IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 31, 2019, accepted January 13, 2020, date of publication January 29, 2020, date of current version February 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2970192

Urban Area Function Zoning Based on User Relationships in Location-Based Social Networks

FEI HAO [1,2,3], JUNZHE ZHANG [2], ZONGTAO DUAN [4], LIANG ZHAO [5], LANTIAN GUO [6], AND DOO-SOON PARK [7], (Member, IEEE)

[1] Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an 710016, China
[2] School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
[3] Department of Computer Science, College of Engineering, Mathematics, and Physical Sciences, University of Exeter, Exeter EX4 4PY, U.K.
[4] School of Information Engineering, Chang'an University, Xi'an 710064, China
[5] School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China
[6] School of Automation and Electronic Engineering, Qingdao University of Science and Technology, Qingdao 266061, China
[7] Department of Computer Software Engineering, Soonchunhyang University, Asan 336745, South Korea

Corresponding author: Fei Hao (feehao@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702317, in part by the EU Horizon 2020 Programme Marie Sklodowska-Curie Individual Fellowship under Grant H2020-MSCA-IF-2018-840922, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2019JM-379, in part by the Fund Program for the Scientific Activities of Selected Returned Overseas Professionals in Shaanxi Province under Grant 2017024, in part by the Ministry of Science and ICT (MSIT), South Korea, under the Information Technology Research Center (ITRC) Support Program supervised by the Institute for Information & communications Technology Planning & Evaluation (IITP) under Grant IITP-2019-2014-1-00720, and in part by the National Research Foundation of Korea under Grant 2017R1A2B1008421.

ABSTRACT With advanced development of Internet communication and ubiquitous computing, Social Networks are providing an important information channel for smart city construction. Therefore, analyzing Location-based Social Network is a very valuable work in achieving reasonable urban zoning. In Social Networks, a main purpose of prestige assessment is to extract influential users who are regarded as the key nodes for community detection from Onine Social Networks (OSNs). However, social relationships of users are rarely used to evaluate the popularity of physical locations and zone physical locations. In order to achieve urban area function zoning by evaluating the prestige of geographic regions based on user relationships in Location based Social Networks (LBSNs), this paper proposes a Prestige Density-Based Spatial Clustering of Applications with Noise algorithm (P-DBSCAN) by improving the existing DBSCAN algorithm. Specifically, the algorithm first calculates the centrality of users in the social network, and then converts the centrality of users into the location-centrality through the users' check-in data. After the centrality of each location is obtained, the discrete locations are clustered according to four constraints of the given radius. After clustering, the result of urban area function zoning can be achieved. Extensive experiments are conducted for demonstrating the effectiveness of our proposed algorithm in this paper. In addition, the visualization results reveal the correctness of our proposed approach.

INDEX TERMS Social network, prestige assessment, density clustering, eigenvector centrality, urban area function.

I. INTRODUCTION

As an important task of urban construction, urban area function zoning, requires the urban area function to be consistent with the actual activities of people [1], [6]. With the rapid development of computer science and technology, OSNs have been growing from nothing to cover more and more potential application fields, in which LBSNs provide an effective way to link individual's activities to geographic regions. LBSNs have the following features: (1) allowing users to form their social networks according to social interactions; (2) allowing users to connect with places by checking in. Therefore, in this paper, we conduct prestige assessment of users according to their social relationships and then the prestige of user

The associate editor coordinating the review of this manuscript and approving it for publication was Shirui Pan.

can be further obtained in different locations according to users' check-in data. After that, we zone urban area function according to prestige of locations [2].

As an important research topic in social network analysis, the major purpose of prestige assessment [7] is to find influential users from social networks. The influential users can be the key nodes for information diffusion and community detection [3]. There have been a number of relevant theoretical and empirical studies on the evaluation of social network prestige. Musial *et al.* [4] compared and summarized the user prestige assessment methods, such as Centrality Based on Node Degree [8], Eccentricity Centrality [9], Closeness Centrality [10], and Betweenness Centrality [11] which are commonly used in social network analysis. Particularly, the Eigenvector Centrality [12] is also an efficient approach for prestige evaluation. However, such prestige assessment only focuses on "*user*" without the influence of user's prestige on geographic regions. It limits the value and scope of applications enabled with prestige assessment results. In real life, we often describe a busy place according to the large flow of people. However, this is only a quantitative description, where the "*quality*" of users who have checked in here, is also an important factor for measuring the prestige of this place. In OSNs, the user's quality is thus characterized as the centrality value, in which the different types of centrality reflect the quality of the user's different aspects. Therefore, the prestige value of the entire geographical area can be evaluated by accumulating the "*quantity*" and "*quality*" of the users who checked in the area [27]. To address that issue, this paper conducts prestige assessment of locations on the basis of social network users' prestige assessment, including prestige assessment for discrete locations and geographic regions derived from density clustering [26].

As we known, density clustering using the DBSCAN algorithm can well divide many discrete locations into related geographic regions [5], [13]. Aiming to achieve the urban area function zoning, this paper pioneers a novel Prestige Density Based Spatial Clustering algorithm (P-DBSCAN) by modifying the existing DBSCAN algorithm [20]. The major contributions of this paper are summarized as follows:

- **Transformation from user centrality to location centrality:** A social network can be represented with multiple nodes (*i.e.*, users) and edges which indicates the social interactions. Normally, a social network [14] is mathematically formalized as a graph $G = (V, E)$ with V indicating the set of vertices and E referring to the set of relationships between vertices. In this paper, the centrality value of user V_i is denoted as $UCV(V_i)$, and the centrality value of location L_i is denoted as $LCV(L_i) = \sum_{h=k}^{j}(UCV(V_h))$ ($V_j,\ldots,V_k$ are the users who had checked in location L_i).
- **Prestige Density based Spatial Clustering algorithm:** Our proposed P-DBSCAN algorithm has four requirements for center point selection: 1) the radius; 2) the requirement for prestige of the center point; 3) the requirement for the number of points around the center point; 4) and the requirement for the sum of prestige of the points around the center point. The outputs are the location clustering results and the prestige of geographical area. That is to say, the clustering results are related not only to the distance between locations, but also to the prestige value.
- **Evaluation:** We conduct the experiments on a real location-based social networking dataset for validating the effectiveness of the proposed approach. The visualization results demonstrate that the clustering results are not only consistent with the real urban area function zoning, but also reflect the importance of each urban functional area.

