IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 4, 2020, accepted March 20, 2020, date of publication March 25, 2020, date of current version April 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2983184

Traffic Congestion Forecasting in Shanghai Based on Multi-Period Hotspot Clustering

CHUNHUI XU, ANQIN ZHANG, AND YU CHEN
College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200090, China
Corresponding author: Chunhui Xu (1904414045@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772327 and Grant 61532021.

ABSTRACT Traffic congestion has become increasingly prominent. Effective prediction of road congestion will provide a great reference for urban road planning and residents' travel. Taxi trajectory data objectively reflect the travel routes of the residents of a city. With good temporal and spatial characteristics and high timeliness, these data have become important in the study of urban spatio-temporal characteristics. In this paper, the PageRank-K clustering algorithm is used to analyse hotspots and cluster hotspots of a region. The shortest-distance matching algorithm based on the transition probability is used for map matching; then, the average speed of the road sections can be calculated, and the direction of taxis can be determined. The Dual_XGBoost model is used for traffic congestion prediction. Finally, we compared our model with similar models. We take roads in Shanghai as the test object. The results show that our method is faster in training, can predict congestion at any time and is more sensitive to long-term features than the other models.

INDEX TERMS Hotspot area, map matching, taxi trajectory, OD points, PageRank, XGBoost.

I. INTRODUCTION

The expansion of urban space and the increase in population density make traffic congestion a particularly urgent problem. Accurate prediction of traffic congestion can provide a reference for urban traffic management. Traffic congestion prediction can help to effectively prevent and mitigate traffic congestion by adjusting the time of red streetlights during peak and normal hours and dispatching traffic police in advance to roads that may be congested. It can also reduce the costs and travel times of urban residents and guide their travel. The emergence of intelligent transportation has made traffic congestion predictions develop more rapidly. An intelligent transportation system is a comprehensive traffic management system proposed in recent years. It can accurately detect a large range of road network situations in real time and can learn independently according to historical data; then, the real-time and predicted road network situations are communicated to users. To achieve this goal, intelligent transportation systems combine many advanced technologies, such as information technology, computer technology and other monitoring technologies. In the future, intelligent transportation systems will be an important tool for transportation operation and management.

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

As an important part of urban traffic, taxis are inexpensive and flexible and are the preferred means of transportation for many city residents. Taxi trajectory data are mainly obtained by collecting GPS signals at fixed intervals for different taxi trajectories. With the wide application of GPS, increasingly more GPS data have been generated. These GPS data can be applied to research on intelligent transportation and behaviour analysis of residents' daily travel. Castro, P. S. *et al.* groups the existing body of work into three categories: social dynamics, traffic dynamics, and operational dynamics [1]. Then, they introduce and analyse the research studying taxi trajectories and the development of these works. Wei Chen *et al.* introduces the basic concept and pipeline of traffic data visualization and provides an overview of related data processing techniques [2].

Taxi trajectories have both a temporal and spatial distribution, so taxi trajectory data analysis can be used to derive changes in the city in the temporal and spatial dimensions. Thanks to 24-hour uninterrupted service and the wide range of movement of taxis in cities, taxi trajectory mining has great research value for urban traffic.

Changes in urban road conditions have periodicity in the temporal dimension and follow a certain law in the spatial dimension. Feng Mao *et al.* proposed that the travel of urban residents has a certain temporal periodicity [3]. On complete daily timetables, there are traffic peak hours, such as rush

hours, as well as late-night calm traffic hours. The change in traffic flow over time is reflected not only over one day but also in the week. During the week, Monday to Friday are working days, so most urban residents travel to their companies for work and go home after work. Relatively few people go to entertainment venues. On Saturdays and Sundays, most urban residents do not need to go to work. The main travel activities of urban residents are to enter entertainment venues, so the traffic flow will have a different distribution in the two time periods. In terms of the spatial distribution, Guande Qi *et al.* initially considered three typical category areas [4]: (1) long-distance bus/train stations; (2) entertainment areas; and (3) scenic spots. The analysis of taxi trajectories can accurately analyse the number of taxis going to and from these areas and the social activity patterns of a region. Nicholas Jing Yuan *et al.* proposed that because of the continuous development of cities and the continuous changes in the urban industry, the layouts of cities have changed accordingly, and different functional areas, such as residential areas, entertainment areas, and industrial areas, have been developed [5]. Every day in cities, residents are shuttling between these functional areas for work [3], social activities, shopping [5] and other activities.

