An Efficient Evolutionary User Interest Community Discovery Model in Dynamic Social Networks for Internet of People

Liang Jiang[1,2,3], Leilei Shi[1,2], Lu Liu[4], Jingjing Yao[5], Bo Yuan[4] and Yongjun Zheng[4]
[1]School of Computer Science and Telecommunication Engineering, Jiangsu University, China
[2]Jiangsu Key Laboratory of Security Technology for Industrial Cyberspace, Jiangsu University, China
[3]Jingjiang College of Jiangsu University, China
[4]School of Electronics, Computing and Mathematics, University of Derby, UK
[5]School of Economy and Finance, Jiangsu University, China
Email: l.liu@derby.ac.uk

Abstract: Internet of People (IoP), which focuses on personal information collection by a wide range of the mobile applications, is the next frontier for Internet of Things (IoT). Nowadays, people become more and more dependent on the Internet, increasingly receiving and sending information on social networks (e.g., Twitter, etc.); thus social networks play a decisive role in IoP. Therefore, community discovery has emerged as one of the most challenging problems in social networks analysis. To this end, many algorithms have been proposed to detect communities in static networks. However, microblogging social networks are extremely dynamic in both content distribution and topological structure. In this paper, we propose a model for Efficient Evolutionary User Interest Community Discovery which employs a nature-inspired genetic algorithm to improve the quality of community discovery. Specifically, a preprocessing method based on Hypertext Induced Topic Search improves the quality of initial users and posts, and a label propagation method is used to restrict the conditions of the mutation process to further improve the efficiency and effectiveness of user interest community detection. Finally, the experiments on the real datasets validate the effectiveness of the proposed model.

Keywords: Community discovery; dynamic; genetic algorithm; Internet of People (IoP); label propagation; nature-inspired algorithm

I. INTRODUCTION

From the standpoint of referring to people as cyber entities, Internet of People (IoP) [39,44] denotes the mapping of social individuals. It is committed to data collection, modeling, and ubiquitous intelligence and has a great application of crowd sourced, Internet-based personal information. Internet of People (IoP), which focuses on personal information collection by a wide range of the mobile applications, is the next frontier for Internet of Things (IoT). Nowadays, people become more and more dependent on the Internet, send and receive information in social networks (e.g., Twitter, etc.). For example, Microblogging network is a network structure formed by a large number of users based on their intricate relationships [1-5]. In a microblogging network, people establish a friendly relationship and form different types of friends' circle. For instance, in the academic cooperation network, researchers jointly publish academic works to form academic circles in various fields of research [38,45,46]; while in protein networks, proteins frequently interact with each other to form structural units [6-9]. In recent years, the problem of community discovery has been widely concerned and deeply studied in many disciplines, such as computer science, sociology, and biology [5][10-16]. Research findings of community discovery have also been successfully applied to many fields, such as friend recommendation, personalized product promotion, protein function prediction, and public opinion analysis and processing [5,6].

In traditional community discovery methods, the network structure is treated as a static topology without considering the interaction between the information factors. Meanwhile, in microblogging networks applications, the interaction information among different users is very frequent; the topology only represents the possibility of interaction between users, and the degree of actual interaction is determined by the flow of information between the users. Thus, those community discovery methods based solely on topology have obvious limitations, as they neglect the information flow in the microblogging network that could be directly applied to the microblogging networks.

To address the limitations of traditional community discovery methods, an Efficient Evolutionary User Interest Community Discovery named EEUICD model is proposed in this paper based on a nature-inspired algorithm. Nature-inspired algorithm [40] is a computational technique used for solving optimization problems based on the functions, characteristics and mechanism of nature by studying the information processing mechanism contained therein. Nature-inspired algorithms include electromagnetism-like mechanism [41], genetic algorithm [42], artificial neural network [43] and so on. In our EEUICD model, a multi-objective genetic algorithm is used. The evolutionary community discovery is transformed into a multi-objective optimization problem, which not only improves the quality of user interest community discovery, but also minimizes the interests drift. Moreover, population initialization based on the label propagation algorithm improves the quality of initial users and posts. In addition, applying the label propagation algorithm to the mutation progress enhances the quality of clustering and the convergence rate. The convergence of the multi-objective genetic algorithm and the label propagation algorithm are scalable concurrently, due to the fact that the running time increases linearly with an increase in the number of users and posts. Finally, we

conduct experiments based on synthesized datasets and real-world datasets to validate the effectiveness of the proposed model.

The main contributions of this paper include the following:

(1) A preprocessing method based on HITS *(Hypertext Induced Topic Search)* is proposed to obtain high-quality users and posts and in order to optimize the sparse adjacency matrix at the same time. Setting step threshold S calculates the similarity between users that can reach each other in the S steps to ensure that the topological information of the microblogging network is enhanced without affecting the quality of community discovery. This method can fully describe the local information of each user and can improve the quality of user interest clustering.

(2) A label propagation algorithm-based mutation is presented for achieving quicker convergence of the algorithm while further improving the quality of user interest clustering. During the initialization of the algorithm, the microblogging network is divided into different community structures to generate users and interest with different community structures. Besides, the initial high-quality population is chosen, crossed and mutated, and finally the superior community is selected.

(3) The multi-objective genetic algorithm combined with label propagation algorithm enhances the scalability of the algorithm. When the initial users, posts and interests are generated, the label propagation algorithm can generate users with a certain community structure. At the same time, a greater hub value of user reflects a greater impact of the corresponding user on the surrounding community, which can be used to affect the label propagation process of these users. Therefore, in order to make the initial users to have good community structure, we choose the label update from the users with larger hub value.

The rest of the paper is organized as follows. In section II we discuss related works of dynamic community discovery in microblogging networks. In Section III, IV, V and VI, we introduce our models concerning dynamic community discovery. We discuss our experiments in Section VII and Section VIII presents the conclusion and future study.

II. Related Work

The discovery of user interest structure in dynamic microblogging networks has become an increasing hotspot, and it has a wide range of applications [5][17-19], for example, information influence mining, customer testimonials and more. Microblogging networks are very dynamic in such a way that the structure of a community will change as time evolves. The connections among the members of the community are relatively close, and the users in the community are relatively sparse. As time evolves, the changes in the community and its structure can be understood through the structure formed by most of the connections on these sub graphs. Besides, incremental community discovery [20] and evolutionary community discovery [21] are the two main methods of dynamic online communities.

