Vehicle Trajectory Clustering Based on Dynamic Representation Learning of Internet of Vehicles

Wei Wang, Feng Xia, *Senior Member, IEEE*, Hansong Nie, Zhikui Chen, *Member, IEEE*, Zhiguo Gong, *Senior Member, IEEE*, Xiangjie Kong, *Senior Member, IEEE*, and Wei Wei, *Senior Member, IEEE*

***Abstract*— With the widely used Internet of Things, 5G, and smart city technologies, we are able to acquire a variety of vehicle trajectory data. These trajectory data are of great significance which can be used to extract relevant information in order to, for instance, calculate the optimal path from one position to another, detect abnormal behavior, monitor the traffic flow in a city, and predict the next position of an object. One of the key technology is to cluster vehicle trajectory. However, existing methods mainly rely on manually designed metrics which may lead to biased results. Meanwhile, the large scale of vehicle trajectory data has become a challenge because calculating these manually designed metrics will cost more time and space. To address these challenges, we propose to employ network representation learning to achieve accurate vehicle trajectory clustering. Specifically, we first construct the k-nearest neighbor-based internet of vehicles in a dynamic manner. Then we learn the low-dimensional representations of vehicles by performing dynamic network representation learning on the constructed network. Finally, using the learned vehicle vectors, vehicle trajectories are clustered with machine learning methods. Experimental results on the real-word dataset show that our method achieves the best performance compared against baseline methods.**

***Index Terms*— Internet of Vehicles, network representation learning, vehicle trajectory clustering.**

Manuscript received January 26, 2020; revised March 28, 2020; accepted April 21, 2020. This work is funded by National Key R&D Program of China (2019YFB1600700), The Science and Technology Development Fund, Macau SAR (SKL-IOTSC-2018-2020, FDCT/0045/2019/A1, FDCT/007/2016/AFJ), Guangzhou Science and Technology Innovation and Development Commission (EF005/FST-GZG/2019/GSTIC), Research Committee of University of Macau (MYRG2017-00212-FST, MYRG2018-00129-FST), and China Postdoctoral Science Foundation (2019M651115). The Associate Editor for this article was Z. Lv. *(Corresponding author: Feng Xia.)*

Wei Wang is with the School of Software, Dalian University of Technology, Dalian 116620, China, and also with the State Key Laboratory of Internet of Things for Smart City and Department of Computer and Information Science, University of Macau, Macau 999078 (e-mail: ehomewang@ieee.org).

Feng Xia is with the School of Science, Engineering, and Information Technology, Federation University Australia, Ballarat 3353, Australia (e-mail: f.xia@ieee.org).

Hansong Nie and Zhikui Chen are with the School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: hansong.nie@outlook.com; zkchen@dlut.edu.cn).

Zhiguo Gong is with the State Key Laboratory of Internet of Things for Smart City and Department of Computer and Information Science, University of Macau, Macau 999078 (e-mail: fstzgg@um.edu.mo).

Xiangjie Kong is with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: xjkong@ieee.org).

Wei Wei is with the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China (e-mail: weiwei@xaut.edu.cn).

Digital Object Identifier 10.1109/TITS.2020.2995856

I. Introduction

WITH the advance of Internet of things, communication technology, and the increasing ability of data collection, we are able to obtain a large amount of spatial trajectory data [24], [31]. These trajectory data are significant for exploring human movement patterns and activity patterns. For example, it can be used to extract relevant information in order, for instance, to calculate the optimal path from position A to B, detect abnormal behavior, monitor the traffic flow in a city, and predict the next position of an object, etc. Exploring the underlying knowledge of trajectory data is an important way to solve urban problems. More recently, trajectory data mining has become a research hotspot in data mining.

Vehicle trajectory data is one of the most common trajectory data which refers to the large amount of real-time trajectory data produced by intelligent connected vehicles and the popularization of onboard positioning terminal equipment [10], [25], [26]. Mining and analyzing vehicle trajectory data are of great significance for vehicle to vehicle communication, intelligent transportation, point of interest recommendation, and location-based services. These goals can be achieved by investigating the different features of vehicle trajectory data characterizing the temporal and spatial information of the vehicles on the road. For this purpose, vehicle trajectory clustering has been proposed [1], [13], [15].

Vehicle trajectory clustering aims to regroup similar vehicle trajectories together into different groups [28], [30]. It is challenging to cluster vehicle trajectories because of the large-scale, time-aware, and changing characteristics of vehicles. With the increasing use of GPS receivers and 5G mobile devices, the size of trajectory data is increasing sharply. Vehicles may move randomly following different paths in a certain area.The road networks of different city regions may be totally different. In addition, the same vehicle may present totally different trajectories over different time periods of a day. Meanwhile, the patterns on weekdays and weekends may also different.

To our knowledge, many studies have been attempted for vehicle trajectories clustering. Most of them are designed

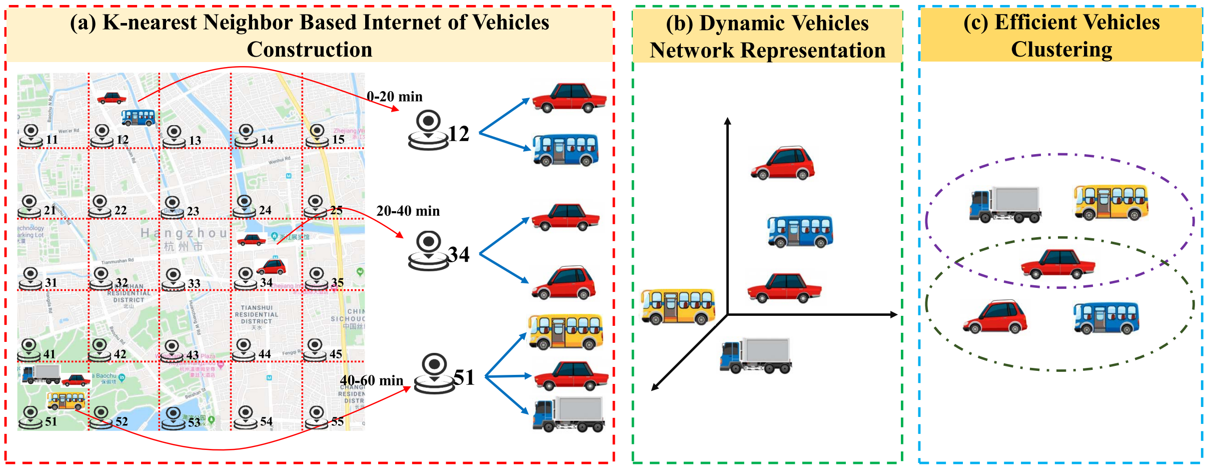

(a) K-nearest Neighbor Based Internet of Vehicles Construction
(b) Dynamic Vehicles Network Representation
(c) Efficient Vehicles Clustering
0-20 min
20-40 min
40-60 min
12
34
51
11
12
13
14
15
21
22
23
24
25
31
32
33
34
35
41
42
43
44
45
51
52
53
54
55