In most traffic congestion predictions, only time series data are considered, such as LSTM and GRU. Z. Abbas *et al.* proposed and compared three models for short-term road traffic density prediction based on long short-term memory (LSTM) neural networks [6]. R. Fu *et al.* used LSTM and gated recurrent unit (GRU) neural network methods to predict short-term traffic flow [7]. In other traffic congestion predictions, only spatial-influencing factors are considered. Toncharoen Ratchanon and Piantanakulchai Mongkut applied a CNN model to predict traffic states [8]. Song Changhee *et al.* constructed a CNN-based model to predict traffic information [9]. There are also prediction methods that consider various factors. Toon Bogaerts *et al.* proposed a deep neural network that simultaneously extracts the spatial features of traffic using a graph convolution network and its temporal features by means of LSTM to make both short-term and long-term predictions [10]. However, they did not combine time-influencing factors, spatial-influencing factors and other factors to predict traffic congestion.

In our method, a traffic congestion prediction algorithm based on multi-period hotspot clustering is proposed. We combine spatio-temporal influencing factors, workdays and angles to predict traffic congestion. The reasons for choosing these factors are described in detail in the following sections of the paper. Finally, Shanghai is taken as a case study to objectively analyse the advantages and disadvantages of the method.

Our work in this paper is as follows:

(1) The combination of the PageRank-K algorithm and intermediate trajectories is proposed to mine hotspots and potential hotspots.

(2) The transition probability is introduced to improve the map-matching algorithm based on the shortest distance, which makes the map-matching algorithm more accurate.

(3) Dual_XGBoost is applied to traffic congestion prediction for the first time.

II. LITERATURE REVIEW

Among current hotspot mining algorithms, there are density-based algorithms. Linjiang Zheng *et al.* first added a grid to the map and set the density threshold of the grid cells, then mapped the trajectory points to the grid cells and extracted the hotspot grid cells based on the density threshold [11]. Finally, the hotspots of the city were found by merging the reachable hotspot grid cells. Gui, Zhiming *et al.* proposed a MapReduce-based DBSCAN [12] framework for discovering traffic hotspots from taxis trajectories, which first extracts the dense part as a stop point from each trajectory and then groups the stop points into traffic hotspots [13]. Y. Shen extracted hotspots for loading and unloading passenger(s) using an improved DBSCAN algorithm (GADBSCAN), which is very suitable for extracting urban tourism hotspots from floating car data in cities [14]. There are also other algorithms based on the measure of global and local spatial autocorrelation. Xu, Zhimin *et al.* proposed spatio-temporal hotspot region discovery methods based on the automatic detection of intensive time intervals [15]. First, they determined which road segment the vehicle is travelling on. Then, they estimated the travel speed and created spatial weight matrices. Finally, measures of the global and local spatial autocorrelation were used to evaluate the spatial distribution of the traffic condition and reveal the traffic hotspots.

Currently, there are many map-matching algorithms; some are based on probability, and some are based on distance. Among the probability-based map-matching algorithms, the HMM algorithm [16] is the most common method. The algorithm undergoes three steps for each trajectory in the dataset: computing the emission probability, computing the transition probability, and computing the result by means of the Viterbi algorithm. Francia, M *et al.* proposed a distributed extension of the enhanced map-matching algorithm with a hidden Markov model for mobile phone positioning and overcame its limitations, which enhances the robustness of the algorithm [17]. Qi, H *et al.* performed junction matching according to road widths, angles and other information and used it to improve the HMM algorithm, which effectively increases the accuracy of the method [18]. Among the shortest-distance-based algorithms, Mohammed Quddus *et al.* proposed the stMM algorithm to improve the matching accuracy, in which the difference between the distance along the shortest path and the distance along the vehicle trajectory and the differences in vehicle trajectory headings are considered [19].