A. Incremental community discovering

The basic idea of incremental community discovery algorithm is to cluster only the network at the first-time point. At a later time, according to the slow change characteristic of the network, the cluster results of the previous time point are used as the basis to form the network. At the same time, the cluster results are adjusted locally according to the features of the current network conditions in order to achieve a smooth cluster results. Meng et al. [22] proposed a community detection method in dynamic networks called iDBLINK which is evolved from an incremental density-based link clustering algorithm. It updates the local link community structure at a given current time by changing the similarity between the edges at the adjacent moments. Guo et al. [23] proposed an Incremental Dynamic Community detection method based on Improved Modularity (ICIM) which uses an improved modularity as the evaluation index of the community. The influence of the community structure on the attribution coefficient of vertex neighbors according to the historical topology and incremental changes is adjusted with the changes of users and edges in the local area.

Ozdikis et al. [24] proposed an incremental community detection method based on detecting similar terms in a temporal context. This is an unsupervised approach that uses symbiosis-based techniques to measure similarity online, further use them in the vector expansion.

To some extent, the methods of incremental community discovery sacrifice the quality of clustering for achieving less time overheads. The method of evolutionary community discovery incorporates the influence of the cluster quality. It ensures a reliable cluster quality whilst achieving cluster results closer to the real community structure.

B. Evolutionary community discovering

In order to mitigate the impact of noise data and to improve stability, some studies have applied the Markov model to evolutionary community discovery. Lin et al. [25] proposed FacetNet framework, which is the most classic algorithm of evolutionary community discovery. This framework uses a random block model to generate associations, and analyze the evolution of associations based on the probability model of Dirichlet distribution. They define the snapshot quality and historical overhead using the KL-divergence algorithm. This method integrates the community discovery and community evolution, during which the data of the time *t* and the historical community structure simultaneously affect the community structure at time *t*. Chen et al. [26] proposed a similarity measurement method based on social balance theory for signed networks. This method removes positive links with low similarity between communities and reconstructs positive neighbor sets. Users in the same community will evolve over time, and when the user's status is stable, the community will eventually be detected. Ma et al. [27] proposed a novel dynamic community detection algorithm based on two evolutionary non-negative matrix factorization (ENMF) frameworks. Two evolutionary non-negative matrix

factorization (ENMF) frameworks, and the equivalence between evolutionary spectral clustering and ENMF has been proved in the proposed algorithm. This algorithm can escape the local optimal solution without increasing the time complexity.

However, there are some problems evident in the number of communities and the trade-off between the two objectives in the above algorithm. To this end, Folino et al. proposed the DYNMOGA algorithm [28] to solve these community detection problems by considering multi-objective optimization problems. DYNMOGA has improved the algorithms using other classic evolutionary communities. In addition, DYNMOGA can automatically discover the number of communities by using genetic algorithms to select the best solution. However, due to its own scalability and longer execution time, DYNMOGA is not widely used in large-scale networks.

To this end, the proposed model EEUICD (Efficient Evolutionary User Interest Community Discovery) employs a genetic algorithm which is a nature-inspired algorithm to improve the quality of community discovery. In our EEUICD model, a multi-objective genetic algorithm is used. The evolutionary community discovery is transformed into a multi-objective optimization problem, which not only improves its own scalability and reduce the execution time of user interest community discovery [28], but also solves some problems regarding the number of communities and balancing the trade-off between the two objectives of the above algorithm [25-27].

III. PREPROCESSING METHOD

Given a microblogging network *G=(V,E)*, *V={v1,v2,...,Vn}* is a set of users, *E={e1,e2,...,Em}* is a set of edges, and *N(u)* is the neighbor user set of user *u*. Adjacency matrix $A = [a_{ij}]_{n\times n}$ denotes the connection relationship among users, the value of the corresponding element of the matrix indicates whether the edge exists: if there is an edge existing between v_i and v_j, then $a_{ij}=1$; if no edge exists between v_i and v_j, then $a_{ij}=0$.

Generally, the adjacency matrix *A* can be used as the similarity matrix of the microblogging network to describe the similarity between users. However, in addition to the similarity between the users, which are directly connected in the network, there are different degrees of similarity between the users which are not directly connected. For example, there is a certain similarity between two users that can reach one another after finite number of steps. The adjacency matrix is used as the similarity matrix of the network which can simply represent the similarity in the relationships between the directly connected users, but cannot express the similarity in the relationships between users not directly connected. Therefore, the adjacency matrix loses the similarity relationship information between many users and cannot reflect a complete local information of each user. Adjacency matrix contains limited information which affects the accuracy of community discovery.

Therefore, in order to describe the local information of each user more adequately, a method based on step number is proposed in this section. According to the adjacency matrix *A* in the network, the similarity relationship between users is calculated, and a new similarity matrix is obtained. The definitions of s-step and similarity matrix are detailed as follows.

Definition 1 (s-step). Given a social network *G=(V,E)*, $u \in V$, For any user $u \in V$ in the point set, if user *u* can arrive at user *v* at least after *s* step, that is, the length of the shortest path from user *u* to user *v* is *s*. In other words, called user *u* can arrive at user *v* through *s* step.

As shown in Figure 1, in a network diagram, user v_1 can reach user v_4 in two steps, user v_5 in 3 steps, and user v_6 in 2 steps.

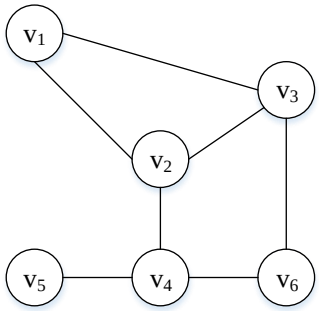

Figure 1. A sample network

Definition 2 (User similarity). Given a social network *G=(V,E)*, $u, v \in V$, the similarity *Sim(u,v)* between *u* and *v* is defined as:

$$Sim(u,v) = e^{\sigma(1-s)} \quad (1)$$

where, user *u* arrives at user *v* after *s* step, $s \geq 1$, σ is a attenuation factor. $\sigma \in (0,1)$. As the number of steps s increases, the similarity decreases. σ controls the degree of similarity attenuation. Larger value of σ reflects a rapidly degrading similarity relationship between users.

Definition 3 (similarity matrix X). Given a social network *G=(V,E)*, $X = [x_{ij}]_{n\times n}$ is a matrix corresponding to G. Using formula (1), the similarity $x_{ij}=Sim(v_i,v_j), v_i, v_j \in V$ between two users v_i and v_j in *X* is calculated, where X is the similarity matrix of *G*.

The step number and the attenuation factor are used to calculate the similarity relationship between two users which are not directly connected, in order to better reflect the community topology structure, and to improve the accuracy of community detection. However, when the number of steps is higher than a certain threshold, two users not belonging to the same community will also obtain a certain similarity value, which makes the boundary of community structure more obscure. Therefore, setting step threshold *S*, only calculates the similarity between users those can reach each other in the *S* steps, so as to ensure that the topological information of the microblogging social network is enhanced without affecting the division of community boundaries. In the experimental section, the step number threshold *S* and the attenuation factor σ are

analyzed, and the influence of different step threshold S and attenuation factor σ on the result is studied.

