IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 17, 2019, accepted December 30, 2019, date of publication January 17, 2020, date of current version January 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966995

Dynamic Path Planning Algorithms With Load Balancing Based on Data Prediction for Smart Transportation Systems

NING SUN[1], HUIZHU SHI[1], GUANGJIE HAN[2,3], (Senior Member, IEEE), BIN WANG[1], AND LEI SHU[2,4], (Senior Member, IEEE)

[1]College of Internet of Things Engineering, Hohai University, Changzhou 213022, China
[2]College of Engineering, Nanjing Agricultural University, Nanjing 210095, China
[3]College of Information and Communication System, Hohai University, Changzhou 213022, China
[4]School of Engineering, University of Lincoln, Lincoln LN6 7TS, U.K.

Corresponding author: Guangjie Han (hanguangjie@gmail.com)

This work was supported in part by the National Key Research and Development Program under Grant 2017YFE0125300, in part by the Fundamental Research Funds for the Central Universities under Grant 2018B23514 and Grant 2018B740X14, in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX18_0551, in part by the National Natural Science Foundation of China-Guangdong Joint Fund under Grant U1801264, and in part by the Jiangsu Key Research and Development Program under Grant BE2019648.

ABSTRACT In modern transportation, traffic congestion has become an urgent problem in large and medium-sized cities. In smart transportation systems, it is an effective solution to design load balancing path planning algorithms that can dynamically adapt to traffic conditions in order to avoid congestion. In this work, a traffic path planning algorithm based on data prediction (TPPDP) is proposed to find the path with the shortest travel time, which is built on a predictive model based on historical traffic data and current traffic information. Furthermore, a path planning algorithm based on data prediction with load balancing (TPPDP-LB) is also proposed, which combines the predicted information and the number of concurrent requests to achieve the path with shortest travel time while maintaining global load balancing. A specific distributed computing framework for TPPDP-LB algorithm is designed to reduce the runtime of the algorithm. The simulation results proved that both TPPDP and TPPDP-LB algorithms have the advantage of shortest travel time, and TPPDP-LB algorithm achieves load balancing of computing. It is also proved that the distributed computing framework designed for TPPDP-LP algorithm can effectively reduce the runtime of system as well as keep the accuracy of algorithm.

INDEX TERMS Path planning algorithm, data prediction, load balancing, distributed computing, smart transportation.

I. INTRODUCTION

With the rapid development of economy, the transportation industry has made remarkable progress and the number of vehicles has also increased dramatically. Due to the restriction of land resources, traffic problems in large and medium cities are continuously exacerbated, which is manifested in serious traffic congestion and frequent traffic accident [1] and causes the waste of resources, environment pollution and huge economic loss to society [2]. Because the urban road resources are very limited, improving road utilization is an effective way to cope with the growing requirements of transportation, in addition to building new transport infrastructures. However, in real life, road utilization is often unsatisfactory due to the lack of information that can help drivers make correct and accurate choice. Therefore, it is meaningful to predict real-time traffic situation, and the predicted data can provide the drivers with effective and timely congestion information so that to obtain the optimal path with shortest travel time [3].

The existing path planning algorithms include classical and dynamic algorithms. The classical shortest path planning algorithms, such as Dijkstra [4] and the A* algorithm [5], take the current state of road segments into consideration,

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

e.g., the travel time is usually taken as the weight of the road segment. The dynamic path planning algorithms consider more influence factors, which usually use big data prediction technology to analyze the characteristics of traffic and predict future situation of roads to get the optimal path [6]. Dynamic path planning algorithms often consider special situations, e.g., how to plan a path with the shortest travel time under the premise of meeting a prescribed [7], how to rearrange a path quickly in the case of deviating from the original path [8], and how to protect the source location privacy in the routing for industrial internet of things (IIoTs) [9]. However, all the existing traffic path planning algorithms have not taken the problem of concurrent requests into consideration, which brings two problems:

1) The load balance of the whole traffic network will be affected, for when a certain road segment is recommended too many times at the same time, its optimality will be lost.
2) The response time will be affected, for a large number of concurrent requests cause a big challenge to the runtime of the traffic path planning system.

Based on the above, a traffic path planning algorithm based on data prediction (TPPDP) is proposed in this paper, in order to achieve the optimal path with the shortest travel time. Also, from the viewpoint of load balancing, an improved path planning algorithm and a specific distributed big data processing framework is designed to improve the processing speed of the system, because the common distributed computing framework cannot meet the parallel processing requirements of the algorithm. The main works of this paper are listed as follows:

1) A model for traffic information prediction is designed to predict the traffic state of the road segments in the future based on historical traffic data and current traffic information, and a traffic path planning algorithm based on data prediction (TPPDP) is proposed to arrange a path with the shortest travel time.
2) A traffic path planning algorithm based on data prediction with load balancing (TPPDP-LB) is proposed, which combines the predicted information and the recommended number of the road segments to plan a path with the shortest travel time while keeping global load balancing.
3) A distributed computing framework is designed for running TPPDP-LB algorithm to respond to a large number of concurrent path planning requests quickly. This distributed processing framework is carried out through simulation.

The results of simulation show that TPPDP algorithm has the advantage of shortest travel time while TPPDP-LB algorithm also achieves load balancing, which can effectively avoid congested sections in the future. It is also proved that the distributed computing framework designed for TPPDP-LB can improve the system operation in terms of big data processing and response time.

The rest of the paper is organized as follows: the related works are introduced in Section II. Section III introduces two proposed path planning algorithms in details: TPPDP and TPPDP-LB algorithm. The distributed computing framework designed for TPPDP-LB algorithm is introduced in Section IV. The simulation is implemented in Section V and the results are analyzed and discussed in terms of the performance of the proposed algorithms and the distributed computing framework. Finally, conclusions are drawn in Section VI.

II. RELATED WORKS

A. TRADITIONAL PATH PLANNING ALGORITHMS

Traditional path planning problems are often simplified to the shortest path finding problem, which aims at find out the shortest path (SP) between two nodes in a graph. The classical shortest path planning algorithms include Dijkstra [4], A-Star (A*) [5] and their family algorithms.

Dijkstra algorithm is used to find out the shortest path between two nodes in a graph. The algorithm flow is described as follows. Firstly, a directed graph is constructed. Starting from a current node, a shortest path is found out from this node to one of its neighboring nodes, taking the length of the path as the weight of each neighboring node. Then the node with minimum weight value is selected as the next hop node and the shortest path from it to its neighbor nodes is chosen. Repeat this step, until the shortest path to the destination is determined.

Wang [10] proposed an improved Dijkstra algorithm, redefining the weight value between two nodes and introducing the concept of equivalent path instead of the actual path which can achieve adaptive improvement of Dijkstra algorithm. However, this paper is applied to find the best evacuation route when a mine is flooded and the weight value between nodes is calculated based on the length, type and condition of the roadways, which cannot be applied to modern transportation system.

Wei *et al.* [11] proposed an improved Dijkstra algorithm, which introduces the concept of congestion weight function which only considers the road conditions at the current time and not realizes real-time performance.

A* algorithm is a popular heuristic search algorithm. It estimates the superiority of the nodes by using function. The innovation of the algorithm is that a known global variable is introduced when selecting the next node, which estimates the distance between the current node and the destination. The global variable is used as a measure to evaluate the possibility of the node on the optimal path, which leads to search for more likely nodes, so that to increase the efficiency of the search process.

Li *et al.* [12] proposed an improved A* search algorithm. Through inverse search and optimized evaluation function, the undirected search is transformed into directed search and the global evaluation is transformed into local evaluation, so that to improve the efficiency of the algorithm and make it more suitable for processing large scale path planning problems. Shi *et al.* [13] took the distance between the head and the end nodes as the cost function of the element instead of the direction, and the algorithm improves the validity of A*.

Classical path planning algorithms are usually used in graph theory to find the shortest path in the graph, which cannot be directly applied to the actual complex traffic environments.

B. DYNAMIC PATH PLANNING ALGORITHMS

With the increasing of real-time traffic information, dynamic spatial networks are getting popular and path planning in dynamic spatial networks is becoming more and more important. Dynamic route planning problem (DRPP) aims at solving the optimization problem of path planning for vehicles traveling from a given source point to given destination. Although path planning algorithms have been widely studied, it is not enough for dynamic route planning in real life, for most of the route planning algorithms are designed by finding the SP. However, SP algorithm is ineffective for dynamic route planning, i.e., due to the difference and complexity of road conditions, the travel time calculated by SP algorithm might not equal to the overall path with shortest time (ST).

In recent years, dynamic path planning algorithms are often applied in special application scenario. He *et al.* [14] and Zhang *et al.* [15] took experiential routes collected from taxis into consideration, and proposed a route planning algorithm that best fits the driver's experience based on the fact that the driver's wisdom is always hidden in a large amount of taxis GPS data. Similarly, Luo *et al.* [16] and Wei *et al.* [17] also proposed a popular path planning algorithm based on taxi trajectory aimed at providing the most frequent route. Han *et al.* [18] proposed a novel charging path planning schema for wireless rechargeable sensor networks, which takes the energy of nodes into consideration to reduce the number of dead nodes and prolong the network lifetime.

Considering that different drivers have different driving preferences, i.e., some prefer to time-efficient driving and the others prefer to fuel-efficient driving, Dai *et al.* [19] proposed a personalized route recommendation algorithm to meet the different preferences of drivers.

Kanoh [20] proposed a dynamic path planning algorithm which is based on the genetic algorithm, and uses the latest recorded travel time as a prediction of travel time at a certain time in the future.

Li *et al.* [21] proposed an optimal route queries algorithm that can deal with the arbitrary order constraints. It uses two different methodologies, namely backward search and forward search, to achieve optimal path query efficiently with the order constraints.