Many scholars have proposed traffic prediction algorithms. Zhang Yong-chuana *et al.* studied a method to analyse traffic congestion by taxi trajectory data [20]. The whole method is divided into several steps, such as data cleaning, map matching, speed estimation, and congestion classification,

which can effectively extract information on traffic congestion in cities. Most traffic congestion algorithms go through the above steps. Shuming Sun *et al.* used a map-matching algorithm based on an HMM to match GPS points and used CNN, RNN and other neural networks to predict traffic congestion [21]. Yuankai Wu *et al.* proposed a new model (DNN-BTF) that makes full use of the time periodicity and spatial characteristics of traffic flow [22]. Then, an attention-based model was introduced to automatically determine the importance of past traffic flow. Nicholas G. Polson *et al.* developed a new deep learning architecture that combines a linear model fitted using a series of tanh layers and L1 regularization to predict short-term traffic flow [23]. The first layer of the architecture identifies spatio-temporal relations among predictors, and the other layers model nonlinear relations. Yu Bing *et al.* proposed a new framework, STGCN, to solve the problem of traffic forecasting [24]. Two layers of spatio-temporal convolutional blocks were used to mine the spatio-temporal characteristics of the data. Each spatio-temporal convolutional block combines a GCN and GLU. This framework effectively improves the prediction effect. Di Zang *et al.* proposed a multi-scale spatio-temporal feature learning network (MSTFLN) that consists of convolutional long short-term memory and convolutional neural networks as the model to predict long-term traffic speed for elevated highways [25]. In the MSTFLN, the speeds of multiple loop detectors over multiple time intervals are used to generate a matrix, and multiple different time scale matrices are used as the learning data to predict traffic speeds on a certain day. Yuanli Gu *et al.* introduced entropy-based grey relation analysis to choose lane sections strongly correlated with the lane section to be predicted and then established a new framework by combining an LSTM neural network and a GRU neural network to predict the traffic speed [26]. Jingyuan Wang *et al.* proposed a new structure (eRCNN) [27]. In the eRCNN, the spatio-temporal traffic speeds of contiguous road segments are integrated as an input matrix. By further introducing separate error feedback neurons to the recurrent layer, eRCNN can learn from the prediction errors. Through these two steps, the prediction accuracy rate is significantly improved. Loan N.N. Do *et al.* used an attention mechanism to mine the temporal and spatial characteristics of data with their proposed model, thereby improving the prediction effect of the model [28]. Xiong, Haoyi *et al.* noted that traffic congestion could propagate across roads [29]. To solve the prediction problem, they predicted the footprint of congestion propagation as propagation graphs and proposed the PPI_Fast algorithm. Zhou, Xun proposed the concept of a G-graph to describe the problem and used the gathering score to judge the possibility of gathering events appearing at a certain position during aggregation [30]. Then, they proposed the SmartEdge algorithm to efficiently discover the top-k G-graphs. A year later, Khezerlou, Amin Vahedian *et al.* proposed a novel simulation-based evaluation approach to study the performance of the SmartEdge algorithm, including the timeliness and location accuracy [31]. Their method can be used to predict not only traffic congestion but also other gathering events, such as traffic accidents, concerts and sports events.

How to define traffic congestion is a difficult issue. Different people have different definitions of congestion, according to a survey by Bertini [32]. Most survey respondents listed time, speed, volume, level of service (LOS) traffic signal cycle failure (meaning that one has to wait through more than one cycle to clear the queue) as the main reference for a congestion definition. Michael A.P. Taylor described traffic congestion as the phenomenon of increased disruption of traffic movement on an element of the transport system [33], [34]. This phenomenon is caused by the interaction of traffic flow or traffic elements in traffic flow. Michael A.P. Taylor also mentioned some of the causes of congestion. These causes can be divided into two types: one is the occurrence of incidents, such as breakdowns, road works, or road crashes, and the other is the peak periods.

