \text{TN}$ [97].

*Area under the receiver operating characteristic curve (*AUCROC*)* identifies the area under the receiver operating characteristic (ROC). It illustrates the performance of binary classifiers relating the true-positive rate $\text{TPR} = \text{TP/TP} + \text{FN}$ to the false-positive rate $\text{FPR} = \text{FP/FP} + \text{TN}$ and provides a visual interpretation useful to compare different models. In other words, it can be construed as the probability that a randomly chosen link correctly predicted is given a higher score than a randomly chosen link wrongly predicted [98].

Jaccard (Jaccard similarity coefficient) is a measurement to compare the similarity and diversity of a sample set. If two nodes are highly similar, they are more likely to be in the same community, and hence the relationship between them should be relatively important [76].

Graph conductance is used as a simple measure to capture community goodness [99]. It corresponds most closely to the intuition that a community comprises a set of densely linked nodes that are sparsely linked to the outside. Particularly, the sets of nodes that closely resemble a community are characterized by a lower conductance. It has many inward edges and/or a few edges pointing outside [96]. Conductance is used in the partitioning of graphs which is the ratio of the total number of graphs cut edges to the total volume of the smallest part [78].

Density can be defined as the number of connections divided by the complete possible number of connections a node could have [100].

4.3 RQ3. Which datasets are used for community detection in social networks?

Data is a valuable resource for any research. Real data means data from a production system, vendor, public records, or any other dataset which otherwise contains operational data. For example, a dataset that is a 10-year-old backup of an existing system and contains data about real individuals, matters, or cases, would be real data [101]. However, sometimes collecting real data is a difficult task due to the cost, sensitivity, and processing time. Thus, synthetic data can be an alternative [102]. Synthetic data is artificial data that mimics real-world observations and is used to train machine learning models when actual data is difficult or expensive to get. Synthetic data recreates something that exists in the real world and obtains their characteristics but do not depict them directly. They reflect real-world data, mathematically or statistically.

They are typically created using algorithms and can be deployed to validate mathematical models and train machine learning models.

Data generated by a computer simulation is an example of synthetic data. Such data does not have to be only generated by computers, as the same face generation could be done by a person who creates new faces with drawings. But even with advances in mathematics and probability theory, generating synthetic data without computer resources is very time-consuming and generally complicated. Synthetic data is often generated to represent the authentic data and allow a baseline to be set. Another benefit of synthetic data is to protect the privacy and confidentiality of authentic data [103]. In the following, the characteristics of the dataset used in the literature are presented. Table 7 presents datasets used for community detection in social networks in the literature.

According to Table 7, the datasets most frequently used in literature are, Zachary Karate, Lancichinetti–Fortunato–Radicchi (LFR), and American College Football with usage rates of 26.42%, 23.98%, and 22.35%, respectively. Among them, Zachary Karate and American College Football are from type real network data, and Lancichinetti–Fortunato–Radicchi (LFR) is a synthetic dataset. Moreover, datasets that are used in less than 10 papers are listed in Table 3 in Online Appendix A. A total of 118 datasets are used in 255 papers, and we calculated[2] the average amount of their usage as 0.87%.

In the following, the datasets are described concisely:

Lancichinetti–Fortunato–Radicchi (LFR) [104] is a synthetic dataset as a benchmark proposed by Girvan and Newman[105]. The LFR generator has several parameters among them, such as the number of nodes, the average degree of incoming edges, the maximum degree of the incoming edges, the fraction between incoming and outgoing edges inside a community, the minimal community size, and the maximal community size [106, 107]. LFR benchmark can generate overlapping communities in directed and weighted networks [104].

*Zachary Karate/Karate Club*as a real network dataset was collected from members of the University Karate Club by Wayne Zachary in 1977. Every node represents a member of the club, and an edge represents a link between the two members of the club. The network has 34 nodes and 78 edges, and is undirected [108, 109].

The American College Football as a real network dataset contains the network of American football games (not soccer) between Division IA colleges during the regular season of Fall 2000. There are 115 nodes representing the football teams while an edge means there is a game between the teams connected by the edge. The teams were divided into 12 conferences, and, except for one conference, they played against the teams in the same conference more frequently than those in other conferences [106, 110].

Bottlenose Dolphin as a real network dataset contains the social communication network of 62 Bottlenose Dolphins living in Doubtful Sound, New Zealand. The network is naturally split into two communities and the larger one can be further divided into some small subgroups [111].

Facebook as a real network dataset is an online social media and social networking service. After registering, users can post text, photographs, and multimedia which are shared with any other users who have agreed to be their friend or, with different privacy settings, publicly. Users can communicate directly with each other with Facebook Messenger, join common-interest groups, and receive notifications on the activities of their Facebook friends and the pages they follow as well [101]. The datasets are actually based on the posts given on Facebook. For each post on Facebook, on a particular period of time, many individuals used

[2] The percentage is the average value for each rarely used dataset. The average value of percentage is calculated as $\left(\frac{Count\ of\ References\ for\ all\ rarely\ used metric}{Count\ of\ All\ References}\right)/Count\ of\ Metrics$

to like, share and give comments on it for review purposes. On Facebook, every post used to have a unique id and every person also used to have a unique id. So, when a particular post is liked, shared, and commented on by an individual, the corresponding individual id in Facebook can be extracted including the id of the corresponding post [76, 100].

DBLP dataset is a citation real network dataset. An academic paper citation network that each paper is associated with an Abstract, Authors, Year, Venue, and Title. The dataset can be used for clustering network and side information, studying influence in the citation network, finding the most influential papers, topic modeling analysis, etc., upon the DBLP repository [76, 100].

Twitter as a real network dataset is a microblogging and social networking service owned by American company Twitter, Inc., on which users post and interact with messages known as *tweets*. Registered users can post, like, and retweet tweets, while unregistered users only have a limited ability to read public tweets. Users interact with Twitter through browser or mobile frontend software, or programmatically via its APIs. Prior to April 2020, services were accessible via SMS [84]. With 100 million daily active users, Twitter is one of the most popular social networks nowadays. It enables users to send short messages (called tweets) up to 280 characters and, has around 6000 tweets per second, which corresponds to over 500 million tweets per day. Counting on this huge volume of data generated, Twitter can be seen as a valuable source of data that is useful for monitoring several social aspects, such as detecting and analyzing communities [112].

Political books as a real network dataset includes the books published around the time of the 2004 presidential election about US politics and distributed through online bookseller Amazon.com. This dataset contains 105 books about US politics. Nodes are books sold by the online bookseller Amazon.com and edges represent frequent co-purchasing of books by the same buyers [65, 113].

Cora as a real network dataset is an Internet portal for computer science research papers. Not only does it provide Keyword search facilities for over 50,000 collected papers, but also places these papers into a computer science topic hierarchy, maps the citation links between papers, provides bibliographic information about each paper, and is growing daily [114].

Amazon is a real online commercial network/graph for purchasing products where nodes represent products and an edge between two products indicates that the above nodes are frequently purchased together [115]. This is a product co-purchasing network based on customers who purchased a product along with another product [116].

Enron Mail as a real network dataset records the email communication data of the employees of Enron in the USA [117]. Nodes and edges in the network represent individual employees and their email communications, respectively [83].

Citeseer as a real network dataset is a citation network with 3312 nodes and 4732 edges. Each node is classified into one of six classes [118].

PolBlogs as a real network dataset is a network of hyperlinks between weblogs on US politics, recorded in 2005 [119]. Each node is represented by its political affiliation, conservative or liberal [120].