The remainder of this paper is structured as follows. Section II overviews the related work. The preliminaries and problem statement are provided in Section III. Section IV proposes our approach for the addressed problem. The experimental results and discussions are presented in Section V. Section VI concludes this paper.

II. RELATED WORK

In this section, we will overview the most related research to this work from the following two aspects: (1) Influential location in LBSNs; and (2) Density based clustering.

A. INFLUENTIAL LOCATION IN LBSNs

Regarding to location promotion problem, Zhu *et al.* [21] formalized it as an influence maximization problem in an LBSNs. They proposed two user mobility models, i.e., Gaussian-based and distance-based mobility models, to capture the check-in behavior of individual user in LBSNs, based on location-aware propagation probabilities. Hai [22] proposed the PMNF model to capture human mobility and the IM greedy algorithms to maximize the influence spread of influential users. Their experiments validated that the PMNF model has covered important areas of human movement behavior. Wang *et al.* [23] first formulated the distance-aware influence maximization problem, then extracted a seed set that maximizes the expected influence over users who are more likely to be the potential customers of the promoted location. In one word, given a target location, their aim is to find the users that should be advertised to attract more visitors to this location. Doan *et al.* [24] evaluated the popularity ranks of locations based on the number of visitors. Aiming to find a single location which attracts most users, Zhou *et al.* [25] investigated the problem of choosing an optimal location for an event such that the event's influence can be maximized.

B. DENSITY BASED CLUSTERING

Density clustering method has been widely used in network analysis and data mining, etc. It considers the internally continuous dense sample subset as the same type, and the representative algorithm is DBSCAN. DBSCAN algorithm first divides all sample points into center points and non-center points, where the center points refer to those points within the radius r where the number of sample points is greater than k, where r and k are both parameters. If the distance between the two center points does not exceed r, it is called the direct density reachable. The center point with multiple direct densities is called density reachable. A class cluster is formed when a center point set with a maximum density reachable is merged with the non-center points within the radius r of each center point. It can be seen that the center point sets between various clusters obtained by DBSCAN algorithm do not intersect, but the non-center point sets may intersect, and there may be some non-center points whose r radius does not contain any center points, which is regarded as noise. Therefore, DBSCAN algorithm allows class cluster overlap and some points do not belong to any class cluster, which is an advantage that most other clustering algorithms do not have. The disadvantage of DBSCAN algorithm is that it utilizes a uniform scale r and k to measure all class clusters, which is often inappropriate [19]. Therefore, an improved version of the OPTICS [15] algorithm has been proposed, which sets the radius r as a flexible parameter to be adjusted by the algorithm itself, but it is ideologically the same as DBSCAN. However, the clustering result obtained by this improved algorithm is still only related to the distance between sample points, and our proposed P-DBSCAN algorithm increases the judgment condition of the center point, so that the clustering result is also related to the attribute (prestige of location) of the sample points.

III. PRELIMINARIES AND PROBLEM STATEMENT

This section firstly introduces the preliminaries on location social network and DBSCAN algorithm. Then, the problem of urban area function zoning based on user relationships in location-based social networks is formally defined.

Definition 1 (Location-Based Social Network [16], [17]): A location-based social network can be represented as a 3-tuple $G = (V, E, C)$, where V is the set of nodes (i.e., users), and E is the set of edges which indicates the social connections between users, and C is the users' check-in data set, each of these pieces of data records a user checking in at a certain time and place.

Figure 1 depicts an architecture of Location-based Social Network that are social networks using GPS features to locate you and that let you broadcast your location and location-tagged media content, such as photos, video, and texts from your mobile device. Therefore, this architecture contains two layers: (1) OSNs Layer: in this layer, users form their social networks according to interactions in LBSNs; (2) Physical Location Layer: mobile users who hold their mobile devices can broadcast their location by checking-in and checking-out with location-based services apps.

Overall, the physical location is composed of the instant location of a mobile user at a given timestamp and the location history that a mobile user has accumulated in a certain period. Further, the interdependency includes not only that two mobile users co-occur in the same physical location or share similar location histories but also the knowledge, *e.g.*, common interests, behavior, and activities, inferred from an user's location and location-tagged data [28].

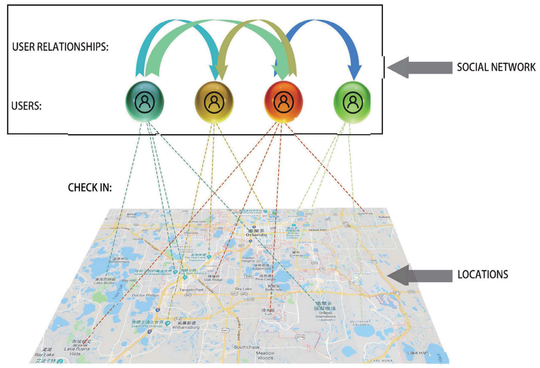


FIGURE 1. Architecture of location-based social network.

DBSCAN Algorithm: It is a density-based clustering algorithm based on high-density connected regions, which can divide regions with sufficient high-density into clusters and find clusters of arbitrary shapes in noisy data. This algorithm needs to select a distance for measuring. For the data set to be clustered, the distance between any two points reflects the density between points, indicating whether points and points can be clustered in the same class. Because the DBSCAN algorithm is difficult to define the density of high-dimensional data, the points in two-dimensional space can be measured by Euclidean distance.

DBSCAN algorithm requires two input parameters: one parameter is radius (Eps), which represents the range of circular neighborhood centered on a given point P; the other parameter is the minimum number of points in the neighborhood centered at point P (MinPts). Point P is called the center point if the number of points in the neighborhood with point P as the center and radius of Eps is no less than MinPts [18]. Unlike existing algorithms, the modified DBSCAN algorithm requires four input parameters. Besides Eps and MinPts, we add two constraints: the requirement for the center point prestige value, and the requirement for the sum of prestige value of the points around the center point. After the center points selected, the center points that can be connected are divided into a group and outliers are obtained according to the set of center points obtained and the value of radius Eps. It puts each group of center points and the points whose distance from the center point is less than the radius Eps into a cluster. Consequently, it completes the cluster division.

Figure 2 shows an example of an existing DBSCAN algorithm, where the clustering result is only related to the distance of points.

