Hindawi
Complexity
Volume 2022, Article ID 2072950, 17 pages
https://doi.org/10.1155/2022/2072950

Research Article

Rough-Set-Based Real-Time Interest Label Extraction over Large-Scale Social Networks

Xiaoling Huang ,[1] Lei Li,[2] Hao Wang,[2] Chengxiang Hu,[1] Xiaohan Xu,[2] and Changlin Wu[3]

[1] *School of Computer and Information Engineering, Chuzhou University, Chuzhou, Anhui 239000, China*
[2] *School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, Anhui 230009, China*
[3] *East China Langyashan Pumped Storage Co. Ltd., Chuzhou, Anhui 239000, China*

Correspondence should be addressed to Xiaoling Huang; hxl@chzu.edu.cn

Received 11 January 2022; Accepted 7 May 2022; Published 10 June 2022

Academic Editor: Manman Yuan

Labels provide a quick and effective solution to obtain people interesting content from large-scale social network information. The current interest label extraction method based on the subgraph stream proves the feasibility of the subgraph stream for user label extraction. However, it is extremely time-consuming for constructing subgraphs. As an effective mathematical method to deal with fuzzy and uncertain information, rough set-based representations for subgraph stream construction are capable of capturing the uncertainties of the social network. Therefore, we propose an effective approach called RS_UNITE_SS (namely, rough-set-based user-networked interest topic extraction in the form of subgraph stream), which is suitable for large-scale social network user interest label extraction. Specifically, we first propose the subgraph division algorithm to construct a subgraph stream by incorporating a rough set. Then, the algorithm for user real-time interest label extraction based on upper approximation (RILE) is proposed by using sequentially characteristics of the subgraph. Empirically, we evaluate RS_UNITE_SS over real-world datasets, and experimental results demonstrate that our proposed approach is more computationally efficient than existing methods while achieving higher precision value and MRR value.

1. Introduction

Traditional batch processing algorithms for extracting user interest usually make it difficult to store and process large-scale data with a complex social relationship at one time, making it more difficult to extract real-time interest for users. In response to these problems, the idea of large-scale complex social network graph decomposition is utilized for studying user interest label extraction [1]. To effectively solve the problem of handling large-scale user data, a "subgraph stream" data structure is proposed for interest label extraction in our previous work, which not only balances the effectiveness and efficiency of user interest extraction to a certain extent but also verifies that the data structure of the subgraph stream can be applied to traditional interest extraction methods. However, the construction of data structure has the following limitations:

(1) The data structure of the subgraph stream cannot consider the dynamic characteristics of social network information thoroughly; only the order of appearance of users in the subgraph is used to reflect the real-time characteristics of user interest when the real-time interests of users are extracted.

(2) In order to achieve the load balance of the number of users in each subgraph, more edges (relationship) are lost in the process of constructing the subgraph stream; meanwhile, in order to construct subgraphs with close user relations, a lot of computational costs is spent to determine the association between candidate users and the current subgraph.

To solve the above-mentioned problems in the construction of the subgraph stream, the subgraph stream is reconstructed by using the rough set theory. Rough set theory [2] is a powerful mathematical method to deal with fuzzy and uncertain information. It has been successfully applied to various fields [3–5]. When constructing a subgraph based on rough sets, the subgraph is defined as a set that cannot be precisely defined. In addition, some nodes, especially the boundary nodes of the subgraph, are grouped into multiple subgraphs [6]. The feasibility of dividing complex networks into communities based on the rough set theory is verified in the literature [7, 8]. Hence, community division and subgraph construction can be classified as cluster analysis based on graph structures. Consequently, the idea of subgraph division based on a rough set is feasible. However, the community division algorithm mentioned above needs to read the global network structure in dividing the community, which results in more computational and storage costs.

In response to the above problems, combined with the dynamic characteristics of social networks, a large-scale social network user interest label extraction method called RS_UNITE_SS (rough-set-based user-networked interest topic extraction in the form of subgraph stream) is proposed to construct subgraph stream with the aim of reducing time complexity and using the idea of "extracting user interest label as soon as the information of user arrives." Specifically, on the basis of the "subgraph stream" data structure, a rough set-based subgraph stream construction strategy is further proposed to better adapt to the real-time label extraction of users under the large-scale social network structure demand. Then, a real-time interest label extraction algorithm for social network users under the subgraph stream structure is proposed by combining the subgraph stream and the temporal characteristics of users in the subgraph, which not only guarantees the accuracy of real-time interest label extraction of social network users but also satisfies the requirement of lower time complexity for interest label algorithms. The main contributions of our work can be summarized as follows:

(1) RS3 (rough-set-based subgraph stream) is constructed based on rough set theory. The subgraph division algorithm is proposed to construct the subgraph stream by using the upper and lower approximations. Compared with the existing method of subgraph stream construction, the time complexity of RS3 is reduced. Therefore, it is more suitable for the division of graph structure in large-scale social networks. Moreover, a relatively stable and orderly subgraph stream strategy is proposed by using the number of intersections between the upper approximation of the subgraphs, which enables extracting user real-time interest labels from the continuously coming large-scale social network information.

(2) The algorithm for user real-time interest label extraction based upper approximation (RILE) is proposed by using sequentially characteristics of subgraph. Specifically, a relatively stable and effective subgraph user interest label update sequence algorithm named TAIL (the update timing algorithm for user interest) is proposed to improve the accuracy of user interest label extraction by combining the subgraph stream and different types of users in each subgraph. Then, combined with the proposed user update timing strategy, a method for extracting social network user interest labels is proposed by using TAIL under the sliding window subgraph environment.

(3) Our proposed method RS_UNITE_SS fully considers the dynamic characteristics of social networks to extract user interest labels. A method for using interest label update is proposed to infer real-time interest labels of social network users by combining user interest labels from the previous moment and modeling social network users' information published. By comparing with a state-of-the-art method for user interest label extraction, the experimental results show the effectiveness of the RS_UNITE_SS method.

In the following, the existing research work closely related to user interest label extraction is briefly reviewed, especially the work on user interest label extraction with a large-scale social network. Then, our related work in terms of the subgraph stream construction strategy are given.

In recent years, the researches on user interest extraction have received significant attention due to its wide range of applications. Several research works [9–13] investigated the methods for extracting user interest labels by analyzing information of user tweets or retweets. For example, Kang et al. [14] proposed a supervised user interest label extraction method based on Wikipedia to compensate for insufficient information and inaccuracy on Weibo. In addition, some research works [15, 16] performed user interest inference based on social relationships among users. For example, Bhattacharya et al. [15] proposed a behavior model for recommending followers. However, the limitations and the sparseness of information sources often lead to inaccurate and incomplete results when extracting user interests. Therefore, the user interest label can be extracted by combining text content information published according to social network users and social network relationship structure information. Wang et al. [17] proposed a user interest mining method based on a random walk graph model by means of user posts and social network relationships. However, this method is more suitable for extracting user interest from new or inactive users. In response to the cold start of user interest in social networks, Zarrinkalam et al. [18] identified user interest topics from users' information and social relationships and combined the association relationship between the interest topics to extract users' potential interests. However, the accuracy of predicting the potential user interest needs to be further improved. Wang et al. [19] proposed a method named UNITE_SS for extracting user interest in conjunction with user posts and social follow relationships, which achieves a balance between the influence of the two types of information on current user interest. Wang et al. [20] proposed an assumption that user similarity cannot only rely on the actual link relationship between social network users for measurement and established a user interest inference method based on two relationship graphs. However, a lot of storage space is required to store user weights in this method. Zheng et al. [21] used the similarity of social user content and social user relationship to detect overlapping communities when extracting user interest so as to solve the problems of sparse information in social networks. Besides, the core users' interests have different influences in diverse communities, which cannot be considered. However, the large-scale social network users and the complex social relationships between users require the high computational and storage costs of extracting social user interest labels, and user interest will shift over time [22, 23]. Therefore, it is an important issue how to extract real-time user interest label under massive, continuous, and dynamic large-scale social network environments.

Although the widespread existence of large-scale social network users and complex social relationships between users, there are relatively few efforts devoted to the use of large graph decomposition to solve the problem of the high cost of calculation and storage of large-scale social network user interest labels. Huang et al. [1] proposed a UNITE_SS method for extracting social network user interest labels based on the data structure of the "subgraph stream" and presented the extension of the traditional user interest label extraction method to the new "subgraph stream" environment. This method guaranteed a certain balance in the accuracy and efficiency of user interest label extraction and proved that the "subgraph stream" is an alternative way to solve problems in a large-scale social network environment. However, the UNITE_SS method only relied on the time sequence of the subgraph to reflect the user real-time interest label extraction, which leads to insufficient real-time interest label extraction. There has been relatively little research work focused on the construction of subgraph streams. Combining the social network characteristics of follower/followee characteristics, the frequency of user posting and forwarding posts, and the most association relationship among users, five subgraph stream construction strategies that meet the goal of subgraph load balancing were proposed [1]. However, due to much attention to the complex equilibrium problem of subgraphs in the construction strategy, the original sparse social relationship loses more social relationship, which affects the accuracy of interest label extraction to a certain extent. Besides, when constructing the subgraph stream, high time complexity is caused due to a large number of calculations to determine the association relationship between the candidate user and the current subgraph.