III. METHODOLOGY

A. ALGORITHM DESCRIPTION

In the method mentioned in Section II, there are many methods for traffic prediction, including those based on deep learning and those based on multi-scale spatio-temporal features. Most of these methods are based on the spatio-temporal characteristics of the data. However, when spatio-temporal data have a large impact on congestion, other factors will also have an impact on congestion to some extent, such as whether it is a workday or not and the angle of the vehicle.

A traffic hotspot area means a high population density, high traffic flow and a high probability of accidents. These factors will have a large impact on traffic. There will be different hotspots in different periods of time in a city. The hotspots of daily commuting times are concentrated in residential areas, work areas and their transportation hubs. During the week, on non-working days, people are mainly concentrated in entertainment areas. In cities, most traffic congestion occurs in hotspots or traffic hubs between hotspots. Therefore, multi-period hotspot area mining can be used for traffic congestion prediction. In our method, a traffic congestion prediction algorithm based on multi-period hotspot clustering is proposed. The algorithm combines spatio-temporal and other influencing factors. Our method pre-processes urban road network information and taxi trajectory data by using Spark. The processed data are divided into 96 15-minute periods. The hotspot clusters in different time periods fully reflect the changes in traffic in the temporal and spatial dimensions. The combination of the PageRank algorithm and intermediate trajectories is used to mine the attractiveness of the regions in each time period, and the K-means clustering algorithm is used to cluster the regions by their attractiveness values in each time interval. Then, the shortest-distance matching algorithm based on the transition probability is used to perform map matching on the track points of the taxi, the road average speed is calculated, and the travelling direction of the taxi is

determined. The results are input into the XGBoost_SVM, Dual_XGBoost, and XGBoost_BP neural network models for traffic congestion prediction. Finally, taking the large dataset of taxi trajectories in Shanghai as a case study, the proposed method is compared with some existing methods to objectively analyse the advantages and disadvantages of our method.

Combining the Shanghai traffic situation with R. L. Bertini's definition of traffic congestion and Michael A.P. Taylor's description of traffic congestion, the conditions for judging traffic congestion in this paper are as follows:

- The speed threshold of traffic congestion is 10 km/h. An average road speed less than 10 km/h can be judged as congestion.
- Two or more cars on a certain section of the road have a large deviation in the driving angle, and these cars stay on the road for more than 15 minutes.

B. HOTSPOT AREA CLUSTERING BASED ON PageRank-K

Hotspots are areas that are attractive to people and have a high population density in a city. Regions are given a value to measure their attractiveness, and traffic congestion is more likely to occur in regions with higher attractiveness values. In this paper, the PageRank-K algorithm is proposed to calculate the attractiveness of the regions, and then the regions are clustered according to their attractiveness.

The PageRank-K algorithm is based on the improved PageRank algorithm and the K-means clustering algorithm. The PageRank algorithm was developed by Google to rank web pages on the Internet according to their popularity [35]. The K-means clustering algorithm is the most widely used partitioning-based clustering algorithm, which divides n objects into K clusters to make them have a high similarity.

There are many hotspot algorithms, but most of them are based on OD points. However, for traffic prediction, the hotspots based on OD points alone cannot fully show the traffic situation. Since residents will travel on many roads in a single trip and these roads may be congested by travelling residents, the analysis of taxi trajectories between the origin point and destination point becomes very important. As shown in Fig. 1, area C, as a departure area and destination area, has low attractiveness; however, because the paths of areas A and B, and D and areas E, F and G intersect in area C, traffic congestion is still prone to occur in area C. A region similar to area C, a transportation hub with low attractiveness, is called a potential hotspot area. In this paper, hotspots and potential hotspots are mined by the improved PageRank algorithm.