Fig. 1. Framework of the proposed method.

based on distance [8], [12]. For example, Besse *et al.* [1] propose a symmetrized segment-path distance based vehicle trajectory clustering method, which is time insensitive by calculation the difference between the shape and physical distance of two vehicle trajectories. Yu *et al.* [29] design a new trajectory clustering algorithm based on multi-feature trajectory similarity measure which utilizes the characteristics of orientation, speed, shape, location, and continuity of each trajectory for clustering. However, although these methods can achieve certain success, their proposed metrics are manually designed which may lead to biased results. Meanwhile, the large-scale characteristic of vehicle trajectory data has become a limitation when using these methods. Calculating these manually designed metrics will cost more running time and computing space. Moreover, vehicle trajectory data are well-structured into networks so that it is promising to cluster vehicle trajectory from a network perspective.

Against this background, in this paper, we propose an efficient vehicle trajectory clustering method by performing network representation learning on k-nearest neighbor-based internet of vehicles. First of all, we represent vehicles in the selected area as nodes and construct the internet of vehicles based on the k-nearest neighbor. As the location of vehicles is constantly changing, the vehicle network is a dynamic network. Then we propose to perform dynamic network representation learning on the constructed dynamic vehicular network to learn the embedding vectors of vehicles in the network. Finally, typical clustering methods are used to cluster the embedding vectors of nodes to find vehicles with similar behavior patterns. Experimental results on a real-world dataset demonstrate that our proposed method can achieve the best performance by comparison with several baseline methods.

Our main contributions of this paper can be summarized as follows:

- We propose to cluster vehicle trajectory from a network perspective where the network is constructed based on the k-nearest neighbor. Moreover, to capture changing positions of vehicles, we construct the network in a dynamic manner.
- We design a dynamic network representation learning based vehicle trajectory clustering method, where the vehicle similarities are calculated based on the represented low-dimensional vehicle vectors.
- We perform extensive experiments on the real-world dataset to evaluate the performance of our method and the experimental results suggest that our method can achieve the best performance.

Our paper is organized as follows. Section 2 presents our method. The experimental setups are introduced in Section 3. Section 4 shows the experimental results. We review related work in Section 5 and conclude this paper in Section 6.

II. Proposed Method

In this paper, we propose a novel method to cluster vehicles in a certain area to find vehicles with similar behavior patterns. The framework of our proposed method can be seen in Fig. 1. First of all, we represent vehicles in the selected area as nodes and construct the vehicle social network. As the location of vehicles is constantly changing, the vehicle social network is a dynamic network. Then we propose to perform dynamic network representation learning on the constructed dynamic vehicular networks to learn the embedding vectors of vehicles in the network. Finally, typical clustering methods are used to cluster the embedding vectors of nodes to find vehicles with similar behavior patterns. The details of our method will be discussed in the following parts.

A. Construction of Dynamic Vehicle Network

In order to construct the dynamic vehicle network, we regard vehicles as nodes in the network, so we get the node set V. For every two nodes (v_i and v_j) and in V, in order to determine whether there is an edge (e_{ij}) between them, we divide the region into many small squares with length and width of 0.001° according to longitude and latitude. The algorithm of calculating positions of vehicles is shown in Algorithm 1. If v_i and v_j in the same small square, there is an edge (e_{ij}) between v_i and v_j, and vice versa.

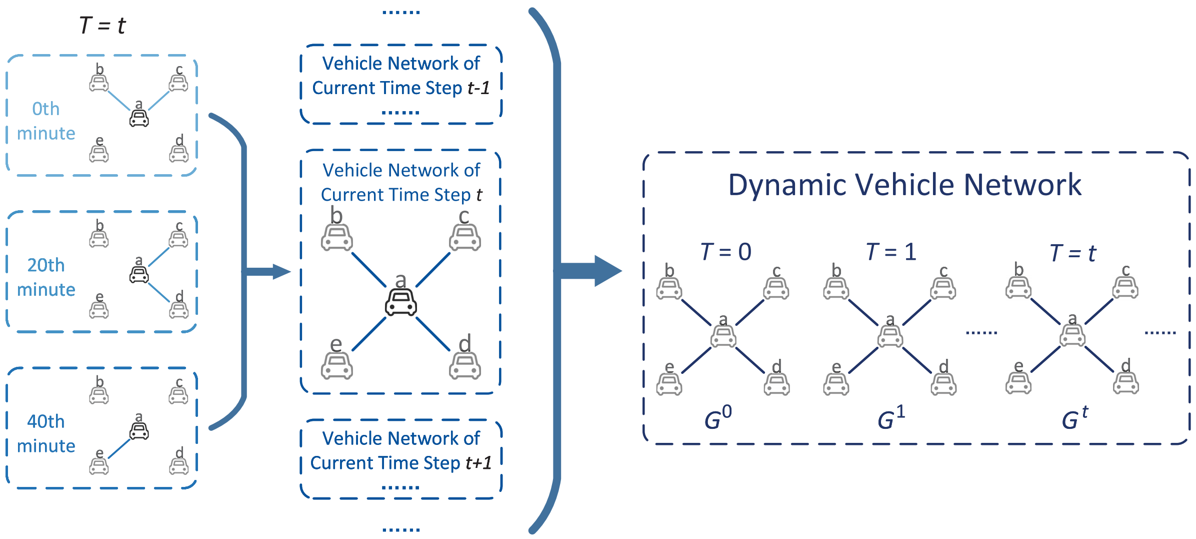

T = t
0th minute
20th minute
40th minute
Vehicle Network of Current Time Step t-1
Vehicle Network of Current Time Step t
Vehicle Network of Current Time Step t+1
Dynamic Vehicle Network
T = 0
T = 1
T = t
G^0
G^1
G^t
a
b
c
d
e

Fig. 2. Formation of the dynamic vehicle network.

Because the locations of vehicles are always changing, the vehicle network is a dynamic network. A dynamic network consists of many snapshots at different time steps $G = (G^0, G^1, \ldots, G^t, G^{t+1}, \ldots)$, where t is the current time step [9]. In this paper, we take one hour as a time step. Considering that the locations of the vehicle within one hour may also be very different, we build the edges for the nodes in the network at the 0-th minute, the 20-th minute, and the 40-th minute in each hour, and keep the edges of these three moments in the current hour's network. The algorithm for constructing the vehicle network is shown in Algorithm 2. The vehicle networks of different time steps form the dynamic vehicle network. This process is shown in Fig. 2.