Except for the subgraph stream construction strategy proposed in the UNITE_SS method, there is no special research on the subgraph stream construction. Nevertheless, there were comprehensive researches on graph partition or community division. Among them, extensive researches focused on graph partition. Particularly, edge-based graph segmentation methods were proposed in [24–26], whereas node-based graph segmentation methods were investigated in [27–30]. However, the graph partition algorithm only takes the number of divided subgraph nodes to be equivalent and the minimum cut of the graph (node or edge) as the goal [31] without considering the similarity between nodes. In recent years, many methods have been proposed to solve the problem of community division in complex networks. Although community division is not specifically used to construct subgraphs, the idea of dividing communities is similar to the division of graphs. They are both clustering analysis problems based on graph structures. Therefore, the idea of community division can be used to construct subgraphs in complex social networks. Raghavan et al. [32] proposed a community tag propagation algorithm that aims to reduce time complexity and discover large-scale social network structures. However, this method is applied to discover overlapping communities. Lancichinetti et al. [33] proposed a local expansion method named LFM to find the overlapping communities, which causes the algorithm to be unstable due to the random selection of nodes. Rodriguez et al. [34] proposed an overlapping community detection method based on density peaks, but the algorithm has high time complexity. A rough set is an effective mathematical method applied to graph theory, which is capable of uncertain information [35–37]. In the traditional community division method, a node is usually classified into a community set. Lingras et al. [6] pointed out that the rough set theory is used to describe the community boundary as a fuzzy set, which is more flexible than the traditional community division set in terms of description. Finally, it is experimentally proved that the boundary nodes are more suitable for grouping into multiple communities. Zuo et al. [7] presented rough clustering methods to divide social networks into communities. Gupta et al. [8] proposed a network community division algorithm using an approximate set theory based on rough sets. Bie et al. [38] developed a community partition algorithm based on density peaks by using fuzzy clustering. However, the algorithm complexity is relatively high. In summary, these community division methods were constructed based on the global network structure, which is not suitable for large-scale social networks. In addition, these methods failed to take the problem of node load balancing into consideration in the community division. The research work related to graph partition and community division has been relatively mature, but they do not take into account the "big graph" characteristics of social networks. When these research results are directly used to construct a subgraph stream based on large-scale social network users, it may cause the following problems:

(1) In the social network environment, the relationship between large-scale social network users is complicated, and it is often difficult or impossible to deal with the "big graph" structure by directly using the subgraph division algorithm based on the global structures.

(2) The existing graph partition and community division methods pay more attention to subgraph partition, which do not consider the order between subgraphs. Therefore, it is extremely difficult to determine the order between these subgraphs when constructing the subgraph stream.

(3) The current methods of graph partition and community division do not take into account the directionality between nodes. Therefore, it is not appropriate to directly use the existing graph partition and community division methods to construct subgraph streams. To solve this issue, it is necessary to develop the subgraph stream with lower time complexity to lay the foundation for extracting higher-precision interest labels.

The remainder of the paper is organized as follows. First, we summarize the related work. Section 2 reveals the details of our proposed approach. Section 3 experimentally compares our proposed approach with the state-of-the-art methods on the Sina Weibo dataset and discusses the experimental results, and finally, Section 4 draws some conclusions and future work.

2. Materials and Methods

2.1. Problem Definition. In this subsection, the definition of the upper and lower approximation sets based on the rough sets are proposed to construct the subgraph; then the definition of the relevant characteristics of the subgraph is presented. Finally, the research problem and its premises are explained.

Definition 1. The similarity between u_i and u_j is defined as follows:

$$\mathrm{sim}\left(u_i, u_j\right) = \frac{\{f(u_i) \cup u_i\} \cap \{f(u_j) \cup u_j\}}{\sqrt{|f(u_i)| \cdot |(u_j)|}}, \tag{1}$$

where $f(u_i)$ represents the followee/follower set of the user u_i.

At the same time, in order to improve the similarity between two users that are directly connected, the user's own node is included when calculating the intersection of each user's neighbors. And 42% of social network users have less than 5 followers [39], so we use the user's follower list to calculate the similarity between users instead of his/her follower list.

Definition 2. The degree of membership $B(u_i, D_k)$ is a metric used to measure that the user u_i belongs to the subgraph D_k. We define $B(u_i, D_k)$ as follows:

$$B(u_i, D_k) = \sum_{j=1}^{j=|D_k|} \mathrm{sim}\left(u_i, u_j\right). \tag{2}$$

According to Definition 1, the similarity between the user u_i and the user u_j in the subgraph D_k is calculated, and then the similarity results are accumulated to obtain the membership degree of the user u_i belonging to the subgraph D_k. The larger the membership degree $B(u_i, D_k)$ is, the more it indicates that u_i more likely belongs to the subgraph D_k.

Definition 3. Subgraph definition. A subgraph is a data structure composed of social network users and their relationships. Here, by using the upper and lower approximations in rough set theory, we define the subgraph D_m as follows:

$$D_m = (\underline{R}(D_m), \overline{R}(D_m)). \tag{3}$$

where $\underline{R}(D_m)$ and $\overline{R}(D_m)$ represent the lower approximation and the upper approximation, respectively, and their detail definitions are represented as follows:

$$\begin{aligned}\underline{R}(D_m) &= \left\{u_i \middle| \frac{B(u_i, D_n)}{B(u_i, D_m)} < \eta \wedge \left(B(u_i, D_m) = \max_{p=1}^{p=|S_p|}\left(B\left(u_i, D_p\right)\right)\right), \forall u_i \in \text{List}, \forall D_n \in D\,(m \neq n)\right\},\\ \overline{R}(D_m) &= \left\{u_i \middle| \frac{B(u_i, D_n)}{B(u_i, D_m)} < \eta \wedge \left(B(u_i, D_m) = \max_{p=1}^{p=|S_p|}\left(B\left(u_i, D_p\right)\right)\right), \forall u_i \in \text{List}, \exists D_n \in D\,(m \neq n)\right\},\end{aligned} \tag{4}$$

where $0 < \eta \leq 1$ is the membership ratio threshold and $B(u_i, D_k) = \sum_{j=1}^{j=|D_k|} \mathrm{sim}(u_i, u_j)$, where $B(u_i, D_k)$ is used to measure the degree of membership of the user u_i belonging to the subgraph D_k.

All subgraphs D_n except D_m must satisfy: If the degree of membership of user u_i belonging to subgraph D_n is smaller than the product of η and the degree of membership of u_i belonging to subgraph D_m , and the degree of membership of u_i belonging to subgraph D_m is the largest membership value among the membership degrees of u_i belonging to other subgraphs, which means that the user u_i must belong to the subgraph D_m, the user u_i is divided into the lower approximation $\underline{R}(D_m)$.

Except for D_m, if there is a subgraph D_n, the degree of membership u_i of belonging to subgraph D_n is smaller than the product of η and the degree of membership of u_i belonging to subgraph D_m, and the degree of membership of u_i belonging to the D_m is the largest membership value among the membership degrees of u_i belonging to other subgraphs. Then the user u_i is divided into the upper approximation $\overline{R}(D_m)$ and the upper approximation $\overline{R}(D_n)$ at the same time.

Definition 4. Subgraph stream. Given a large-scale network structure $G = (V, E)$, where V and E represent the social network users and the relationship between users, respectively, the subgraph stream is a sequence of ordered subgraph collections. According to Definition 3, the subgraph stream S is defined as follows:

$$S = \{(\underline{R}(D_1), \overline{R}(D_1)), (\underline{R}(D_2), \overline{R}(D_2)), \ldots, (\underline{R}(D_m), \overline{R}(D_m))\}. \tag{5}$$

Definition 5. User interest. In a given time range $[t_a, t_b]$, the interest label I_{u_i} of the user u_i is expressed as a weight vector with topic Γ as follows:

$$I_{u_i} = \{\langle \tau_1^{u_i}, s_1^{u_i} >; \cdots; < \tau_\Gamma^{u_i}, s_\Gamma^{u_i}\rangle\}. \quad (6)$$

Hypothesis 1. The list of candidate users and the list of follower/followee by each user is known, mining large-scale social network user interest labels.

Hypothesis 2. If there are more common followers/followees between two users, the similarity of interest topic labels between users will be higher.

Hypothesis 3. The interests of VIP (very important person) users in the social networking platform are not easily affected by the outside world. Hence, their interests are relatively accurate, clear, and stable.

Problem 1. The problem of extracting real-time interest labels of social network users is studied based on the large-scale social network information stream. That is, we must solve the problem of how to construct a relatively stable and ordered subgraph stream *S*. Then, under the subgraph stream *S*, how to extract the user's real-time interest labels over time, and satisfy high label extraction accuracy and low time complexity at the same time is a big challenge.

2.2. Rough-Set-Based User Real-Time Interest Label Extraction Method. In this subsection, RS_UNITE_SS is proposed on the basis of the work of UNITE_SS [1] to better meet the needs of large-scale social network users' real-time interest label extraction. First, a more reasonable and stable subgraph stream based on the rough set theory RS3 is constructed. Then, the algorithm of TAIL is used to define the user interest label update timing in the subgraph, and finally based on user timing characteristics and rough sets, a large-scale social network user real-time interest label extraction algorithm RILE is proposed. As shown in Figure 1, the relationship between the RS_UNITE_SS method and the algorithms involved in this section is clarified.

2.2.1. Rough-Set-Based Subgraph Stream Construction Strategy RS3. In order to construct a more reasonable and stable subgraph stream, a strategy is proposed based on rough set theory to construct a subgraph stream RS3 (rough-set-based subgraph stream). This strategy is specifically divided into the following three steps:

(1) An effective algorithm for selecting the center node of the subgraph named SelectCenterNode is proposed to obtain a list of *M* center nodes using attribute information such as the number of users' followers and the follower list.