Sun *et al.* [22] proposed a robust optimal path selection method, which mainly considers the reliability of the road section. It aims to find a relatively stable path to ensure that the arrival time is within a certain range.

Sommer *et al.* [23] evaluated a traffic information system based on vehicle-to-infrastructure communications that provide information about traffic jams for vehicles and help to optimize route planning.

Li *et al.* [7] defined a path discovery problem named temporal-sensitive optimal route discovery (TORD), and they aimed to find an optimal route that covers a set of user-specified locations and satisfies the specified sorting constraints and time constraints.

Liu *et al.* [8] proposed spatio-temporal outliers (STO) trees, an algorithm for discovering spatiotemporal anomalies and their causal relationship. Similarly, Shang *et al.* [24] proposed a dynamic shortest path monitoring algorithm in spatial network. When the cost of driving changes or the route deviates from the planned route, the algorithm can update the path efficiently using the pre-planned route.

Most the above algorithms aim at solving specific problems in certain scenarios, not the path planning problem with shortest time which is the most fundamental demand in real traffic environment. Furthermore, these algorithms do not take the impact of the recommended number of road segment for consideration. When a certain road segment is recommended too many times at the same time, its optimality will be lost, for in the following time this section may be blocked because of too many recommendations. Therefore, the research of dynamic path planning algorithm in the actual traffic environment is still significant.

C. PREDICTION METHODS OF TRAFFIC INFORMATION

For dynamic path planning algorithm, it is meaningful to predict real-time traffic situation, and the forecast data can provide the driver with effective and timely congestion information to obtain the optimal path which usually means the path with the shortest travel time. Short-term traffic prediction refers to estimating basic parameters of traffic within 15 minutes in the future. It is the key to realizing intelligent transport system (ITS) control and induction. Vlahogianni *et al.* [25] have proposed many solutions to deal with parameter prediction problems of short-term traffic flow. However, the performance of prediction is disturbed by various random conditions, such as weather, human activities, etc. Besides, short-term traffic flow has its own random and fuzzy pattern. Based on the above reasons, the problem of short-term traffic forecasting cannot be completely solved [26].

There are two kinds of short-term traffic state prediction schemes. The first type is parameter prediction method, which is based on a certain mathematical model. It includes linear regression, time series and kalman filter model prediction methods, etc. Linear regression method analyzes historical data to find out the functional relationship between independent variables and dependent variables.

Fontanelli *et al.* [27] proposed a dynamic route planning algorithm based on three future travel estimation. They used the combination of the latest travel time and the number of vehicles currently in a road segment as a prediction of future travel time.

Kong *et al.* [28] proposed a TLR model which uses Gaussianprocess regression and statistical approaches to predict traffic information including average travel time and so on.

Guo *et al.* [29] proposed a prediction algorithm including an adaptive Kalman recursion for stochastic short-term traffic flow rate prediction.

In view of various traffic conditions, Fei *et al.* [30] proposed an adaptive dynamic linear prediction model, which can identify traffic conditions and use real-time data to adjust parameters to achieve accurate prediction. However, this kind of method cannot reflect the non-linearity and uncertainty in traffic prediction and cannot avoid the influence of random factors [31].

The second type is the nonparametric prediction method, which is a knowledge-based intelligent prediction method. Its structure is not predefined compared with the parameter prediction method. In addition, this method relies on the relationship between related or non-related business variables and is regarded as a dynamic classification model with strong non-linear processing ability. Among them, artificial neural network has attracted much attention due to its superior prediction performance.

Kong *et al.* [32] proposed a congestion prediction method based on support vector machine to predict traffic flow parameters including traffic volume and average traffic speed. The particle swarm optimization sub-module is used to optimize the punish coefficients and the parameters.

Kang *et al.* [33] considers that the traffic flow, speed and occupancy of existing and neighboring stations can be used as input to improve traffic flow prediction performance. They used the long short-term memory (LSTM) model to analyze the impact of different input settings on traffic information prediction performance and proposed a traffic information prediction algorithm with LSTM recurrent neural network.

Extreme learning machine (ELM) is an efficient learning paradigm for a wide range of fields. Xing *et al.* [34] used the kernel function method instead of the hidden layer method to overcome the variation problem and proposed a traffic flow prediction method to improve the accuracy of traffic flow prediction.

Due to the complexity of the transportation system, it may be interfered by random factors, such as random weather, traffic events, etc. These uncertain factors increase the uncertainty of traffic, and it is difficult to find a suitable mathematical model to describe the traffic characteristics. The k-nearest neighbor (KNN) model is one of the most commonly used methods in nonparametric regression. Cai [35] and Su [36] both used KNN-based methods to predict traffic flow. The results showed that compared with the other prediction model, the model could improve the prediction accuracy and avoid the influence of subjective classification in traffic state forecasting.

Due to the excellent performance in traffic prediction, the KNN-based model has been chosen as the prediction model in this paper. Also, the input parameters of KNN is selected in this paper to improve the accuracy of prediction.

III. TPPDP: TRAFFIC PATH PLANNING ALGORITHM BASED ON DATA PREDICTION

A. NETWORK MODEL

In this paper, the actual traffic road is abstracted as a graphic structure and referred to as a road network. The definition used in this paper is as follows:

Definition 1: Road network: As shown in Figure 1, the urban road network is defined as a directed graph $G = (V, E)$, where V is a set of vertices, representing the starting node of the road or the road intersection based on the real road situation, and $E \subseteq v \times v$ is a set of edges between two vertices with direction.

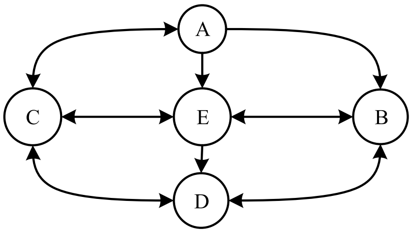


FIGURE 1. Directed graph for road network.

Definition 2: Road segment: a continuous road between two intersections in the road network. For example, $\overrightarrow{AB}$, $\overrightarrow{CD}$ and $\overrightarrow{DC}$ in Figure 1 all represent road segment.

Definition 3: Path: $P = (r_1, r_2, \ldots, r_i)$, the sequential set of road segments, indicates the road segments that a traveler will pass through in sequence, where $r \subseteq E$. As shown in Figure 2, $P_{CB} = \{(\overrightarrow{CB}), (\overrightarrow{CA}, \overrightarrow{AB}), (\overrightarrow{CD}, \overrightarrow{DB})\}$.

Definition 4: Index$_T$: Tral $= \{Seg_i | i\epsilon(1, 2, \ldots, n)\}$, that represents all possible road segments of the path that planned for a traveler.

B. OVERALL DESIGN

The overall process of TPPDP algorithm is shown in Figure 2. The users initiate path planning requests to the system, and the input value is the departure place and the destination. Once receiving parallel data requests, the system sequentially executes the data prediction algorithm and path planning algorithm. The data prediction algorithm runs based on the current traffic data an historical data, and finally computes the predicted average travel speed of each road segment. Then the travel time of the road segment is regarded as the selection criterion of the road segment in the path planning algorithm. The travel time of each segment is calculated based on the segment speed, which is determined by two factors: one is the predicted average travel speed, and the other is the recommended times of each road segment which is based on the path planning results at the same time. The optimal path planning result is the path with shortest travel time.

TPPDP algorithm studies the dynamic and global path planning algorithm, combines the path planning algorithm with traffic prediction technology. In addition, considering the overall coordination of the road network,

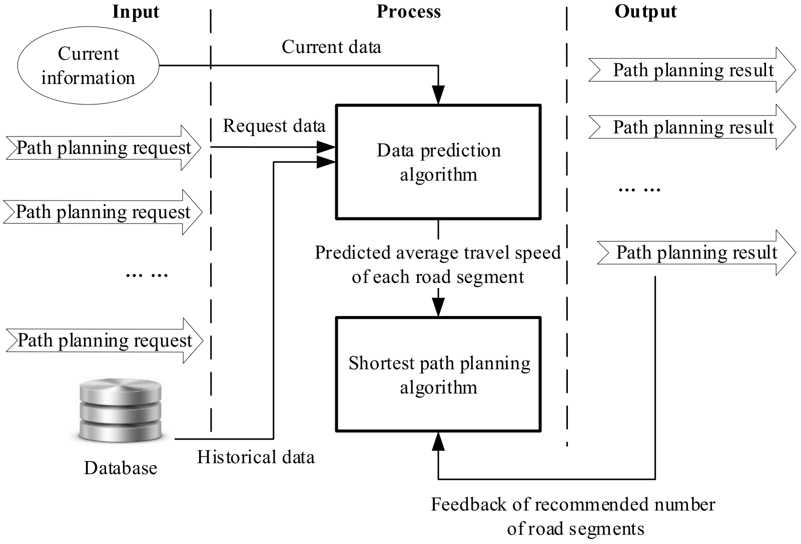


FIGURE 2. Overall TPPDP algorithm.

load-balanced paths are recommended dynamically from a global perspective. On the one hand, it avoids recommending the same section to too many users, causing possible traffic congestion in the future; On the other hand, it can respond to unexpected situations such as traffic control and traffic accidents and update the planned path in time.

C. TRAFFIC SPEED PREDICTION METHOD BASED ON HISTORICAL DATA

Since the driving time reflects the effects of traffic situation, the path planning algorithm usually uses travel time of the path as the criterion for determining the optimal path, and the driving time is determined by the average travel speed. Therefore, in this paper a prediction algorithm is proposed to predict the speed on the road. The traffic prediction described in this paper refers to the prediction of the average travel speed of a road segment in a specific time slot.