IV. Extracting High-quality Posts and Influential Users with the HITS Algorithm

In this paper, the HITS algorithm [37] is extended to exploit the inseparable connection between the interests and their corresponding users for distilling the high-quality users and the popular interests. The proposed Filtering User-Interest method can effectively filter random low-quality users and ordinary interests, as shown in Figure 2.

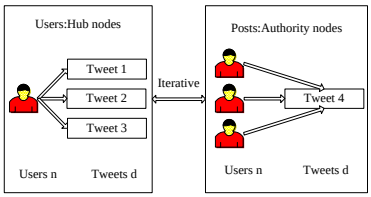

Figure 2. Iterative Model for Determining the Authority Score of Posts and the Hub Score of Users.

Many hot interests generally attract high-quality users. Intuitively, hot interests can be recommended or commented by a greater number of high-quality users as compared to that of the ordinary interests. In addition, high-quality users can draw an increased level of attention to the hot interests, which are usually spread or broadcast over the microblogging network. The authority value of the HITS algorithm has been completely utilized with more importance, to identify the high-quality users, alongside the hub value of the interests. Furthermore, a special emphasis has been given to the theory that there is a strong possibility of hot interests attracting many high-quality users.

V. Clustering Algorithm Framework Based on Multi Objective Optimization

The proposed EEUICD model is based on genetic algorithms [35,36] optimal users are posts are exploited to solve multi-objective optimization problem, and each user-post pair can aid a potential solution. Therefore, the algorithm needs to implement an image from phenotype to genotype at the initial stage of coding. Individuals evolve continuously through successive generations. Offspring are usually produced in this way: the two parents are crossed to inherit the genetic structure of the father, and then mutate the parent gene to produce a good result. In each generation, the calculation of the fitness of users should be carried out, i.e. the calculation of the objective function, and the users with higher fitness will be chosen to the next generation iteration. After several generations, the new generation tends to satisfy a given condition, i.e. the final pairs of user and post is considered as the optimal or near-optimal solution for all the objective functions, as shown Figure 3.

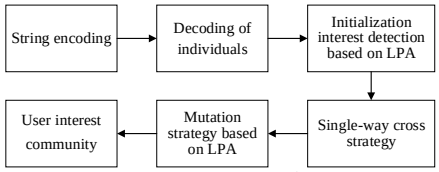

Figure 3. EEUICD model flow chart

A. *Encoding and decoding of individuals*

Existing community structure discovery algorithms are mainly based on string encoding and graph encoding. In comparison with graph encoding, string encoding can represent the community structure more intuitively and effectively. An arbitrary division of a network represents an individual, containing N posts *pe1*, *pe2,…, pen*, *n* represents the number of users. Each post corresponds to a value of *j*. These posts constitute the network, and each value *j* corresponds to the i^{th} post *pei*, and *j* denotes the community of the i^{th} post *pei*, which means that the posts with the same label belong to the same user interest community. In string representation, the community of the network user is only an identifier.

B. *Initialization interest detection algorithm based on label propagation*

When the initial pairs of user and post are generated, the efficiency of the algorithm can be improved by increasing the community structure of the initial high-quality users and posts and the diversity of the initial population. In order to achieve this goal, this paper uses the idea of label propagation algorithm to generate the initial population.

The label propagation algorithm for microblogging network based on semi-supervised learning (LPA) is proposed by Zhu et al [29], the basic method adopted by the LPA algorithm uses the users' interest label information to predict the label information of other unlabeled users. The label propagation algorithm are classified based on transfers between user labels, it is not limited to the shape of the distribution of the data, as long as the data is the same type of spatial distribution, the algorithm can divide them into the same class, to achieve low time complexity, clustering effect, good extensibility. Raghava et al [30] proposes the application of LPA to community discovery for quicker processing. This algorithm is abbreviated as RAK algorithm. In the RAK algorithm, first of all, each user is assigned a unique label, and each user has several neighbors; in each iteration, each user, according to their neighbor users' label, are constantly updated for most of their neighbor user's label. The algorithm terminates until the user labels are no longer changed. Finally, the corresponding community is divided according to the label of each user. The main steps of the RAK algorithm are described as follows:

(1) For network $G=(V,E)$, $\forall x \in V$, the algorithm initially gives any user x a unique label value Lx, and Lx denotes the community number of the user x.

(2) According to the interest label of user x's neighbor set $N(x)$, the user x iteratively updates its label value Lx, which is the label value of most neighbors. In the iterative update process, if there is more than one optional label, the

label of one of the neighbors is randomly updated to the new label value of the user *x*. After K iterations, the label change of each user tends to be stable.

(3) $\forall x, y \in V$, if two users share the same label value, i.e. *Lx=Ly*, then the user *x* and the user *y* belong to the same community and generate the community division.

The time complexity of the RAK algorithm is *O(km)*, *k* represents the number of iterations of the algorithm, and *m* represents the number of edges of the network. Because the time complexity of the RAK algorithm is almost linear, it exhibits very good efficiency in dealing with large-scale data. However, it can be seen from the steps described above that the RAK algorithm depends on some random factors, such a way that during the user label update process; if there are more than one optional labels, a label is selected randomly for update. In these optional labels, different labels will lead to a certain difference in the structure of the community result. Therefore, these random factors of the RAK algorithm will make the algorithm structure unstable. When dealing with large-scale data, such a situation will be more obvious and frequent.

Because of the good cluster quality and near linear time of label propagation algorithm, this paper uses the idea of the RAK label propagation algorithm as the baseline. When the initial individual is generated, the label propagation algorithm can generate users with a certain community structure. At the same time, a greater degree of user implies a greater the impact of the corresponding user on the surrounding community, as it can be used as the neighbor of more users to affect the label propagation process of these users. Therefore, in order to make the initial users to have good community structure, we choose the label update from the users having a larger connection degree. The following experimental results show that the label update from the users with larger degree will improve the quality of the clustering algorithm.

In the initialization process, each user in the microblogging network is assigned with a unique label, which represents the user belongs to the community number. In order to enhance the stability of the label propagation algorithm, the order of label update starts from the user with the largest degree, uses asynchronous update, and only updates the label once. The initialization algorithm based on label propagation is shown in algorithm 1.