(2) The subgraph partition strategy named Cnode_subgraph. First, initial *M* subgraph sets are constructed with each node in the central node list as the core of the subgraph, respectively, and then the candidate user set is divided into upper or lower approximations of each subgraph based on Definition 3.

(3) The subgraph stream construction strategy named ConSubStream, which is used to define the order of subgraphs by using the overlap set between the subgraphs to construct a subgraph stream.

(1) Select the Strategy of Center Node of Subgraph. Select the center node of the subgraph is equivalent to finding the center node in the user set of the subgraph. There are many alternative ways to select the central node of the set, such as the classic degree centrality [40] and the density cluster centrality [13]. However, the selection of these central nodes needs to read the global network structure to describe or represent their own nodes, which is very computationally expensive under the large-scale social network structure. Therefore, a method for selecting the center node of the subgraph based on the characteristics of the social network is proposed, which select the center node of the subgraph by using the local network structure formed by the follower/followee list information of the social network user, as seen in Algorithm 1 for details.

Algorithm 1 presents the selection algorithm of the center node of the subgraph named SelectCenterNode, and the time complexity is $O(\text{calist} * M)$. The candidate user list caList is sorted in descending order of the number of each user's followers (line 1 in Algorithm 1). According to the sorted result, the default first user u_1 is selected as the initial central node list and stored in the scList, and the user is deleted from the caList at the same time (line 2 in Algorithm 1). If the number of central nodes in the scList does not reach *M*, then select the user u_i from the caList in turn to determine whether the current user u_i is the central node (lines 3–10 in Algorithm 1). If the number of common followers between the follower of the current user u_i and that of each central node in the scList does not exceed λ, then u_i is the central node and is stored in scList. Then, the user is deleted from caList. The directionality of the following/followed relationship is considered when we determine whether the current user u_i is the central node (line 6 in Algorithm 1). If u_i and u_j follow each other, the number between user u_i and his/her neighbor user u_j is recorded as 2 to avoid that the selected central nodes belong to the same subgraph.

(2) Divide the Rough Set Upper and Lower Approximations Based on the Central Node. According to the central node list and the definition of the rough set upper and lower approximations, the candidate users are divided into the upper approximation or the lower approximation centered on these central nodes. Finally, each subgraph is constructed using the attribute of the following/followed list between users, as seen in Algorithm 2 for details.

Algorithm 2 gives the upper and lower approximations algorithm Cnode_subgraph based on the central node partition, and its time complexity is $O(\text{caList} * M)$. In lines 1–5 in Algorithm 2, the candidate user u_i is selected from the calist in turn, and the degree of membership to each

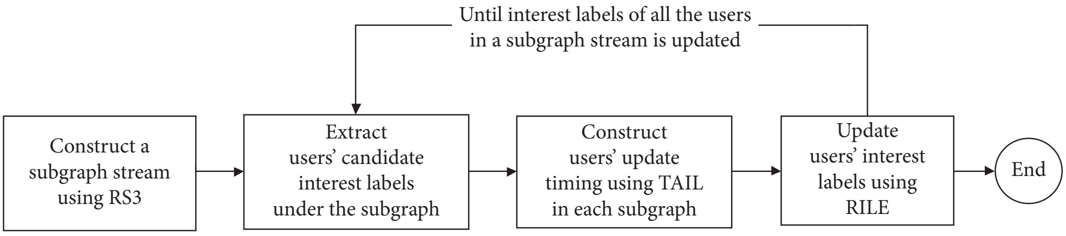
Until interest labels of all the users in a subgraph stream is updated
Construct a subgraph stream using RS3
Extract users' candidate interest labels under the subgraph
Construct users' update timing using TAIL in each subgraph
Update users' interest labels using RILE
End

Figure 1: An illustration diagram showing links between RS_UNITE_SS and the proposed algorithm.

```
Input: candidate user list caList = $\{u_1, u_2, \ldots, u_N\}$ and the followees/followers of each user.
Output: coreList composed of $M$ ($M \ll ca$List.size) central user nodes.
sort users in by user followees;
scList = $\{u_1\}$, caList = caList/$u_i$;
while calist.size $\neq 0$ && coreList.size < $M$ do
    $u_i$ = ca List.get ($i$) ($0 \le i \le$ core List.size)
    for $u_j \in$ coreList do
        if $\{f(u_i) \cup u_i\} \cap \{f(u_j) \cup u_j\} < \lambda$. then
            core List = core List $\cup\, u_i$; ca List = caList/$u_i$;
        end
    end
end
Output coreList.
```

Algorithm 1: Selecting center node of subgraph algorithm.

```
     Input: candidate user list named and subgraph list named gSet = $\{D_1, D_2, \ldots, D_M\}$.
     Output: divide caList each user into a subgraph.
(1)  for $u_i \in$ caList do
(2)      for $D_j \in g$Set do
(3)          Calculate the degree of membership $B(u_i, D_j)$ from $u_i$ to each subgraph $D_j$;
(4)          Find the maximum membership degree Max $(B(u_i, D_j))$ between $u_i$ and the subgraph, and store the subgraph number and membership degree of the current maximum membership degree into $m$ and $B(u_i, D_m)$ respectively;
(5)      end
(6)      count = 1;
(7)      for $D_j \in g$Set do
(8)          if $B(u_i, D_j)/B(u_i, D_m) < \eta$ then
(9)              count + +
(10)         else
(11)             tmp Upper List = tmp Upper List $\cup \{j\}$;
(12)         end
(13)     end
(14)     if count == $M$ then
(15)         $\underline{R}(D_m) = \underline{R}(D_m) \cup u_i$;
(16)         for $k \in$ tmpUpperList do
(17)             $\overline{R}(D_k) = \overline{R}(D_k) \cup u_i$;
(18)         end
(19)     end
(20) end
(21) return gSet = $\{(\underline{R}(D_1), \overline{R}(D_1)), (\underline{R}(D_2), \overline{R}(D_2)), \ldots, (\underline{R}(D_m), \overline{R}(D_m))\}$.
```

Algorithm 2: Algorithm for dividing upper and lower approximations of subgraph based on the central node named Cnode_subgraph.

subgraph is calculated. Then, the subgraph with the maximum degree of membership to which the user u_i belongs to is selected, and the number of the subgraph and the maximum degree of membership value are, respectively, stored in m and $B(u_i, D_m)$. Lines 6–20 in Algorithm 2 remove the maximum membership subgraph and count the subgraphs that do not satisfy the range of $\eta * \text{Max}(B(u_i, D_j))$. If the number of counts is equal to the number of subgraphs M, then the user u_i is divided into the lower approximation $\underline{R}(D_m)$ of the subgraph D_m; otherwise, the user u_i is divided into the upper approximation of the subgraph numbers listed in tmpUpperList. Among them, tmpUpperList is used to temporarily store a list of all subgraph numbers that satisfy the condition that the user u_i to a certain subgraph's membership degree is greater than $\eta * \mathrm{Max}(B(u_i, D_j))$.

(3) Constructing the Subgraph Stream Based on the Overlap Degree of Approximate on Rough Set. According to the constructed subgraphs, the multiple subgraphs are sorted according to the overlap between the subgraphs to form a subgraph stream. The idea of subgraph stream construction: (1) intersect the upper approximations of the two subgraphs to obtain the number of overlapping users, (2) select the two subgraphs with the largest number of overlapping users as the initial subgraphs and add them to the subgraph stream S, and (3) take the subgraph selected from the remaining subgraphs, which can ensure that the number of intersections between the subgraph and the last subgraph of the current subgraph stream S is the maximum, and the subgraph is regarded as a subsequent subgraph and added to the subgraph stream S. Finally, iterate successively until all the subgraphs are sorted to form a subgraph stream.

Algorithm 3 presents Con Sub Stream, an algorithm for constructing a subgraph stream based on the overlap degree of approximations in rough sets. Lines 1–8 in Algorithm 3 calculate the upper approximation intersection between two subgraphs in the subgraph list gSet. Lines 9–10 in Algorithm 3 select the two subgraphs with the largest number of intersections as the initial two subgraphs. If the number is the same, then a group of two subgraphs is selected as the initial two subgraphs, which are stored in S and are deleted from gSet at the same time. In lines 11–21 in Algorithm 3, the remaining subgraph list gSet is sorted to form a subgraph stream. The subgraph selected from the gSet can guarantee the maximum overlap with the last subgraph block in the current subgraph stream S as a principle, and the subgraph is added to S and is deleted from the gSet at the same time. And iterate until all subgraphs are added to S.

Compared with the subgraph stream construction strategy of the UNITE_SS, the subgraph stream constructed by the algorithm of Con Sub Stream is orderly and stable. Not only there is a closer relationship within the subgraph but also there is a relatively stable relationship between adjacent subgraphs, which is conducive to the extraction of interest labels. Among them, the algorithm complexity of Con Sub Stream is $O(M)$, and that of subgraph stream construction in UNITE_SS is $O(\mathrm{caList.size})$.

```
     Input: subgraph list gSet = $\{D_1, D_2, \ldots, D_M\}$.
     Output: subgraph stream $S$.
(1)  for $D_i \in g$Set do
(2)      for $j = i + 1s$ to gSet.size do
(3)          if $|\overline{R}(D_i) \cap \overline{R}(D_j)| > \max$ then
(4)              $\max = \overline{R}(D_i) \cap \overline{R}(D_j)$;
(5)              max Position = $\{i, j\}$;
(6)          end
(7)      end
(8)  end
(9)  $S = \{\overline{R}(D_i), \overline{R}(D_j)\}$;
(10) $g$Set = $g$Set/$\{\overline{R}(D_i), \overline{R}(D_j)\}$;
(11) count = 2
(12) while count $\leq M$ do
(13)     for $k = 1$ to $g$Set.size do
(14)         if $|\overline{R}(D_k) \cap \overline{R}(D_j)| > \max$ then
(15)             $\max = \overline{R}(D_k) \cap \overline{R}(D_j)$;
(16)             max Position = $\{k\}$;
(17)         end
(18)     end
(19)     $S = S + \{D_{\mathrm{maxPosition}}\}$, gSet = gSet/$\{D_{\mathrm{maxPosition}}\}$
(20)     count + +;
(21) end
(22) return $S$.
```

Algorithm 3: Constructing the subgraph stream algorithm based on the overlap degree of approximations in rough sets named ConSubStream.