This algorithm is based on the historical data of traffic, which mainly records the average travel speed of road segment. In order to summarize the rules for prediction, a traffic speed forecasting model based on KNN is proposed in this paper, which takes the factors that affect the travel speed as the characteristic attributes of input data, such as the time slot, current speed and so on, and takes the prediction speed of a specific time slot in the future as the output data.

In this paper, a data prediction model was used. As shown in Figure 3, the current road segment data is filtered out from the historical database and the step is based on the ID of road segment. Then the k nearest neighbors of input data (five-dimensional eigenvectors) are found from the historical database using the KNN algorithm. After that, the output Y, speed of a specific time slot in the future, is obtained by weighting and averaging the values of k nearest neighbors from Y_1 to Y_K. The traffic prediction described in this paper mainly refers to the prediction of the average travel speed Y of a road segment in a specific time slot.

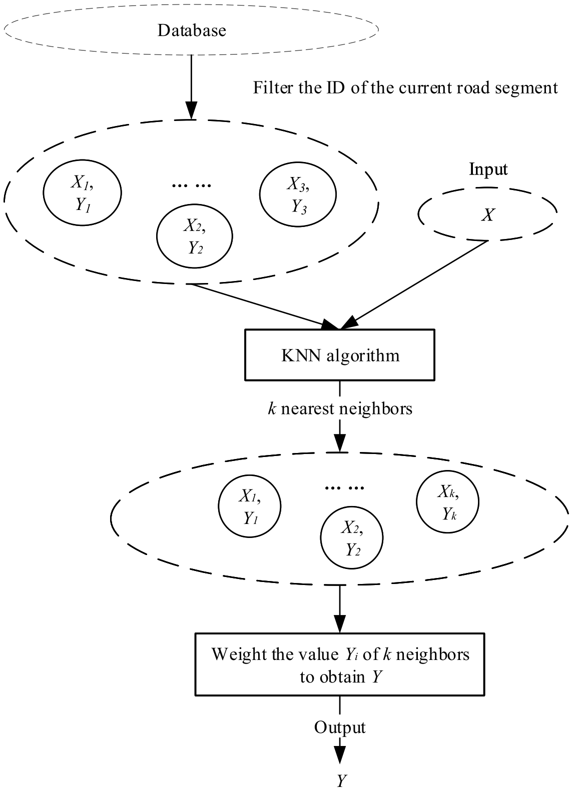


FIGURE 3. Data prediction method based on KNN algorithm.

1) PARAMETERS IN KNN

There are two parameters in KNN algorithm: the dimension n of the input data (eigenvector) and the number k of nearest neighbors. In order to determine the value of k and n, Yao *et al.* [37] used mean absolute percentage error(MAPE) as the evaluation index, and made experiments on different combinations of k and n. The results showed that the parameter pair $\{n = 5, k = 6\}$ produces lower MAPE. Therefore, the

TABLE 1. Input eigenvector.

Name	Value	Name	Value
Time slot	1-144	Holiday	0 or 1
Day	1-7	Rainfall	1-6
Current speed (km/h)	/		

parameter pair $\{n = 5, k = 6\}$ is used as the parameter of the KNN algorithm in this paper.

2) INPUT DATA (X)

The attributes of the input data are the factors that affect driving on the road segment. Usually, the factors affecting the driving on the road segment include driving time slot, holidays, rainfall and so on. Therefore, the input instance of the KNN algorithm in this paper is determined as a 5-dimensional eigenvector: {Time slot, Day, Holiday, Rainfall, Current speed}, as shown in Table 1.

1) According to [38], 10 minutes is regarded as a unit of traffic status, so one day is divided into 144 time slots as a time interval is 10 minutes. The reason for choosing this attribute is that the travelling of urban residents is regular and the traffic pattern of urban roads is also periodic, similar and dynamic.
2) Day represents the day of the week, and the value is from 1 to 7. Since the urban residents' travel within a week has certain regularity, the road traffic mode also has periodicity, similarity and dynamics with the unit of day.
3) Holiday is a boolean value to mark it's a holiday or not. People have different travel habits on holidays or non-holidays. Therefore, there is also a significant difference between holiday and non-holiday traffic flows.
4) The values of Rainfall consist of 6 levels:
   a) Light rain: 10 millimeter (24h).
   b) Moderate rain: 10-24.9 millimeter (24h).
   c) Heavy rain: 25-49.9 millimeter (24h).
   d) Rainstorm: >50 millimeter (24h).
   e) Heavy rainstorm: >100 millimeter (24h).
   f) Torrential rain: >250 millimeter (24h).
5) The current speed reflects the average travel speed in this road segment in this time slot, and it generally does not exceed the maximum speed limit of the road. Since the current speed has an important influence on the speed of the next time slot, this attribute can combine current traffic data with historical data to make the prediction more realistic.

The prediction of traffic requires robust to short-term and long-term changes in traffic conditions. If these changes are unexpected, such as traffic accidents and bad weather, it is very difficult to construct a responsive prediction. Above all, unexpected changes are not considered in this paper.

3) OUTPUT DATA (Y)

The output data is the predicted speed v for a road segment in next time slot. When determining k nearest neighbors by KNN algorithm, the first step is to calculate the distances between the input data X and the historical data. The Euclidean distance is used here:

$$d_i = \sqrt{\sum_{j=1}^{5}(x_j - x_{i,j})^2} \tag{1}$$

where X is the 5-dimensional eigenvector ($X = \{x_1, x_2, x_3, x_4, x_5\}$), and d_i represents the Euclidean distance between the input data X and the ith historical data ($X_i = \{x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}, x_{i,5}\}$), i refers to the relevant data in the database, and j is the serial number of the eigenvector. According to the value of d_i, k historic data with minimum d_i can be selected as the k nearest neighbors of the input data. that is, the output of the KNN algorithm. Because the parameter number of the KNN is $\{n = 5, k = 6\}$, the neighbors obtained by the KNN algorithm is $\{(X_1, Y_1), \ldots, (X_6, Y_6)\}$. Using the method of voting to obtain the final value, that is, each Y_i is multiplied by a weight, and then added to obtain the final value Y.

The weight of each neighbor is the reciprocal of the distance: $1/d_i$.

Sum these weights to get D:

$$D = \sum_{i=1}^{k}(\frac{1}{d_i}) \tag{2}$$

The output Y is:

$$D = \sum_{i=1}^{k}(\frac{1/d_i}{D} \times y_i) \tag{3}$$

The output data is the predicted speed v, that is, $v = Y$.

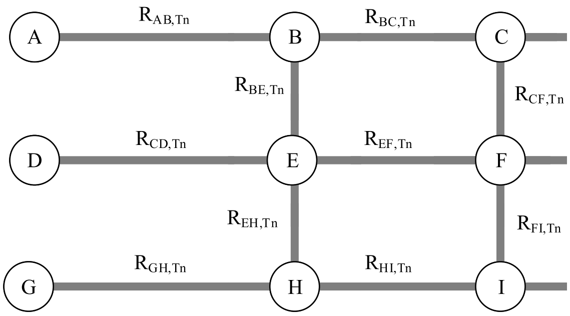


FIGURE 4. Road Network with weight.

D. PROCESS OF TPPDP ALGORITHM

The TPPDP algorithm is an improved Dijkstra algorithm. The road network is abstracted as a graph shown in Figure 4. The point indicates the starting point of the road segment and the edge refers to road segment. The algorithm is designed as follows.

1) EDGE WEIGHT SETTING

The travel time intuitively reflects the real time traffic condition and also meets the query requirements of most people, so the weight of the road segment in Figure 4 is set as the travel time. The weight of the road segment is expressed as R_{i,T_n}, where i represents the ID of the road segment and T_n represents the time slot of current time.

The cost of the node is the weight of road segment, that is, the travel time of its upstream road segment. The optimal path planned by this way is with the least travel time.

In real life the traffic situation of the road segment changes over time, and correspondingly the weight of the road segment is different at each time slot of one day. According to Section III-C, one day can be divided into 144 time slots, with every 10 minutes as one unit. Therefore, each road segment has 144 weights: $R_{i,1-144}$, and the corresponding weights can be selected according to current time.

The average travel time of the road segment is determined by the average travel speed v, which is calculated from the speed predicted in Section III-C.

2) PATH PLANNING ALGORITHM

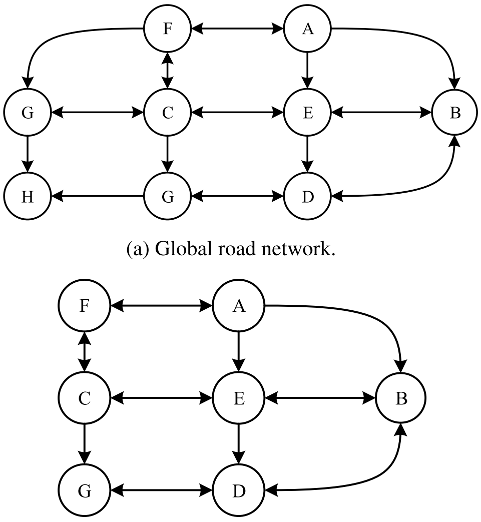
The process of path planning executes as follows: Firstly, a graph for road network is constructed as Figure 4 shown and the weight of edge is set as described in Section III-D.1. The path planning request from user is received, including the start point and the destination. Starting from the requested source node, the optimal path with minimum weights from the current node to its neighboring node is obtained. Then take the node with the least cost as the current node, recursively, until the optimal path to the destination point is determined. Since the average travel time is the weight of the road segment, the optimal path obtained by this way is the path with the minimum total travel time.

The above algorithm is a global query algorithm, and in every recursive step it is possible to scan all the nodes to find the shortest path to the destination. In the case of big data for urban traffic, the global query will seriously restrict the calculation speed of the path planning algorithm proposed above. Therefore, it is necessary to design the method of accelerating calculation.