Algorithm 1 Vehicle Position

```
Require: Vehicle set, $V$; Longitude of vehicles, $Long$; Latitude of vehicles, $Lat$;
Ensure: Positions corresponding to vehicles, $P$;
1: $maxLong = \mathbf{max}(Long)$
2: $minLong = \mathbf{min}(Long)$
3: $minLat = \mathbf{min}(Lat)$
4: Number of squares per row, $n = \mathbf{ceil}((maxLong - minLong) / 0.001)$
5: for $v$ in $V$ do
6:     $P[v] = (Lat[v] - minLat)$ // 0.001 * $n + (Long[v] - minLong)$ // 0.001
7: end for
```

Algorithm 2 Construction of Vehicle Network

```
Require: Vehicle set, $V$; Longitude of vehicles at 0th, 20th, 40th minute, $Long$; Latitude of vehicles at 0th, 20th, 40th minute, $Lat$;
Ensure: vehicle network, $G$;
1: $G$.add_nodes_from($V$)
2: for $minute$ in $list(0, 20, 40)$ do
3:     $P = $ VehiclePosition($Long[minute]$, $Lat[minute]$)
4:     for $v_i$ in V do
5:         for $v_j$ in V do
6:             if $P[v_i] == P[v_j]$ then
7:                 $G$.add_edge($v_i$, $v_j$)
8:             end if
9:         end for
10:    end for
11: end for
```

B. Dynamic Network Representation Learning

After getting the dynamic vehicle network, it is possible to calculate the vector similarity by network-based clustering methods. However, with the increasing size of network, traditional clustering methods cannot be finished in an efficient manner. Recently, with the advance of network representation learning technologies, it is possible to achieve node clustering more efficiently and accurately. Meanwhile, considering the dynamic nature of the vehicular network, it is essential to measure the changing network topology. Inspired by previous work [9], we propose to learn the embedding vectors of vehicles by performing dynamic network representation learning on the previously constructed k-nearest neighbor-based vehicular network.

We name the dynamic network representation learning method as DynWalks. Specifically, the idea of DynWalks is motivated by DeepWalk [20]. DeepWalk is a typical network representation learning method which takes the sequence of nodes obtained by random walk as a sentence, and obtains needed information of the network from the truncated random walk sequence, and then learns the potential representation of the node through some information.

Similarly, DynWalks performs truncated random walks with length l on each selected node for r times. By using a silding window with length $w + 1 + w$ to slide on each random walk sequence, the training pairs set D consisting of $(v_{center}, v_{center+i})$ is formed, where $i \in [-w, +w]$, $i \neq 0$. DynWalks uses the Skip-Gram Negative Sampling (SGNS) model [18] to train node embedding vectors ($Vec \in \mathbb{R}^d$) over each node pair in D, i.e.,

$$\max \log \delta(Z_i \cdot Z_j) + q \cdot E_{v_k \sim P_D}[\log \delta(-Z_i \cdot Z_k)], \quad (1)$$

where q denotes the number of negative sampling number, v_k denotes negative sample from the P_d distribution, δ denotes the Sigmoid Function, vehicle v_i and v_j's representations are denoted as Z_i and Z_j, respectively.

Meanwhile, based on the previous modified DeepWalk, DynWalks adopts an online manner to capture the dynamics of the network, which can be denoted as:

$$Z^t == \begin{cases} f(G^t, SGNS^t_{rand}) & t = 0 \\ f(G^t, G_{t-1}, SGNS^{t-1}) & t \geqslant 1, \end{cases}$$

where $SGNS^{t-1}$ denotes the learning results of last step and Z^t denotes the embedding matrix of time t. In other words, when $t = 0$, DynWalks performs random walks on all nodes of G^0 and obtains the embedding vectors of all nodes. When $t \neq 0$, DynWalks only performs random walks on selected nodes and updates the embedding vecotrs of selected nodes. The embedding vectors of other nodes remains unchanged. Since the new nodes don't have corresponding embedding vectors, they must be selected. Except for new nodes, the number of selected nodes on time step t is $\alpha|V^t|$. Selected nodes contain $\beta\alpha|V^t|$ most affected nodes and $(1-\beta)\alpha|V^t|$ diverse nodes which are generated by random selection [9]. α and β are two hyper-parameters introduced by authors.

It is worth mentioning that the vehicle network on each time step must be a connected network because DynWalks performs random walks on the network. Therefore, after constructing the vehicle network, we should take the maximal connected subgraph.

C. Node Clustering via Machine Learning

After using DynWalks on the dynamic vehicle networks, we get the embedding vectors of each vehicle. In order to find vehicles with similar behavior patterns, various clustering methods can be used to cluster the embedding vectors of nodes. Here, we mainly explore the following clustering algorithms:

- **K-means:** K-means is a classical clustering algorithm proposed in 1967 [16]. It is an iterative clustering analysis algorithm. It randomly selects K nodes as the initial clustering centers firstly, then calculates the distance between each node and each clustering center, and assigns each node to the nearest clustering center. Cluster centers and nodes assigned to them represent different clusters. Every time a node is allocated, the cluster center will be recalculated according to the existing nodes in the cluster. This process will be repeated until a termination condition is met. The termination condition can be that no node is reassigned to different clusters, no cluster center changes again, or the sum of the squared errors is local minimum.
- **GMM:** Gaussian Mixture Model [23] is the extension of the Gaussian model. GMM uses a combination of multiple Gaussian distributions to characterize the data distribution because Gaussian distribution has good mathematical properties and good computing performance. The Gaussian mixture model can be regarded as a model composed of K single Gaussian models, and the K sub-models are hidden variables of the hybrid model. In general, a mixed model can use any probability distribution. In fact, GMM is similar to k-means, but GMM learns some probability density functions. GMM gives the probability that these data points are assigned to each cluster, also known as soft assignment.
- **K-mediods:** K-mediods [19] algorithm is a clustering algorithm based on the partitioning method. To be precise, it is an improved algorithm of the K-means algorithm. The K-mediods algorithm has the ability to process large data sets, and the clusters are quite compact. The distinct advantages between clusters are the same as the K-means algorithm. Compared with K-means, the K-mediods algorithm is less sensitive to noise, so that the outlier will not cause the deviation of the division result to be too large, and a small amount of data will not cause significant impact.

With the help of clustering algorithms, we can fin vehicles with similar behavior patterns. The whole process of our clustering method of dynamic vehicle network is shown Algorithm 3.