2.2.2. RILE Based on the Upper Approximation User Real-Time Interest Label Extraction Strategy under the Subgraph Stream. After constructing the subgraph stream S based on the RS3 method, the user interest label update timing characteristics and the upward approximation-oriented user interest label update timing strategy is proposed, and then a more flexible label weighting mechanism is designed. Finally, using the update timing strategy and the label weight mechanism, the algorithm RILE is proposed based on the upper approximation to extract real-time interest labels of large-scale social network users.

The RILE method mainly includes three steps: (1) obtain the initial candidate interest labels, (2) define the user interest label update timing strategy TAIL, and (3) propose the user's real-time interest label extraction algorithm.

(1) Extraction Candidate Interest Labels. For each user $u_i (u_i \in U)$, the attribute information of each user u_i is defined as $\mathrm{Attribute}(u_i) = (\mathrm{Posts}_{u_i}, FS_{u_i}, F_{u_i}, t)$, where t represents the current time, Posts_{u_i} represents the posts posted by the user up to time t, FS_{u_i} represents the number of followers of the user u_i, and F_{u_i} is the followee collection of u_i.

If the interest label of the user u_i at time $t + \Delta t$ is to be extracted, it is necessary to determine whether the interest label of the user at time t has been extracted. If the interest label of user u_i at time t has been extracted, then the interest label at time t is directly used as the candidate interest label, and the information of all users in the subgraph $D^w_{t+\Delta t}$ are crawled from t to $t + \Delta t$; hence, nearly q posts published within the time period are recorded as $\mathrm{Posts}\, q/\Delta t_1$; if the interest label of user u_i at time t has never been extracted, then the nearest w subgraphs up to time $t + \Delta t$ are collected and are recorded as $\mathrm{Posts}\, q/t + \Delta t_1$.

$$D^w_{t+\Delta t} = \{D_{t+\Delta t+w-1}, \ldots, D_{t+\Delta t}\} \tag{7}$$

For w subgraphs within adjacent sliding windows, the topic modeling method is used to extract candidate interest labels [19].

```
    Input: $S = \{(\underline{R}(D_1), \overline{R}(D_1)), (\underline{R}(D_2), \overline{R}(D_2)), \ldots, (\underline{R}(D_m), \overline{R}(D_m))\}$.
    Output: Users'update timing for each subgraph.
 (1) while $k < m$ do
 (2)     for $u_i \in \overline{R}(D_k)$ do
 (3)         if $FS_{u_i} > \lambda_1$ then
 (4)             $u_i \in U_c$;
 (5)         else if $F_{u_i} \cap U_c > \lambda_2$ and degree $(u_i) > \lambda_3$ then
 (6)             $u_i \in U_p$;
 (7)         else
 (8)             $u_i \in U_n$;
 (9)         end
(10)     end
(11)     tmp Timing List.add $(U_c \longrightarrow U_p \longrightarrow U_n)$.
(12) end
(13) return tmp Timing List.
```

Algorithm 4: TAIL algorithm.

(2) The Upper-Approximation-Oriented User Interest Label Update Timing Strategy TAIL. An influential person is often followed by others, so the interest label of an influential person is not only relatively stable but also easily affects the interest label of his/her followers or fans. Therefore, the sooner you update the interest label of influential people, it is of great significance to improve the overall user interest label extraction accuracy. Based on the above motivation and the combination of the number of users' followers and the degree center, the TAIL algorithm for updating user interest labels in the subgraph is proposed. That is, the users in the upper approximation corresponding to the subgraphs are divided into different user types, and then the order of updating the interest labels of different types of users is given.

Algorithm 4 presents the upper approximation-oriented user interest label update timing algorithm TAIL. The specific execution process of Algorithm 4 is shown as follows:

Step 1: Define the user type. Three types of users are defined, namely: big V user (very important person) list U_c, ordinary active user list U_p, and ordinary inactive user list U_n. According to the number of users' follower and their degree centers in the subgraph, the users in the upper approximation $\overline{R}(D_m)$ corresponding to the current subgraph are divided to identify the user type. In lines 3–4 of Algorithm 4, if the number of user' followers FS_{ui} is greater than the threshold λ_1, then user u_i belongs to the list of celebrities or V users U_c; in lines 5–6, if the intersection between the follower of user u_i named F_{ui} and the big V user list U_c is greater than the threshold λ_2 and the degree center of u_i in the subgraph is greater than the threshold λ_3, then the user belongs to the common active user list U_p; else, the user u_i belongs to the ordinary inactive user list U_n in line 8 of Algorithm 4.

Step 2: According to the different types of users, the order of user updating labels is given. That is, on the basis of obtaining the initial user candidate labels, priority is given to updating interest labels of influential users or celebrities. For the interest labels of ordinary users, the interest labels of their own are often updated after the labels of the influential users concerned have been updated. In line 11 of Algorithm 4, the order of the updating user interest label in each subgraph is shown as follows: big V user list U_c, ordinary active user list U_p, ordinary inactive user list U_n. tmp Timing List is used to temporarily store the sequence of the user label update in the subgraph. That is, according to the attention set of big V users, priority is given to the updating interest labels of users in the big V user list U_c; then, according to the updated big V user interest labels, the interest labels of users in the common active user list U_p are updated; finally, according to the updated interest labels of big V users and ordinary active users, the interest labels of users in the list of ordinary inactive users U_n are updated.

(3) User's Real-Time Interest Label Extraction Algorithm RILE. According to user interest label update timing features and flexible label weighting mechanism, a real-time interest label extraction algorithm RILE is proposed for large-scale social network users based on the upper approximation.

Before explaining the RILE algorithm, the motivation of the new label weight update formula definition is firstly introduced based on the UNITE algorithm. Different from the UNITE method, when the user interest label is updated, the RILE method not only considers the influence of the update sequence of the user interest label on the accuracy of interest label extraction but also considers that the influence of different neighbor followers on the current user interest label is also different, such as the interests of the current user influenced by that of their friends or celebrities.

$$s_{ip} = s_{ip} + \sum_{u_k \in \text{neighbor}(u_i)} W_{ik} * s_{ip} * \text{is Over Lap}(u_{ik}), \tag{8}$$

where

$$\text{is Over Lap}(u_{ik}) = \begin{cases} 1, & \text{if overlap List}(u_{ik}) \neq \varnothing; \\ 0, & \text{if overlap List}(u_{ik}) = \varnothing, \end{cases} \tag{9}$$

and

$$W_{ik} = \frac{w_{ik}}{\sum w_{ik}}. \tag{10}$$

In addition, the weight W_{ik} used in [1] is more suitable for the global network structure. However, the definition of the weight under the subgraph structure is more limited because the network structure is sparser than the global structure when the subgraph is constructed, which affects the accuracy of label update. Therefore, it is of practical significance to redefine the user interest label weight update formula (10) through different social neighbor user types, as shown in the following formula:

$$w_{ik} = \begin{cases} \dfrac{\alpha * C_f\left(u_i, u_k^c\right)}{\alpha * \sum_{k=1}^{|c|} C_f\left(u_i, u_k^c\right) + \beta * \sum_{k=1}^{|p|} C_f\left(u_i, u_k^p\right) + \gamma * \sum_{k=1}^{|n|} C_f\left(u_i, u_k^n\right)}, \\[2ex] \dfrac{\beta * C_f\left(u_i, u_k^p\right)}{\alpha * \sum_{k=1}^{|c|} C_f\left(u_i, u_k^c\right) + \beta * \sum_{k=1}^{|p|} C_f\left(u_i, u_k^p\right) + \gamma * \sum_{k=1}^{|n|} C_f\left(u_i, u_k^n\right)}, \\[2ex] \dfrac{\gamma * C_f\left(u_i, u_k^n\right)}{\alpha * \sum_{k=1}^{|c|} C_f\left(u_i, u_k^c\right) + \beta * \sum_{k=1}^{|p|} C_f\left(u_i, u_k^p\right) + \gamma * \sum_{k=1}^{|n|} C_f\left(u_i, u_k^n\right)}, \end{cases} \tag{11}$$

where $\alpha + \beta + \gamma = 1$, $C_f(u_i, u_k^*) = f(u_i) \cap f(u_k^*)$, and $f(*)$ represents the user's followees list; u_k^c represents the big V user; u_k^p means normal active users; and u_k^n means normal inactive users. $C_f(u_i, u_k^c)$ represents the number of common attention sets between the user u_i and the big V user, where $C_f(u_i, u_k^p)$ represents the number of common attention sets between the user u_i and the ordinary active user u_k^p ($u_k^p \in U_p$ and $u_k^p \in \text{neighbor}(u_i)$) and $C_f(u_i, u_k^n)$ represents the number of common attention sets between the user u_i and the nonactive user u_k^n ($u_k^n \in U_n$ and $u_k^n \in \text{neighbor}(u_i)$).