According to [19], the method of building a local road network can be used to accelerate the calculation speed of path planning algorithm. The local road network is a sub graph of the global road network which contains all the nodes in the city map. In this paper, the local road network can be constructed by searching for the set of nodes $Node_{SD}$ that are included in the paths from the requested source node to destination in the historical database. $Node_{SD}$ is defined as follows:

$$Node_{SD} = \{node_i | node_i \text{ in the paths from } S \text{ to } D\} \quad (4)$$

The process is described by the following example. Suppose that Figure 5(a) represents the global road network, now the path from node C to B needs to be planned. Instead of scanning all nodes in the graph, the algorithm firstly searches for the nodes include in the paths starting from C to B in the historical database. The nodes formed by search result construct the local road network. Then the path planning algorithm executes in the local graph. Assuming that the paths from C to B include the following:



(a) Global road network.

(b) Local road network.

FIGURE 5. Builting local road network.

$$\begin{aligned} P_{CB1} &= \{\overrightarrow{CE}, \overrightarrow{EB}\} \\ P_{CB2} &= \{\overrightarrow{CF}, \overrightarrow{FA}, \overrightarrow{AB}\} \\ P_{CB3} &= \{\overrightarrow{CG}, \overrightarrow{GD}, \overrightarrow{DB}\} \end{aligned} \quad (5)$$

The nodes contained in these paths can be added to the set of $Node_{CB}$:

$$Node_{CB} = \{A, E, D, F, G\} \quad (6)$$

Therefore, the local road network from C to B is constructed by $Node_{CB}$, as shown in Figure 5(b).

The above process aims to simplify the original road network. By using local road network instead of global one for searching, this algorithm greatly reduces the computation load.

E. TPPDP-LB: TRAFFIC PATH PLANNING ALGORITHM BASED ON DATA PREDICTION WITH LOAD BALANCING

The average road speed required by the path planning algorithm is studied in this section. Ge [39] considers that path recommendation algorithm is usually targeted at a single vehicle, and it leads to overlapping segments recommended by the path planning system to different users. E.g., the same optimal path is recommended to a large number of users without considering the overall coordination, when they request path planning from same source and destination simultaneously. It causes overlapping recommended road segments could generate new traffic congestion and the capacity of these sections will decrease as the vehicles grow in the next time slot. The path planning system is eventually caught in the problem of local optimization.

For this reason, the recommendation factor is introduced as a correction of the average speed predicted in Section III-C.

It represents the impact of the recommended times on the average speed of the road segment. Combined with the recommendation factors, the average speed predicted in Section III-C can be effectively adjusted, avoiding certain road segments being repeatedly recommended. The combination of average prediction speed and recommendation factors reflects the load balancing of path planning algorithm.

According to Lu *et al.* [40], the relationship between driving speed and the traffic density is as follows:

$$V(\rho) = V_{\rho=10} - 13.375 \times ln(\rho) + 30.797 \tag{7}$$

where ρ is the density of the road segment, expressed as the number of vehicles per lane per kilometer, and $V_{(\rho=10)}$ indicates the average speed of the vehicles when the traffic density is 10 veh/(km*lane).

The average speed of a certain segment in time slot t_m is expressed as $V(\rho)_m$:

$$V(\rho)_m = V_{\rho=10} - 13.375 \times ln(\rho) + 30.797 \tag{8}$$

where ρ is the density of the road segment in the corresponding time slot, which can be predicted by KNN algorithm.

The system continuously recommends the path to the users and these paths could contain the same road segments. The number of recommended times r can be converted to the increment of the density of road segments ρ_{inc} and ρ_{inc} will have a certain impact on the predicted speed $V(\rho)_m$

$$\rho_{inc} = r/L \tag{9}$$

where L is the length of corresponding road segment.

Considering this impact, the average speed of the road segment is corrected to $V(\rho + \rho_{inc})_m$, as shown in the following:

$$V(\rho+\rho_{inc})_m = V_{\rho=10} - 13.375 \times ln(\rho+\rho_{inc}) + 30.797 \tag{10}$$

It can be obtained:

$$V(\rho + \rho_{inc})_m - V(\rho)_m = 13.375 \times ln(\frac{\rho}{\rho + \rho_{inc}}) \tag{11}$$

That is:

$$V(\rho + \rho_{inc})_m = V(\rho)_m + 13.375 \times ln(\frac{\rho}{\rho + \rho_{inc}}) \tag{12}$$

$$V = V(\rho + \rho_{inc})_m \tag{13}$$

As mentioned above, the average travel speed is corrected to $V(\rho + \rho_{inc})_m$ after considering the recommended times of the road segment. Therefore, the weight of the road segment is the average travel time, which is expressed as follows:

$$weight = L/V \tag{14}$$

F. DYNAMIC ADAPTION

TPPDP and TPPDP-LB algorithms only consider the static path planning problem, and cannot be adapted to the situation where the traffic situation changes after the navigation starts. Therefore, the dynamic adaption to sudden traffic changes, such as traffic accident and control, is required. In order to obtain real-time traffic information and change the path planning dynamically, an index of drivers and road segments is used. The format is defined as follows:

$$Tral = \{Seg_i | i\epsilon(1, 2, \ldots, n)\} \tag{15}$$

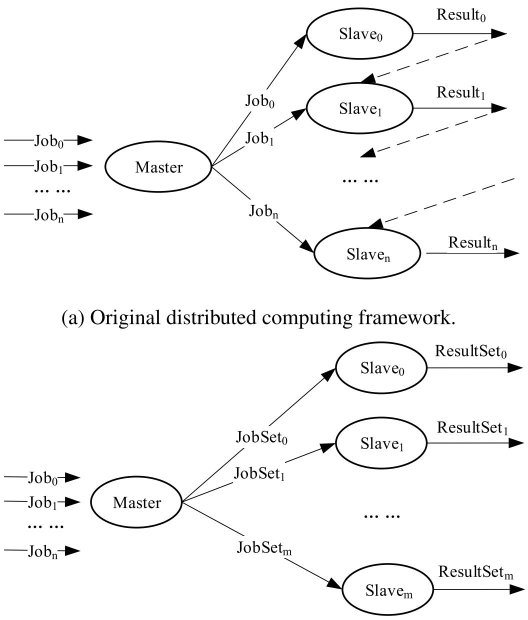
The above formula indicates that the planned path for the driver is composed of the road segments from 1 to n. When a traffic accident occurs on the road segment i, and the driver hasn't arrived at this segment, the path planning algorithm for the driver will re-execute in time. By this way, the algorithm can be adapt to the sudden change if traffic.

IV. DISTRIBUTED COMPUTING FRAMEWORK FOR TPPDP-LB

A. THE REQUIREMENT OF DISTRIBUTED COMPUTING FOR TPPDP-LB

In modern transportation systems, TPPDP and TPPDP-LB algorithms usually face many concurrent path planning requests from different uses, so this is a problem of big data processing and it is particularly important to improve the execution speed of the algorithms by using a distributed computing framework. The condition of a distributed parallel computing framework is that the processing between each part of the data set is independent and isolated, such as Map-Reduce [41]. However, in TPPDP-LB algorithms, the number of recommended times is a key factor to determine the predicted average speed of road segment in next time slot, that is, the result of each path planning request will have an impact on the next path planning request. Therefore, TPPDP-LB algorithm is actually a serial calculation as shown in Figure 6(a): One job corresponds to one path planning request and the jobs are divided into n parts which are assigned to n distributed compute nodes. When TPPDP-LB algorithm is executed, the result of Job i affects Job $i + 1$. The current distributed computing frameworks like Map-Reduce cannot satisfy the requirement of TPPDP-LB, so it is necessary to design a special distributed computing framework for this algorithm.

The improved distributed computing framework proposed in this section mainly solves the problem of computation decomposition. The concept of "correlation" is defined firstly. If the path based on a path planning request does not overlap or less overlaps with the path based on another path planning request, the two requests are considered to have low correlation. Path planning requests with less correlation have little effect on each other, so they can be distributed to different compute nodes for parallel computing. As shown in Figure 6(b), the path planning requests with high correlation are distributed to the same compute node for serial



(a) Original distributed computing framework.

(b) Improved distributed computing framework.

FIGURE 6. Distributed computing framework for TPPDP-LB.

computing, while the path planning requests with low correlation are distributed to different compute nodes for parallel computing. This framework can both take advantage of the distributed computing framework and handle data-dependent calculations.

B. DESIGN OF DISTRIBUTED COMPUTING FRAMEWORK FOR TPPDP-LB

1) CLUSTERING METHOD

In order to solve the problem of computation decomposition, K-means++ algorithm [42] is chosen, for it augments the K-means algorithm with a randomized seeding technique and improves the speed and accuracy of clustering efficiently. The path planning requests are classified by using K-means++ algorithm and the purpose of grouping the requests with larger correlation is realized. K-means++ algorithm divides *n* objects into *k* clusters according to their attributes to achieve the results that the objects in the same cluster is similar and little different while the objects in different clusters are less similar.

The clustering method in this paper is based on the path planning request which includes starting node and destination node. It is regarded that the path planning requests with close starting node and destination node have larger correlation. Two mappings in this clustering method are defined as follows.

1) *The first mapping*: the path planning requests are mapped to vectors in a two-dimensional space, and the coordinates are composed of latitude and longitude of the starting node and the destination node. One vector in the coordinates corresponds to one path planning request:

$$Req_i = \overrightarrow{(S_{LOi}, S_{LAi}), (D_{LOi}, D_{LAi})} \tag{16}$$

2) *The second mapping*: the vector Req_i is mapped to a point with four-dimensional coordinates, which are respectively the longitude of the starting point S_{LOi}, the latitude of the starting point S_{LAi}, the longitude of the destination point D_{LOi}, the latitude of the starting point D_{LAi}.