In the general PageRank algorithm, links between pages are hyperlinks and do not consist of other pages. In Fig. 2, the connection between web page A and web page B indicates that A has hyperlinks to B, but web pages and hyperlinks are disparate. Algorithms for web pages cannot be applied to hyperlinks between web pages. However, the connections between the various areas on a map consist of other areas between the two areas. As shown in Fig. 3, a person from area A travelling to another area B must pass through the regional sequence ⟨C, D, E, F, G⟩, which is the route of departure area A and destination area B. Areas C, D, E, F and G can also be departure areas and destination areas, so each area itself is not only a departure and destination area but also a part of the route between other departure areas and destination areas. That is, each area has both attractiveness as a departure area and destination area and attractiveness as a part of the route between other departure areas and destination areas. In the following sections, the attractiveness of the area as a transport hub is called the potential attractiveness. The route is composed of areas, so the areas that make up the route can also have regional characteristics that can be calculated

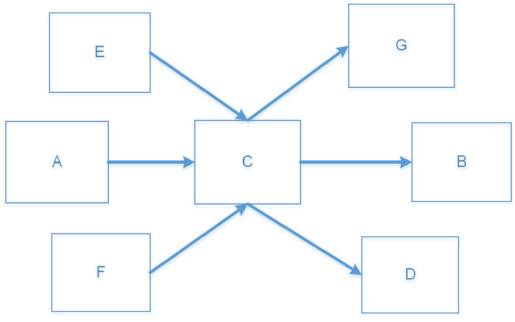


FIGURE 1. Hot spots and potential hot spots. A, B, D, E, F, G are hot spot, and C is a traffic hub. The arrow lines in the figure represent the traffic flow.

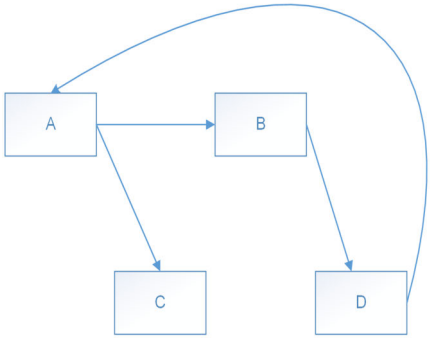


FIGURE 2. PageRank on the web. A, B, C, D are webpages, The arrow line in the figure represent hyperlinks.

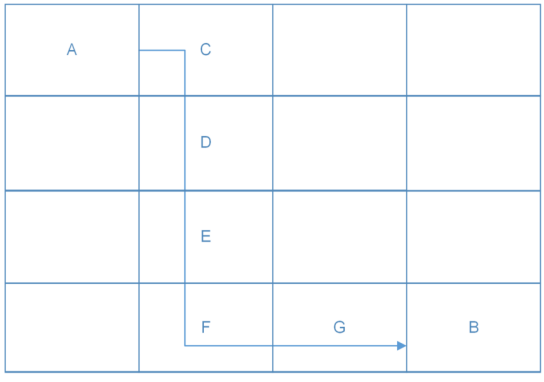


FIGURE 3. The taxi's mobile route. A is the departure area, B is the destination area. C, D, E, F, G make up the path from A to B.

using the algorithms for regions. The PageRank algorithm is used to calculate the attractiveness value and the potential attractiveness value of a region; these PR values together constitute the complete PR value of the area.

The detailed algorithm of PageRank-K is given below.

First, the entire urban space is divided into multiple polygons, which are called ''grid cells''. The longitude range of Shanghai is [120°51', 122°12'], and the latitude range is [30°40', 31°53']. The longitudes and latitudes are divided into sections in intervals of 0.02°. The dividing lines of the latitudes and longitudes divide the map into multiple cells. Each cell is a quadrilateral with a distance between the four vertices of approximately 2.23288888 km. The latitude and longitude coordinates of the four vertices are recorded, which is the process of traffic gridding.

Second, each cell acts as an area. The OD points are clustered according to the grid cells. The OD points within the same area are gathered as a cluster. Each area is used as a cluster to calculate the attractiveness to people based on the number of OD points in the cluster. Let $num_O^i = |O_i|$ denote the number of O points in area i, where $num_D^i = |D_i|$ is the number of D points in area i.