Algorithm 1: The initialization algorithm based on label propagation

Input: population number p, neighbor set Nt and degree Dt of microblogging social network Gt

output: initial solution

method:

(1) **For *i=1* to *p***;

(2) $g^i = [g_1^i, g_2^i, ..., g_n^i], i \in \{1,2,..., p\}, g_j^i = j, j \in \{1,2,..., n\}$, n is the number of users;

(3) Arbitrary individual $\forall g^i \in g, i \in \{1,2,..., \mathrm{p}\}$ randomly generated sequence $X = [\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_n], \mathrm{x}_i \in N_t(\mathrm{i})$;

(4) Sort(D_t), start from a large degree user, update with an asynchronous update policy, and update only once;

(5) Asynchronous update:
$g^i(t) = f(g_1^i(t), ..., g_h^i(t), g_{h+1}^i(t-1), ..., g_l^i(t-1)), t \in \{1,2,..., k\}$
k is the number of iterations;

(6) **End For**

(7) **Return** *p* initial populations, the initial population at time *t* is defined as $g_t = \{\mathrm{g}_t^1, \mathrm{g}_t^2, ..., \mathrm{g}_t^p\}$;

The process of updating users includes a synchronous update mode and an asynchronous update mode. The experimental results given in [30] showed that the asynchronous update method is more stable than the synchronous update method, but need more update time. This article uses the asynchronous update strategy, in which the label of user *x* during *t* is updated according to the user previously updated in the t^{th} update and also the user previously not updated in the t^{th} update.

C. Single crossing strategy

Because the decoding process of the above algorithm solves the situation of incompatible labels among different users, this paper uses the single-way cross strategy proposed by Tasgin et al [31] in order to further maintain a good community structure. The cross strategy is shown in Table 1, where *s* is the number of the user. Table 1 shows a network with 6 users. Assuming that user *2* is randomly selected, the community users *{1,2,6}* where user 2 is located in individual *A* will propagate to individual *B {1,2,6}*, That is, updating the label of the user *{1,2,6}* in individual *B* to the label of user 2 in individual *A*, thus obtaining the new individual C after crossing. This crossover operation propagates the community structure information in *A* to B. In this algorithm, each cross operation is performed twice, one is spreading from *A* to *B*, and the other is spreading from *B* to *A*.

Table 1. One-Way crossover

S(user)		A(source)		B(target)		C(new)
1		1	→	1	→	1
2	→	1	→	2	→	1
3		2		2		2
4		3		1		1
5		2		2		2
6		1	→	1	→	1

D. Mutation strategy based on label propagation algorithm

The improvement of the mutation algorithm is a novel point of this paper. This paper does not take the traditional classical mutation algorithm, but transforms the mutation process into an intermediate process of label propagation, which further improves the user interest clustering quality and accelerates the convergence rate of the algorithm. In mutation operation, label propagation is updated only once. This method differs from the initialization based on label propagation in two perspectives. Firstly, the updated user interest label is the one after the crossover algorithm; secondly, the label update is started from the most influential users in the direction of propagation. In this way, the mutation process implements a nested genetic algorithm and the label propagation algorithm. The mutation strategy based on the label propagation algorithm is shown in algorithm 2.

Algorithm 2: A mutation algorithm based on label propagation

Input: selecting users $g=\{g^1,g^2,...,g^M\}$, population number M, neighbor set N_t of microblogging social network G_t

output: crossing users $g=\{g^{1'},g^{2'},...,g^{M'}\}$

method:

(1) **For *i=1* to *M***;

(2) Arbitrary individual $\forall g^i \in g, i \in \{1,2,...,M\}$ randomly generated quence $X=[x_1,x_2,...,x_n], x_i \in N_t(i)$;

(3) Start from the user with user number *1*, update with an asynchronous update policy, and update only once.

(4) Asynchronous update: $g^i(t)=f(g_1^i(t),...,g_h^i(t),g_{h+1}^i(t-1),...,g_l^i(t-1)), t\in\{1,2,...,k\}$,

k is the number of iterations;

(5) **End For**

(6) **Return** *M* crossed users $g=\{g^{1'},g^{2'},...,g^{M'}\}$;

VI. EEUICD MODEL FLOW

A. Objective function

As proposed by Chakrabarti et al. [21], the evolutionary community found a framework that defines a slowly changing dynamic network, where the cost of snapshot (SC) is used to measure the quality of the community structure and the time cost (TC) is used to measure the similarity between clusters of two consecutive time steps. The framework is used to measure cluster quality, as shown in the table below, to control the trade-off between the two objectives:

$$cost = \alpha \cdot SC + (1 - \alpha) \cdot TC \quad (2)$$

In this formula, α is the balance factor, and its value is defined by the user. When $\alpha=1$, the result only considers the clustering quality. When $\alpha=0$, the result of clustering is achieved closer to the previous iteration. When the value of α is between *0* and *1*, two objectives can be controlled in order to reach the optimal equilibrium point in order to find the optimal clustering result.

As shown in formula (2), this algorithm focuses on optimizing snapshot quality *SC* and historical cost *TC* to achieve the ultimate goal of cost optimization. Because snapshot quality *SC* measures the quality of community structures at *t* moments, an objective function is needed to maximize the number of edges in each community and to minimize the number of edges between communities. Therefore, this paper uses the standard module degree *Q* which is widely used in the community structure discovery domain.

A Microblogging Network $G_t=(V_t,E_t)$ has *n* users and *m* edges on time *t*, and its community structure is denoted as $C=\{C_1,C_2,...,C_k\}$, *k* is the number of communities. l_s represents the number of edges between all the users in the community C_s, and d_s represents the sum of the degree of the users in the community.

The module degree *Q* is defined as follows:

$$Q=\sum_{s=1}^{k}\left[\frac{l_s}{m}-\left(\frac{d_s}{2m}\right)^2\right] \quad (3)$$

In the module of the module degree Q, the first component shows the probability of the edge in a interest community, and the second component shows that if the edge is randomly assigned and the interest community structure is not considered. The probability value of the edge is in the entire network.

The second objective function must minimize the historical cost *TC*. In this paper, Normalized Mutual Information (NMI) [32] is used to measure the similarity between the user interest community structure at time *t* and the social structure at the previous moment by using a matrix. Suppose a network is divided into two partitions $A=\{A_1,A_2,...,A_a\}$ and $B=\{B_1,B_2,...,B_b\}$, *C* is a matrix, where the element C_{ij} is the number of users in the interest community $A_i \in A$ and the interest community $B_i \in B$ at the same time. The definition of *NMI* is as follows:

$$NMI(A,B)=\frac{-2\sum_{t=1}^{C_A}\sum_{j=1}^{C_B}C_{ij}\log\left(C_{ij}N/C_{i.}C_{.j}\right)}{\sum_{t=1}^{C_A}C_{i.}\log\left(C_{i.}/N\right)+\sum_{t=1}^{C_B}C_{.j}\log\left(C_{.j}/N\right)} \quad (4)$$

where, C_A represents the number of communities in *A*, and C_B represents the number of communities in *B*, C_i represents the sum of rows in matrix *C*, $C_{.j}$ represents the sum of columns in matrix *C*, and *N* is the number of users. If *A=B*, then *NMI (A,B)=1*. If *A* and *B* are completely different, then *NMI (A,B)=0*. So, the second goal of this paper is to maximize $NMI(C_t,C_{t-1})$ at time *t*.