Algorithm 3 Clustering of Dynamic Vehicle Network

```
Require: Time range, $T$; Vehicle set over $T$, $V$; Longitude of vehicles during $T$, $Long$; Latitude of vehicles during $T$, $Lat$;
Ensure: Clustering results of dynamic vehicle network, $C_{dyn}$;
1: Dynamic vehicle network $G_{dyn}$ = list()
2: for $t$ in $T$ do
3:     $G$ = Construct Vehicle Network($V[t]$, $Long[t]$, $Lat[t]$)
4:     $G$ = max_connected_component_subgraphs($G$)
5:     $G_{dyn}$.append($G$)
6: end for
7: Node embedding vectors of dynamic vehicle network $E$ = DynWalks($G_{dyn}$)
8: Clustering results of dynamic vehicle network, $C_{dyn}$ = list();
9: for $t$ in $T$ do
10:     $C$ = $K$-means($E[t]$)
11:     $C_{dyn}$.append($C$)
12: end for
```

III. Experimental Setup

In order to show the good performance of our method, we also compare it with different network representation algorithms over three evaluation metrics

A. Datasets

In this paper, we use the taxi dataset of Hangzhou, China, to carry out the experiment. This taxi dataset contains the information of all taxis in Hangzhou from March 1, 2014 to March 31, 2014, including license plate number, GPS time, longitude, and latitude. For simplicity, we only select taxi data

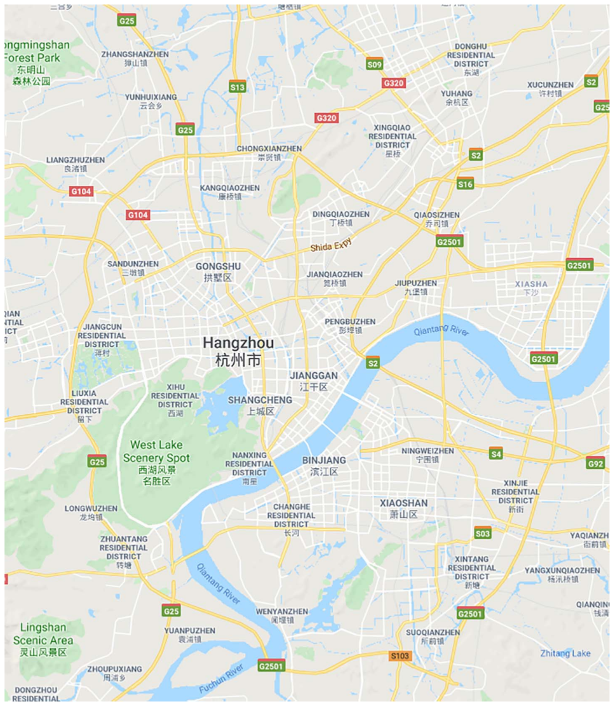


Fig. 3. Selected geographical area of Hangzhou, China.

TABLE I
NUMBER OF NODES AND EDGES PER HOUR IN THE DYNAMIC VEHICLE NETWORK

Hour	#nodes	#edges
8	1786	4370
9	1858	4861
10	1876	4616
11	1879	4856
12	1858	4750
13	1936	4212
14	1933	4539
15	2049	4870
16	1979	5024
17	1923	3654
18	1845	3706
19	1909	4050

from 8 a.m. to 7 p.m. in March 1, 2014, and build the vehicle networks every hour. Due to the large latitude and longitude range of the data set, in order to reduce the sparsity of the network, we choose 30.07°N∼30.47°N, 120.0°E∼120.4°E as the geographic area of the dynamic vehicle network (Fig. 3). It covers most of the urban areas. The number of nodes and edges per hour in the dynamic vehicle network is shown in Table I.

B. Baseline Methods

In order to demonstrate the superiority of this method, we also compare it with two other network representation learning algorithms, including DeepWalk [20] and LINE [22]).

- **DeepWalk** DeepWalk is a typical network representation learning algorithm which is designed by adopting information gained from truncated random walks to learn the low-dimensional representations of nodes.
- **LINE** LINE can retain first- and second-order neighbor information of nodes for representation. It finds the first-order similarity and empirical probability, second-order similarity, and empirical probability of the nodes in the network, respectively, and minimizes their KL divergence as the targets for neural network training, and then combines the outputs as the node's low-dimensional vector representation.

For all the clustering algorithms, the number of clusters k is set from 2 to 50. For simplicity, we only compare the clustering results of each method to the last network (7 p.m.) of the dynamic vehicle network. It means that DeepWalk and LINE only learn the embedding vectors of nodes on the last vehicle network, because they are static network representation algorithms.

C. Evaluation Metrics

We used three metrics to evaluate the clustering results: Silhouette Coefficient (SC) [21], Davies-Bouldin Index (DBI) [3],and Calinski-Harabaz Index (CHI) [2]. Detailed descriptions of these indicators are as follows:

- Silhouette Coefficient: the silhouette coefficient formula of node v_i is:

$$S(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{2}$$

where $a(i)$ is the average distance between node v_i and other nodes in the cluster to which it belongs, $b(i)$ is the minimum value of the average distances between node v_i and all nodes in other clusters. The Silhouette Coefficient of a method is the mean value of the silhouette coefficient of all nodes. Its value range is $[-1, 1]$. The closer it is to 1, the better the clustering result.

- Davies-Bouldin Index: The formula of Davies-Bouldin Index is as follows:

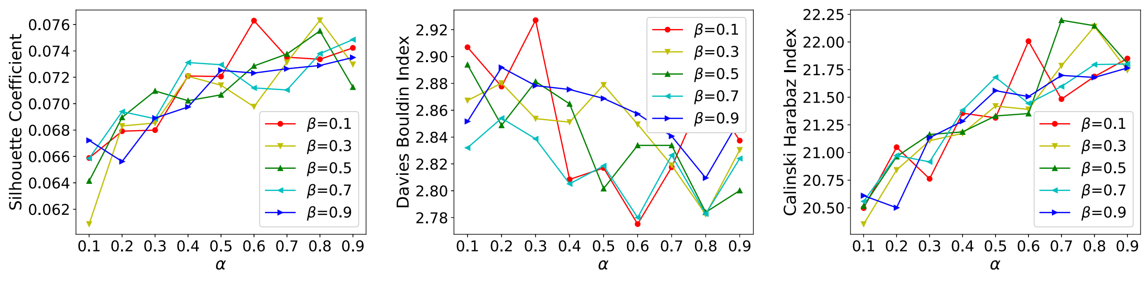
$$DBI = \frac{1}{N}\sum_{i=1}^{N} \max_{j \neq i}\left(\frac{\overline{S_i} + \overline{S_j}}{\|w_i - w_j\|_2}\right) \tag{3}$$

where $\overline{S_i}$ is the average distance of nodes in cluster i to the centroid of cluster i, w_i is the centroid of cluster i, N is the number of clusters. It is the mean value of the maximum value of the ratio of the sum of the average distance within any two categories to the distance between the centers of mass of the two clusters. The smaller the DBI value, the closer the clustering result is with the inner cluster, and the farther the different clusters are separated, the better the clustering result.