Problem 1 (Urban Area Function Zoning Based on User Relationships in Location-Based Social Networks): Figure 3 shows the map of Orlando, Florida, USA presented by Google map.[1] This paper will cluster the locations of Orlando and

[1] https://www.google.com/maps/place/orlando.html

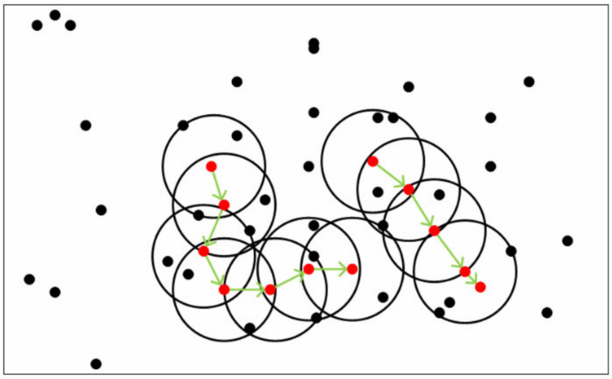


FIGURE 2. An example of DBSCAN algorithm.

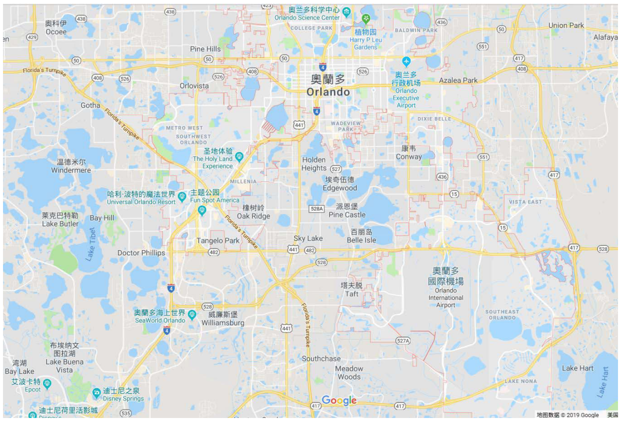


FIGURE 3. The map of Orlando, Florida, USA.

evaluate its geographical prestige. The popularity ratings are based on public location-based social networks data sets that are available from Stanford university.[2] The social network among users active in Orlando is $G(V, E)$, where V is the node set, representing all users in the social network, and E is the edge set. If there is an edge (V_1, V_2) in E, it means there is a connection between these two users. Another user's check-in data table Q contains the time and place the user checked in. The user's reputation value can be calculated via social network G, and the user's reputation can be assigned to the location via Q. After clustering the locations, the prestige of the geographical area can be further calculated.

Gowalla is a location-based social network where users share their location by checking in. The user's relational network is undirected, consisting of 196,591 nodes and 950,327 edges. Between February 2009 and October 2010, a total of 644,890 user check-ins are collected. In this paper, we only use the check-in data within Orlando city (longitude −81.15° to −81.65°, latitude 28.3° to 28.7°). The users' check-in data, named *gowalla_totalcheckins.txt*, is shown in Figure 4. For each behavior, one check-in data is user id, check-in time, check-in location latitude, check-in location longitude, and check-in location id from left to right.

[2]http://snap.stanford.edu/data/loc-Gowalla.html

Gowalla_totalCheckins.txt

1	0	2010-10-19T23:55:27Z	30.2359091167	-97.7951395833	22847
2	0	2010-10-18T22:17:43Z	30.2691029532	-97.7493953705	420315
3	0	2010-10-17T23:42:03Z	30.2557309927	-97.7633857727	316637
4	0	2010-10-17T19:26:05Z	30.2634181234	-97.7575966669	16516
5	0	2010-10-16T18:50:42Z	30.2742918584	-97.7405226231	5535878
6	0	2010-10-12T23:58:03Z	30.261599404	-97.7585805953	15372
7	0	2010-10-12T22:02:11Z	30.2679095833	-97.7493124167	21714
8	0	2010-10-12T19:44:40Z	30.2691029532	-97.7493953705	420315
9	0	2010-10-12T15:57:20Z	30.2811204101	-97.7452111244	153505
10	0	2010-10-12T15:19:03Z	30.2691029532	-97.7493953705	420315
11	0	2010-10-12T00:21:28Z	40.6438845363	-73.7828063965	23261
12	0	2010-10-11T20:21:20Z	40.74137425	-73.9881052167	16907
13	0	2010-10-11T20:20:42Z	40.741388197	-73.9894545078	12973
14	0	2010-10-11T00:06:30Z	40.7249103345	-73.9946207517	341255
15	0	2010-10-10T22:00:37Z	40.729768314	-73.9985353275	260957
16	0	2010-10-10T21:17:14Z	40.7285271242	-73.9968681335	1933724
17	0	2010-10-10T17:47:04Z	40.7417466987	-73.993421425	105068
18	0	2010-10-09T23:51:10Z	40.7341933833	-74.0041635333	34817
19	0	2010-10-09T22:27:07Z	40.7425115937	-74.0060305595	27836
20	0	2010-10-09T21:39:26Z	40.7423961659	-74.0075433254	15079
21	0	2010-10-09T21:36:05Z	40.7423961659	-74.0075433254	15079
22	0	2010-10-09T21:05:23Z	40.7358847426	-74.0049684048	22806
23	0	2010-10-09T20:55:47Z	40.7275253534	-73.9853990078	1365909
24	0	2010-10-09T01:37:03Z	40.7568799674	-73.9862251282	11844
25	0	2010-10-08T21:48:37Z	40.7074172208	-74.0113627911	11742
26	0	2010-10-08T21:45:48Z	40.7071727167	-74.0105454333	19822
27	0	2010-10-08T21:43:52Z	40.7070708167	-74.0119528667	15169
28	0	2010-10-08T21:43:02Z	40.705823135	-73.9966964722	11794
29	0	2010-10-08T19:28:36Z	40.7693780407	-73.9630830288	1567837
30	0	2010-10-08T17:24:27Z	40.7808054632	-73.9764726162	35513

FIGURE 4. The sample of *Gowalla_totalcheckins.txt*.

Gowalla_edges.txt

1	0	1
2	0	2
3	0	3
4	0	4
5	0	5
6	0	6
7	0	7
8	0	8
9	0	9
10	0	10
11	0	11
12	0	12
13	0	13
14	0	14
15	0	15
16	0	16
17	0	17
18	0	18
19	0	19
20	0	20
21	0	21
22	0	22
23	0	23
24	0	24
25	0	25
26	0	26
27	0	27
28	0	28
29	0	29
30	0	30

Gowalla_edges.txt

605792	12060	92486
605793	12060	92487
605794	12060	92488
605795	12060	92489
605796	12061	203
605797	12061	483
605798	12061	1545
605799	12061	3584
605800	12061	3824
605801	12061	9747
605802	12061	12101
605803	12061	15377
605804	12061	24641
605805	12061	28794
605806	12061	31623
605807	12061	36417
605808	12061	36434
605809	12061	36789
605810	12061	68785
605811	12061	76609
605812	12062	203
605813	12062	1413
605814	12062	1431
605815	12062	2149
605816	12062	3587
605817	12062	3702
605818	12062	6536
605819	12062	12033
605820	12062	44447
605821	12062	47469

FIGURE 5. The sample of *Gowalla_edges.txt*.