In fact, after equation (4), two sets of communities *A* and *B* are given, the former identifies the community of interest extracted by the algorithm (with size *a*) and the latter represents the ground-based community set (with size *b*) in order to calculate the *NMI*. It is necessary to determine the best community match by cost *O(ab)*. Assuming $a \square b$, the *NMI* calculations require $O(a^2)$ comparisons, making them unsuitable for large-scale networks. In order to reduce

the computational complexity and speed up the evaluation process, the method proposed in [33] is adopted: given an algorithm community $x \in A$, (1) its users with their corresponding ground truth community $y \in B$ are labeled, then (2) community x with the ground truth community with the highest number of labels in community x is matched. Two measures are defined:

Community Precision represents the percentage of users in algorithm community x labeled with ground truth community y, computed as:

$$Precision = \frac{|x \cap y|}{|x|} \tag{5}$$

Community Recall represents the percentage of users in the ground truth community y covered by the algorithm community x, computed as:

$$Recall = \frac{|x \cap y|}{|y|} \tag{6}$$

The two measures describe the overlap between algorithm community x and ground truth community y for each pair (x,y): A perfect match is obtained when both *precision* and *recall* are 1.We also define the quality score of the algorithm community set by calculating *precision* and *recall* of all the communities in the set and then calculate their average *F-measure* that is the harmonic mean of *precision* and *recall*:

$$F\text{ - }measure = 2\frac{precision * recall}{precision + recall} \tag{7}$$

To reduce the time complexity of the EEUICD model, a novel community evaluation technique able to cope with the computational issues that arise when calculating *NMI* on large community sets is adopted. F-measure is used instead of NMI as the objective function.

B. EEUICD model flow

Given a dynamic microblogging network sequence $G=\{G1,G2,...,GT\}$. EEUICD model firstly finds the division of network G_1 and performs single-objective optimization algorithm on the microblogging network. The roulette betting algorithm is used to calculate and optimize the first objective function value, that is, the value of Q, and then the label-based genetic algorithm is used to obtain the online community division at time $T=1$. When the time $T>1$, the multi-objective optimization genetic algorithm first uses a label initialization algorithm to generate a group of users, calculates two objective function values, and performs non-dominated sorting to rank each user. Using elitist retention strategy, users with low ranks are selected, new users are generated by cross-mutation, the offspring and the parent are mixed and sorted non-dominated, and the better users are selected to enter the next iteration. The algorithm terminates after a fixed number of iterations; meanwhile, the algorithm returns a set of solutions, all of which are Pareto front. Each solution corresponds to a different equilibrium point between two objective functions, and the division of each network contains a different number of communities. These solutions are satisfying the snapshot quality and historical overhead in the non-dominated solutions. In these solutions, the best solution of the interest community structure is selected, that is, the maximum partition of the module value is chosen as the result returned at the final time t. The EEUICD model is shown in Algorithm 3.

Algorithm 3: EEUICD (G,T)

Input: Microblogging social network sequence,$G=\{G_1,G_2,...,G_T\}$, Time point T
output: Community partition on each network $C_t=\{C_{t1},C_{t2},...,C_{tk}\}$

method:

(1) Based on the label initialization algorithm, p initial solutions are obtained $g_t = \{g_t^1, g_t^2, ..., g_t^p\}$;
(2) $\forall g^i \in g, i \in \{1,2,..., p\}$ the interest community is obtained by decoding $Ct=\{C_{t1},C_{t2},...,C_{tk}\}$, k is the number of communities;
(3) Calculate two objective functions Q, *F-measure*;
(4) When $t=1$, roulette betting algorithm is selected, only first objective functions are optimized;
(5) **For *t=2* to *T***;
(6) **While** termination conditions are not satisfied **do**
(7) Performs non-dominated sorting to rank each individual;
(8) Choose the best individual to generate the offspring;
(9) The offspring are evolved with single cross-operation and label-based mutation operation;
(10) The offspring and the parent are sorted non-dominated and ranked for each individual;
(11) Elite reserve, select low level users into the next generation;
(12) **End While**
(13) **Return** the individual with the largest Q value as the solution $Ct=\{C_{t1},C_{t2},...,C_{tk}\}$;
(14) **End For**

VII. EXPERIMENTS

A. Experiment Settings and Dataset

The experiments are conducted on a machine with Intel I7 4.2 GHz CPU and 16G memory. Our dataset is collected from Twitter (http://twitter.com/) [34] via Twitter API. The collected dataset is composed of 1,000,000 posts from July 25, 2018 to July 28, 2018.

B. Comparative Methods

The algorithm proposed in this paper is compared with the existing typical algorithms, detailed as follows:

(1) iDBLINK algorithm [22]: It can update the local link community structure in the current moment through the change of similarity between the edges at the adjacent moments, which includes the creation, growth, merging, deletion, contraction, and division of link communities.

(2) FacetNet algorithm [25]: It uses a random block model to generate associations, and analyze the evolution of associations based on the probability model of Dirichlet

distribution. It uses the KL-divergence algorithm to define the snapshot quality and historical overhead. This model integrates community discovery and community evolution; the data of the time *t* and the historical community structure simultaneously affect the interest community structure at time t,

(3) DYNMOGA algorithm [28]: It improved algorithms from other classic evolutionary communities, using non-dominated sorting to balance snapshot quality and time costs. In addition, it is able to automatically find the number of communities by using genetic algorithms to select the best solution.

C. Parameter Experiment

The effect of the step number threshold *S* and attenuation factor σ are evaluated in this section. 1000 users' data in the database are randomly selected for experiments, and the *F-measure* score mentioned above is a measure of the index.

In the experiment, the value of one parameter is fixed, and the influence of the change of the other parameter value on the *F-measure* is analyzed to determine the final value of the parameters.

(1) Step number threshold *S*

In view of the dataset, the attenuation factor $\sigma=0.5$ is set up, and the effect of the step number threshold *S* on the *F-measure* is analyzed.