YouTube as a real network dataset represents the YouTube social network graph, where each vertex is a user, and two users are linked if they have established a friendship relation. Communities are defined by the groups created by the users, which are formed by those users that joined the group [121–123].

LiveJournal is a real network dataset as free online blogging community where users declare friendship to each other. It is formed as a graph representing the social network around LiveJournal. Similar to the YouTube network, the nodes represent the users establishing a friendship with other users. Users can create groups, which define the ground truth

communities, and which are formed by those users that joined that group. A list of friends for a user includes several communities to individual users [76, 101, 121].

Wikipedia page as a real network dataset use the Wikipedia website as a source to extract its data consisting of 2405 web pages from 19 categories and 17,981 edges. It is an online collaboration encyclopedia written by its users and is widely used in social network analysis. The social relationship in this dataset is relatively dense [118, 124].

The Email-Eu-Core dataset is a real network dataset that represents the communication patterns of a large European research institution. It consists of email exchanges between individuals over a period of time, providing valuable insights into the dynamics and structure of email communication networks [125].

The Orkut dataset is a real network dataset and it is a publicly available dataset containing anonymized information from the Orkut social network. It includes user profiles, friendship connections, community affiliations, and user activities [123, 126].

NetScience is a real network dataset that represents the scientific collaboration network among researchers working on network theory and experiments. It includes information about researchers, their collaborations, and the papers they have co-authored [127].

The electrical grid stability simulated dataset also known as the Power grid dataset, is a real simulated dataset that provides valuable information for analyzing and studying the stability of electrical grids. It offers insights into the behavior and dynamics of power systems, allowing researchers to investigate and understand the factors that contribute to grid stability or instability [128].

The Jazz musicians dataset is real and it is utilized in network analysis to explore the relationships and collaborations among jazz musicians. By examining this dataset, researchers can gain insights into the structure of the jazz community, identify influential musicians, and study patterns of musical collaboration within the genre [129].

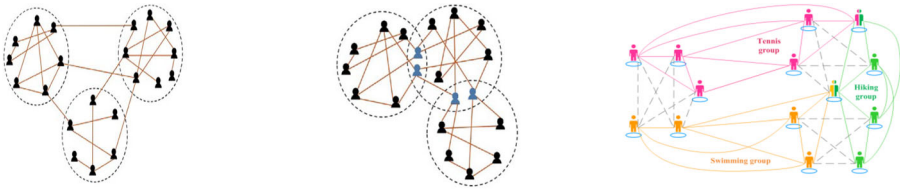
The PubMed dataset is a real network and it is a comprehensive collection of scientific literature in the field of biomedical and life sciences [130]. Table 8 statistically presents the usage of real networks and synthetic/artificial datasets in the literature. As can be seen, most of the papers in the literature (67.88%) have used real network datasets while 2.03% have used synthetic/artificial datasets.

It is noteworthy that some researches (74 papers rated as 30.08%) have used both types of real dataset and synthetic benchmark.

4.4 RQ4. What types of community are on social networks?

There is no specific categorization of communities on social networks. The categorization may be helpful to list communities in a directory to make it easier for users to find ones to join by tagging each community with one or more types. Users can get a better understanding of the landscape, and how communities are being used in the organization, and suggest different approaches for each type of community to improve their effectiveness. Based on the review of the literature [24, 26, 131], as presented in Table 9, three types of communities can be extracted: a) disjoint, b) overlap, and c) hidden.

a. **Disjoint community** The non-overlapping network is familiar with nodes of community in the network in which they do not have a relation with another network. Figure 6a shows an example of disjoint community detection [26]. It is named disjoint, while the overlapping network is a condition of nodes in the network to be a part of another community and it is called a node with many communities. In other words, a disjoint



(a) Non-overlap community (b) Overlap community (c) Hidden community

Fig. 6 Different types of communities

community assumes a network that can be partitioned into dense regions where nodes have more connections among each other than those with the rest of the networks [132].

b. **Overlapping community** Communities can share one or more common nodes in some real-world networks. For example, in social networks, actors may be part of different communities: work, family, friends, and so on. All these communities will share a common member, and usually more since a work colleague can also be a friend outside the working environment [9]. Figure 6b shows an example of possible overlapping community partitions: the central node is shared by the two communities [26].

c. **Hidden community** In a social network, individuals may belong to multiple strong social communities, corresponding to groups such as families, colleagues and friends. Though overlapping, the connections inside these communities are strong and numerous enough that existing overlapping community detection algorithms can perceive and uncover these latent but dominant modular structures. However, in these strong communities, individuals may also belong to some weaker communities, such as a group of medical patients that see each other at the doctor's office and communicate infrequently. The hidden community structure is sparser and harder to detect. A hidden community structure can be contemplated as a special type of overlapping community structure. It is a set of users that have a lot of implicit connections. Unlike "friends" on social networks or members of a real community, users of hidden communities may not know each other, they may live in different regions of the same country, but their interests coincide. These interests may be reflected in users' posts. Figure 6c shows a model of hidden community detection [131].

5 Discussion

5.1 Principal findings

In this section, the findings of the study are discussed. In the context of the social network, communities refer to entities (such as users) that are related in some aspects. Such relations can be detected by analyzing the structure of social networks. This study attempted to present a systematic map of machine learning-based community detection approaches. Based on the results presented in previous sections, there are three types of community detection in social networks (non-overlapping, overlapping, and hidden). Based on the literature, unsupervised learning has the greatest number of machine learning algorithms that are used for all types of community detection in social networks.

In the non-overlapping type of community detection, deep learning (40 references) is the most used after unsupervised learning (64 references). But in overlapping community detection, semi-supervised learning (18 references) is the most used after unsupervised learning (35 references). Reviewing the literature shows that a few numbers of papers used hidden community detection for social networks (six references) which most of them (three references) used unsupervised learning as their method (see Fig. 7). This study indicates in unsupervised learning, clustering is the most used approach that presents a lot of information about hidden attributes, relationships, and properties of the participants.

Figure 8 shows the usage of different types of community detection using machine learning in social networks in recent years. As can be seen, using machine learning is increasing after 2016. Among different types of community detection, non-overlap is used more than other types such that there has been a significant growth in usage since 2017. Some machine learning approaches such as deep learning based on auto-encoder algorithms [A110] [A147] have been seen in non-overlap community detection since 2017, so most papers published in the scope of deep learning can be seen in 2020. [A1], [A8], [A30], [A31], [A40], [A75], [A80], [A84] and it is increasing in 2023; refer to Table 4 in Online Appendix A.