For example, the vectors in Figure 7 are mapped to the points in a four-dimensional space. Corresponding vector and coordinate point represent the same path planning request.

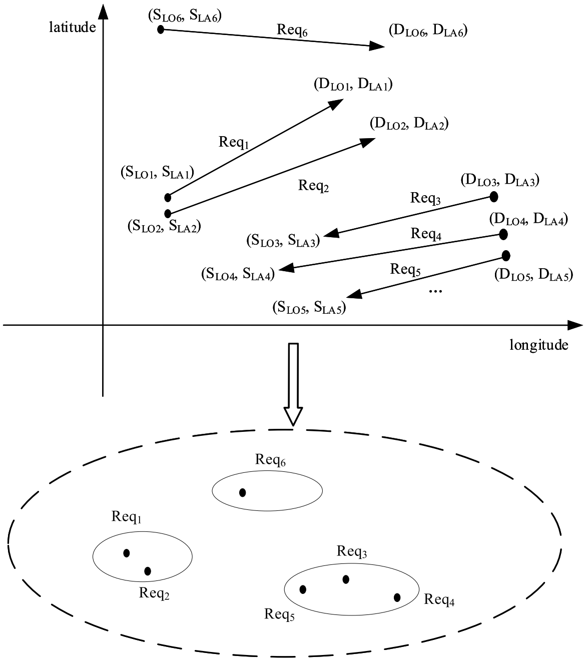


FIGURE 7. Two mapping method.

Then the points in the coordinate system are clustered by using K-means++ algorithm and the points close to each other are divided into one cluster according to Euclidean distance. That is, the path planning requests with close starting point and the destination point are divided into a cluster.

Since the path planning requests in the same cluster have close starting points and the destination, the correlation between them is large. Therefore, the results of these requests have more overlapping road segments, so each result of a request will affect results of other requests in the cluster. According to TPPDP-LB, serial computing is required in one cluster.

While the path planning requests are in different clusters, the correlation between them is small. Then the results of these requests will not affect each other, that is, they have less overlapping road segments. Therefore, parallel computing can be performed between different clusters to increase executing speed.

2) OPTIMIZATION OF CLUSTERING METHOD

The value k in K-means++ usually affects the clustering results and it can be determined by pre-processing based on the historical data of urban roads. The purpose of load balancing in TPPDP-LB is to avoid the same road segments being recommended to too many vehicles, resulting in congestion in next time slot. Conversely, from the perspective of congestion, the purpose of load balancing is to divert vehicles on congested road segments. Taking the overall situation of the road network into account, the number of crowed areas, which are defined as consecutive crowded road segments, can be chosen as the value k of the clustering algorithm. And the value k can be adjusted according to the number of congested areas in different periods, e.g., there are more crowded areas during rush hour and the value k can be set to a larger value, while other time crowded area is less and k can be set to a smaller value. In this way, elasticity of computing resources can be achieved, which can not only meet the requirement of distributed computing, but also save computing resources.

V. SIMULATION

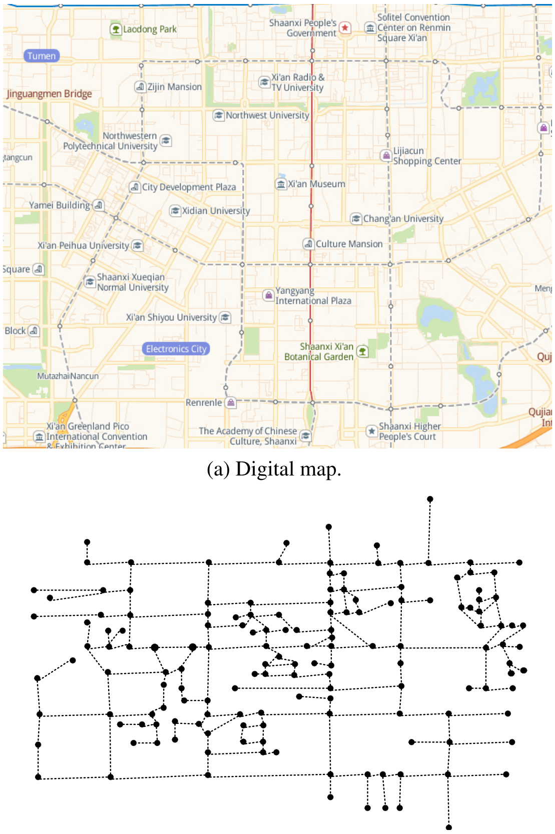
A. SIMULATION SETTINGS

The digital map data used in the simulation is downloaded from the website OpenStreetMap [43], where the map data is uploaded by the public and free for public non-commercial use. As shown in Figure 8(a), the simulation map selects the area of Yanta West Road in Xi'an, China, and the map size is about $2.4\,km * 4\,km$. The exported data of digital map is in XML format and the structure of file is shown in Table 2. The data of map can be abstracted and mapped to a road network graph as shown in Figure 8(b), where the node refers to the vertex of road section while the link represents a road segment. The traffic history data is taken from the traffic data on the map area within one month and the data structure is shown in Table 1.

TABLE 2. Structure of map file.

```
<?xml version="" encoding=""?>
<bounds minlat="" minlon="" maxlat="" maxlon=""/>
<node id="" lat="" lon="" version="" timestamp="" changeset=""
uid="" user="">
        <tag k="source" v="PGS"/>
</node>
<way id="" version="" timestamp="" changeset="" uid="" user="">
        <nd ref=""/>
        <nd ref=""/>
        <tag k="" v=""/> <tag k="" v="ąś/>
        <tag k="" v=""/>
</way>
```

The path planning algorithms are simulated in the above map, and the distributed computing framework is implemented in multiple virtual machines with the same configuration on two physical computers. All simulation works are implemented with Matlab R2014a.


Laodong Park
Shaanxi People's Government
Sofitel Convention Center on Renmin Square Xi'an
Tumen
Jinguangmen Bridge
Zijin Mansion
Xi'an Radio & TV University
Northwest University
Northwestern Polytechnical University
angcun
Lijiacun Shopping Center
City Development Plaza
Xi'an Museum
Yamei Building
Xidian University
Chang'an University
Xi'an Peihua University
Culture Mansion
quare
Shaanxi Xueqian Normal University
Yangyang International Plaza
Menj
Xi'an Shiyou University
Block
Electronics City
Shaanxi Xi'an Botanical Garden
Quj
MutazhaiNancun
Renrenle
Xi'an Greenland Pico International Convention & Exhibition Center
The Academy of Chinese Culture, Shaanxi
Shaanxi Higher People's Court
Qujia Int

(a) Digital map.

(b) Road network graph.

FIGURE 8. Digital map and corresponding road network.

B. SIMULATION OF PATH PLANNING ALGORITHMS

In this part, the simulation compared the following four path planning algorithms: Dijkstra, improved Dijkstra (i-Dijkstra) [10], TPPDP and TPPDP-LB in terms of travel time, dynamic adaptability and load balancing, etc. The simulation experiments in this part run on a single computing node, without considering distributed parallel computing.

1) TRAVEL TIME

The simulations in this part compare the travel time planned by four algorithms, in the cases of different time periods and different number of path planning requests. It is assumed that all requests start from source point S and end at destination node D as shown in Figure 9. The first simulation compare the travel time with different departure times, because according to historical data, traffic congestion is different in different time periods. The simulation parameters are set as Table 3, which selects several typical departure times, and path planning requests are set proportionally based on historical data. As shown in Figure 10, the simulation result reveals that TPPDR and TPPDR-LB algorithms have shorter travel time than Dijkstra and i-Dijkstra on average, while during rush hour, such as at 7:00, TPPDR-LB has obvious superior performance in terms of travel time compared with

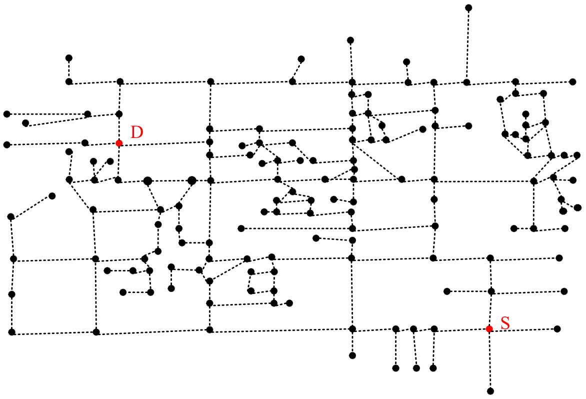


FIGURE 9. Source and destination nodes in the road network.

TABLE 3. Simulation parameters.

	Departure time	Concurrent requests
1	7:00	63
2	8:00	18
3	9:00	38
4	10:00	10
5	11:00	49
6	12:00	38
7	13:00	20
8	21:00	39

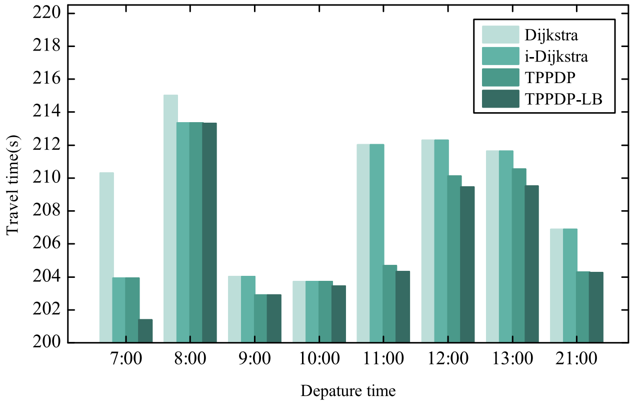


FIGURE 10. Travel time with different departure time.

other algorithms. The reason is that the number of path planning requests during the rush hour is the largest, and TPPDR-LB algorithm can well achieve load balancing, thereby reducing the overall travel time.