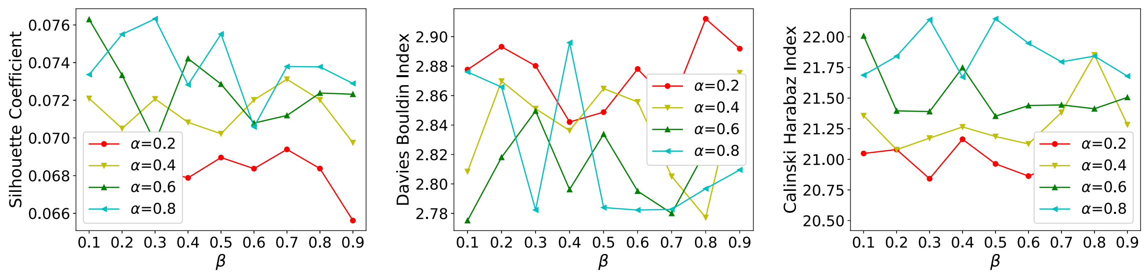
- Calinski-Harabaz Index: The formula of Davies-Bouldin Index is:

$$CHI = \frac{tr(B_k)}{tr(W_k)} \frac{m-k}{k-1} \tag{4}$$

where m is the number of nodes, k is the number of clusters, B_k is the covariance matrix between clusters, W_k is the covariance matrix of data in the cluster, tr is

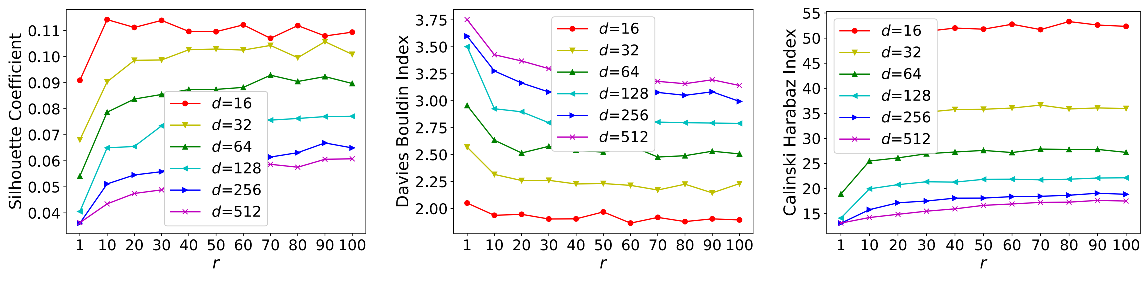


(a) The proportion of selected nodes

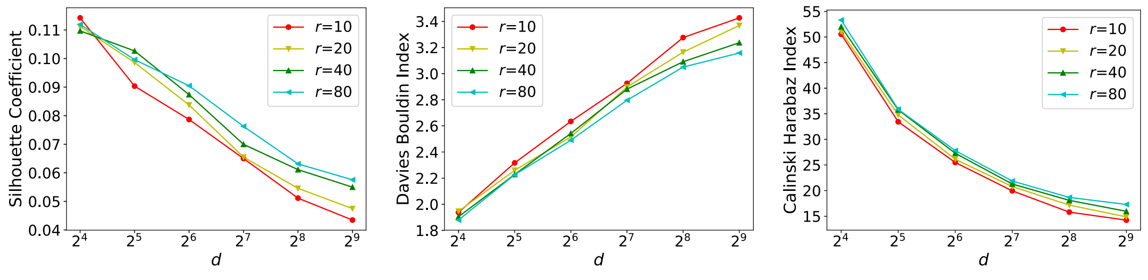


(b) The proportion of affected nodes

Fig. 4. Parameter sensitivity of α and β.



(a) Number of walks



(b) Dimensions

Fig. 5. Parameter sensitivity of r and d.

the trace of matrix. It is the ratio of the between-cluster variance and the within-cluster variance. The larger CHI is, the closer the cluster itself is, the more dispersed the clusters are, the better the clustering results.

IV. Experimental Results

In this section, we show the experimental results from the perspectives of parameter sensitivity, clustering method sensitivity, and accuracy comparison.

A. Parameter Sensitivity Analysis

In order to study how the parameters of DynWalks affect the clustering result, we also carried out experiments by changing the values of hyper-parameters. For simplicity, we fix the walk length ($l = 80$) and window size ($w = 10$) and vary the proportion of selected nodes α, the proportion of affected nodes β, dimensions of embedding vectors d, the number of walks on per node r to study the effects of parameters.

Fig. 4(a) shows the effects of increasing the proportion of selected nodes to our model. Although the curve of Davies-Bouldin Index is a little concussion, generally speaking, with the increase of α, the clustering result will be better. It's consistent with our intuition. The increase of α means that we will perform random walks on more nodes. So the clustering result will be better than random walks on fewer nodes.

The effect of increasing β is as shown in Fig. 4(b). When we increase β, the clustering result curves show a fluctuating state,