According to the constructed subgraph stream S, user interest label update timing TAIL and label update weight formula (12), a real-time user interest label extraction algorithm RILE based on the upper approximation is proposed, as shown in Algorithm 5.

Algorithm 5 presents a large-scale social network user interest label extraction algorithm named RILE. In Algorithm 5, in lines 1–5, a subgraph set is constructed with a sliding window, then construct a subgraph based on the user's attention list attributes in the subgraph set, and then use the TALE algorithm to divide each user in the subgraph into one of the three types, which ultimately constitutes three sets of big V user list U_c, ordinary active user list U_p, and ordinary inactive user list U_n. In lines 6–13, if the user's interest label at time t has not been extracted, then we crawl all the user' most recent Weibo content in the subgraph up to time $t + \Delta t$ and perform topic modeling on the Weibo content to obtain the initial interest labels, and finally update user interest label based on the UNITE algorithm according to the subgraph, the new weight formula, and the user update timing characteristics. If the user has extracted the user interest label at time t, then the user interest label at time t is used as the user's initial user interest label at time $t + \Delta t$; at the same time, only the Weibo content published by the user at $[t, t + \Delta t]$ is crawled, and topic modeling has to be conducted. Lines 14–26, according to big V user list U_c, ordinary active user list U_p, and ordinary inactive user list U_n order, use the label extraction method UNITE to update the user interest label, except that the interest label update score formula is replaced with formula (12).

The comparisons between RS_UNITE_SS and UNITE proposed in this study are analyzed as follows:

(1) Real time. When UNITE extracts user interest labels, it needs to crawl all the microblogs posted by users until $t + \Delta t$. Using these microblogs and the global network structure, the interest labels of all users will be extracted uniformly in a batch process. Even if you want to extract the labels of a specified user at a certain moment, you still need to read the global network structure, so extracting labels of interest often requires a lot of storage and time. However, when RS_UNITE_SS extracts a specific user interest label, it only needs to crawl the microblogs posted by users in the subgraph where the user u_i is located within Δt to extract a real-time user interest label.

(2) Complexity. UNITE extracts interest labels at different moments, and the required time complexity is $O(\text{calist.size})$; RS_UNITE_SS uses a subgraph stream to extract user interest labels. Therefore, when the subgraph stream is constructed, the time complexity of extracting the user interest label is $O(M)$ $(M \ll \text{calist.size})$.

3. Results

3.1. Data Sets. The two real-world Sina Weibo platform datasets named SubGraphData [1] and WeiboData [41] are applied to evaluate the effect of RS_UNITE_SS.

```
Input: subgraph stream $S = \{(\underline{R}(D_1), \overline{R}(D_1)), (\underline{R}(D_2), \overline{R}(D_2)), \ldots, (\underline{R}(D_m), \overline{R}(D_m))\}$.
Output: interest labels of all user at the time of $t + \Delta t$.
for $j = 1$ to $S$.size do
    for $k = j$ to $j + w - 1$ do
        $\overline{R}(D_j) = \overline{R}(D_j) \cup \overline{R}(D_k)$;
        obtain subgraph $G_j$ using the followees of each user in $\overline{R}(D_j)$;
        put users in $\overline{R}(D_j)$ into one of the user types called $\{U_c, U_p, U_n\}$ using $G_j$ and TALE algorithm;
        if the user interest label has not been extracted at the time of $t$ then
            crawl the nearly $q$ posts posted by all users in $\overline{R}(D_j)$ until the time of $t + \Delta t$, named Posts $q/t + \Delta t_1$;
            topic modeling on $Posts^{q}_{t+\Delta t_1}$ to get candidate user interest label;
            use formulas (9), (10), and (12) to update the interest label of users in $\overline{R}(D_j)$ according to the sequence of $U_c$, $U_p$, and $U_n$;
        else
            crawl the nearly $q$ posts posted by all users in $\overline{R}(D_j)$ between the time of $t$ and $t + \Delta t$, named Posts $q/\Delta t_1$ and topic
    modeling on Posts $q/\Delta t_1$;
            user interest label at the time of $t$ is used as candidate user interest label;
        end
        while $u_i \in U_c$ && new interest labels extracted by user $u_i$ are detected do
            update interest labels of $u_i$;
            use UNITE, formulas (9), (10), and (12) to update the interest label of users who follow $u_i$ according to $G_j$;
        end
        while $u_i \in U_p$ && new interest labels extracted by user $u_i$ are detected do
            update interest labels of $u_i$;
            use UNITE, formulas (9), (10), and (12) to update interest label of users who follow $u_i$ according to $G_j$;
        end
        while $u_i \in U_n$ && new interest labels extracted by user $u_i$ are detected do
            update interest labels of $u_i$;
            use UNITE, formulas (9), (10), and (12) to update interest label of users who follow $u_i$ according to $G_j$;
        end
    end
end
return interest labels of all users and each interest of $u_i$ is formalized as $I_{u_i} = \{\langle \tau_1^{u_i}, s_1^{u_i} >; \cdots; < \tau_\Gamma^{u_i}, s_\Gamma^{u_i} \rangle\}$.
```

ALGORITHM 5: User real-time interest label extraction based upper approximation algorithm named RILE.

(1) SubGraphData. There are more following relationships among users in SubGraphData, and a large number of original microblogs posted by users are crawled in the dataset. In order to construct SubGraphData, we start from the Sina Weibo user "Li Kaifu" and crawl the information such as the current user's followee list, the number of followers, and the posts published by the user in the followee list until October 30, 2017. After 4 iterations and denoising processing, a social network user dataset consisting of about 2.07 million user information, and more than 1.92 million microblogs are generated with an average of 54 following relationships for each user.

(2) WeiboData. The original intention of the public dataset WeiboData was to verify the user's social influence and the effect of user forwarding behavior modeling. Therefore, there are more forwarding relationships between users in this dataset. The dataset of WeiboData initially starts from 100 random Sina Weibo users; the follower list and the followee list of each user are collected and iterated in turn; and finally, a list of approximately 1.7 million users, 300,000 Weibo, and more than 23.75 million reposted microblogs are generated. After data preprocessing, the dataset contains 6 subparts such as dynamic follow relationship network and forwarding relationship network to meet different experimental needs.

3.2. Evaluation Metrics.

In order to verify the effect of real-time interest labeling of large-scale social network users under the subgraph stream, the metrics of social relationship change rate RChange, modularity, perplexity, precision, and MRR are used to evaluate the efficiency of user interest label extraction. The definitions of modularity, precision, and MRR can be found in literature [19].

3.2.1. Social Relationship Change Rate RChange.

In order to verify the influence of social network relationships on the construction of the subgraph stream, the average change rate of user relationship indicator RChange is proposed, which is used to measure the frequency of changes in user social relationships per unit time under large-scale social networks, as shown in the following formula:

$$R\,\text{Change} = \frac{\sum_{i=1}^{i=n}\left(c_{u_i}/tr\left(u_i\right)\right)}{n}, \tag{12}$$

where n denotes the total number of social network users, $tr(u_i)$ represents the total number of users followed by user

u_i in a period of time t, and c_{u_i} is the number of changes in the following relationship within a period of time t.

3.2.2. Perplexity. The metics of perplexity widely used in language models [12, 18, 20] is introduced to evaluate the user interest label, which is defined as follows:

$$\text{Perplexity}\left(D_{\text{test}}\right) = \exp\left\{-\frac{\sum_{d_u \in D_{\text{test}}} \sum_{w \in W_{d_u}} \log\left(\sum_{z \in Z} P_z(w) S_u(Z)\right)}{\sum_{d_u \in D_{\text{test}}} \left|W_{d_u}\right|}\right\}, \tag{13}$$

where d_u is the test document collection of user u, W_{d_u} is a series of words or phrases in d_u, $P_z(w)$ is the probability of word w on topic $z \in Z$, and $S_u(Z)$ is the score of user u on topic Z. Here, the data of September 2012 are randomly selected from WeiboData as the training set, and the remaining data are used as the test set. Similarly, the data of May 2017 are randomly selected from SubGraphData as the training set, and the remaining data are used as the test set. If a method has a lower perplexity value in the test set, the method has a better effect on extracting user interest labels.

3.2.3. Evaluation of Reposting Behavior. To verify the accuracy of the user interest label, we use the evaluation of reposting behavior [20] to judge whether the user is interested in a topic on the Weibo.

Suppose that a Weibo p published by a user is represented as a vector $S(p) = \left\{s_p(\tau_1), \ldots, s_p(\tau_\Gamma)\right\}$ based on the topic of $T = \{\tau_1, \ldots, \tau_\Gamma\}$, where

$$s_p(\tau_k) = \frac{\prod_{w \in W_p} P_{\tau_k}(w)}{\sum_{i=1}^{i=\Gamma} \prod_{w \in W_p} P_{\tau_i}(w),} \tag{14}$$

where W_p denotes the set of words w in microblog p and $P_{\tau_i}(w)$ represents the probability of word w on topic τ_i. The semantic similarity value is calculated between $S(p)$ and the interest tag vector $S(u_i)$ of user u_i, where is calculated according to formula (7); then the Weibo posted by user u_i is calculated according to the similarity, and all the values of similarity are sorted in descending order; finally, the metrics of precision and MRR are used to compare the sorted microblogs with the most expected results of the microblogs actually reposted by the user to predict the user's reposting behavior and achieve the goal of identifying the effect of user interest label extraction.