As shown in Figure 5, each behavior has two associated users'id, representing one edge of the relational network.

IV. PROPOSED APPROACH

In this section, we introduce how to implement the urban area function zoning of Orlando based on user relationships in LBSNs. The process of implementation is shown in Figure 6, which is composed of five steps: (1) check-in data filtering (*Red module*); (2) user relationship network filtering (*Yellow module*); (3) calculating the node centrality of location (*Blue module*); (4) calculating the eigenvector centrality of location (*Green module*); (5) density clustering (*Purple modules*).

Step 1: Check-in data filtering

By invoking Algorithm 1, this step works as follows: It opens and reads file *Gowalla_totalCheckins.txt* line by line, and converts each line into a list named *check_out*[] (Line 2). Then it checks whether the second and third item of *check_out*[] (the latitude and longitude value of the check-in place) meet the requirements. If it matches, the line

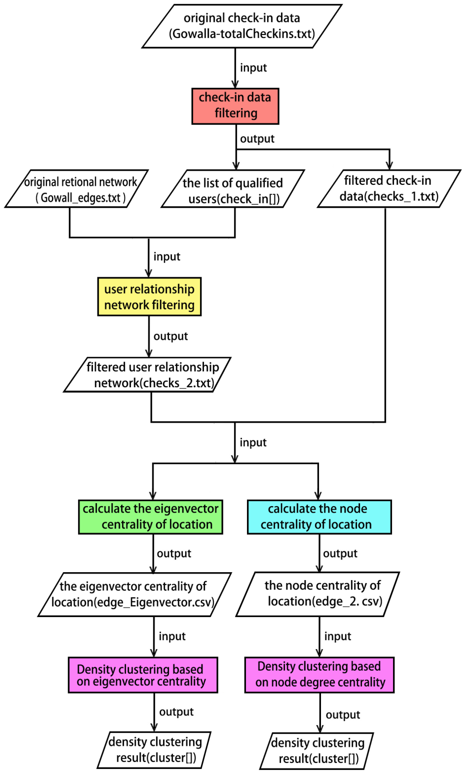


FIGURE 6. The framework of our proposed approach.

Algorithm 1 Check-In Data Filtering Algorithm

Input:
Original check-in data(*Gowalla_totalCheckins.txt*)

Output:
List of qualified users(*checks_in*[]), filtered check_in data(*checks_1.txt*)

1: **begin**
2: **for** each line in *Gowalla_totalCheckins.txt*
3:     **if** this line of check-in data **in** Orlando
4:         write this line into *checks_1.txt*
5:         *check_in.append*(the user of this line)
6: **end**

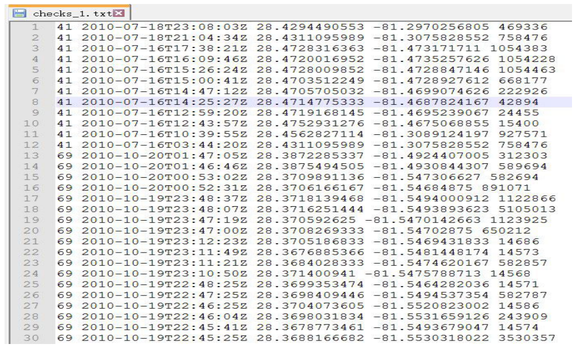
is written to the newly created document *checks_1.txt* (as shown in Figure 7), and the first item of *check_out*[] (user id number) added to the list *check_in*[] (Lines 3-5).

Step 2: User relationship network filtering

This step works by invoking Algorithm 2: this step opens and reads file *Gowalla_edges.txt* line by line, and converts each line into a list named *select_out*[] (Line 2). If both the zeroth and first item of *select_out*[] are contained in *check_in*[] (user *a* and user *b* have both been checked in the scope), that line is written to the document *checks_2.txt* (Lines 3-4) (as shown in Figure 8).

Step 3: Calculating the node centrality of location

This step works as follows: we take *checks_2.txt* and *checks_1.txt* as inputs of this step, then we create a file named *edge_2.csv* with four columns for "*local_id*", "*longitude*",



```
checks_1.txt
1  41 2010-07-18T23:08:03Z 28.4294490553 -81.2970256805 469336
2  41 2010-07-18T21:04:34Z 28.4311095989 -81.3075828552 758476
3  41 2010-07-16T17:38:21Z 28.4728316363 -81.473171711 1054383
4  41 2010-07-16T16:09:46Z 28.4720016952 -81.4735257626 1054228
5  41 2010-07-16T15:26:24Z 28.4728009852 -81.4728847146 1054463
6  41 2010-07-16T15:00:41Z 28.4703512249 -81.4728927612 668177
7  41 2010-07-16T14:47:12Z 28.4705705032 -81.4699074626 222926
8  41 2010-07-16T14:25:27Z 28.4714775333 -81.4687824167 42894
9  41 2010-07-16T12:59:20Z 28.4719168145 -81.4695239067 24455
10 41 2010-07-16T12:43:57Z 28.4752931276 -81.4675068855 15400
11 41 2010-07-16T10:39:55Z 28.4562827114 -81.3089124197 927571
12 41 2010-07-16T03:44:20Z 28.4311095989 -81.3075828552 758476
13 69 2010-10-20T01:47:05Z 28.3872285337 -81.4924407005 312303
14 69 2010-10-20T01:46:46Z 28.3875494505 -81.4930844307 589694
15 69 2010-10-20T00:53:02Z 28.3709891136 -81.547306627 582694
16 69 2010-10-20T00:52:31Z 28.370616167 -81.5468475 891071
17 69 2010-10-19T23:48:37Z 28.3718139468 -81.5494000912 1122866
18 69 2010-10-19T23:48:07Z 28.3716251444 -81.5493893623 5105013
19 69 2010-10-19T23:47:19Z 28.370592625 -81.5470142663 1123925
20 69 2010-10-19T23:47:00Z 28.3708269333 -81.54702875 650212
21 69 2010-10-19T23:12:23Z 28.3705186833 -81.5469431833 14686
22 69 2010-10-19T23:11:49Z 28.367688536 6 -81.5481448174 14573
23 69 2010-10-19T23:11:21Z 28.3684028333 -81.5474620167 582857
24 69 2010-10-19T23:10:50Z 28.371400941 -81.5475788713 14568
25 69 2010-10-19T22:48:25Z 28.3699353474 -81.546428 2036 14571
26 69 2010-10-19T22:47:25Z 28.3698409446 -81.5494537354 582787
27 69 2010-10-19T22:46:25Z 28.3704073605 -81.5520823002 14586
28 69 2010-10-19T22:46:04Z 28.3698031834 -81.5531659126 243909
29 69 2010-10-19T22:45:41Z 28.367877346 1 -81.5493679047 14574
30 69 2010-10-19T22:45:25Z 28.3688166682 -81.5530318022 3530357
```