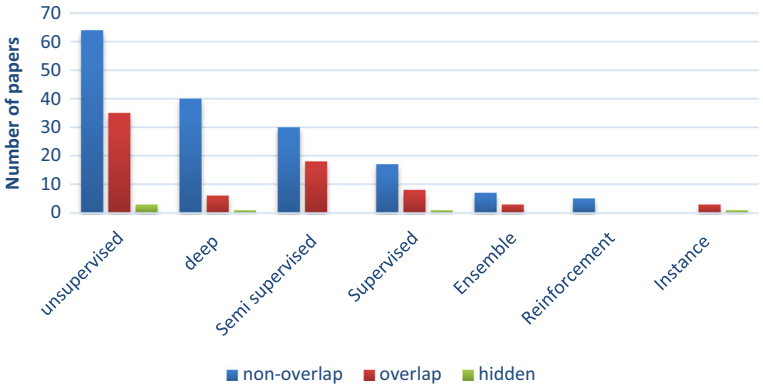


Fig. 7 Usage of machine learning approaches for different types of community detection in social networks

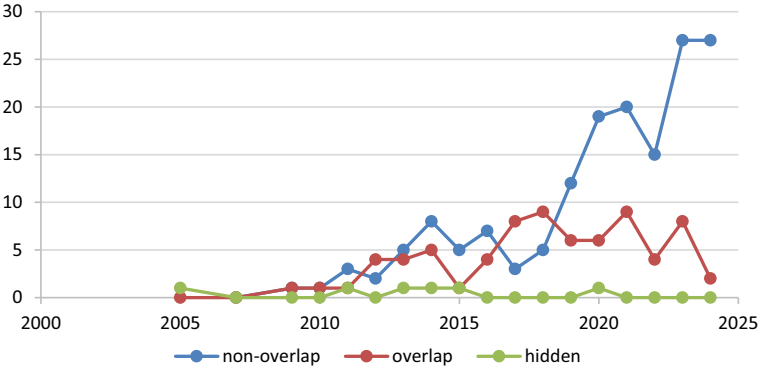


Fig. 8 Different types of community detection in social networks using machine learning in recent years

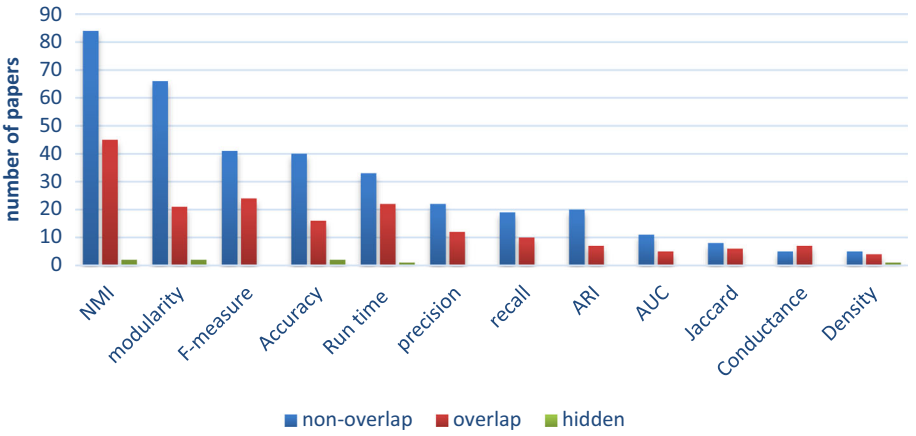


Fig. 9 The usage of different metrics distinguished by each type of community detection (non-overlap, overlap, hidden) in social networks

Using ensemble learning and boosting method [A104] has been seen in non-overlap community detection since 2013 and in overlapping [A66] since 2016. Deep learning methods for overlapping community detection have been seen since 2018 [A96]. Semi-supervised learning based on graph theory is the most used approach for overlap community detection in 2018 and it has been used the most in 2022 for both non-overlap and overlap community detection. Hidden community detection has been the least used type in social networks during the last years such that it has been seen only in 6 references from 2005 until now [A21], [A22], [A68], [A71], [A126], [A158]; refer to Table 4 in Online Appendix A.

As shown in Table 6 of Sect. 4, NMI is the most commonly used metric in different community detection related references in the literature. It shows the overall statistics of used metrics in the literature. Figure 9 shows the usage of different metrics distinguished by each type of community detection (non-overlap, overlap, hidden) in social networks. Analyzing the statistics presented in Fig. 9 gives us more information about metric usage so that a metric such as NMI, which is the most common overall, is not necessarily the most common in all types of community detection in social networks.

As can be seen in Fig. 9, in non-overlap community detection, NMI (84 references), modularity (66 references), and F-measure (41 references) are the most common metrics, respectively. In overlap community detection, NMI (45 references), F-measure (24 references), and run-time (22 references) are the most common metrics, respectively. Furthermore, in the hidden community detection, NMI, modularity, and accuracy all with 2 references are the most common metrics. It is noteworthy, for all metrics, the number of non-overlap-based references is more than overlap-based ones except the conductance. More details of the references and metrics used are presented in Table 5 of Online Appendix A.

Figure 10 shows the statistics of papers used different datasets distinguished by non-overlap, overlap, and hidden community detection. As can be seen in Table 7, Karate (65 references), LFR (59 references), and Football (55 references), and are the most used datasets overall. Also, if each type of community detection is considered separately, Karate (46 references), Football and Dolphin (37 references), and LFR (35 references) are the most used datasets in non-overlap detections. Furthermore, in overlap detections, LFR (24 references) as a synthetic dataset has been used in more papers. After that, Karate (19 references), and

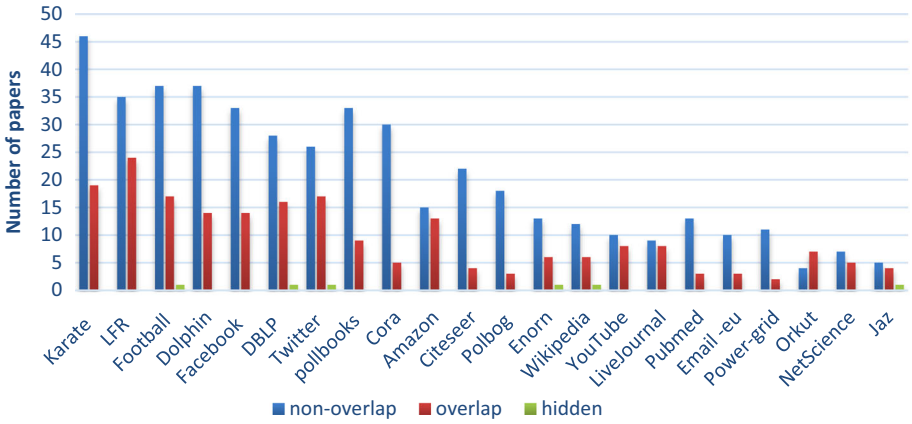


Fig. 10 The usage of different datasets distinguished by each type of community detection (non-overlap, overlap, hidden) in social networks

Football and Twitter (17 references) are the most used as real datasets. In hidden detections, paper [A21] used more well-known datasets such as football, Wikipedia and Jaz. For all datasets, the non-overlap community detections had more using references except the Orkut. More details of information about the datasets are presented in Table 6 of Online Appendix A.

Based on the research in the literature, until 2022, methods based on unsupervised learning and also clustering have been widely used in the detection of communities in social networks. The reason for this is that such methods are able to independently identify patterns, relationships and structures in data without any predefined labels. Such algorithms (e.g., k means) can identify complex patterns and relationships that may not be apparent through human observation or predefined classifications. This can identify groups or clusters in the data, providing insights into categories or variants that may not have been previously known.

It was later revealed that unsupervised learning and clustering methods are not efficient enough for large-scale networks because it is sometimes challenging and even impossible to examine the large amount of content, interactions, or behaviors generated by the user. Also, considering that social networks contain heterogeneous data such as texts, images, and links, unsupervised learning-based methods face problems in processing and extracting meaningful patterns. They also are not completely adapted to the dynamic nature of social networks. Therefore, in many cases, unsupervised learning and clustering-based methods are used in combination with other methods.

From 2022 onwards, the use of methods based on semi-supervised learning to detect communities in social networks has increased, and recently, deep learning methods are frequently used. Semi-supervised learning methods act efficiently with labeled or unlabeled data and provide a more accurate view of the structure of communities compared to unsupervised methods. Also, deep learning methods are able detect the communities in social networks accurately using neural networks to analyze patterns and extract complex relationships in the data.