3.3. Experiments That Affect the Construction of Subgraph Streams

3.3.1. The Influence of Social Relationships on the Construction of Subgraph Stream. Social relations mainly include following relations, forwarding relations and @ relations, so this section mainly tries these three relations to construct a subgraph stream. In order to verify the stability of constructing subgraph streams based on different social relations, on the WeiboData dataset, the changes in different social relations among social users in a unit of time are counted. Among them, the text named "graph_170 w_1 month.txt" is used as the WeiboData subdataset to crawl the information of each user's watch list in the time period from September 28, 2012, to October 29, 2012, in units of days. The experimental results are shown in Table 1.

According to the statistical data of the following relationship in Table 1, the subgraph stream based on the following relationship is the most stable social relationship structure, and the relationship change rate is about 3%. The subgraph stream constructed based on the forwarding relationship changes the most in unit time because the forwarding relationship is an unstable and relatively sparse dynamic relationship structure. In addition, the change rate of the subgraph stream based on the @ relationship is lower than the change rate of the subgraph stream based on the forwarding relationship. Because @ relationship is a dynamic relationship, its @ relationship structure in the dataset is sparser than the forwarding relationship.

Although users with following or forwarding relationships often share the same interests, we need a more stable and effective attention relationship to construct the subgraph stream to solve the complex, sparse, and dynamic characteristics of large-scale social network data. Since the change rate of constructing the subgraph stream based on the following relationship is the lowest, even if the relationship structure of the following relationship changes when the subgraph stream is constructed, the influence of this change in the relationship on the accuracy of user interest label extraction is minimal. Therefore, during the experiment, we avoid doing much processing on the changes in the subgraph relationship and pay more attention to the changes in the attributes of user nodes in the subgraph structure.

3.3.2. The Influence of Subgraph Partition and Subgraph Sequence on the Construction of Subgraph Stream. On the basis of experiments that the attention relationship is of great significance for constructing a stable subgraph stream, the subgraph stream algorithm RS3 is developed based on rough sets and following relations, which is in multimetrics with the existing multiple subgraph construction-related algorithms such as MaxAssociation & HighFrequency, CommunityDetection, and the rough set community division algorithm proposed by Gupta. Among them, MaxAssociation & HighFrequency and CommunityDetection strategies are introduced in UNITE_SS [1].

Based on the model of Gupta et al. [8], this method that approximates the two-hop neighbor connection network of each node as the upper approximation is proposed based on the rough set theory. Then, the "relative connection" and other calculation formulas are used to continuously iterate. Nodes in the network are divided into communities, and overlapping communities can be found.

The parameter values involved in the RS3 method are defined as follows: in the strategy of constructing the center node of the subgraph, the number of the neighbor node intersection λ is set to 5; the parameter of the upper and lower approximations η of the subgraph is set to 0.8, where the larger η indicates that the greater the degree of distinction between the subgraphs, the fewer the overlapping

Table 1: Changes in different social relations on WeiboData.

Metrics	Following (%)	Retweet (%)	@ (%)
Relationship change rate within a month	**3.16**	47.63	25.46

Table 2: Comparison of subgraph partitioning algorithms on SubGraphData.

Metrics	MaxAssociation & HighFrequency	CommunityDetection	Gupta et al.	RS3
Number of subgraphs (M)	2,000	306	1,792	**1,875**
Whether to read the global structure	No	Yes	Yes	**No**
Time complexity (h)	$O(n)$	$O(n^2)$	$O(n^2)$	$\mathbf{O(n)}$
Running time	8.73	104.28	111.29	**5.32**
Modularity	0.1345	0.2454	0.2867	**0.2228**

Table 3: Comparison of subgraph partitioning algorithms on WeiboData.

Metrics	MaxAssociation & HighFrequency	CommunityDetection	Gupta et al.	RS3
Number of subgraphs (M)	1,800	278	1,423	**1,620**
Whether to read the global structure	No	Yes	Yes	**No**
Time complexity (h)	$O(n)$	$O(n^2)$	$O(n^2)$	$\mathbf{O(n)}$
Running time	7.62	97.89	100.16	**4.27**
Modularity	0.1211	0.2198	0.2535	**0.2051**

Table 4: Comparison of user interest label extraction effects under different subgraph stream construction strategies on SubGraphData.

Extract user interest label base on different subgraph stream	RS3 (sort by overlap)	RS3 (sort by random)
MRR	**0.842**	0.723
Precision@3	**0.756**	0.643

users. The number of subgraph M for the MaxAssociation & HighFrequency method is set to 2,000 and 1,800 in SubGraphData and WeiboData, respectively.

Tables 2 and 3 show the experimental results of each subgraph construction algorithm on the two datasets. As shown in Tables 2 and 3, the model of Gupta et al. has the best modularity value and the best community division effect. However, the model of Gupta et al. [8] needs to read the global network structure to obtain each result when subgraphs are divided. The two-hop neighbor network of a point requires multiple iterations to obtain the optimal subgraph, which takes a long time; the time complexity of the algorithm is $O(n^2)$, which is very unsuitable for large-scale social network subgraph construction. Analogously, the CommunityDetection method has similar problems. In addition, taking the SubGraphData dataset as an example, although this method has a high modularity value, the number of nonoverlapping subgraphs is divided into only 306 pieces, and there are 5 pieces containing about 30,000 users, which lead to the unbalanced load of the subgraphs. RS3 is superior to the MaxAssociation & HighFrequency method on the metrics of modularity because the MaxAssociation & HighFrequency strategy is more focused on ensuring the load balance of the number of users in the subgraph when facing the conflict between the balance of the number of users in the subgraph and the retention of tightness between users, while RS3 is more biased to keep close social relationship between users and then obtains overlapping subgraphs. In summary, according to the number of subgraphs, the metrics of modularity, running time, and time complexity, RS3 is a promising subgraph construction strategy with the best comprehensive metrics effect and provides a feasible solution for decomposing large-scale social network graphs.

In the following, in order to verify whether the subgraph stream composed of different subgraph orders will affect the extraction of social user interest labels, the subgraphs are divided based on the RS3 method. Then, these subgraphs are sorted in random order and sorted based on Algorithm 3 to form a subgraph stream. Finally, the RS_UNITE_SS method is used to extract the user's real-time interest label on both subgraph stream environments. Table 4 shows the extraction of user interest in two subgraph streams under the SubGraphData to evaluate its MRR and precision. Among them, the SubGraphData contains the core interest label manually annotated by each user as the expected result, which facilitates the evaluation of the experiment.

As shown in Table 4, the subgraph stream sorted based on the degree of overlap between the subgraphs is 16.46% and 17.57%, higher than the subgraph stream based on the random sorting on the metrics of MRR and precision for extracting user real-time interest labels.

Because the subgraph stream construction method based on the order of overlap degree not only uses the relationship between subgraphs but also uses the association relationship between adjacent subgraphs to provide more sufficient social relations for user real-time interest label extraction.

Table 5: Comparison of perplexity metrics on SubGraphdata.

K	50	75	100
RS_UNITE_SS	**6,792**	**6,774**	**6,705**
EIUI(SR)_S	6,845	6,823	6,804
CoReg_SS	6,897	6,845	6,819
UNITE_SS	6,912	6,871	6,855

Table 6: Comparison of perplextiy metrics on WeiboData.

K	50	75	100
RS_UNITE_SS	**8,652**	**8,641**	**8,626**
EIUI(SR)_S	8,771	8,764	8,749
CoReg_SS	8,892	8,875	8,856
UNITE_SS	8,977	8,965	8,947

Therefore, in the following experiments, we use the subgraph stream construction method RS3 based on the degree of overlap to construct the subgraph stream for extracting user real-time interest label.

3.4. Comparison with Benchmark Methods. In order to further verify the effect of the RS_UNITE_SS method in extracting user real-time interest labels, we compare RS_UNITE_SS with state-of-the-art user interest extraction methods, including UNITE_SS, EIUI(SR) (abbreviated as EIUI(SR)_SS), and CoReg based on subgraph stream (abbreviated as CoReg_SS).

CoReg_SS: a social user interest modeling method named CoReg, is proposed based on the assumption that there are similar interests among users who have a common referent and pointed link relationship [20]. EIUI(SR) method is used to extract the user's real-time interest label within the sliding window under the subgraph stream environment constructed based on CoReg. The detail parameters in the experiment are referred to in the literature [20].

EIUI(SR)_SS: EIUI(SR) [18] is proposed to extract users' potential interest labels by combining user-published text content, social relationship between users, and association information between interest label topics. Except for the different user interest label extraction methods used under the subgraph stream, we use the same subgraph stream as the CoReg based on the subgraph stream to extract the user's real-time interest label. In addition, the parameters involved in the experiment can be found in literature [15].

RS_UNITE_SS method: the method is proposed in Section 4 of this study. The values of parameters are defined as follows: in the strategy of constructing the center node of the subgraph, the neighbor node intersection number λ is set to 5; the threshold λ_1 used to determine the big V user is set as 4,000; the threshold λ_2 for determining the number of intersections between ordinary active users and large V users is set to 5; and the degree center of the subgraph where ordinary active users located λ_3 is set to 3. In the weight calculation formula (12) involved in the RILE algorithm, $\alpha = 0.7$, $\beta = 0.2$, and $\gamma = 0.1$; the sliding window w is set to 2.

3.4.1. Evaluating User Interest Topic Based on Perplexity. In order to evaluate user interest topics, the metrics of perplexity are used to compare the extraction effects of user interest labels under different K on two datasets, as shown in Tables 5 and 6. The smaller the value of perplexity, the better the performance.