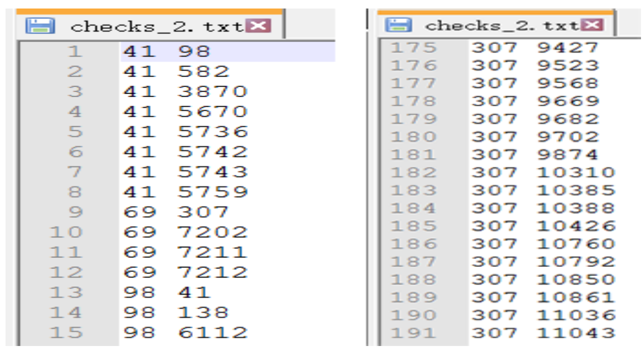
FIGURE 7. The sample of *checks_1.txt*.

Algorithm 2 User Relationship Network Filtering Algorithm

Input:
Original relational network(*Gowalla_edges.txt*), list of qualified users(*check_in*[])

Output:
Filtered user relationship network(*checks_2.txt*)

1: **begin**
2: **for** each line in *Gowalla_totalCheckins.txt*
3:     **if** both users of this line **in** *check_in*[]
4:         write this line into *checks_2.txt*
5: **end**



```
checks_2.txt
1  41 98
2  41 582
3  41 3870
4  41 5670
5  41 5736
6  41 5742
7  41 5743
8  41 5759
9  69 307
10 69 7202
11 69 7211
12 69 7212
13 98 41
14 98 138
15 98 6112
```

```
checks_2.txt
175 307 9427
176 307 9523
177 307 9568
178 307 9669
179 307 9682
180 307 9702
181 307 9874
182 307 10310
183 307 10385
184 307 10388
185 307 10426
186 307 10760
187 307 10792
188 307 10850
189 307 10861
190 307 11036
191 307 11043
```

FIGURE 8. The sample of *checks_2.txt*.

"*latitude*" and "*degree_local*". After that, we calculate the node degree centrality of users according to *checks*_2.txt (Line 2 in Algorithm 3), and assign the node degree centrality of users to check-in locations according to *checks_1.txt*. At last, the location data is stored in *edge_2.csv* (as shown in Figure 9), and each line of data contains four components: location id, location longitude value, location latitude value, and node degree centrality of location (Lines 3-5 in Algorithm 3).

Step 4: Calculating the eigenvector centrality of location

In this step, user's node degree centrality is taken as the initial value to calculate user's eigenvector centrality. After that, we assign the eigenvector centrality of users to check-in locations by the same way we carried out in the previous step. At last, the location information data is stored in *edge_Eigenvector.csv*.

Algorithm 3 Calculating the Node Centrality of Location Algorithm

Input:
Filtered user relationship network(*checks_2.txt*), filtered check_in data(*checks_1.txt*)
Output:
Node centrality of location(*edge_2.csv*)

begin
calculate the node degree centrality of users
for each location
  $LCV(L_i) = \sum_{h=k}^{j}(UCV(V_h))$ ($V_j,\ldots,V_k$ are the users who had checked in location L_i)
    write this line into *checks_2.txt*
end

Algorithm 4 P-DBSCAN Based Clustering Algorithm

Input:
Node centrality of location(*edge_2.csv*),or,eigenvector centrality of location(*edge_Eigenvector.csv*)
Output:
Density cluster result(*clusters*[])

1: *data*[] = *edge_2.csv* or *edge_Eigenvector.csv*
2: **begin**
3: **for** each line in *data*[]
4:   **if** the location in this line meets the four constraints
5:     *center_p.append*(this line)
6: def *divi_centerP*(data, eps, minPts, deg1, deg2):
7: def *add_otherP*(data, eps, minPts, deg1, deg2):
8: **end**

	A	B	C	D
1	local_id	longitude	latitude	degree_local
2	469336	-81.297026	28.4294491	76
3	758476	-81.307583	28.4311096	1625
4	1054383	-81.473172	28.4728316	106
5	1054228	-81.473526	28.4720017	169
6	1054463	-81.472885	28.472801	204
7	668177	-81.472893	28.4703512	26
8	222926	-81.469907	28.4705705	89
9	42894	-81.468782	28.4714775	152
10	24455	-81.469524	28.4719168	297
11	15400	-81.467507	28.4752931	381
12	927571	-81.308912	28.4562827	19

FIGURE 9. The sample of *edge_2.csv*.

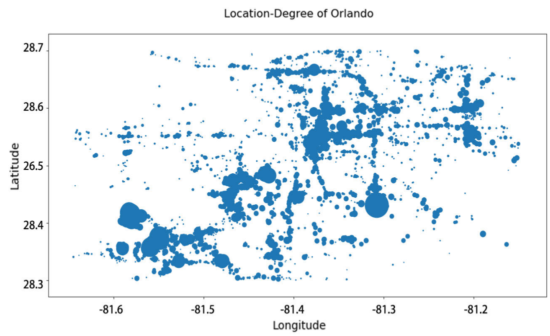


FIGURE 10. The visualization of file *edge_2.csv*.

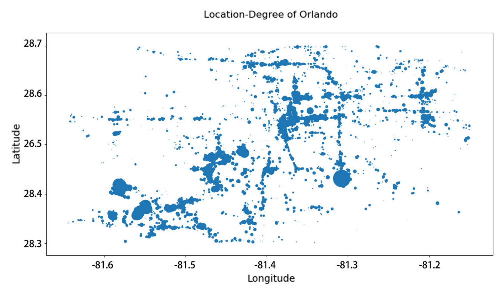


FIGURE 11. The visualization of file *edge_Eigenvector.csv*.