The results presented in papers such as [A216], [A221], [A226], [A232], [A237], [A239], [A241], [A242], [A243], [A244] indicates that semi-supervised learning and deep learning methods have been used more than other methods in recent years to detect communities in

social networks. Moreover, NMI is the most widely used metric, karate and LFR are the most common artificial and real datasets, respectively. In addition, non-overlap is the most used approach for detecting communities.

The complexity of algorithms in graph theory typically depends on the size of the graph's nodes and edges. These measurements represent the graph's magnitude and intricate nature, affecting how algorithms traverse nodes and edges for computations or validations. Comparing algorithms also depends on these metrics to determine which algorithm could be more effective for various graph sizes. Therefore, the quantity of nodes and edges play a fundamental role in examining and enhancing the computational complexity of graph algorithms. Table 10 outlines the factors affecting the computational complexity for each machine learning approach and their specific impacts on community detection in social networks.

As can be seen in Table 10, several factors impact the computational complexity of community detection in social networks when using different machine learning approaches. In supervised learning, increased training data sizes need greater resources for training. The processing time and memory usage are greatly impacted by the quantity and intricacy of features, including node attributes. The performance and rate at which training algorithms, like gradient descent, converge differ and affect overall computational requirements like

Table 10 The factors affecting the computational complexity for each machine learning approach and their potential benefits/drawbacks on community detection in social networks

Approach	Factor	Potential benefit	Potential drawbacks
Supervised learning	Training data size, Feature quantity, Convergence rate	High accuracy with sufficient data, High performance	Resource-intensive, High complexity, Slow convergence, High cost
Unsupervised learning	Number of communities, Starting value approaches	Finding hidden patterns, Detailed clustering, Accurate results,	High cost, Highly depends on to starting values, More needed iterations
Semi-supervised learning	Labeled and unlabeled data ratio, Algorithm sophistication	Utilizes less labeled data, Can handle complex tasks	High complexity, High memory usage
Deep learning	Size and complexity of networks, Batch size during training	High accuracy and adaptability, Fast training for large batch sizes, Prevents overfitting	Slow convergence, High memory usage, Low scalability
Ensemble learning	Quantity of base learners	High performance, Robust predictions, Enhanced accuracy	High processing cost, Resource-intensive, Low scalability
Instance -based learning	Calculations for prediction	Simple and intuitive, Fast predictions	Over-fitting, High preprocessing cost
Reinforcement learning	Size of state and action spaces, Balance between exploration and exploitation	Able to solve complex tasks, Optimized learning,	Resource-intensive, Lack of balance, Slow convergence, Low consistency

[A39], [A44]. In the realm of unsupervised learning, computational complexity is heavily impacted by the algorithm selected. For example, spectral clustering [A238] requires eigenvalue decompositions, which are more computationally costly than clustering [A22], [A29]. The desired number of communities or clusters impacts both the number of iterations and the overall complexity of the process. Different approaches to setting the starting values can greatly influence how quickly iterative algorithms converge, while the rigor of the convergence criteria impacts the number of iterations needed to find a solution.

In methods like [A140], [A241] for semi-supervised learning, the complexity is greatly influenced by the ratio and incorporation of labeled and unlabeled data. Graph-based techniques like [A63] frequently utilize Jaccard matrices, resulting in higher memory and computational requirements. Sophisticated algorithms are needed to effectively mix labeled and unlabeled data, further complicating the process. The neural network's design is essential in deep learning, with the size and complexity of networks impacting computational needs for community detection using GNN and GCNs in [A218], [A242], [A244]. Training deep networks requires a significant amount of time and computational resources. The memory usage and training speed are influenced by the batch size used during training, as larger batch sizes may require more memory but could also potentially accelerate the process like [A215], [A234]. Moreover, regularization methods such as dropout add additional calculations during the training process, leading to additional computational requirements.

In ensemble learning, the computational complexity is greatly impacted by the quantity of base learners, with a higher number of learners (e.g., in boosting or bagging) leading to a linear or even greater increase in complexity like [A61]. Having a variety of models in an ensemble can result in increased computational needs, since managing and incorporating diverse models necessitates additional resources. The method for integrating predictions, like voting or stacking, increases the computational load, also affecting the overall complexity of the learning process. In case-based learning like [A192], there can be significant storage needs, as algorithms such as KNN have to hold onto all instances, which can be demanding on memory, particularly with extensive networks. Calculating distances between instances for prediction is costly in terms of computation, and improving search operations can help but necessitates extra preprocessing, leading to increased overall computational complexity.

The complexity of learning in social networks increases with larger state and action spaces in reinforcement learning. Finding the right balance between exploring new options and exploiting known strategies can affect how many iterations are needed to learn the best policies. The computational requirements also differ depending on the learning algorithm chosen, like Q-learning or policy gradient methods. Furthermore, limited or delayed rewards require additional iteration for successful learning, adding to the complexity of reinforcement learning techniques in identifying social network communities. Table 9 in Online Appendix A provides a list outlining the computational complexities of the algorithms mentioned in the paper.

5.2 Study limitations

This research is restricted to valid journal and conference papers related to different approaches to community detection in social networks. By applying our search phrase, we obtained many papers, the majority of which were found to be irrelevant, duplicated, or invalid. We attempted to choose a small number of papers to ensure that all the selected documents fully matched the research objectives. We applied accurate quality assessment criteria to select the related documents able to provide synthesized results. One of the limitations is

finding social network-related datasets with appropriate features and contents regarding the unstructured and heterogeneous nature of datasets.

One of the most important limitations in our research was selecting the scientific search database and preparing an appropriate search phrase regarding the differences between search databases. There are multiple valid scientific search databases such as IEEE, ACM, etc. After investigating the capabilities and validity of existing scientific search databases, the authors decided to use ACM, Scopus, Web of Science, IEEE Xplore, and Springer to gather the related references. They are widely recognized by the scientific community and are able to provide a sufficient set of results. ScienceDirect was discarded since it did not allow authors to ensure a similar search to that carried out in the other digital libraries.

The search phrase has been prepared to find the most related and limited number of references using the metadata Title, Abstract, and Keywords. The selection process was carried out in pairs in order to minimize potential biases and conflicts were resolved by a third party. If necessary, a debate then took place in order to reach a consensus, and all the reasons for the inclusion and exclusion of the studies were recorded at each stage.

The Springer search database was not able to limit the search based on Title, Keyword, and Abstract; thus, the authors had to search the documents as full text and then filtered the results based on Title, Keyword, and Abstract using the Endnote software capabilities. Also, the ACM search database had two databases to search. Therefore, the authors had to search within both of them and merge the results and after that remove the duplications that appeared. Moreover, IEEE and ACM search databases were not capable of searching within Title, Keyword, and Abstract simultaneously. Thus, the authors had to search the documents separately on the above fields and then merge the results and subsequently remove the duplications.

To keep uniformity for all the searches, the authors prepared the search phrase as consisting of Keywords derivations (e.g., social network, social networks, social networking). Given that the required information is not necessarily provided in the Title, Abstract, or Keywords of the documents, authors read other sections of the documents, and the risk of missing a relevant paper was mitigated by the first two authors independently reading all of the papers found in the initial search. Finally, the lack of SMS papers in the literature about social network community detection had harder the process of carrying out the research. Lack of consensus among researchers on some aspects such as categories of social network communities added to the research difficulties.