The second simulation compares the travel time with different number of concurrent requests in same time slot to evaluate the abilities of four algorithms to handle concurrent requests. The departure time is set at 7am, when the traffic is heavy. From Figure 11, it can be concluded that the number of concurrent requests has a certain impact on the travel time of the path planned by different algorithm. The four algorithms all cost more travel time as the number of concurrent requests increases, because the road will become congested as the number of concurrent requests increases.

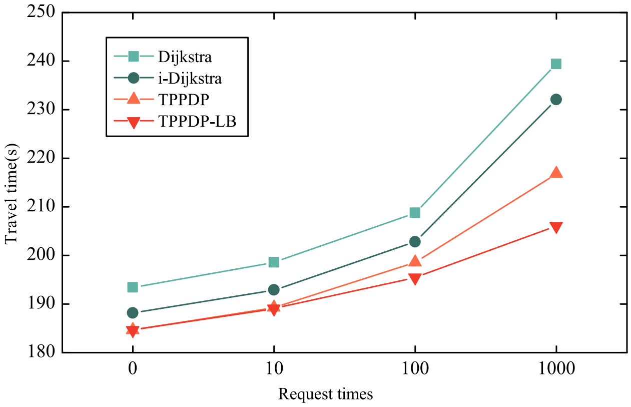


FIGURE 11. Travel time with different number of concurrent requests.

It is also proved that TPPDP and TPPDP-LB algorithms achieve shorter travel time in any case while TPPDP-LB algorithm has best performance when number of requests reaches the maximum. TPPDP-LP algorithm is least affected by the number of concurrent requests because the algorithm takes the recommended times of road segments into account as that to implement load balancing of path planning.

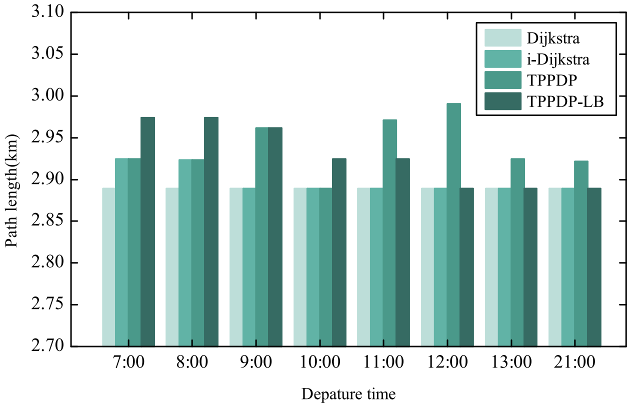
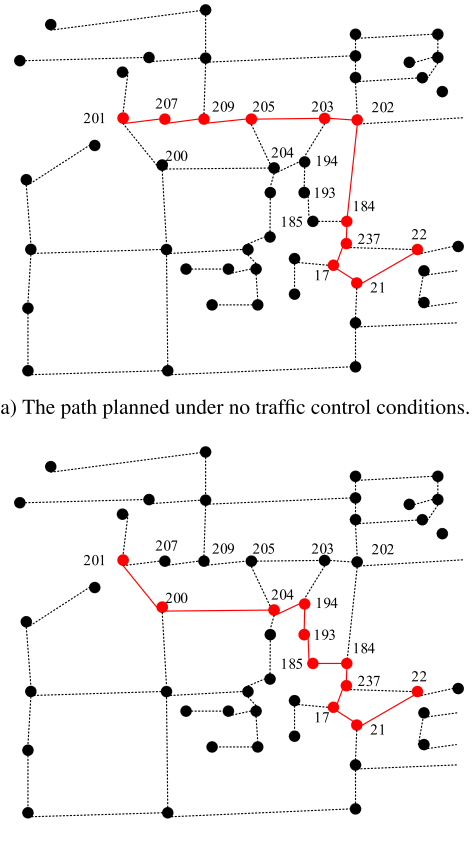


FIGURE 12. Length of the planned path.

As is shown in Figure 12, the path length generated by four algorithms is compared. The path planned by the Dijkstra algorithm is always the shortest, and the paths planned by the TPPDP and TPPDP-LB algorithm are longer because the two algorithms concern more about the travel time instead of the length of path. It can be seen from the results that the path lengths calculated by TPPDP and TPPDP-LB algorithms are not much longer than Dijkstra algorithm, and are even the same at some time slot. Therefore, these two algorithms sacrifice the cost of a little distance and achieve the minimum travel time, and as mentioned above, travel time is the most important issue for querying users.

2) DYNAMIC ADAPTABILITY

In Section III-F, a dynamic adaptive optimization algorithm is proposed to improve the TPPDP and TPPDP-LB algorithms, in order to deal with road sudden conditions in time.



(a) The path planned under no traffic control conditions.

(b) The path planned under traffic control conditions.

FIGURE 13. Dynamic adaptability of path planning algorithm.

The simulation in this part is implemented to verify the dynamic adaptability in the case of traffic control.

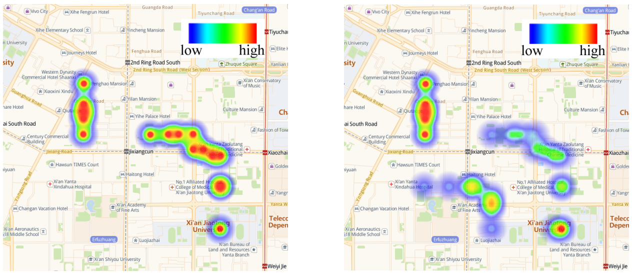
The path planning request is set from node 22 to 201 and the departure time is 9 am. By running TPPDP algorithm, a planned path represented by red solid lines is obtained as shown in Figure 13(a), which has the shortest travel time of 34.1466 seconds. Then we set up road traffic control between sections 184 to 202, and by using dynamic adaptive optimization algorithm, the planned route becomes as shown in Figure 13(b), avoiding the traffic control sections, and the travel time reaches 37.0532 seconds. Therefore, it can be inferred that the algorithm proposed in this paper can realize the dynamic planning path. When sudden traffic control or traffic accidents happen, paths can be update in time to avoid congestion.

3) LOAD BALANCING

As mentioned in Section III, TPPDP-LB algorithm implements load balancing for path planning by considering the number of concurrent requests. The simulation in this part aims to verify the capacity of load balancing of TPPDP-LB compared to TPPDP. The settings of the simulation experiment are shown in Table 4. 20 concurrent path planning requests from node 30 to node 122 are sent to the simulation system at the same time, and then TPPDP algorithm and TPPDP-LB algorithm respectively plan different paths to these requesting users.

TABLE 4. Simulation parameters.

Name	Value
Starting node	30
Destination	122
Departure time	10:00
Number of concurrent requests	20



(a) Hot zone map caused by TPPDP.

(b) Hot zone map caused by TPPDP-LB algorithm.

FIGURE 14. Traffic hot zone map.

Heat map is used to describe the traffic density in the next period, and the result is shown in Figure 14. Figure 14(a) is the traffic hot zone map generated by the path planned by TPPDP Algorithm, while Figure 14(b) is generated by TPPDP-LB Algorithm. The hot zone map uses different colors to represent different traffic density. Red represents the most congested traffic conditions, and yellow, green, cyan, and blue in turn represent a decrease in traffic density. Comparing Figure 14 (a) and (b), it shows that the paths planned by TPPDP-LB algorithm achieves better load balancing than the paths planned by TPPDP algorithm. It is because TPPDP-LB algorithm will recommend users to other paths if a road segment is recommended to many users at the same time, so as to ensure the load balancing of these road segments in the next time. The larger the number of concurrent path planning requests is, the better the performance of load balancing is.

C. SIMULATION OF DISTRIBUTED COMPUTING FRAMEWORK

In Section IV, a distributed computing framework is designed for implementing TPPDP-LB algorithm to solve the problem that the algorithm is difficult to implement in a distributed mode because of serial computing. The simulation experiments in this section run the TPPDP-LB algorithm on the distributed computing framework to verify the performance of the framework in terms of runtime and result validity. The frequency of algorithm execution can be adjusted according to the frequency of request generation. The algorithm is executed once per second in the simulation.

TABLE 5. Clustering results.

	k=1 (number)	k=4 (number)	k=8 (number)
cluster 1	4980	1257	565
cluster 2	/	1404	938
cluster 3	/	1311	986
cluster 4	/	1008	305
cluster 5	/	/	604
cluster 6	/	/	764
cluster 7	/	/	579
cluster 8	/	/	239

1) RUNTIME

Reducing algorithm runtime is the main purpose of using a distributed computing framework, therefore the first simulation experiment is to compare the runtime with different numbers of distributed compute nodes. As mentioned in Section IV, the number of distributed compute nodes depends on the k value of the KNN clustering algorithm. The simulation parameters are set as shown in Table 5: 4980 path planning requests are randomly generated with different starting nodes and destinations, and the k values are set to 1, 4, and 8, respectively. The request data are clustered in the case of $k = 1$, $k = 4$ and $k = 8$, and the clustering results are shown in Table 5. In the case of $k = 1$, there is one compute node that execute all the path planning requests. In the case of $k = 4$, the data are divided into 4 different clusters and there are 4 compute nodes to process 4 groups of the path planning requests simultaneously. In the case of $k = 8$, the data are divided into 8 different clusters and there are 8 compute nodes to process 8 groups of the path planning requests simultaneously. The path planning requests that are divided into the same cluster are processed serially, as well as the path planning requests that are divided into different clusters are processed parallelly.