Step 5: P-DBSCAN based clustering

As shown in Algorithm 4, *edge_2.csv* or *edge_Eigenvector.csv* is the input of this step, and we store the file contents in *data*[] as a list. Then we select center points from *data*[] and store it in the list *center_p*[] (Lines 3-5). After center points selected, the center points that can be connected are divided into a group. We put each group of center points and the points whose distance from the center point is less than the radius into a cluster. Thus complete the clustering division (Lines 6-7). And the total prestige value of the region is obtained by adding the prestige values of all locations in same cluster.

V. EXPERIMENTS

This section mainly carries out the experiments and evaluates the proposed approach. All the experiments are implemented on a quad-core computer with 2.50-GHZ and 8G memory.

A. VISUALIZATION OF EXPERIMENTAL RESULTS

In order to better exhibit the experimental results, we employed the bubble chart as the visualization of experimental results.

Figure 10 shows the visualization result of *edge_2.csv*, each circle in the figure represents a place in Orlando, and the larger the circle's radius, the higher the prestige value of the place.

When calculating the eigenvector centrality, we need to multiply the adjacency matrix and the initial vector many times, since the elements in the matrix and vector are both positive, the value obtained by multiple times multiplication is very large, and the output of the final result needs to be scaled down by 10^{61} times. And Figure 11 is the visualization result of *edge_Eigenvector.csv*.

According to the clustering result, list *clusters*[], obtained in step 5, we respectively visualize the results of density clustering based on node degree centrality and density clustering based on eigenvector centrality (as shown in Figure 12 and Figure 13).

B. ANALYSIS OF EXPERIMENTAL RESULTS

In this experiment, more than 5,000 locations in Orlando are divided into 7 zones. By comparing with the real

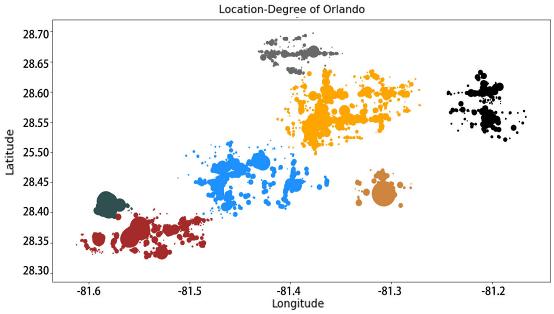


FIGURE 12. Clustering result based on node centrality of location.

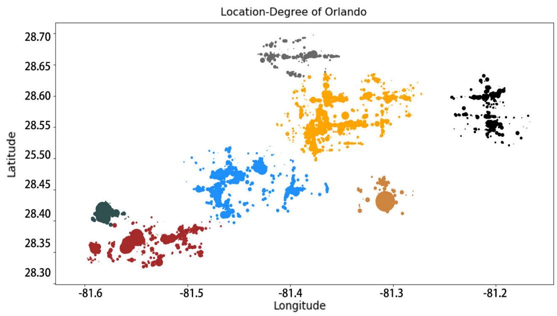


FIGURE 13. Clustering result based on eigenvector centrality of location.

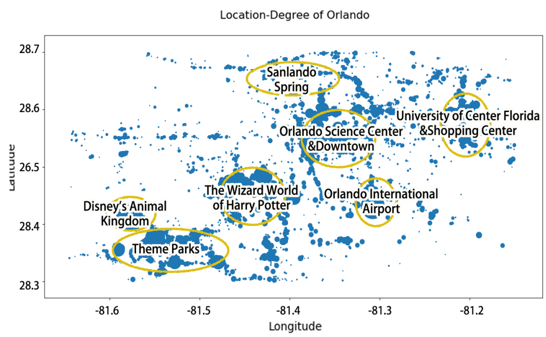


FIGURE 14. The important functional point-of-interests in each zone.

map[3] (Figure 3), we found out the important locations in each zone and calculated the total value of each zone (as shown in Figure 14). Figures 15 and 16 show the percentage of prestige for area functions (*i.e.*, the popularity of area functions) with two approaches. Apparently, it is found that there exists a little bit difference in the relative prestige between approaches.

The running time of each step is shown in Table 1:

TABLE 1. Running time of each part of the process.

Process step	Step 1	Step 2	Step 3
Running time	100.31s	461.45s	45.72s
Process step	Step 4	Step 5 (node degree centrality)	Step 5 (eigenvector centrality)
Running time	481.85s	216.15s	218.88s

[3]https://www.google.com/maps/place/orlando.html

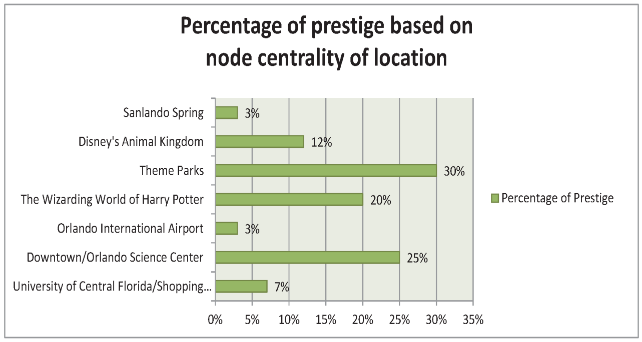


FIGURE 15. Prestige assessment result based on node centrality of location.

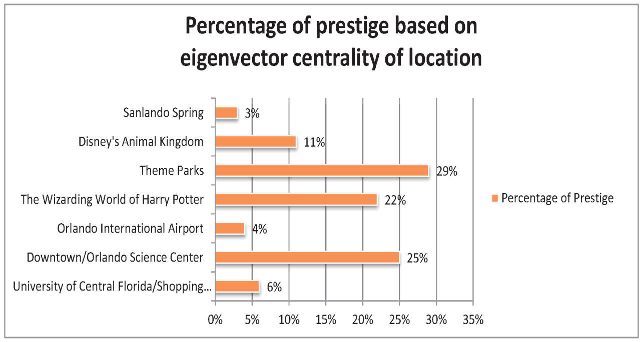


FIGURE 16. Prestige assessment result based on eigenvector centrality of location.

C. ANALYSIS OF INPUT CONDITION

The key problem to be solved in this paper is to use the P-DBSCAN algorithm to evaluate the popularity of geographic regions and zone them based on user's prestige.

The input of the P-DBSCAN algorithm is the location information data set. We have the following four requirements: the given radius (**Requirement 1**), the requirement for the center point reputation value (**Requirement 2**), the requirement for the number of locations around the center point (**Requirement 3**), and the requirement for the sum of the location reputation value around the center point (**Requirement 4**); and the output is the result of urban area function zoning and geographical region prestige value.