5.3 Implications for research and practice

The results of the research indicate that many different algorithms have been proposed and implemented for social network community detection and each of them has various pros and cons. In clustering, similar data points are grouped into the same cluster based on their properties and relations of the members and their interactions as well. According to the nature and features of clustering, it is a repetitive method that has been seen in most papers to identify the community. Hierarchical clustering is highly used in community detection and the creation of hierarchy can be agglomerative or partitioning [133]. As a result, knowing the number of groups as input is not necessary by using agglomerative algorithms, creating small groups of clusters is the disadvantage of agglomerative algorithms though. Because such small groups usually do not present important information. The results of the research also indicate that the partitioning algorithm such as k means is used frequently in community detection. It is also an efficient auxiliary algorithm to improve community detection in other methods. High speed, ease of use, and implementation of big data have been among the

benefits of algorithm k means in the papers investigated in this research even though the dependence of its results on the initial grouping of central points and the necessity of giving a specific number of communities as input to the algorithm can be mentioned its disadvantage. There are some methods such as greedy heuristics algorithms used to improve the quality of clustering.

The DBSCAN algorithms contain the noise factor and are resistant to outlier data; however, it is not able to cluster datasets with different densities well. It is noteworthy that though SCAN-based methods are fast, their dependency on the minimum similarity parameter harder their estimation. Our study also showed that some unsupervised community detection methods are used rarely such as spectral methods. Spectral methods have a higher quality compared to similarity-based methods such as clustering in separating communities. These methods minimize the cut function and are applied to small networks or simple cases, but their high computational complexity cannot be ignored in real-world networks.

The results of our study also highlight traditional models are less applicable for practical applications with complex structures. Modularity-based algorithms detect communities in the network by considering a dynamic process occurring in the network or statistical structures. Also, a result of this research shows that instances of modularity optimization methods showed more successful results based on accuracy in community detection. Some of these methods investigated in this scope based on label propagation, Node2vec, Infomap, and Louvain algorithms have been mentioned in this research.

Deep learning has been introduced as another emerging algorithm for detecting communities in real networks. This algorithm can model the relations using both high and low-dimensional data [24]. Our findings in this research indicate that this algorithm has sufficient performance for community detection in large-volume data by capturing complex features or unstructured information even though the approach based on deep learning is limited only by their input data patterns.

Our study shows that using conventional approaches to social network community detection often considers the structure of the network, and usually the characteristics of nodes are not taken into account while many real-world social networks are affected by additional features such as age or gender that can present interests and help to distinguish different communities. The literature review indicates that making a balance between local information and local search and global and semantic views can prevent falling into local optimum situations and it potentially improves the accuracy and time complexity of community detection.

The results of this research on metrics used to evaluate the quality of community detection show that NMI and modularity are two commonly used metrics for evaluating the quality of community structure. One of the reasons for the popularity of NMI is the ability to compare two clustering with different numbers of clusters. Also, more density of links in the actual community compared to the expected density in the conditions that users are connected randomly results in more modularity of the community in the way that modularity considers the most popular and widely accepted measure of the fitness of communities. The results of the research on datasets used in community detection show that five real-world networks with known community structures (i.e., Zachary's Karate Club, American College Football, Dolphin, Facebook, and DBLP) are the most used ones. Our investigations highlighted that in terms of edge type, loop, and availability, the dataset Zachary's Karate Club is the most popular in community structure identifications with no multiple edges and loops; meanwhile, it is publicly available on the internet. Also, LFR is the most used synthetic dataset with power-law distribution. It resembles real-world networks and its advantage over other datasets is supporting the heterogeneity in the distribution of node degrees and its different size of the community.

Community detection, a crucial aspect of network analysis, is progressing to address the issues presented by more intricate and expansive networks in different fields. The future of research in community detection is ready to tackle scalability and efficiency issues by creating algorithms that can efficiently manage large networks using parallel processing and distributed computing techniques. The need for algorithms that can offer timely insights is increasing in importance for tasks like social media monitoring and cybersecurity, making real-time community detection crucial. Researchers are currently researching dynamic networks, with the goal of adjusting algorithms to monitor changing community structures over time. Investigating multilayer and multiplex networks requires creating algorithms that are able to examine various network layers at the same time in order to reveal interconnected communities and grasp the dynamics of complex systems.

Incorporating node attributes and metadata boosts accuracy and relevance in community detection algorithms, and expanding methods to heterogeneous networks enhances modeling potential for various real-world uses. Utilizing machine learning and deep learning techniques, such as supervised and semi-supervised methods and graph neural networks, is helping to improve the accuracy and scalability of community detection by capturing complex network patterns and relationships. Game theory is becoming increasingly important in understanding strategic interactions in communities, leading to the creation of algorithms that encourage honest reporting of affiliations and study stable community structures. Strong evaluation measurements and thorough benchmarking initiatives are crucial for evaluating algorithm effectiveness in dynamic, multilayer, and attributed networks, promoting progress in community detection research in diverse fields like biology, sociology, economics, and beyond.

6 Conclusion

Our systematic map study investigated journal and valid conference papers that applied machine learning techniques in social network community detection between 2000 and 2024; 246 papers were selected to answer the four RQs of this study. Our results indicated that among the main categories of machine learning algorithms, unsupervised learning was more prevalent until 2022 and after that, semi-supervised and deep learning algorithms are the most used ones. Also, the efficiency metrics most frequently used are NMI, modularity, and F-measure. In terms of the used datasets, Zachary Karate and LFR are the most common ones. This paper presented all the above results statistically to provide a quantitative perspective of the mapped study.

Community detection in social networks varies in computational complexity across machine learning methods. Supervised learning scales with data size and feature complexity. Unsupervised methods like spectral clustering are computationally intensive, affected by cluster count and initialization. Furthermore, semi-supervised and deep learning involve complex data integration and network size considerations. Regarding to the dynamic nature of social networks, handling the high dynamicity is one of the gaps in proposed methods. Also, modern social networks involve diverse data types, including text, images, videos, and user behavior. Integrating and effectively leveraging multimedia data for community detection remains as a gap in the literature. Therefore, the recent methods for community detection in social networks have tended toward hybrid methods to handle accuracy, dynamicity, and heterogeneity simultaneously.

For future work, we propose investigating and reviewing the nature-inspired and artificial intelligence-based methods for community detection in social networks. The relationships in social networks are not always explicitly detectable, because of incomplete network topology and isolated subgraphs. Thus, one of the issues to investigate is the analysis of network topological structure in community detection. Furthermore, focusing on scalability with parallel processing and distributed computing, real-time detection, and exploring dynamic networks and multilayer structures to understand complex system dynamics, integration of node attributes, and game theory for stable community structures, are valuable issues to be investigated more in future works.

Community detection in social networks is a valuable technique for better understanding the structure and dynamics of interactions within social networks, which has important applications in areas such as marketing, politics, sociology, and data science.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10115-024-02201-8.