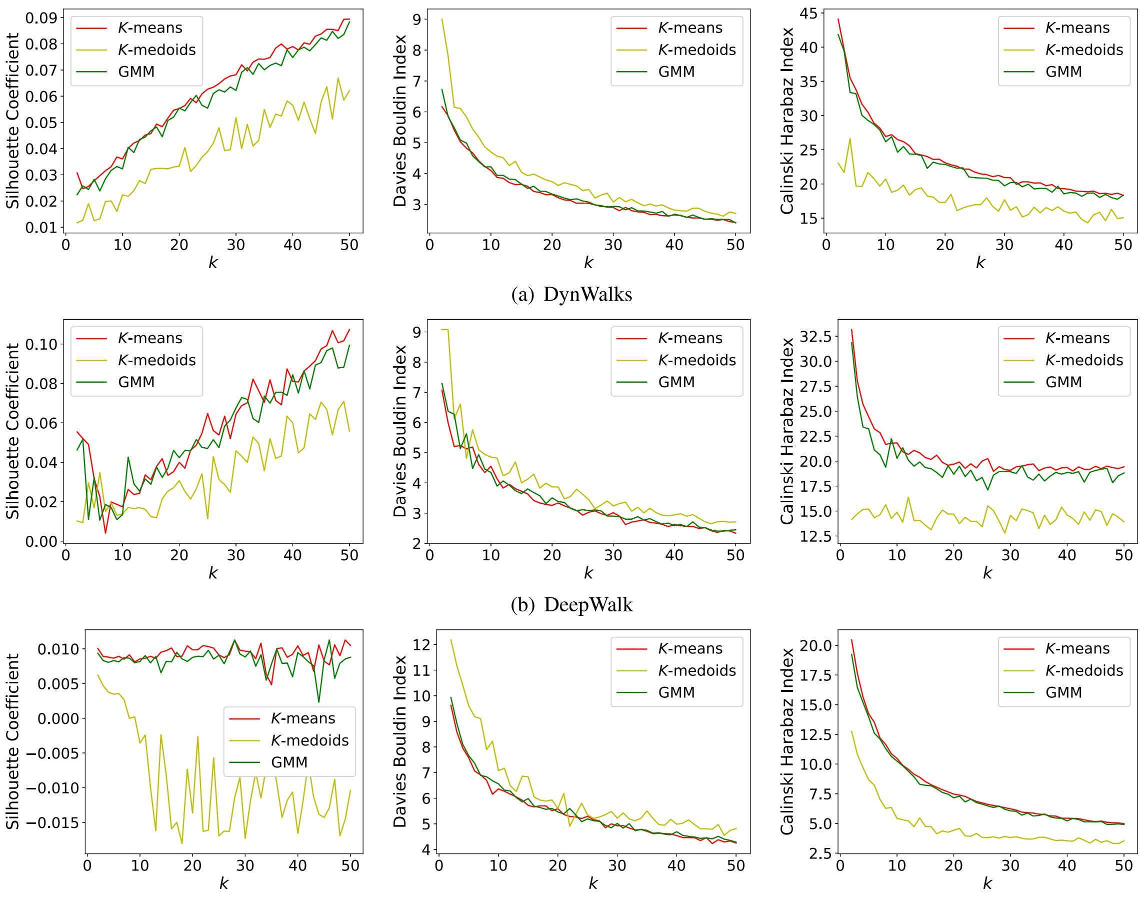


Fig. 6. Performance of clustering algorithms in different network representation learning algorithms.

and there is no obvious upward or downward trend. Therefore, the influence of β on the clustering result is uncertain. We need to find a suitable β by tuning parameters.

B. Clustering Algorithms Sensitivity

Fig. 5(a) shows the effects of increasing the number of walks on nodes to our model. We found that clustering result does not increase with the increase of r. When r increases to a certain value, the clustering result is the best. For example, when the dimensions d equals to 16 and r equals to 10, the clustering result is the best. If we continue to increase r, the clustering performance will not be significantly improved. This trend is consistent over different d.

The clustering result of increasing the dimensions of the embedding vector is shown in Fig. 5(b). It reveals that there has been a steady decrease in the clustering result when we increase d. This trend is consistent on different r. It means that a small embedding vector dimension is more suitable for the dynamic vehicle network.

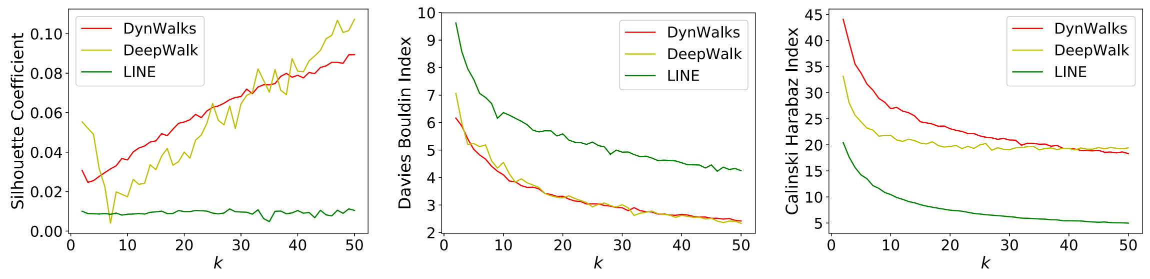
In order to verify the clustering performance of the K-means algorithm used in the vehicle network, we compare it with K-medoids and Gaussian Mixture Model (GMM). At the same time, in order to avoid contingency, we use the embedding vectors learned by DynWalks, DeepWalk, LINE in these clustering algorithms. The comparison of clustering result is shown in Fig. 6. Red line is the performance of K-means algorithm we choose in our method.

Overall, K-means outperforms K-medoids in all three metrics and all three network representation learning algorithms. Compared with GMM, K-means performs better in some k values, and shows competitiveness in other k values. It shows that the K-means is more suitable for the dynamic vehicle network.

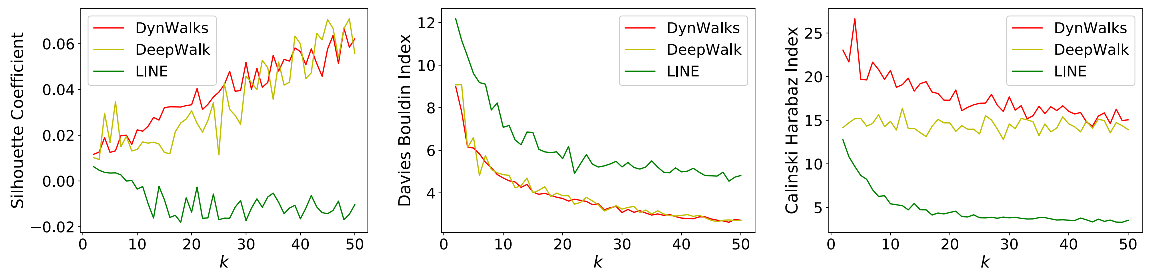
C. Accuracy Comparison

In order to verify the performance of the network representation learning algorithm we used, we compare DynWalks with DeepWalk and LINE on the same vehicle network. They use the same parameters: $r = 20$, $l = 80$, $w = 10$, $d = 128$. In addition, α and β of dynWalks are set to 0.2 and 0.5, respectively. We use three different clustering algorithms to cluster the embedded vectors generated by them. The clustering result is shown in Fig. 7. Red line is the performance of DynWalks we use.

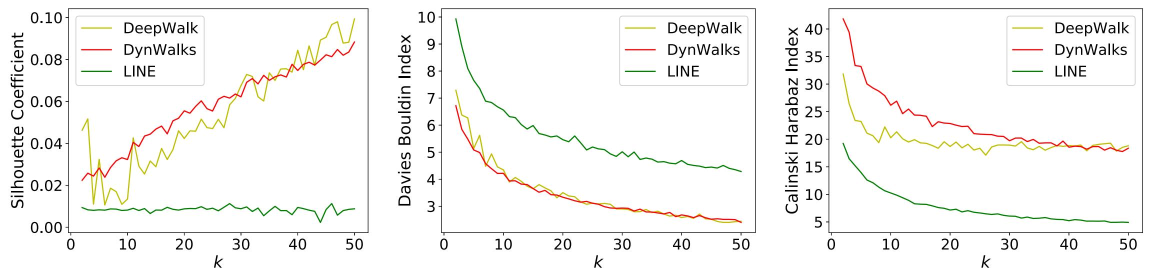
As can be seen from Fig. 7, DynWalks performs better than LINE in all three metrics and all three clustering algorithms. Compared to DeepWalk, DynWalks performs competitive performance in Davies-Bouldin Index. And its performance in Calinski-Harabaz Index is better than DeepWalk. In Silhouette Coefficient, when k is less than 5 or more than 39, DeepWalk



(a) K-means



(b) K-medoids



(c) Gaussian Mixture Model

Fig. 7. Performance of network representation learning algorithms in different clustering algorithms.

is better than DynWalks. While in other k values, DynWalks performs better.