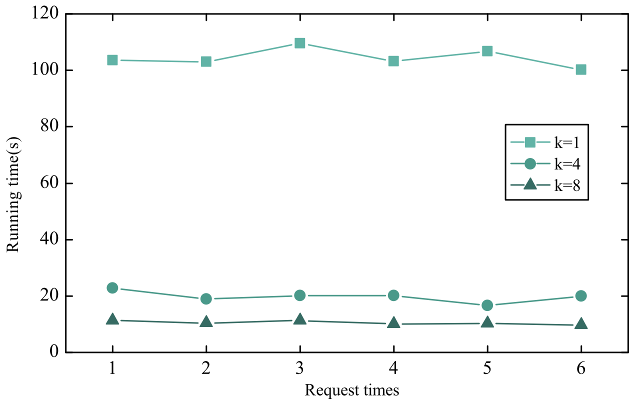


FIGURE 15. Comparison of runtime with different clusters.

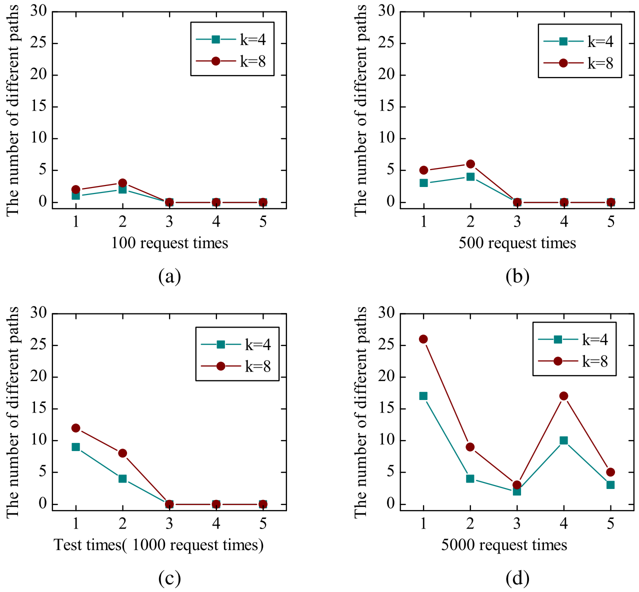
Figure 15 shows the runtime of TPPDP-LB algorithm with different k values ($k = 1$, $k = 4$, $k = 8$). The average runtime of three conditions is 104.41 seconds, 20.31 seconds and 10.52 seconds, which shows the effectiveness of the distributed computing framework in improving the processing speed. It can be concluded that the runtime of whole algorithm decreases as the number of k increases, because as the number of clusters increases, the amount of data that each compute node needs to process decreases accordingly. The whole runtime of each condition depends on the compute node with the longest runtime.

2) ALGORITHM ACCURACY

The TPPDP-LB algorithm is essentially a serial calculation process, because each path planning request needs take into account the results of the previous requests. By dividing the request data into clusters with no correlation, the distributed computing framework achieves parallel computing. However, the actual path planning request is difficult to achieve no correlation completely, and some road sections will always overlap. In this case, simulation experiments are required to prove that the difference between the recommended path generated by the distributed computing framework and the path generated by serial computing is very small.

The simulation experiment setting is as follows: Concurrent path planning requests are randomly generate, and the number of requests are 100, 500, 1000 and 5000 respectively. Then the distributed computing framework is used to execute the TPPDP-LB algorithm, taking the values of k 4 and 8 respectively. Each computing was performed five times. Finally, the experimental results were compared with the paths generated by the serial computing to verify the difference of results. When $k = 8$, the different number of paths get more, compared to the condition of $k = 4$. However, from the results of all experiments in the figure, compared with the results produced by parallel computing, the results produced using distributed computing framework have very little difference, which are controlled within 1%. Therefore, it is concluded that using parallel computing framework can guarantee the accuracy of TPPDP-LB algorithm.

The simulation experiment setting is as follows: Concurrent path planning requests are randomly generate, and the number of requests are 100, 500, 1000 and 5000 respectively. Then the distributed computing framework is used to execute the TPPDP-LB algorithm, taking the values of k 4 and 8 respectively. Each computing was performed five times. Finally, the experimental results were compared with the paths generated by the serial computing to verify the difference of results. The simulation result is shown in Figure 16, which proved that there is a small difference between the path produced by distributed computing framework and the path produced by serial computing. When the number of concurrent requests increases from 100 to 5000, the difference between the execution results by distributed computing and serial computing framework also increases. It is because the more the number of requests gets, the greater the correlation between the paths may exist. Furthermore, the value of k also affects the accuracy of the algorithm. When $k = 8$, the different number of paths get more, compared to the condition of $k = 4$. However, from the results of all experiments in the Figure 16, compared with the results produced by parallel computing, the results produced using distributed



(a)

(b)

(c)

(d)

FIGURE 16. **The difference between results produced by distributed computing framework and serial computing.**

computing framework have very little difference, which are controlled within 1%. Therefore, it is concluded that using parallel computing framework can guarantee the accuracy of TPPDP-LB algorithm.

In summary, the simulation results proved that TPPDP algorithm can produce the planned path with the shortest travel time, based on historical traffic data, as well as the TPPDP-LB algorithm can achieve load balancing and avoid localized congestion by considering the situation where some road sections are reused by concurrently requested paths. In addition, the simulation also proves that the distributed computing framework to execute TPPDP-LB algorithm can implement parallel computing to reduce runtime, without affecting the superiority of the TPPDP-LB algorithm.

VI. CONCLUSION

In this work, a traffic path planning algorithm based on data prediction named TPPDP is proposed. A prediction model is designed to predict the traffic state of the road segment in the future based on historical traffic data and current traffic information, and based on the predicted result, which is the average travel speed of road segments in next timeslot, TPPDP algorithm is proposed to plan a path with the shortest travel time. Furthermore, considering that there are overlaps in path segments between planned paths produced by concurrent requests, a path planning algorithm based on data prediction with load balancing named TPPDP-LB is proposed, which combines the predicted information and the concurrent recommended number of the road segments to plan a path with the shortest travel time as well as keeping global load balancing. A specific distributed computing framework for TPPDP-LB algorithm is designed to improve the processing speed of the system, because the processing of TPPDP-LB algorithm is serial and the common distributed computing framework cannot meet the parallel processing requirements of TPPDP-LB algorithm.

The results of simulation show that TPPDP algorithm can find the path with shortest travel time while TPPDP-LB algorithm can achieve load balancing of computing, which can effectively avoid congested sections in the future. It is also shown that the distributed computing framework designed for TPPDP-LP algorithm can effectively reduce the runtime of system operation by 80% as well as keep the accuracy of TPPDP-LB algorithm.

REFERENCES

[1] B. S. Meghana, S. Kumari, and T. P. Pushphavathi, “Comprehensive traffic management system: Real-time traffic data analysis using RFID,” in *Proc. Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, vol. 2, Apr. 2017, pp. 168–171.

[2] Y. Gu and L. Xie, “Fast sensitivity analysis approach to assessing congestion induced wind curtailment,” *IEEE Trans. Power Syst.*, vol. 29, no. 1, pp. 101–110, Jan. 2014.

[3] J. Zhao, Y. Gao, Z. Bai, H. Wang, and S. Lu, “Traffic speed prediction under non-recurrent congestion: Based on LSTM method and BeiDou navigation satellite system data,” *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 2, pp. 70–81, Mar. 2019.

[4] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[5] P. Hart, N. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[6] Y. Wang, H. Zhou, and Y. Wang, “Mobile robot dynamic path planning based on improved genetic algorithm,” in *Proc. AIP Conf. Proc.*, vol. 1864, no. 1. New York, NY, USA: AIP, 2017, Art. no. 020046.

[7] M. Li, X. Li, and J. Yin, “TORD problem and its solution based on big trajectories data,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1666–1677, Jun. 2016.

[8] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing, “Discovering spatio-temporal causal interactions in traffic data streams,” in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2011, pp. 1010–1018.

[9] G. Han, X. Miao, H. Wang, L. Liu, J. Jiang, and Y. Peng, “A dynamic multipath scheme for protecting source-location privacy using multiple sinks in WSNs intended for IIoT,” *IEEE Trans. Ind. Inf.*, to be published.

[10] R. Wang, “A research on the weighted improvement of Dijkstra algorithm in optimal path calculation,” in *Proc. 5th Int. Conf. Frontiers Manuf. Sci. Measuring Technol. (FMSMT)*, 2017.

[11] M. Wei and Y. Meng, “Research on the optimal route choice based on improved Dijkstra,” in *Proc. IEEE Workshop Adv. Res. Technol. Ind. Appl. (WARTIA)*, Sep. 2014, pp. 303–306.

[12] D. Li, B. Han, and Y. Han, “Conversely improved a star route search algorithm,” *J. Syst. Simul.*, vol. 22, pp. 5175–5177, Nov. 2007.

[13] H. Shi, W. Cao, S. Zhu, and B. Zhu, “Applications of the improved A* algorithm for route planning,” in *Proc. 2nd Int. Conf. Intell. Comput. Technol. Automat.*, vol. 1, 2009, pp. 299–302.

[14] Z. He, K. Chen, and X. Chen, “A collaborative method for route discovery using taxi drivers’ experience and preferences,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2505–2514, Aug. 2018.

[15] J. Zhang, W. Meng, Q. Liu, H. Jiang, Y. Feng, and G. Wang, “Efficient vehicles path planning algorithm based on taxi GPS big data,” *Optik*, vol. 127, no. 5, pp. 2579–2585, Mar. 2016.

[16] W. Luo, H. Tan, L. Chen, and L. M. Ni, “Finding time period-based most frequent path in big trajectory data,” in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2013, pp. 713–724.

[17] L. Wei, Y. Zheng, and W. Peng, “Mining popular routes from social media,” in *Multimedia Data Mining and Analytics*, A. Baughman, Ed. Springer, 2015, pp. 93–116.