Author contributions Author 1 did the review of the literature and analyzed/prepared all the sections of the paper. Authors 2 and 3 supervised all the parts of this research and guided other authors to prepare and improve the research report. Author 4 cooperates with Author 1 in the process of literature review, steps of systematic mapping, and writing the research report.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Mohamed E-M et al (2019) A comprehensive literature review on community detection: approaches and applications. Proced Comput Sci 151:295–302
2. Alotaibi N, Rhouma D (2022) A review on community structures detection in time evolving social networks. J King Saud Univ-Comput Inf Sci 34(8):5646–5662
3. Chunaev P (2020) Community detection in node-attributed social networks: a survey. Comput Sci Rev 37:100286
4. Fani H, Bagheri E (2017) Community detection in social networks. Encycl Semant Comput Robot Intell 1(01):1630001
5. Enugala R et al (2015) Community detection in dynamic social networks: a survey. Int J Res Appl 2(6):278–285
6. Khatoon M, Banu WA (2015) A survey on community detection methods in social networks. Int J Educ Manag Eng 5(1):8
7. Kumar P, Singh D (2024) Community detection algorithms tools and applications. In: 2024 11th international conference on reliability, infocom technologies and optimization (Trends and Future Directions)(ICRITO). IEEE
8. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174

9. Coscia M, Giannotti F, Pedreschi D (2011) A classification for community discovery methods in complex networks. Stat Anal Data Min ASA Data Sci J 4(5):512–546
10. Plantié M, Crampes M, (2013) Survey on social community detection. In: Social media retrieval. pp 65–85
11. Malliaros FD, Vazirgiannis M (2013) Clustering and community detection in directed networks: a survey. Phys Rep 533(4):95–142
12. Azaouzi M, Rhouma D, Romdhane LB (2019) Community detection in large-scale social networks: state-of-the-art and future directions. Soc Netw Anal Min 9:1–32
13. Vieira VDF, RibeiroXavier C, Evsukoff AG (2020) A comparative study of overlapping community detection methods from the perspective of the structural properties. Appl Netw Sci 5(1):1–42
14. Yassine S, Kadry S, Sicilia M-Á (2021) Detecting communities in online learning repository. Anal Users' Interact Khan Acad Repos 12(1):57–64
15. Garey MR (1997) Computers and intractability: a guide to the theory of np-completeness, freeman. Fundamental
16. Naik D et al (2022) Parallel and distributed paradigms for community detection in social networks: a methodological review. Expert Syst Appl 187:115956
17. Cazabet R, Rossetti G, Amblard F (2017) Dynamic community detection. In: Alhajj Reda, Rokne Jon (eds) Encyclopedia of social network analysis and mining. Springer New York, New York, NY, pp 1–10. https://doi.org/10.1007/978-1-4614-7163-9_383-1
18. Falih I et al (2018) Community detection in attributed network. In: Companion proceedings of the web conference 2018
19. Wang Y et al (2023) Dual structural consistency preserving community detection on social networks. IEEE Trans Knowl Data Eng 35(11):11301–11315. https://doi.org/10.1109/TKDE.2022.3230502
20. Betzel RF (2023) Community detection in network neuroscience. Connectome analysis. Elsevier, Amsterdam, pp 149–171
21. Christopoulos K, Tsichlas K (2022) State-of-the-art in community detection in temporal networks. In: Maglogiannis I, Iliadis L, Macintyre J, Cortez P (eds) Artificial Intelligence Applications and Innovations. AIAI 2022 IFIP WG 12.5 International Workshops: MHDW 2022, 5G-PINE 2022, AIBMG 2022, ML@HC 2022, and AIBEI 2022, Hersonissos, Crete, Greece, June 17–20, 2022, Proceedings. Springer International Publishing, Cham, pp 370–381. https://doi.org/10.1007/978-3-031-08341-9_30
22. Cazabet R, Rossetti G (2019) Challenges in community discovery on temporal networks. In: Holme P, Saramäki J (eds) Temporal network theory. Springer International Publishing, Cham, pp 181–197. https://doi.org/10.1007/978-3-030-23495-9_10
23. Wang Y et al (2022) Temporal dual-attributed network generation oriented community detection model. IEEE Trans Emerg Top Comput
24. Su X et al (2022) A comprehensive survey on community detection with deep learning. IEEE Trans Neural Netw Learn Syst
25. Souravlas S et al (2021) A classification of community detection methods in social networks: a survey. Int J Gen Syst 50(1):63–91
26. Javed MA et al (2018) Community detection in networks: a multidisciplinary review. J Netw Comput Appl 108:87–111
27. Jonnalagadda A, Kuppusamy L (2016) A survey on game theoretic models for community detection in social networks. Soc Netw Anal Min 6:1–24
28. Wang Y et al (2024) Position matters: play a sequential game to detect significant communities. IEEE Trans Knowl Data Eng 36(7):3402–3416. https://doi.org/10.1109/TKDE.2023.3323567
29. Yang J, McAuley J, Leskovec J (2013) Community detection in networks with node attributes. In: 2013 IEEE 13th international conference on data mining. IEEE
30. Jin D et al (2021) A survey of community detection approaches: from statistical modeling to deep learning. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2021.3104155
31. Liu F, Xue S, Wu J, Zhou C, Hu W, Paris C, Nepal S, Yang J, Yu PS (2020) Deep learning for community detection: progress, challenges and opportunities. In: 29th international joint conference on artificial intelligence (IJCAI 20)
32. Dhilber M, Bhavani SD (2020) Community detection in social networks using deep learning. In: Distributed computing and internet technology: 16th international conference, ICDCIT 2020, Bhubaneswar, India, January 9–12, 2020, Proceedings 16. Springer
33. Souravlas S, Sifaleras A, Katsavounis S (2019) A parallel algorithm for community detection in social networks, based on path analysis and threaded binary trees. IEEE Access 7:20499–20519
34. Souravlas S, Sifaleras A, Katsavounis S (2020) Hybrid CPU-GPU community detection in weighted networks. IEEE Access 8:57527–57551