As shown in Tables 5 and 6, the perplexity of the RS_UNITE_SS is the lowest. In RS_UNITE_SS, the current user's neighbor types are classified, and the weights of different neighbor types are set to update the current user's initial interest labels, so it has the best performance among all methods. The UNITE_SS method performs the worst on the perplexity. Since more social relationships among users are lost when constructing the subgraph in order to ensure the load balance of the number of user nodes in the subgraph, the accuracy of interest label extraction is affected due to insufficient information when the user interest label is updated. RS_UNITE_SS, EIUI(SR)_SS, and CoReg_SS use the goal that have a close relationship between each other and preserve the association between adjacent subgraphs to construct subgraphs as much as possible, so the social relationship loss is relatively small and the user interest topics are clear. EIUI(SR)_SS performs better than CoReg_SS. Although the real-time nature of the user interest label is not considered, CoReg_SS can further extract the potential user interest label of active users through the correlation between topics. UNITE_SS performed the worst because the strategy of MaxAssociation & HighFrequency is used to construct a load balancing subgraph stream and more association relationships are lost, which lead to the small influence of neighbor users and reduce the accuracy of interest label extraction. In addition, we can conclude that the number of topics K has little effect on the perplextiy of each method.

3.4.2. User Interest Label Evaluation Based on Predicted Behavior. The method of predicting user reposting behavior is used to evaluate the effect of user interest label extraction. In the experiment, we use the precision [9] and MRR [19] to calculate users' reposting behavior on the datasets. Among them, Table 7 shows the comparison results of MRR based on user reposting behavior with different methods on SubGraphData and WeiboData datasets. Figures 2 and 3

Table 7: Comparison results of MRR based on user retweeting behavior of different methods under different dataset.

Methods	RS_UNITE_SS	EIUI(SR)_SS	CoReg_SS	UNITE_SS
SubGraphData	**0.229**	0.221	0.216	0.209
WeiboData	**0.268**	0.249	0.233	0.224

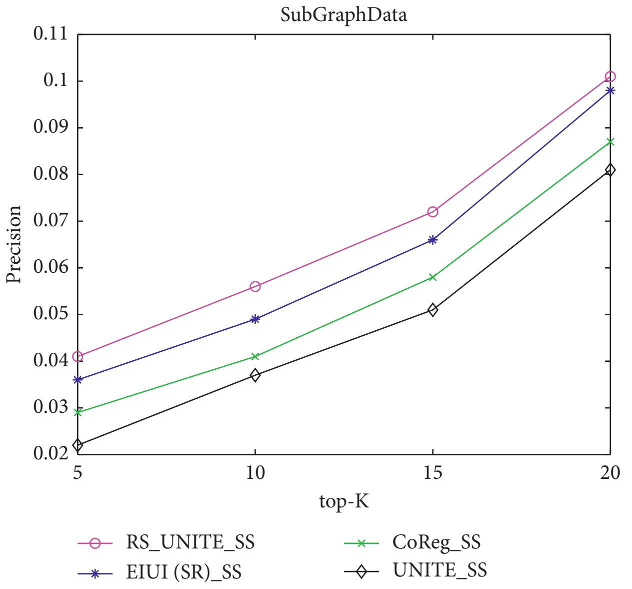

Figure 2: Comparison of methods based on user interest behavior prediction accuracy on SubGraphData.

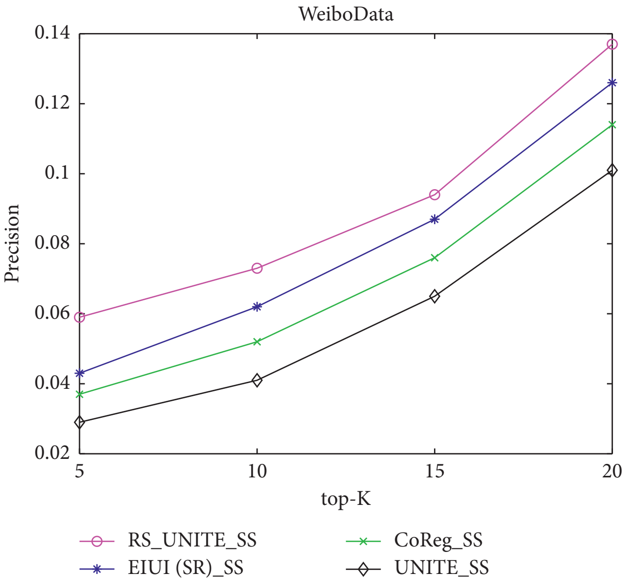

Figure 3: Comparison of methods based on user interest behavior prediction accuracy on WeiboData.

show the comparison results of different methods on SubGraphData and WeiboData based on the user's top-K interest behavior prediction under the metrics of precision, where K is set as 5, 10, 15, and 20, respectively.

As can be seen from Table 7 and Figures 2 and 3, RS_UNITE_SS performs better than the benchmark method with promising performance in terms of MRR and precision.

Due to RS_UNITE_SS extracts users' labels based on the user interest label extracted from the content posted by the current user and its initial score, and the score of the current user interest label is updated based on the overlap between the current users and neighbor users. When both EIUI(SR)_SS and CoReg_SS update the user interest label, the score of the neighbor user interest label is mainly used to affect the current user interest label, which may cause the deviation of the user interest label due to excessive dependence on neighbors. The EIUI(SR)_SS method performs better than CoReg_SS when extracting interest labels. Because CoReg_SS only relies on similar user structures identified by the relationship of followees or followers to update current user interest labels and does not make full use of the actual social relationship between social network users. UNITE_SS performs the worst in all methods. On the one hand, the UNITE_SS method has a large loss of relationship when constructing the subgraph stream, which results in the lack of neighbor user information when the current user's interest label is extracted. This problem actually affects the accuracy. On the other hand, the different influences of neighbor users and the update sequence of neighbor users' labels are not considered to extract real-time interest labels.

3.4.3. Evaluation of User Interest Label Extraction Efficiency Based on Running Time. As shown in Tables 8 and 9, all methods contain the time spent in constructing the subgraph stream in the first real-time interest label extraction. Taking the SubgraphData dataset as an example, 7 hours, 14 minutes, and 18 seconds is taken to construct a subgraph stream based on a rough set. In the UNITE_SS method, the subgraph stream constructed based on the MaxAssociation & HighFrequency strategy took 8 hours, 15 minutes, and 3 seconds because it costs a lot of time to calculate the current user and both the previous subgraph and the current subgraph to form a joint subgraph, which performs association calculations to select the user who is most associated with the current joint subgraph. Both EIUI(SR) and CoReg have low processing efficiency due to a large amount of data when extracting user interest labels. Taking the SubGraphData as an example, a part of the data (6.96 G) is intercepted for measurement, and they are, respectively, calculated according to the total size of the data (25G) at 0.9942 G/h and 0.96 G/h processing speed. When the subgraph stream is constructed, the traditional user interest label extraction

Table 8: CRunning time (h) comparison of user interest label extraction methods on SubGraphData.

Methods	RS_UNITE_SS	EIUI(SR)_SS	CoReg_SS	UNITE_SS	EIUI(SR)	CoReg
The running time required for the first real-time interest label extraction	**11.34**	12.23	12.43	11.55	24.58	25.87
The running time required for the second real-time interest label extraction (Δt = 1 month)	**3.45**	4.63	4.77	4.73	25.96	26.32

Table 9: Running time (h) comparison of user interest label extraction methods on WeiboData.

Methods	RS_UNITE_SS	EIUI(SR)_SS	CoReg_SS	UNITE_SS	EIUI(SR)	CoReg
The running time required for the first real-time interest label extraction	**9.45**	10.11	11.04	9.67	22.93	23.12
The running time required for the second real-time interest label extraction (Δt = 1 month)	**2.89**	3.22	3.71	2.96	23.37	24.26

method based on the subgraph stream only needs to focus on the interest label extraction, so it takes less time. However, EIUI(SR) and CoReg still need to read global data for user interest label extraction.

As can be seen from Tables 8 and 9, when RS_UNITE_SS first extracts real-time interest labels of users, the time it takes is similar to that of other interest label extraction methods based on subgraph streams. However, RS_UNITE_SS took the least time among all the baseline methods in the second user real-time interest label extraction (Δt = 1 month). Because the interest label update algorithm is defined in the RS_UNITE_SS method, we only need to pay attention to crawl the user's microblog information updated in the past month. This provides a meaningful solution to construct a stable subgraph stream when extracting the user's real-time interest label.

The existence of user interest label extraction methods do not consider the large-scale and dynamic characteristics of social networks simultaneously, which lead to difficulties in large-scale user interest label extraction. Not to mention, it can maintain the real-time extraction of user interest labels. The main reason for that is existing user interest extraction methods are batch algorithms, which cannot handle large-scale user interest labels. By contrast, our proposed approach RS_UNIT_SS incorporates a rough set to construct a subgraph stream, which not only lays the foundation for extracting large-scale users' interest labels but also is able to cope with the uncertainties of social networks. Therefore, our proposed approach RS_UNIT_SS achieves a better balance between efficient and efficiency.

4. Conclusions

We propose a rough-set-based real-time interest label extraction approach RS_UNITE_SS for large-scale social network users, which can adapt to large-scale and dynamic characteristics of social networks simultaneously. Specifically, the large-scale social network structure is divided into subgraphs based on the approximations of rough set theory and the follow relationship between social users, and then a relatively stable sequential subgraph stream is constructed based on the degree of overlap between the subgraphs. Furthermore, under the data structure of the subgraph stream, the user interest label update timing strategy is developed according to the different user types between the subgraphs. On the basis of this strategy, the user's real-time interest label extraction algorithm is proposed. Experimental results illustrate that RS_UNITE_SS not only guarantees the accuracy of label extraction on large-scale social network data but also ensures that the extracted labels are real-time. Our proposed approach is only applicable to social network data built by relatively stable relationships and cannot adapt to social network data constructed by dynamic relationships such as @ and comment. In our future work, we plan to study the improvement of the performance of our proposed approach.