In addition, DeepWalk and LINE need to learn embedding vectors of all nodes in the vehicle network, because they are static network representation learning algorithms. However, DynWalks only needs to learn the embedding vectors for partial nodes in each time t when $t > 0$. This means that DynWalks has less time complexity. Therefore, DynWalks is better than DeepWalk and LINE in the dynamic vehicle network.

V. Related Works

In this section, we review the related works from the perspectives of vehicle clustering and network representation learning.

A. Vehicle Trajectory Clustering

In the current context of big data, trajectory data mining [17], [31] is significant to explore human movement patterns and activity patterns. Exploring the underlying knowledge of trajectory data is an important way to solve urban problems. With the widespread application of mobile internet, handheld mobile devices, satellite navigation systems, and geographic information systems, and the continuous improvement of data collection efficiency, scholars are now able to obtain a large amount of spatial trajectory data, which represents mobile objects such as people, vehicles, and animals. In recent years, trajectory data mining has become a research hotspot in data mining. These trajectory data can be used to extract relevant information in order, for instance, to calculate the optimal path from position A to position B, detect abnormal behavior, monitor the traffic flow in a city, predict the next position of an object, etc.

Trajectory data mining tasks mainly include trajectory data preprocessing, trajectory data management, trajectory pattern mining, abnormal trajectory detection, and trajectory classification. Meanwhile, vehicle trajectory data mining is receiving more and more attention from both academia and industry. For example, Kong *et al.* [11] propose a time-location-relationship combined taxi service recommendation model to improve taxi drivers' profits. Their model IS designed to acquire passenger volume, mean trip distance, and average trip time in functional regions during every period on weekdays and weekends by taking advantage of the Gaussian process regression and statistical approaches. So that their method allows drivers to pick up more passengers within a short time frame.

One of the simplest and most powerful methods to obtain knowledge from trajectory data is trajectory clustering [4]. In general, trajectory clustering is based on similarity measures between trajectories. Vehicle clustering is one of the main tasks of trajectory clustering. Vehicle clustering aims to regroup similar vehicles together into groups that are different from one another. The basic foundation of vehicle clustering is the vehicle similarity. However, existing similarity measurements can not well utilize specific vehicles' trajectories, which are merely based on distance [29].

B. Network Representation Learning

Traditional network analysis methods based on graph structure have the problems of high computational complexity and low parallelism. Meanwhile, they cannot be applied to machine learning methods. In order to solve these problems brought by traditional network analysis methods, learning the low-dimensional representation of nodes in the network is a new solution. In the network embedded space, the initial relationships between nodes, such as edges or other higher-order topological metrics are transformed into a distance in the vector space. The topology and structural properties of the nodes are encoded as embedded vectors. At the same time, traditional network representation directly uses the observed adjacency matrix, which often contains noise and redundant information. Network embedding first learns the dense and continuous representation of the network in low-dimensional space so that noise and redundant information can be reduced [6], [23].

By representation learning large-scale complex networks, not only can the problem of network data sparseness be effectively solved, but also different types of heterogeneous information in the network can be merged into the same matrix, which can more effectively solve certain network-based specific problems. The purpose of network representation learning is to map the structural characteristics of any node in the network to a continuous dense vector at low latitude. In this process, the node information and network topology information in the network can be well retained. It can effectively solve some network-based issues such as link prediction, vertex classification, personalized recommendation, and community detection.

Typical network representation learning methods include structure preserving network embedding and side information enhanced network representation learning. Structure preserving network embedding methods include DeepWalk [20], Node2vec [7], and LINE [22]. DeepWalk [20] learns the representation of nodes in the network while preserving the neighbor structure of the nodes. The algorithm finds that the distribution of nodes in short random walks is similar to the distribution of words in NLP. Node2vec [7] is proposed by pointing out that DeepWalk cannot capture the diverse connection modes in the network well. It defines a flexible concept of node neighbors, and combines depth-first sampling and breadth-first sampling, and uses second-order random walks to sample neighbor nodes. LINE [22] is suitable for large-scale network embedding problem which can retain the first-order and second-order approximations of nodes at the same time. First-order approximation is the node pair relationship that can be directly observed, such as side information. Second-order approximation consists of two nodes' context (neighbor) similarity.

Side information enhanced network representation learning methods try to combine the labels, attributes, and even the semantics of nodes to perform representation learning. Some typical methods include TADW [27], DANE [5], and STNE [14]. TADW [27] is a framework that can combine features obtained from two different types of information. Specifically, it can incorporate text features of vertices into network representation learning. DANE [5] can capture the high nonlinearity and preserve various proximities in both topological structure and node attributes based on a deep model. STNE [14] is a novel sequence-to-sequence model-based network representation learning framework that can learn node representation with a content-to-node seq2seq model. Considering the superiority of network representation learning in dealing with large-scale complex networks, we adopt it as the basic framework for our proposed vehicle clustering task.

VI. Conclusion

Mining and analyzing various trajectory data is important in understanding human mobility patterns and further constructing smart cites. Among various trajectory data, vehicle trajectory data have drawn extensive attention from both academia and industry. A typical way of mining these trajectory data is clustering, i.e., vehicle trajectory clustering. To better meet the requirement of dealing with large scale network-based trajectory data, this paper proposes a novel vehicle trajectory clustering method based on dynamic network representation learning. Our proposed methods can not only utilize the network topology of vehicle networks but also capture the dynamic characteristic of the vehicle network. In addition, our method relies on machine learning methods for clustering without using manually designed metrics, which can avoid biased results.

Nevertheless, in order to cluster vehicle trajectory, we have constructed a k-near neighbor based internet of vehicles, which should be further explored in the future. Other network construction methods should be evaluated for comparison. Meanwhile, due to the limitation of data acquisition, we merely run experiments on one dataset. We will further explore the universality of the proposed method upon other accessible datasets.

References

[1] P. C. Besse, B. Guillouet, J.-M. Loubes, and F. Royer, "Review and perspective for distance-based clustering of vehicle trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3306–3317, Nov. 2016.

[2] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Statist.-Simul. Comput.*, vol. 3, no. 1, pp. 1–27, Jan. 1974.

[3] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

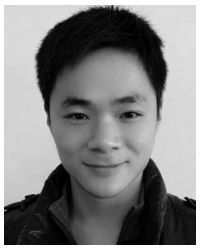
[4] Z. Fu, W. Hu, and T. Tan, "Similarity based vehicle trajectory clustering and anomaly detection," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Sep. 2005, p. II-602.