[18] G. Han, H. Guan, J. Wu, S. Chan, L. Shu, and W. Zhang, “An uneven cluster-based mobile charging algorithm for wireless rechargeable sensor networks,” *IEEE Syst. J.*, vol. 13, no. 4, pp. 3747–3758, Dec. 2019.

[19] J. Dai, B. Yang, C. Guo, and Z. Ding, "Personalized route recommendation using big trajectory data," in *Proc. IEEE 31st Int. Conf. Data Eng.*, Apr. 2015, pp. 543–554.

[20] H. Kanoh, "Dynamic route planning for car navigation systems using virus genetic algorithms," *Int. J. Knowl.-Based Intell. Eng. Syst.*, vol. 11, no. 1, pp. 65–78, Feb. 2007.

[21] J. Li, Y. D. Yang, and N. Mamoulis, "Optimal route queries with arbitrary order constraints," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 5, pp. 1097–1110, May 2013.

[22] S.-C. Sun, Z.-Y. Duan, and D.-Y. Yang, "Optimal paths planning in dynamic transportation networks with random link travel times," *J. Cent. South Univ.*, vol. 21, no. 4, pp. 1616–1623, Apr. 2014.

[23] C. Sommer, A. Schmidt, Y. Chen, R. German, W. Koch, and F. Dressler, "On the feasibility of UMTS-based traffic information systems," *Ad Hoc Netw.*, vol. 8, no. 5, pp. 506–517, Jul. 2010.

[24] S. Shang, L. Chen, Z.-W. Wei, D.-H. Guo, and J.-R. Wen, "Dynamic shortest path monitoring in spatial networks," *J. Comput. Sci. Technol.*, vol. 31, no. 4, pp. 637–648, Jul. 2016.

[25] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 3–19, Jun. 2014.

[26] W. Xiangxue, X. Lunhui, and C. Kaixun, "Data-driven short-term forecasting for urban road network traffic based on data processing and LSTM-RNN," *Arab J. Sci. Eng.*, vol. 44, no. 4, pp. 3043–3060, Apr. 2019.

[27] S. Fontanelli, E. Bini, and P. Santi, "Dynamic route planning in vehicular networks based on future travel estimation," in *Proc. IEEE Veh. Netw. Conf.*, Dec. 2010, pp. 126–133.

[28] 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. Inf.*, vol. 13, no. 3, pp. 1202–1212, Jun. 2017.

[29] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 50–64, Jun. 2014.

[30] X. Fei, C.-C. Lu, and K. Liu, "A Bayesian dynamic linear model approach for real-time short-term freeway travel time prediction," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 6, pp. 1306–1318, Dec. 2011.

[31] W. Li and L. Yang, "Application of wavelet neural network in prediction of short time traffic flow prediction," *Comput. Simul.*, vol. 133, no. 9, pp. 647–651, 2012.

[32] X. Kong, Z. Xu, G. Shen, J. Wang, Q. Yang, and B. Zhang, "Urban traffic congestion estimation and prediction based on floating car trajectory data," *Future Gener. Comput. Syst.*, vol. 61, pp. 97–107, Aug. 2016.

[33] D. Kang, Y. Lv, and Y.-Y. Chen, "Short-term traffic flow prediction with LSTM recurrent neural network," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.

[34] Y.-M. Xing, X.-J. Ban, and R. Liu, "A short-term traffic flow prediction method based on kernel extreme learning machine," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2018, pp. 533–536.

[35] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 21–34, Jan. 2016.

[36] Z. Su, Q. Liu, J. Lu, Y. Cai, H. Jiang, and L. Wahab, "Short-time traffic state forecasting using adaptive neighborhood selection based on expansion strategy," *IEEE Access*, vol. 6, pp. 48210–48223, 2018.

[37] B. Yao, Z. Wang, M. Zhang, P. Hu, and X. Yan, "Hybrid model for prediction of real-time traffic flow," *Proc. Inst. Civil Eng. Transp.*, vol. 169, no. 2, pp. 88–96, Apr. 2016.

[38] X. Yang, "Systematic approach of dynamic traffic control for urban expressway," Ph.D. dissertation, Dept. Road Traffic Engineer, Tongji Univ., Shanghai, China, 1996.

[39] Y. Ge, J. Wang, Y. Meng, and F. Jiang, "Research progress on dynamic route planning of vehicle navigation," *J. Highway Transp. Res. Develop.*, vol. 27, no. 11, pp. 113–117, 2010.

[40] S. Lu, H. Lu, Z. Tang, and J. Lu, "Analysis of influencing factors on urban travel speed," *J. Highway Transp. Res. Develop.*, vol. 1, no. 20, pp. 89–92, 2003.

[41] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proc. 6th Symp. Operating Syst. Design Implement. (OSD)*, 2004.

[42] D. Arthur and S. Vassilvitskii, "k-means plus plus: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms (ACM SIGACT)*. New Orleans, LA, USA: SIAM, Jan. 2007, pp. 1027–1035.

[43] S. Coast. *Open Street Map*. Accessed: Jan. 2019. [Online]. Available: https://www.openstreetmap.org/

NING SUN received the Ph.D. degree in computer science from Chungbuk National University, South Korea, in 2013. She is currently an Associate Professor with the Department of Internet of Things Engineering, Hohai University, Changzhou, China. Her research interests include network security, big data analysis, and the network architecture of IoTs.

HUIZHU SHI received the B.S. degree from Hohai University, China, in 2017, where she is currently pursuing the M.S. degree with the College of Internet of Things Engineering. Her research interests include big data analysis and network security, such as instruction detection using machine learning algorithm.

GUANGJIE HAN (Senior Member, IEEE) received the Ph.D. degree from Northeastern University, Shenyang, China, in 2004.

In February 2008, he finished his work as a Postdoctoral Researcher at the Department of Computer Science, Chonnam National University, Gwangju, South Korea. From October 2010 to October 2011, he was a Visiting Research Scholar with Osaka University, Suita, Japan. From January 2017 to February 2017, he was a Visiting Professor with the City University of Hong Kong, China. He is currently a Professor with the Department of Information and Communication System, Hohai University, Changzhou, China, and a Distinguished Professor of the Dalian University of Technology, Dalian, China. He is the author of over 350 articles published in related international conference proceedings and journals, including the IEEE JSAC, IEEE COMST, IEEE TMC, IEEE TIE, IEEE TII, IEEE TCC, IEEE TPDS, IEEE TVT, IEEE IoT Journal, the IEEE Systems, the IEEE Wireless Communications, the IEEE Communications, the IEEE Network, and so on. He is the holder of 130 patents. His current H-index is 36 and i10-index is 115 in Google Citation (Google Scholar). Total citation of his articles by other people is more than 5400 times. His current research interests include the Internet of Things, the industrial internet, machine learning and artificial intelligence, mobile computing, and security and privacy. He has served as the Co-Chair for more than 50 international conferences/workshops and as a Technical Program Committee member for more than 150 conferences. He has served on the Editorial Boards of up to 16 international journals, including the IEEE JSAC, the IEEE Network, the IEEE Systems, IEEE Access, the IEEE/CCA JAS, *Telecommunication Systems*, and so on. He has guest edited a number of special issues in the IEEE Journals and Magazines, including the IEEE Communications, the IEEE Wireless Communications, the IEEE Transactions on Industrial Informatics, *Computer Networks*, and so on. He has served as a Reviewer for more than 60 journals. He had been awarded the ComManTel 2014, ComComAP 2014, as well as Chinacom 2014 and Qshine 2016 Best Paper Awards.

BIN WANG received the B.S. degree from Hohai University, Changzhou, China, in 2019, where he is currently pursuing the M.S. degree with the College of Internet of Things Engineering. His research interests include big data analysis, the industrial internet, machine learning, and artificial intelligence.

LEI SHU (Senior Member, IEEE) received the B.Sc. degree in computer science from the South Central University for Nationalities, China, in 2002, the M.Sc. degree in computer engineering from Kyung Hee University, South Korea, in 2005, and the Ph.D. degree from the Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland, in 2010. Until 2012, he was a Specially Assigned Researcher with the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, Japan. He is currently a Distinguished Professor with Nanjing Agricultural University, China, and a Lincoln Professor with the University of Lincoln, U.K. He is also the Director of the NAU-Lincoln Joint Research Center of Intelligent Engineering. His main research fields are wireless sensor networks and the Internet of Things. He has published over 400 papers in related conferences, journals, and books in the areas of sensor networks and Internet of Things. His current H-index is 48 and i10-index is 178 in Google Scholar Citation. He was a recipient of the 2014 Top Level Talents in Sailing Plan of Guangdong Province, China, the 2015 Outstanding Young Professor of Guangdong Province, and the GLOBECOM 2010, ICC 2013, ComManTel 2014, WICON 2016, SigTelCom 2017 Best Paper Awards, the 2017 and 2018 IEEE Systems Journal Best Paper Awards, the 2017 Journal of Network and Computer Applications Best Research Paper Award, and the Outstanding Associate Editor Award of 2017, 2018 IEEE ACCESS. He has also served over 50 various Co-Chair for international conferences/workshops, such as IWCMC, ICC, ISCC, ICNC, Chinacom, especially the Symposium Co-Chair for IWCMC 2012, ICC 2012, the General Co-Chair for Chinacom 2014, Qshine 2015, Collaboratecom 2017, DependSys 2018, and SCI 2019, the TPC Chair for InisCom 2015, NCCA 2015, WICON 2016, NCCA 2016, Chinacom 2017, InisCom 2017, WMNC 2017, and NCCA 2018; and a TPC member of over 150 conferences, such as ICDCS, DCOSS, MASS, ICC, GLOBECOM, ICCCN, WCNC, and ISCC.