Data Availability

The data used to support the findings of this study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under grant no. 2016YFB1000900, the National Natural Science Foundation of China (NSFC) under grant nos. 61503114 and 61906060, the Key Project of the Natural Science Foundation of Educational Commission of Anhui Province under grant nos. KJ2021A1065 and KJ2019A0642, and the Anhui Provincial Natural Science Foundation under grant nos. 1808085MF177 and 2008085MF224.

References

[1] X. Huang, H. Wang, L. Li, Y. Zhu, and C. Hu, "Microbloggers' interest inference using a subgraph stream," *Intelligent Data Analysis*, vol. 25, no. 2, pp. 397–417, 2021.

[2] Z. Pawlak, "Rough sets," *Rough Sets and Data Mining*, vol. 11, no. 5, pp. 3–7, 1997.

[3] J. Zhao and N. Qi, “Analysis of international competitiveness of multinational corporations based on rough sets,” *Complexity*, vol. 8, pp. 1–10, 2021.

[4] X. Q. Liao, S. Nazir, J. X. Shen, B. L. Shen, and S. Khan, “Rough Set Approach toward Data Modelling and User Knowledge for Extracting Insights,” *Complexity*, vol. 2021, Article ID 7815418, 9 pages, 2021.

[5] H. An, W. Yang, J. Huang, A. Huang, Z. Wan, and M. An, “Identify and Assess Hydropower Project’s Multidimensional Social Impacts with Rough Set and Projection Pursuit Model,” *Complexity*, vol. 2020, Article ID 9394639, 16 pages, 2020.

[6] P. Lingras and G. Peters, “Rough clustering,” *WIREs Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 64–72, 2011.

[7] W. Zhe, “Research in social network based on rough set clustering algorithm,” *International Journal of Advancements in Computing Technology*, vol. 4, no. 15, pp. 295–301, 2012.

[8] S. Gupta, P. Kumar, and B. Bhasker, “A rough connectedness algorithm for mining communities in complex networks,” in *Proceedings of the International Conference on Big Data Analytics and Knowledge Discovery*, pp. 34–48, Springer, New York , NY, USA, August 2016.

[9] M. Michelson and S. A. Macskassy, “Discovering users’ topics of interest on twitter: a first look,” in *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data*, pp. 73–80, Association for Computing Machinery, Toronto, Canada, October 2010.

[10] W. Wu, B. Zhang, and M. Ostendorf, “Automatic Generation of Personalized Annotation Tags for Twitter Users”” in *Proceedings of the 2010 Annual Conference Of the North American Chapter of the Association for Computational Linguistics*, pp. 689–692, Association for Computing Machinery, Los Angeles, CA, U S A, June 2010.

[11] T. Vu and V. Perez, “Interest mining from user tweets,” in *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pp. 1869–1872, Association for Computing Machinery, San Francisco, CA, USA, October 2013.

[12] Z. H. Xu, R. Lu, L. Xiang, and Q. Yang, “Discovering user interest on twitter with a modified author-topic model,” in *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pp. 422–429, IEEE, Lyon, France, October 2011.

[13] N. F. Rajani, K. Mcardle, and J. Baldridge, “Extracting topics based on authors, recipients and content in microblogs,” in *Proceedings of the 37th International ACM Conference on Research and Development in Information Retrieval*, pp. 1171–1174, Association for Computing Machinery, New York, NY, U S A, July 2014.

[14] J. Kang, H. Choi, and H. Lee, “Deep recurrent convolutional networks for inferring user interests from social media,” *Journal of Intelligent Information Systems*, vol. 52, no. 1, pp. 191–209, 2019.

[15] P. Bhattacharya, M. B. Zafar, N. Ganguly, S. Ghosh, and K. P. Gummadi, “Inferring user interests in the Twitter social network,” in *Proceedings of the 8th ACM Conference on Recommender Systems*, pp. 357–360, Association for Computing Machinery, Silicon Valley, CA, USA, October 2014.

[16] W. He, H. Y. Liu, J. He, S. Tang, and X. Y. Du, “Extracting interest tags for non-famous users in social network,” in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 861–870, Association for Computing Machinery, Melbourne, Australia, October 2015.

[17] T. T. Wang, H. Y. Liu, J. He, and X. Y. Du, “Mining user interests from information sharing behaviors in social media,” in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 85–98, Springer, Gold Coast, Australia, May 2013.

[18] F. Zarrinkalam, M. Kahani, and E. Bagheri, “Mining user interests over active topics on social networks,” *Information Processing & Management*, vol. 54, no. 2, pp. 339–357, 2018.

[19] H. Wang, X. Huang, and L. Li, “Microblog oriented interest extraction with both content and network structure,” *Intelligent Data Analysis*, vol. 22, no. 3, pp. 515–532, 2018.

[20] J. Wang, W. X. Zhao, Y. He, and X. Li, “Infer user interests via link structure regularization,” *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 2, pp. 1–22, 2014.

[21] J. Zheng, S. Wang, D. Li, and B. Zhang, “Personalized recommendation based on hierarchical interest overlapping community,” *Information Sciences*, vol. 479, pp. 55–75, 2019.

[22] F. M. Facca and P. L. Lanzi, “Mining interesting knowledge from weblogs: a survey,” *Data & Knowledge Engineering*, vol. 53, no. 3, pp. 225–241, 2005.

[23] Z. G. Zhu, Y. H. Zhou, X. Y. Deng, and X. H. Wang, “A Graph-Oriented Model for Hierarchical User Interest in Precision Social Marketing,” *Electronic Commerce Research & Applications*, vol. 41, Article ID 100845, 2019.

[24] N. Jain, G. Liao, and T. L. Willke, “Graphbuilder: scalable graph ETL framework,” in *Proceedings of the First International Workshop on Graph Data Management Experiences and Systems*, vol. 4, June 2013.

[25] C. Xie, L. Yang, W. J. Li, and Z. H. Zhang, “Distributed power-law graph computing: theoretical and empirical analysis,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1673–1681, MIT Press, Montreal, Canada, December 2014.

[26] F. Petroni, L. Querzoni, K. Daudjee, S. Kamali, and G. Iacoboni, “HDRF: stream-based partitioning for power-law graphs,” in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 243–252, Association for Computing Machinery, Montreal, Canada, October 2015.

[27] I. Stanton and G. Kliot, “Streaming graph partitioning for large distributed graphs,” in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1222–1230, Association for Computing Machinery, Beijing, China, August 2012.

[28] I. Stanton, “Streaming balanced graph partitioning algorithms for random graphs,” in *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithm*, pp. 1287–1301, Society for Industrial and Applied Mathematics, Portland, ORE, USA, January 2014.

[29] C. E. Tsourakakis, C. Gkantsidis, B. Radunovic, and M. Vojnovic, “FENNEL: streaming graph partitioning for massive scale graphs,” in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pp. 333–342, Association for Computing Machinery, New York, NY, U S A, February 2014.

[30] C. Tsourakakis, “Streaming graph partitioning in the planted partition model,” in *Proceedings of the 2015 ACM International Conference on Online Social Networks*, pp. 27–35, Association for Computing Machinery, New York, NY, U S A, November 2015.

[31] Z. Abbas, V. Kalavri, P. Carbone, and V. Vlassov, “Streaming graph partitioning,” *Proceedings of the VLDB Endowment*, vol. 11, no. 11, pp. 1590–1603, 2018.

[32] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Physical Review*, vol. 76, no. 3, Article ID 036106, 2007.

[33] A. Lancichinetti, S. Fortunato, and J. Kertész, “Detecting the overlapping and hierarchical community structure in complex networks,” *New Journal of Physics*, vol. 11, no. 3, Article ID 033015, 2009.

[34] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[35] M. Akram and F. Zafar, “Hybrid soft computing models applied to graph theory,” *Studies in Fuzziness and Soft Computing*, vol. 380, pp. 1–419, 2020.

[36] M. Akram and F. Zafar, “Multi-criteria decision-making methods under soft rough fuzzy knowledge,” *Journal of Intelligent and Fuzzy Systems*, vol. 35, no. 3, pp. 3507–3528, 2018.

[37] F. Zafar and M. Akram, “A novel decision-making method based on rough fuzzy information,” *International Journal of Fuzzy Systems*, vol. 20, no. 3, pp. 1000–1014, 2018.

[38] R. Bie, R. Mehmood, S. Ruan, Y. Sun, and H. Dawood, “Adaptive fuzzy clustering by fast search and find of density peaks,” *Personal and Ubiquitous Computing*, vol. 20, no. 5, pp. 785–793, 2016.

[39] J. Hopcroft, T. Lou, and J. Tang, “Who will follow you back?” *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM ’11*, pp. 1137–1146, 2011.

[40] E. Lazega, S. Wasserman, and K. Faust, “Social network analysis: methods and applications,” *Revue Française de Sociologie*, vol. 36, no. 4, pp. 781–783, 1995.

[41] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Z. Li, “Social influence locality for modeling retweeting behaviors,” in *Proceedings of the 23th International Joint Conference on Artificial Intelligence*, pp. 2761–2767, (IJCAI), U S A, August 2013.