Let us take the regional prestige evaluation based on node degree centrality as an example, the locations are divided into 7 clusters, which is consistent with the location relationship in the actual map. At this time, the given radius is 0.022 longitude and latitude, at least 160 points within the radius are required, the site reputation value is required to be at least 30, and the sum of the surrounding sites reputation value is required to be at least 2000. We use the control variable method to study the relationship between each input value and the clustering division result, change the value of one constraint condition and keep the other three input values unchanged, and obtain the relationship between each input value and the number of clustering and the number of center points (as shown from Table 2 to Table 5).

It can be seen that the higher the requirement of the constraint condition, the fewer points meet the center

TABLE 2. Requirements for the sum of reputation values around the central point.

Requirement 4	0	2000	3000	5000	7000	9000
♯ of center points	2134	2134	2131	2090	1957	1586
♯ of clusters	7	7	7	6	7	5

TABLE 3. Central point reputation requirements.

Requirement 2	0	30	50	100	200	500
♯ of center points	2134	2134	1511	419	176	56
♯ of clusters	7	7	9	11	12	3

TABLE 4. Requirements for the number of locations around the central point.

Requirement 3	60	100	160	200	300	500
♯ of center points	2398	2327	2134	1984	1035	94
♯ of clusters	7	8	7	8	3	2

TABLE 5. Requirements for radius.

Requirement 1	0.01	0.02	0.022	0.03	0.05	0.1
♯ of center points	983	2064	2134	2393	2613	2688
♯ of clusters	10	9	7	7	2	1

point condition. When the center point becomes small, the number of clustering may increase or decrease. The increase is caused by the disappearance of the center point acting as the "*bridge*" in the original cluster, which results in the splitting of the original cluster. The reduction is due to the disappearance of a large number of center points and the disappearance of the entire cluster.

In fact, the relationship between the input and output of the P-DBSCAN algorithm can be viewed as a function with four independent variables and two dependent variables. Since such a function cannot be directly represented with a graph, the above tables only partially reflect the relationship between the input and output of the algorithm.

VI. CONCLUSION

This paper aims to evaluate the reputation of geographical areas in reality based on social networks, so as to provide a support for the actual urban planning and supporting facilities construction. We propose a modified DBSCAN algorithm, termed P-DBSCAN which is a novel Prestige Density Based Spatial Clustering algorithm. The algorithm first calculates'the centrality of users in the social network, and then converts the centrality of users into the location-centrality through the user check-in data. After obtaining the centrality of each location, the discrete locations are clustered according to the four constraints of the given radius. After clustering, the result of urban area function zoning can be achieved. The reputation of the interior points of the geographical area divided is the total reputation of the geographical area. In order to test the performance of the proposed algorithm, we also used the control variable method to study the influence of each constraint condition on the final output. Experimental results show that the output of our algorithm is consistent with the real map.

In this paper, we have completed the geographical area prestige assessment, however this is based on a relatively simple static social network. The original check-in data set contains user check-in time information, which is not utilized by our algorithm. In the subsequent algorithm improvement, it can be considered to construct a dynamic social network by using the time information in the check-in data, and change the user's prestige from a constant to a dynamic function, so as to change the location reputation evaluation from static to dynamic.

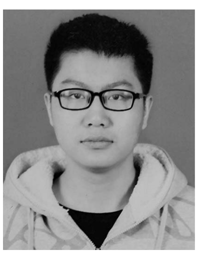
In addition, with the rapid growth of various online social networking services, the types of social networks are increasingly diversified. In this paper, we use only data from the location-based social networking service Gowalla. From a development point of view, we plan to conduct a prestige assessment based on data from online platforms in other potential fields to verify the applicability and universality of our approach.

REFERENCES

[1] A. Founoun and A. Hayar, "Evaluation of the concept of the smart city through local regulation and the importance of local initiative," in *Proc. IEEE Int. Smart Cities Conf. (ISC2)*, Sep. 2018, pp. 1–6.
[2] S. B. Kylasa, G. Kollias, and A. Grama, "Social ties and checkin sites: Connections and latent structures in location based social networks," in *Proc. IEEE/ACM Int. Conf. ASONAM*, Aug. 2015, pp. 194–201.
[3] S. Y. Bhat and M. Abulaish, "OCTracker: A density-based framework for tracking the evolution of overlapping communities in OSNs," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2012, pp. 501–505.
[4] K. Musiał, P. Kazienko, and P. Bródka, "User position measures in social networks," in *Proc. 3rd Workshop Social Netw. Mining Anal.-SNA-KDD*, 2009, pp. 1–9.
[5] S. Beri and K. Kaur, "Hybrid framework for DBSCAN algorithm using fuzzy logic," in *Proc. Int. Conf. Futuristic Trends Comput. Anal. Knowl. Manage. (ABLAZE)*, Feb. 2015, pp. 383–387.
[6] L. Wang, F. Fang, X. Yuan, Z. Luo, Y. Liu, B. Wan, and Y. Zhao, "Urban function zoning using geotagged photos and openstreetmap," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 815–818.
[7] L. Huang and Z. Xia, "Measuring user prestige and interaction preference on social network site," in *Proc. 8th IEEE/ACIS Int. Conf. Comput. Inf. Sci.*, Jun. 2009, pp. 1161–1166.
[8] I. Tugal and A. Karci, "Centrality of nodes with Karcı entropy," in *Proc. Int. Conf. Artif. Intell. Data Process. (IDAP)*, Sep. 2018, pp. 1–7.
[9] K. Iwabuchi, G. Sanders, K. Henderson, and R. Pearce, "Computing exact vertex eccentricity on massive-scale distributed graphs," in *Proc. IEEE Int. Conf. Cluster Comput. (CLUSTER)*, Sep. 2018, pp. 257–267.
[10] R. Mittal and M. Bhatia, "Cross-layer closeness centrality in multiplex social networks," in *Proc. 9th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2018, pp. 1–5.
[11] H. B. M. Shashikala, R. George, and K. A. Shujaee, "Outlier detection in network data using the Betweenness centrality," in *Proc. SoutheastCon*, Apr. 2015, pp. 1–5.
[12] A. Bihari and M. K. Pandia, "Eigenvector centrality and its application in research professionals' relationship network," in *Proc. Int. Conf. Futuristic Trends Comput. Anal. Knowl. Manage. (ABLAZE)*, Feb. 2015, pp. 510–514.
[13] U. Angkhawey and V. Muangsin, "Detecting points of interest in a city from taxi GPS with adaptive DBSCAN," in *Proc. 7th ICT Int. Student Project Conf. (ICT-ISPC)*, Jul. 2018, pp. 1–6.
[14] J. Hu, M. Liu, and J. Zhang, "A semantic model for academic social network analysis," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2014, pp. 310–313.