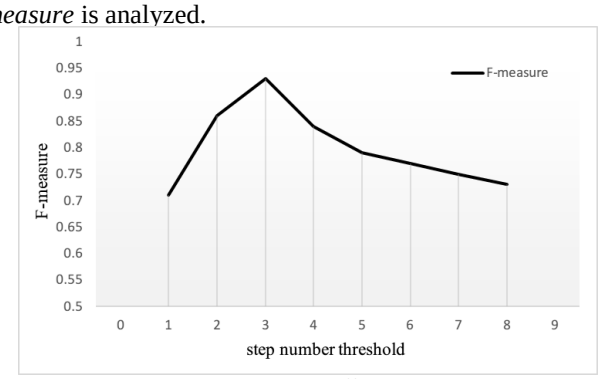

Figure 4. *F-measure* score under the different step number threshold S

As shown in Figure 4, with the increase of the step number threshold *S*, the trend of *F-measure* increases first and then decreases. The experimental results show that considering the similarity of user pairs which are not directly connected but reachable within a certain number of steps, the local structure information of each user can be effectively reflected. However, if the threshold is too large, the distance between the users in the same community will also increase a certain similarity value, which is not conducive to the identification of the interest community boundaries, and the accuracy of the interest community will be reduced. For small datasets, a small step number threshold 3 is selected, and for big datasets, we select a slightly larger step threshold of 8 to achieve the optimal result. The threshold selection in this paper is 3.

(2) Attenuation factor σ

In view of the dataset, the step number threshold $S=0.5$ is set up, and the effect of the attenuation factor σ on the *F-measure* is analyzed.

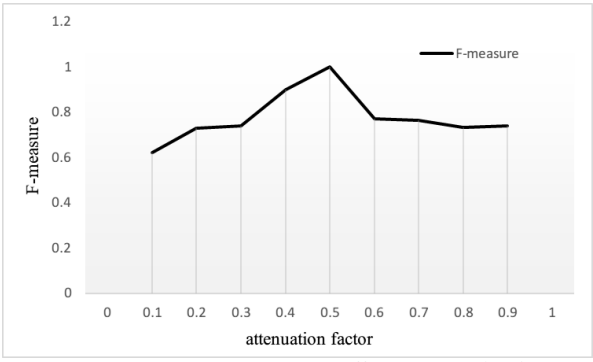

Figure 5. *F-measure* value under the different attenuation factor σ

As shown in Figure 5, with an increase in the attenuation factor, the trend of *F-measure* overall increases first and then decreases. This is due to the fact that the attenuation factor controls the attenuation degree of similarity with an increase in the hop counts. For small datasets, a slight attenuation factor $\sigma=0.5$ is selected to avoid the vagueness of community boundary when the attenuation factor is too large. For a large dataset, a small attenuation factor $\sigma=0.1$ is selected to enhance the quality of local feature of the user to achieve the optimal result.

D. Result Analysis

Our dataset is collected from Twitter. These records make up a microblogging network, where users represent each user, indicating the following: forwarding, replying, and other connections between users. We compared EEUICD model with the chosen models based on a large-scale real-world network, to evaluate their *Precision*, *Recall*, *F-measure*.

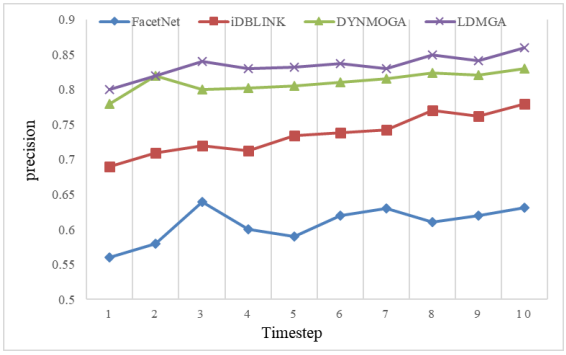

Figure 6. Precision rate comparison

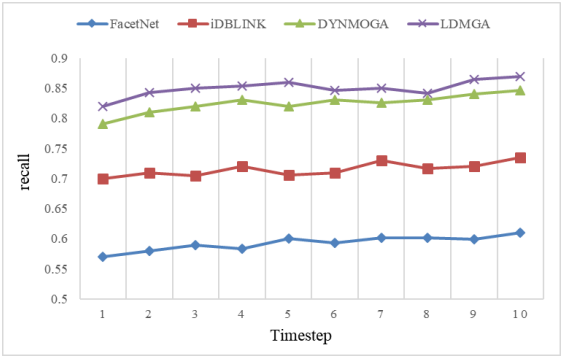

Figure 7. Recall rate comparison

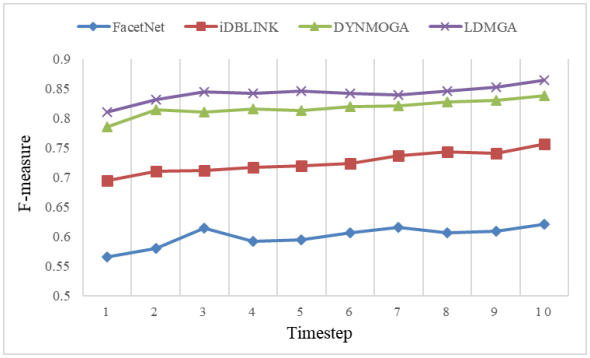

Figure 8. F-measure comparison

As we can see from Figures 6, 7 and 8, the FacetNet algorithm is far worse than the other three algorithms in terms of precision. This is due to the fact that the FacetNet algorithm uses a random block model to generate associations, and analyses the evolution of associations based on the probability model of the Dirichlet distribution, which leads to very low accuracy. Thus, this comparison result shows that the FacetNet algorithm is not suitable for the user interest community discovery in a dynamic microblogging network. The iDBLINK algorithm can update the local link community structure in the current moment through the change in the similarity between the edges at the adjacent moments. Hence the iDBLINK algorithm overcomes the drawbacks of the FacetNet algorithm, but it still has the problem of having no initialization method and no optimization algorithm being used for user interest community discovery. Besides, the DYNMOGA algorithm utilizes the optimization algorithm of genetic algorithms for community detection to obtain a better result than iDBLINK algorithm. Finally, the interest community obtained by the EEUICD model is more accurate than the other typical algorithms, which not only improves its own scalability and reduces the execution time of user interest community discovery, but also solves some problems around the number of communities and the trade-off between the two objectives of the above algorithm. This is due to the use of a HITS based pre-processing method to effectively improve the quality of local information of the users and posts, and the use of a mutation algorithm based on label propagation to enhance the clustering effect and convergence speed of the whole process of user interest community detection.

E. Case Study

In order to verify the advantages of the proposed algorithm in terms of its accuracy of user interest community, the results of the interest community evaluation index *F-measure* obtained for the EEUICD model and the existing typical algorithms are shown in Table 2.