[5] H. Gao and H. Huang, "Deep attributed network embedding," in *Proc. IJCAI*, vol. 18, Jul. 2018, pp. 3364–3370.

[6] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl.-Based Syst.*, vol. 151, pp. 78–94, Jul. 2018.

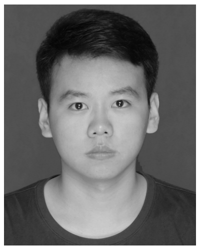
[7] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[8] B. Han, L. Liu, and E. Omiecinski, "Road-network aware trajectory clustering: Integrating locality, flow, and density," *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, pp. 416–429, Feb. 2015.
[9] C. Hou, H. Zhang, K. Tang, and S. He, "DynWalks: Global topology and recent changes awareness dynamic network embedding," 2019, *arXiv:1907.11968*. [Online]. Available: http://arxiv.org/abs/1907.11968
[10] X. Kong *et al.*, "Big trajectory data: A survey of applications and services," *IEEE Access*, vol. 6, pp. 58295–58306, 2018.
[11] X. Kong, F. Xia, J. Wang, A. Rahim, and S. K. Das, "Time-location-relationship combined service recommendation based on taxi trajectory data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1202–1212, Jun. 2017.
[12] D. Kumar, H. Wu, S. Rajasegarar, C. Leckie, S. Krishnaswamy, and M. Palaniswami, "Fast and scalable big data trajectory clustering for understanding urban mobility," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 11, pp. 3709–3722, Nov. 2018.
[13] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: A partition-and-group framework," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2007, pp. 593–604.
[14] J. Liu, Z. He, L. Wei, and Y. Huang, "Content to node: Self-translation network embedding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1794–1802.
[15] Y. Liu, C. Kang, S. Gao, Y. Xiao, and Y. Tian, "Understanding intra-urban trip patterns from taxi trajectory data," *J. Geograph. Syst.*, vol. 14, no. 4, pp. 463–483, Oct. 2012.
[16] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Oakland, CA, USA, 1967, vol. 1, no. 14, pp. 281–297.
[17] J. D. Mazimpaka and S. Timpf, "Trajectory data mining: A review of methods and applications," *J. Spatial Inf. Sci.*, no. 13, pp. 61–99, Dec. 2016.
[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[19] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3336–3341, Mar. 2009.
[20] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.
[21] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.
[22] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.
[23] W. Wang, J. Chen, J. Wang, J. Chen, and Z. Gong, "Geography-aware inductive matrix completion for personalized point-of-interest recommendation in smart cities," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4361–4370, May 2020.
[24] W. Wang, J. Chen, J. Wang, J. Chen, J. Liu, and Z. Gong, "Trust-enhanced collaborative filtering for personalized point of interests recommendation," *IEEE Trans. Ind. Informat.*, early access, Dec. 9, 2019, doi: 10.1109/TII.2019.2958696.
[25] F. Xia, J. Wang, X. Kong, D. Zhang, and Z. Wang, "Ranking station importance with human mobility patterns using subway network datasets," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 14, 2019, doi: 10.1109/TITS.2019.2920962.
[26] F. Xia, A. Rahim, X. Kong, M. Wang, Y. Cai, and J. Wang, "Modeling and analysis of large-scale urban mobility for green transportation," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1469–1481, Apr. 2018.
[27] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Chang, "Network representation learning with rich text information," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 1–7.
[28] D. Yi, J. Su, C. Liu, and W.-H. Chen, "Trajectory clustering aided personalized driver intention prediction for intelligent vehicles," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3693–3702, Jun. 2019.
[29] Q. Yu, Y. Luo, C. Chen, and S. Chen, "Trajectory similarity clustering based on multi-feature distance measurement," *Int. J. Speech Technol.*, vol. 49, no. 6, pp. 2315–2338, Jun. 2019.
[30] G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang, "A review of moving object trajectory clustering algorithms," *Artif. Intell. Rev.*, vol. 47, no. 1, pp. 123–144, Jan. 2017.
[31] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, p. 29, 2015.



Wei Wang received the B.Sc. degree from Shenyang University, Shenyang, China, in 2012, and the Ph.D. degree in software engineering from the Dalian University of Technology, Dalian, China, in 2018. He is currently a Postdoctoral Research Fellow with the University of Macau, Macau, and an Assistant Researcher with the Dalian University of China, Dalian, China. His research interests include big scholarly data, social network analysis, and computational social science.

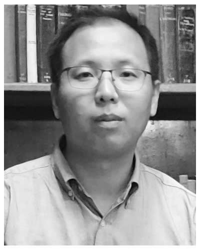
Feng Xia (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China. He is currently an Associate Professor with the Data Science and Discipline Leader (Information Technology), School of Science, Engineering, and Information Technology, Federation University Australia. He has authored two books and over 300 scientific articles in international journals and conferences. His research interests include data science, knowledge management, social computing, and systems engineering. He is a Senior Member of ACM.



Hansong Nie received the B.Sc. degree in electronic information engineering from Dalian Maritime University, Dalian, China, in 2018. He is currently pursuing the master's degree with the School of Software, Dalian University of Technology, Dalian. His research interests include big scholarly data, social network analysis, and science of success.

Zhikui Chen (Member, IEEE) received the B.E. degree in mathematics and the Ph.D. degree in solid mechanics from Chongqing Normal University, Chongqing, China, in 1990 and 1998, respectively. He is currently a Professor with the Dalian University of Technology, Dalian, China. His research interests include the Internet of Things and big data.

Zhiguo Gong (Senior Member, IEEE) received the M.Sc. degree from Peking University, Beijing, China, in 1988, and the Ph.D. degree from the Department of Computer Science, Institute of Mathematics, Chinese Academy of Science. He is currently a Professor and the Head of the Department of Computer and Information Science, University of Macau, Macau. His research interests include machine learning, data mining, database, and information retrieval.



Xiangjie Kong (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China. He is currently a Full Professor with the College of Computer Science and Technology, Zhejiang University of Technology. Previously, he was an Associate Professor with the School of Software, Dalian University of Technology, China. He has published over 100 scientific articles in international journals and conferences (with over 70 indexed by ISI SCIE). His research interests include network science, data science, and computational social science. He is a Senior Member of CCF and a member of ACM.

Wei Wei (Senior Member, IEEE) received the M.S. and Ph.D. degrees from Xi'an Jiaotong University in 2011 and 2005, respectively. He is currently an Associate Professor with the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. His research interests are in the area of wireless networks, wireless sensor networks applications, image processing, mobile computing, distributed computing, and pervasive computing, the Internet of Things, and sensor data clouds. He is a Senior Member of CCF.