[15] D. N. Christodoulides and G. I. Stegeman, "Discreteness in optics," in *Proc. Conf. Lasers Electro-Opt./Int. Quantum Electron. Conf.*, 2009, pp. 1–2.
[16] Z. Zou, X. Xie, and C. Sha, "Mining user behavior and similarity in location-based social networks," in *Proc. 7th Int. Symp. Parallel Archit., Algorithms Program. (PAAP)*, Dec. 2015, pp. 167–171.
[17] Z. Wu, S. Pan, G. Long, C. Zhang, P. S. Yu, and F. Chen, "A comprehensive survey on graph neural networks," 2019, *arXiv:1901.00596*. [Online]. Available: https://arxiv.org/abs/1901.00596
[18] Z. Wang, M. Huang, H. Du, and H. Qin, "A clustering algorithm based on FDP and DBSCAN," in *Proc. 14th Int. Conf. Comput. Intell. Secur. (CIS)*, Nov. 2018, pp. 145–149.
[19] L. Meng'Ao, M. Dongxue, G. Songyuan, and L. Shufen, "Research and improvement of DBSCAN cluster algorithm," in *Proc. 7th Int. Conf. Inf. Technol. Med. Edu. (ITME)*, Nov. 2015, pp. 537–540.
[20] D. Pan and L. Zhao, "Uncertain data cluster based on DBSCAN," in *Proc. Int. Conf. Multimedia Technol.*, Jul. 2011, pp. 3781–3784.
[21] W. Y. Zhu, W. C. Peng, L. J. Chen, K. Zheng, and X. Zhou, "Modeling user mobility for location promotion in location-based social networks," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1573–1582.
[22] N. T. Hai, "A novel approach for location promotion on location-based social networks," in *Proc. IEEE RIVF Int. Conf. Comput. Commun. Technol.-Res., Innov., Vis. Future (RIVF)*, Jan. 2015, pp. 53–58.
[23] A. Wang, A. Zhang, A. Zhang, and A. Lin, "Distance-aware influence maximization in geo-social network," in *Proc. IEEE 32nd Int. Conf. Data Eng. (ICDE)*, May 2016, pp. 1–12.
[24] T.-N. Doan, F. C. T. Chua, and E.-P. Lim, "Mining business competitiveness from user visitation data," in *Proc. Int. Conf. Social Comput., Behav.-Cultural Modeling, Predict.*, 2015, pp. 283–289.
[25] T. Zhou, J. Cao, B. Liu, S. Xu, Z. Zhu, and J. Luo, "Location-based influence maximization in social networks," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.-CIKM*, 2015, pp. 1211–1220.
[26] Q. Qiu, L.-P. Chen, T. Yao, Q.-Y. Zheng, and Z. Yang, "The effect of country of brand image on symbolic value: Brand prestige as a mediator," in *Proc. Int. Conf. Manage. Sci. Eng. 21th Annu. Conf.*, Aug. 2014, pp. 565–570.
[27] Q. Hu and Y. Zhang, "An effective selecting approach for social media big data analysis—Taking commercial hotspot exploration with Weibo check-in data as an example," in *Proc. IEEE 3rd Int. Conf. Big Data Anal.*, Mar. 2018, pp. 28–32.
[28] Y. Zheng, "Location-based social networks: Users," in *Computing With Spatial Trajectories*. New York, NY, USA: Springer, 2011, pp. 243–276.

FEI HAO received the B.Sc. degree in information and computing science and the M.Sc. degree in computer software and theory from Xihua University, China, in 2005 and 2008, respectively, and the Ph.D. degree in computer science and engineering from Soonchunhyang University, South Korea, in 2016. Since 2016, he has been with Shaanxi Normal University, Xi'an, China, where he is currently an Associate Professor. He is also taking a Marie Sklodowska-Curie Individual Fellowship with the University of Exeter, Exeter, U.K. His research interests include social computing, ubiquitous computing, big data analysis and processing, and mobile cloud computing.



JUNZHE ZHANG graduated from the School of Computer Science, Shaanxi Normal University, in June 2019. His research interests include location-based social networks and data mining.

ZONGTAO DUAN received the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2006. He is currently a Professor and the Vice Director with the School of Information and Engineering, Chang'an University, Xi'an. His research interests include parallel computing and intelligent transportation systems.

LIANG ZHAO received the Ph.D. degree from the School of Computing, Edinburgh Napier University, in 2011. He is an Associate Professor with Shenyang Aerospace University, China. Before joining Shenyang Aerospace University, he worked as an Associate Senior Researcher with Hitachi (China) Research and Development Corporation, from 2012 to 2014. He has published more than 60 articles in international journal and conferences, including the IEEE TPDS, the *IEEE ITS Magazine*, the IEEE TMC, and the IEEE ICC. His research interests include VANETs, SDVN, FANETs, and WMNs.

LANTIAN GUO received the Ph.D. degree from Northwestern Polytechnical University, Xi'an, China. He is currently an Assistant Professor with the School of Automation and Electronic Engineering, Qingdao University of Science and Technology, China. He was a Visiting Researcher with the School of Computing, Queen's University, Canada. His current research interests include big data, recommendation systems, machine learning, and artificial intelligence.

DOO-SOON PARK (Member, IEEE) received the Ph.D. degree in computer science from Korea University, in 1988. He was the President of the Korea Information Processing Society (KIPS), in 2015, and the Director of the Central Library, Soonchunhyang University, from 2014 to 2015, where he was also the Dean of the Engineering College, from 2002 to 2003. He is currently a Professor with the Department of Computer Software Engineering, Soonchunhyang University, South Korea, where he is also the Director of the Wellness Service Coaching Center. His research interests include data mining, big data processing, and parallel processing. He is a member of ACM, KIPS, KMS, and KIISE. He was the Editor-in-Chief of the *Journal of Information Processing Systems* (JIPS) at KIPS, from 2009 to 2012. He has served as an Organizing Committee Member of international conferences, including FutureTech 2016, MUE 2016, WORLDIT 2016, GLOBAL IT 2016, CUTE 2015, and CSA 2015.