There are six interested communities obtained by the algorithm proposed in this paper and the other three existing typical algorithms. The F-measure scores produced by each algorithm are reported. As shown in Table 2, the FacetNet algorithm characterizes the worst results, because it is not suitable for the user interest community discovery in dynamic microblogging networks. The iDBLINK algorithm overcomes the drawbacks of the FacetNet algorithm, but it still has the problems of having no initialization method and no optimization algorithm being used. Therefore, its results are not satisfactory. Besides, the DYNMOGA algorithm utilizes the optimization algorithm of genetic algorithm for community detection to obtain better result than the iDBLINK algorithm. The EEUICD algorithm not only improves its own scalability and reduces the execution time of user interest community discovering, but also solves some problems around the number of communities and the trade-off between the two objectives of the above algorithm. Therefore, it outperforms the other compared models. The results further illustrate that the EEUICD model proposed in this paper exhibits the best performance for user interest community discovery.

Table 2. Analysis of community detection results

Algorithm	Community					
	Sports	Economy	Diet	Tourism	Music	Technology
FacetNet	0.61	0.60	0.59	0.62	0.61	0.61
iDBLINK	0.72	0.72	0.73	0.74	0.73	0.71
DYNMOGA	0.82	0.82	0.82	0.81	0.83	0.82
EEUICD	0.83	0.84	0.82	0.85	0.84	0.84

VIII. CONCLUSION

To deal with the problems that traditional community discovery methods face in resolving the interest community detection of dynamic networks (such as microblogging networks), a new multi-objective approach based on the label propagation algorithm, named EEUICD model is proposed in this paper. Employing the idea of multi-objective genetic algorithm, the evolutionary community discovery algorithm is transformed into a multi-objective optimization problem, which not only improves the user interest clustering quality, but also minimizes the clustering drift from one-time step to the successive one. A pre-processing method based on the HITS algorithm improves the cluster quality of initial influential users and posts, and optimizes the sparse adjacency matrix. In addition, by applying the label propagation algorithm to the mutation progress the proposed model enhances the quality of clustering and increases the convergence rate. At the same time, the combination of the multi-objective genetic algorithm and the label propagation algorithm makes the algorithm more scalable. In comparison with other community detection algorithms on real networks, EEUICD

shows a higher correspondence with the ground truth communities.

ACKNOWLEDGEMENT

This work was partially supported by the National Natural Science Foundation of China under Grants No. 61502209, 61502207 and 71701082, Natural Science Foundation of Jiangsu Province under Grant BK20170069, UK-Jiangsu 20-20 World Class University Initiative programme, UK-China Knowledge Economy Education Partnership and Postgraduate Research & Practice Innovation Program of Jiangsu Province No. KYCX17_1808.

REFERENCES

[1] Q. Gui, R. Deng, P. Xue, and X. Cheng, "A community discovery algorithm based on boundary nodes and label propagation," Pattern Recognition Letters, 2017.

[2] M. Sattari and K. Zamanifar, "A spreading activation-based label propagation algorithm for overlapping community detection in dynamic social networks," Data & Knowledge Engineering, 2018.

[3] G. Rossetti, D. Pedreschi, and F. Giannotti, "Node-centric Community Discovery: From static to dynamic social network analysis," Online Social Networks and Media, vol. 3, pp. 32-48, 2017..

[4] K. Berahmand and A. Bouyer, "LP-LPA: A link influence-based label propagation algorithm for discovering community structures in networks," International Journal of Modern Physics B, vol. 32, no. 06, p. 1850062, 2018.

[5] Fortunato and Santo, "Community detection in graphs," Physics Reports, vol. 486, no. 3, pp. 75-174, 2009.

[6] F. Huang, S. Zhang, and X. Zhu, "Discovering network community based on multi-objective optimization," Journal of Software, vol. 24, no. 9, pp. 2062-2077, 2013.

[7] G. Li, K. Guo, Y. Chen, L. Wu, and D. Zhu, "A dynamic community detection algorithm based on Parallel Incremental Related Vertices," in Big Data Analysis (ICBDA), 2017 IEEE 2nd International Conference on, 2017, pp. 779-783: IEEE.

[8] K. Liu, Y. Zhang, K. Lu, X. Wang, and X. Wang, "Reinforcement Label Propagation Algorithm Based on History Record," in International Conference on Neural Information Processing, 2017, pp. 348-356: Springer.

[9] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," Physical Review E Statistical Nonlinear & Soft Matter Physics, vol. 78, no. 4 Pt 2, p. 046110, 2008.

[10] W. Liu, T. Suzumura, L. Chen, and G. Hu, "A generalized incremental bottom-up community detection framework for highly dynamic graphs," in Big Data (Big Data), 2017 IEEE International Conference on, 2017, pp. 3342-3351: IEEE.

[11] H. Sun, J. Liu, J. Huang, G. Wang, X. Jia, and Q. Song, "LinkLPA: A Link ' Based Label Propagation Algorithm for Overlapping Community Detection in Networks," Computational Intelligence, vol. 33, no. 2, pp. 308-331, 2017.

[12] X. Chen, H. Sun, H. Du, J. Huang, and K. Liu, "A Centrality-Based Local-First Approach for Analyzing Overlapping Communities in Dynamic Networks," in Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2017, pp. 508-520: Springer.

[13] H. S. Cheraghchi and A. Zakerolhosseini, "Mining Dynamic Communities based on a Novel Link-Clustering Algorithm," International Journal of Information & Communication Technology Research, vol. 9, no. 1, pp. 45-51, 2017.

[14] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," Physical review E, vol. 69, no. 2, p. 026113, 2004.

[15] M. E. Newman, "The structure and function of complex networks," SIAM review, vol. 45, no. 2, pp. 167-256, 2003.

[16] Z. Xiaoping, L. Xun, and Z. Haiyan, "User community detection on micro-blog using RC model," Journal of Software, vol. 25, no. 12, pp. 2808-2823, 2014.

[17] M. Coscia, F. Giannotti, and D. Pedreschi, "A classification for community discovery methods in complex networks," Statistical Analysis and Data Mining: The ASA Data Science Journal, vol. 4, no. 5, pp. 512-546, 2011.

[18] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 44-54: ACM.

[19] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe, "A framework for community identification in dynamic social networks," in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007, pp. 717-726: ACM.

[20] B. Shan, S.-X. Jiang, S. Zhang, H. Gao, and J. Li, "IC: Incremental algorithm for community identification in dynamic social networks," J. Softw., vol. 20, pp. 184-192, Dec. 2009.

[21] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker Graphs: An Approach to Modeling Networks," Journal of Machine Learning Research, vol. 11, no. 3, pp. 985-1042, 2010.