35. Souravlas S, Anastasiadou S, Katsavounis S (2021) A survey on the recent advances of deep community detection. Appl Sci 11(16):7179
36. Wohlin C et al (2013) On the reliability of mapping studies in software engineering. J Syst Softw 86(10):2594–2610
37. Petersen K, Vakkalanka S, Kuzniarz L (2015) Guidelines for conducting systematic mapping studies in software engineering: an update. Inf Softw Technol 64:1–18
38. Kitchenham BA, Budgen D, Brereton OP (2011) Using mapping studies as the basis for further research–a participant-observer case study. Inf Softw Technol 53(6):638–651
39. Petersen K et al (2008) Systematic mapping studies in software engineering. In: 12th International conference on evaluation and assessment in software engineering (EASE) 12
40. Keele S (2007) Guidelines for performing systematic literature reviews in software engineering. Technical report, ver. 2.3 ebse technical report. ebse
41. Boyd DM, Ellison NB (2007) Social network sites: Definition, history, and scholarship. J Comput-Mediat Commun 13(1):210–230
42. Bar-Ilan J (2018) Tale of three databases: the implication of coverage demonstrated for a sample query. Front Res Metr Anal 3:6
43. De Sutter B, Van Den Oord A (2012) To be or not to be cited in computer science. Commun ACM 55(8):69–75
44. Mahesh B (2020) Machine learning algorithms - a review. Int J Sci Res (IJSR) 9(1):381–386. https://doi.org/10.21275/ART20203995
45. Muhamedyev R (2015) Machine learning methods: an overview. Comput Modell New Technol 19(6):14–29
46. Alzubi J, Nayyar A, Kumar A (2018) Machine learning from theory to algorithms: an overview. J Phys Conf Ser 1142:012012. https://doi.org/10.1088/1742-6596/1142/1/012012
47. Muhammad I, Yan Z (2015) Supervised machine learning approaches: a survey. ICTACT J Soft Comput 5(3)
48. Jiang T, Gradus JL, Rosellini AJ (2020) Supervised machine learning: a brief primer. Behav Ther 51(5):675–687
49. Usama M et al (2019) Unsupervised machine learning for networking: techniques, applications and research challenges. IEEE Access 7:65579–65615
50. Saravanan R, Sujatha P (2018) A state of art techniques on machine learning algorithms: a perspective of supervised learning approaches in data classification. In: 2018 Second international conference on intelligent computing and control systems (ICICCS). IEEE
51. Zhang Q et al (2018) A survey on deep learning for big data. Inf Fus 42:146–157
52. Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT press, Cambridge
53. Zhou Z-H (2012) Ensemble methods: foundations and algorithms. CRC Press
54. Dong X et al (2019) A survey on ensemble learning. Front Comput Sci 14(2):241–258. https://doi.org/10.1007/s11704-019-8208-z
55. Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. Machine learning 6:37–66
56. Kumari A et al (2022) Supervised link prediction using structured-based feature extraction in social network. Concurr Comput Pract Exp 34(13):e5839
57. Balaji T, Annavarapu CSR, Bablani A (2021) Machine learning algorithms for social media analysis: a survey. Comput Sci Rev 40:100395
58. Khatoon M, Banu WA (2021) Unsupervised algorithms comparison in the perspective of community detection from social networks. In: 2021 Third international conference on inventive research in computing applications (ICIRCA). IEEE
59. Brusco M, Steinley D, Watts AL (2022) A comparison of spectral clustering and the walktrap algorithm for community detection in network psychometrics. Psychol Methods. https://doi.org/10.1037/met0000509
60. De Luca M et al (2023) A community detection approach based on network representation learning for repository mining. Expert Syst Appl 231:120597
61. Song Z et al (2023) Graph-based semi-supervised learning: a comprehensive review. IEEE Trans Neural Netw Learn Syst 34(11):8174–8194. https://doi.org/10.1109/TNNLS.2022.3155478
62. Van Engelen JE, Hoos HH (2020) A survey on semi-supervised learning. Mach Learn 109(2):373–440
63. Liu D et al (2020) The network representation learning algorithm based on semi-supervised random walk. IEEE Access 8:222956–222965
64. Qin M et al (2023) Towards a better tradeoff between quality and efficiency of community detection: an inductive embedding method across graphs. ACM Trans Knowl Discov Data 17(9):1–34. https://doi.org/10.1145/3596605

65. Zhang Suqi JW, Li J, Junhua Gu, Tang X, Xinyun Xu (2019) Semi-supervised community detection via constraint matrix construction and active node selection. IEEE Access 8:39078–39090
66. Wang N, Chen P, Li X (2014) Community detection in heterogeneous multi-mode social network via Co-training. In: Foundations of intelligent systems: proceedings of the eighth international conference on intelligent systems and knowledge engineering, Shenzhen, China, Nov 2013 (ISKE 2013). Springer
67. Guo K et al (2023) An attentional-walk-based autoencoder for community detection. Appl Intell 53(10):11505–11523
68. Gao J et al (2021) ICS-GNN: lightweight interactive community search via graph neural network. Proc VLDB Endowment 14(6):1006–1018
69. Yu Y et al (2019) A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput 31(7):1235–1270
70. Wu L et al (2020) Deep learning techniques for community detection in social networks. IEEE Access 8:96016–96026
71. Decelle A, Furtlehner C (2021) Restricted Boltzmann machine: recent advances and mean-field theory. Chin Phys B 30(4):040202
72. Dahlin J, Svenson P (2013) Ensemble approaches for improving community detection methods. arXiv preprint arXiv:1309.0242
73. Rajita B et al (2020) Spark-based parallel method for prediction of events. Arab J Sci Eng 45:3437–3453
74. Jiang M et al (2023) Random forest clustering for discrete sequences. Pattern Recogn Lett 174:145–151. https://doi.org/10.1016/j.patrec.2023.09.001
75. Dong S, Sarem M (2022) NOCD: a new overlapping community detection algorithm based on improved KNN. J Ambient Intell Humaniz Comput 13(6):3053–3063
76. Wu Hang-Yang, Y-LC (2020) Graph sparsification with generative adversarial network. In: In 2020 IEEE international conference on data mining (ICDM). IEEE, pp 1328–1333
77. He Q et al (2022) Reinforcement learning-based rumor blocking approach in directed social networks. IEEE Syst J 16(4):6457–6467
78. Chakraborty T et al (2017) Metrics for community analysis: a survey. ACM Computing Surv (CSUR) 50(4):1–37
79. Strehl A, Ghosh J (2002) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617
80. Choudhary C, Singh I (2022) Community Detection techniques and metrics: a state-of-the-art survey. In: Futuristic sustainable energy & technology: proceedings of the international conference on futuristic sustainable energy &technology (ICFSE, 2021), 19–20 September, 2021. CRC Press
81. Grass-Boada DH et al (2020) Overlapping community detection using multi-objective approach and rough clustering. in Rough sets: international joint conference, IJCRS 2020, Havana, Cuba, June 29–July 3, 2020, Proceedings. Springer
82. Fang-Ju A (2019) Research on a large-scale community detection algorithm based on non-weighted graph. Clust Comput 22(Suppl 2):2555–2562
83. Newman ME, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69(2):026113
84. Manju G, Geetha T (2013) Concept similarity based academic tweet community detection using label propagation. In: Mining intelligence and knowledge exploration: first international conference, MIKE 2013, Tamil Nadu, India, December 18–20, 2013. Proceedings. Springer
85. Cao J et al (2021) Compactness preserving community computation via a network generative process. IEEE Trans Emerg Top Comput Intell 6(5):1044–1056
86. Awal GK, Bharadwaj KK (2019) Leveraging collective intelligence for behavioral prediction in signed social networks through evolutionary approach. Inf Syst Front 21:417–439
87. Chen Y, Qiu X (2013) Detecting community structures in social networks with particle swarm optimization. In: ICoC. Springer
88. Televnoy A, Ivanov SE, Gorlushkina N (2020) Hybrid method of multiple factor data clusterization. In: Digital transformation and global society: 5th international conference, DTGS 2020, St. Petersburg, Russia, June 17–19, 2020, Revised Selected Papers 5. Springer
89. Dhelim S, Ning H, Aung N (2020) ComPath: user interest mining in heterogeneous signed social networks for Internet of people. IEEE Internet Things J 8(8):7024–7035
90. Kafeza E et al (2019) T-PCCE: twitter personality based communicative communities extraction system for big data. IEEE Trans Knowl Data Eng 32(8):1625–1638
91. Sattar NS, Arifuzzaman S (2020) Community detection using semi-supervised learning with graph convolutional network on GPUs. In: 2020 IEEE international conference on big data (Big Data). IEEE
92. Makris C, Pispirigos G, Rizos IO (2020) A distributed bagging ensemble methodology for community prediction in social networks. Information 11(4):199