[22] F. Meng, F. Zhang, M. Zhu, Y. Xing, Z. Wang, and J. Shi, "Incremental Density-Based Link Clustering Algorithm for Community Detection in Dynamic Networks," Mathematical Problems in Engineering,2016,(2016-1-12), vol. 2016, no. 6, pp. 1-11, 2016.

[23] K. Guo, T. Zhu, and G. H. Li, "Incremental dynamic Community discovery algorithm based on Improved Modularity," in Computer and Communications (ICCC), 2016 2nd IEEE International Conference on, 2016, pp. 2536-2541: IEEE.

[24] O. Ozdikis, P. Karagoz, and H. Oğuztüzün, "Incremental clustering with vector expansion for online event detection in microblogs," Social Network Analysis & Mining, vol. 7, no. 1, p. 56, 2017.

[25] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Facetnet: a framework for analyzing communities and their evolutions in dynamic networks," in Proceedings of the 17th international conference on World Wide Web, 2008, pp. 685-694: ACM.

[26] J. Chen, H. Wang, L. Wang, and W. Liu, "A dynamic evolutionary clustering perspective: Community detection in signed networks by reconstructing neighbor sets," Physica A Statistical Mechanics & Its Applications, vol. 447, pp. 482-492, 2016.

[27] X. Ma and D. Dong, Evolutionary Nonnegative Matrix Factorization Algorithms for Community Detection in Dynamic Networks. IEEE Educational Activities Department, 2017, pp. 1045-1058.

[28] F. Folino and C. Pizzuti, "An Evolutionary Multiobjective Approach for Community Discovery in Dynamic Networks," IEEE Transactions on Knowledge & Data Engineering, vol. 26, no. 8, pp. 1838-1852, 2014.

[29] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Technical Report CMU-CALD-02-107, Carnegie Mellon University, pp. 02-107, 2002.

[30] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," Physical review E, vol. 76, no. 3, p. 036106, 2007.

[31] M. Tasgin, A. Herdagdelen, and H. Bingol "Community detection in complex networks using genetic algorithms," in Proc. Eur. Conf. Complex Syst., Apr. 2006.

[32] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," Journal of Statistical Mechanics: Theory and Experiment, vol. 2005, no. 09, p. P09008, 2005.

[33] G. Rossetti, L. Pappalardo, and S. Rinzivillo, "A novel approach to evaluate community detection algorithms on ground truth," in Complex Networks VII: Springer, 2016, pp. 133-144.

[34] Twitter, REST API v1.1 Resources, 2018, Available at: https://dev.twitter.com/.

[35] X. Niu, W. Si, and C. Q. Wu, "A Label-based Evolutionary Computing Approach to Dynamic Community Detection," Computer Communications, vol. 108, 2017.

[36] X. Niu, W. Si, and K. She, " Dynamic network community discovery based on evolutionary clustering," Journal of software, vol. 28, no. 7, pp. 1773-1789, 2017.

[37] L. L. Shi, L. Liu, Y. Wu, L. Jiang, and J. Hardy, "Event Detection and User Interest Discovering in Social Media Data Streams," IEEE Access, vol. 5, no. 99, pp. 20953-20964, 2017.

[38] L. Shi, Y. Wu, L. Liu, X. Sun, and L. Jiang, "Event detection and identification of influential spreaders in social media data streams," Big Data Mining & Analytics, vol. 1, no. 1, pp. 34-46, 2018.

[39] M. Conti, A. Passarella, and S. K. Das, "The Internet of People (IoP): A new wave in pervasive mobile computing," Pervasive & Mobile Computing, vol. 41, 2017.

[40] Mart, V. Iacute, and L. M. Robledo, "Multi-Verse Optimizer: a nature-inspired algorithm for global optimization," Neural Computing & Applications, vol. 27, no. 2, pp. 495-513, 2016.

[41] E. H. Bouchekara, M. A. Abido, and A. E. Chaib, "Optimal Power Flow Using an Improved Electromagnetism-like Mechanism Method," Electric Machines & Power Systems, vol. 44, no. 4, pp. 434-449, 2016.

[42] K. Zhang, H. Du, and M. W. Feldman, "Maximizing influence in a social network: Improved results using a genetic algorithm," Physica A Statistical Mechanics & Its Applications, vol. 478, 2017.

[43] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta, "Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science," Nature Communications, vol. 9, no. 1, p. 2383, 2018.

[44] D. Miao, L. Liu, R. Xu, J. Panneerselvam, Y. Wu, and W. J. I. T. o. I. I. Xu, "An efficient indexing model for the fog layer of industrial internet of things," 2018.

[45] Y. Guo, L. Liu, Y. Wu, and J. J. A. T. o. I. T. Hardy, "Interest-aware content discovery in peer-to-peer social networks," vol. 18, no. 3, p. 39, 2018.

[46] B. Yuan, L. Liu, and N. J. F. G. C. S. Antonopoulos, "Efficient service discovery in decentralized online social networks," vol. 86, pp. 775-791, 2018.

LIANG JIANG received the B.S. degree from the Nanjing University of Posts and Telecommunications, China, in 2007, and the M.S. degree from Jiangsu University, Zhenjiang, China, in2011, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Telecommunication Engineering. His research interests include OSNs, computer networks, and network security.

LEILEI SHI received the B.S. degree from Nantong University, Nantong, China, in 2012, and the M.S. degree from Jiangsu University, Zhenjiang, China, in 2015, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Telecommunication Engineering. His research interests include event detection, data mining, social computing, and cloud computing.

LU LIU received the M.S. degree from Brunel University and the Ph.D. degree from the University of Surrey. He is currently a Professor of Distributed Computing with the University of Derby, U.K, and an Adjunct Professor with Jiangsu University, China. His research interests are in areas of cloud computing, social computing, service-oriented computing, and peer-to-peer computing. Prof. Liu is a fellow of the British Computer Society.

JINGJING YAO received the B.E. degree from Jiangsu University, Zhenjiang, China, in 2011, and the D.M. degree from Jiangsu University, Zhenjiang, China, in 2016. Her research interests include coplex network, information dissemination.

BO YUAN received the Ph.D. degree from University of Derby, U.K in 2018, and BSc. degree in computer science and technology from the Tongji University, China in 2011. He is currently a Post-doctoral researcher with the University of Derby. His research focuses on Decentralized Computing, Cloud Computing, Online Social Networks and Deep Learning.

YONGJUN ZHENG received the PhD, MSc and BSc in Computer Science at Nottingham Trent University, he is employed as a Lecturer in Computer Science in Derby University now and used to work in York St John University; he also worked as a researcher at the University of Cambridge and Middlesex University before joined York St John University. His primary research interest concerns HCI including data visualization, mobile computing and intelligent systems.