93. Singh D, Verma A (2020) Extracting community structure in multi-relational network via deepwalk and consensus clustering. In: Intelligent human computer interaction: 11th international conference, IHCI 2019, Allahabad, India, December 12–14, 2019, Proceedings 11. Springer
94. Yamak Z, Saunier J, Vercouter L (2018) SocksCatch: automatic detection and grouping of sockpuppets in social media. Knowl-Based Syst 149:124–142
95. Lingam G, Ranjan Rout R, Somayajulu DVLN, Das SK (2020) Social botnet community detection: a novel approach based on behavioral similarity in Twitter network using deep learning. In: Proceedings of the 15th ACM Asia conference on computer and communications security. pp 708–718
96. Salehi S, Pouyan A (2020) Detecting overlapping communities in social networks using deep learning. Int J Eng 33(3):366–376
97. Singh J, Singh AK (2021) NSLPCD: Topic based tweets clustering using Node significance based label propagation community detection algorithm. Ann Math Artif Intell 89:371–407
98. Rahul et al (2021) Community detection using graphical relationships. In Inventive communication and computational technologies: proceedings of ICICCT 2020. Springer
99. Verma A, Bharadwaj KK (2015) Discovering communities in heterogeneous social networks based on non-negative tensor factorization and cluster ensemble approach. In: mining intelligence and knowledge exploration: third international conference, MIKE 2015, Hyderabad, India, December 9–11, 2015, Proceedings 3. Springer
100. Khatoon M, Aisha Banu W (2019) An efficient method to detect communities in social networks using DBSCAN algorithm. Soc Netw Anal Min 9(1):9
101. Pallis G, Zeinalipour-Yazti D, Dikaiakos MD (2011) Online social networks: status and trends. New Direct Web Data Manag 1:213–234
102. Sengan S et al (2021) The optimization of reconfigured real-time datasets for improving classification performance of machine learning algorithms. Math Eng Sci Aerosp (MESA), 12(1)
103. Assefa SA et al (2020) Generating synthetic data in finance: opportunities, challenges and pitfalls. In: Proceedings of the First ACM international conference on AI in finance
104. Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Phys Rev E 78(4):046110
105. Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci 99(12):7821–7826
106. Toujani R, Akaichi J (2019) An approach based on mixed hierarchical clustering and optimization for graph analysis in social media network: toward globally hierarchical community structure. Knowl Inf Syst 60(2):907–947
107. Xie Y et al (2019) Sim2vec: node similarity preserving network embedding. Inf Sci 495:37–51
108. Acharya DB, Zhang H (2020) Community detection clustering via gumbel softmax. SN Comput Sci 1(5):262
109. Zhang Z (2013) Community structure detection in social networks based on dictionary learning. Sci China Inf Sci 56:1–12
110. Li Z et al (2017) An efficient semi-supervised community detection framework in social networks. PLoS ONE 12(5):e0178046
111. Pang Z, Wang G, Yang J (2018) A multi-granularity decomposition mechanism of complex tasks based on density peaks. Big Data Mining Anal 1(3):245–256
112. Barros P et al (2018) Identifying communities in social media with deep learning. In: Social computing and social media. Technologies and analytics: 10th international conference, SCSM 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15–20, 2018, Proceedings, Part II 10. 2018. Springer
113. Kumar S, Panda B, Aggarwal D (2021) Community detection in complex networks using network embedding and gravitational search algorithm. J Intell Inf Syst 57:51–72
114. Chen Y et al (2018) Sequential sampling enhanced composite likelihood approach to estimation of social intercorrelations in large-scale networks. Quant Mark Econ 16:409–440
115. Basu T, Murthy C (2015) A similarity assessment technique for effective grouping of documents. Inf Sci 311:149–162
116. Leskovec J, Krevl A (2014) SNAP datasets: stanford large network dataset collection. MI, USA, Ann Arbor
117. Wu L, Ouyang Y, Shi C, Chen C-H (2021) Deep learning-based dynamic community discovery. In: Database systems for advanced applications. DASFAA 2021 International Workshops: BDQM, GDMA, MLDLDSA, MobiSocial, and MUST. Springer International Publishing, Taipei, pp 237–248
118. Huang L, Li R, Li Y, Gu X, Wen K, Xu Z (2012) ℓ 1-graph based community detection in online social networks. In: Web technologies and applications: 14th Asia-Pacific Web Conference, APWeb 2012. Springer Berlin Heidelberg, Kunming, China, pp 644–651

119. Adamic LA, Glance N (2005) The political blogosphere and the 2004 US election: divided they blog. In: Proceedings of the 3rd international workshop on Link discovery
120. Zare H, Hajiabadi M, Jalili M (2021) Detection of community structures in networks with nodal features based on generative probabilistic approach. IEEE Trans Knowl Data Eng 33(7):2863–2874
121. Prat-Pérez A, Dominguez-Sal D, Larriba-Pey J-L (2014) High quality, scalable and parallel community detection for large real graphs. In: Proceedings of the 23rd international conference on World wide web. pp 225–236
122. Luo J, Du Y (2020) Detecting community structure and structural hole spanner simultaneously by using graph convolutional network based Auto-Encoder. Neurocomputing 410:138–150
123. Mislove A et al (2007) Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement
124. Li M, Shuyi Lu, Zhang L, Zhang Y, Zhang Bo (2021) A community detection method for social network based on community embedding. IEEE Trans Comput Soc Syst 8(2):308–318
125. Paranjape A, Benson AR, Leskovec J (2017) Motifs in temporal networks. In: Proceedings of the tenth ACM international conference on web search and data mining
126. Chen W-Y et al (2009) Collaborative filtering for orkut communities: discovery of user latent behavior. In: Proceedings of the 18th international conference on World wide web
127. Chejara P, Godfrey WW (2017) Comparative analysis of community detection algorithms. In: 2017 conference on information and communication technology (CICT). IEEE
128. Nishikawa T, Motter AE (2015) Comparative analysis of existing models for power-grid synchronization. New J Phys 17(1):015012
129. Balke S et al (2022) JSD: a dataset for structure analysis in jazz music. Trans Int Soc Music Inf Retr 5(1):156–172
130. Namata G et al (2012) Query-driven active surveying for collective classification. In: 10th international workshop on mining and learning with graphs
131. He K et al (2018) Hidden community detection in social networks. Inf Sci 425:92–106
132. Negara ES, Andryani R (2018) A review on overlapping and non-overlapping community detection algorithms for social network analytics. Far East J Electron Commun 18(1):1–27
133. Bedi P, Sharma C (2016) Community detection in social networks. Wiley Interdiscip Rev Data Min Knowl Discov 6(3):115–135

Mahsa Nooribakhsh has completed her MSc degree in computer engineering in 2016. She is currently pursuing the PhD degree in Business Management and Administration with the Universitat politècnica de valència, Spain. Her current research interests include data mining, social network analysis, machine learning, deep learning, and meta-heuristic algorithms.

Marta Fernández-Diego received the European PhD degree in electronics and telecommunications engineering from the Lille University of Science and Technology, France, in 2001. She was, for several years, a member of a Software Development Team for mobile phone applications at an international information technology service company. She is currently an Associate Professor with the Department of Business Organization, Universitat Politècnica de València, Spain, where she teaches at the School of Computer Science. Her research interests include empirical software engineering, software effort estimation, and project risk management.

Fernando González-Ladrón-De-Guevara received the PhD degree in industrial engineering, in 2001. He has worked at several universities and IT companies across Europe and Latin America. He is currently an Associate Professor with the Telecommunications Engineering School, Universitat Politècnica de València (UPV). He has coauthored several articles published in well-known international journals. He regularly participates in the organizing committees of several national and international conferences. His research interests include crowdsourcing, ERP systems, and software engineering. He has participated in 27 research projects and contracts with different organizations and was responsible for seven of them.

Mahdi Mollamotalebi has completed his PhD at UTM, Malaysia, in 2013. He is currently an Assistant Professor at the Islamic Azad University of Qazvin (QIAU), Iran. He is a member of young and elite researchers club, Iranian computer society, and Iranian informatics association, Iran. He is also a member of IEEE. His areas of interest include cluster/grid/cloud computing, computer networks and wireless sensors, Internet of Things, machine learning, network security, and social network analysis.