Third, the attractiveness of the area as a hotspot is calculated. num_O^i is used as the initial attractiveness PR value of area i because the number of O points in each area reflects the initial attractiveness of the area. With each area as the departure area, the number of people from the area travelling to each of the remaining areas (except itself) is calculated: $num_{i,j}$ represents the number of people starting from area i and ending at area j. The new attractiveness PR value of the area as a hotspot is as in (1).

$$PR(area_j) = (1-d)/n + d\sum_{i=0}^{n}(num_{i,j}/num_O^i)\cdot PR(area_i), \quad i = 1, 2, \cdots, n; j = 1, 2, \cdots, n \qquad (1)$$

where PR($area_j$) is the PR value of area j; d is the damping factor, which takes a value of 0.85; and n is the total number of areas. Some drivers will wander in the same area, such as in front of supermarkets, around airports, to find passengers. The GPS signals are recorded continuously in the same area, which is similar to a web surfer constantly clicking on the same link. The GPS recorder in some taxis may be defective, causing data to be missed. Some GPS records with large deviations were removed during data pre-processing. These situations are as if the surfer suddenly quit browsing the web. Therefore, we also need the damping factor. According to the PageRank algorithm, we also set it to 0.85.

Similarly, M, which is the intermediate taxi trajectory between the origin point and the destination point, is clustered according to the traffic grid. The M points within the same area are gathered as a cluster. Let $sum_M^i = |M_i|$ indicate the number of M points in region i. Let $sum_{i,j}$ denote the number of people travelling from area i to area j during the trip. An area will be part of multiple paths. For the same reason, we also need damping factor here. The new potential attractive PR value of the area as the traffic hub is as in (2).

$$PR(side_j) = (1-d)/n + d\sum_{i=0}^{n}(sum_{i,j}/sum_M^i)\cdot PR(side_i), \quad i = 1, 2, \cdots, n; j = 1, 2, \cdots, n \qquad (2)$$

The complete PR value of a region is a combination of the potential attractiveness PR value and the attractiveness PR value. The calculation formula of the PR value is as in (3).

$$PR(i) = a\cdot PR(area_i) + b\cdot PR(side_i), \quad i = 1, 2, \cdots, n \qquad (3)$$

where a and b are parameters and both of them are set to 0.5. Setting different values of a and b will produce different attractiveness values, and there will be different clustering results when clustering in hotspots. There is no clear criterion to judge which clustering effect is better. However, here, we need to emphasize that traffic hubs and hotspots have the same impact on traffic congestion, so we use equal weights for combining the PR scores.

Finally, the results of the PageRank algorithm are recorded as the final and complete attractiveness of the region, and the K-means clustering algorithm is used to cluster the attractiveness values of the region. The sum of the squared errors (SSE) is used as a core indicator for selecting the number of clusters. K will have different optimal values for different time intervals. In most cases, when k is equal to 4, the trend of SSE changes the most. To facilitate the calculation of statistics and the implementation of the method below, k is set to 4.

The pseudocode of the improved PageRank is illustrated in Algorithm 1.

Algorithm 1 The Pseudocode of the Improved PageRank

Require: sum[][],sumO[],lis1.size(),edgeWeight[], edge[][];
Ensure: The PR value of each area P[k];
P0 []=sumO[]; //Initialize attractiveness for each area;
P1 []=edgeWeight[]; //Initialize potential attractiveness for each area;
P []; //Initialize total attractiveness for each area;
k=1;
repeat
p0[k]=(1-0.85)E+0.85A^Tp0[k-1]; // A is the matrix form of sum[][],E is a square matrix of n rows and n columns with all 1;
p1[k]=(1-0.85)E+0.85B^Tp1[k-1]; // B is the matrix form of edge[][];
p[k]=(p1[k]+p0[k])/2;
k=k+1;
until || P[k]-p[k-1]|| < ε

C. CALCULATING AVERAGE SPEED BASED ON ROAD MATCHING ALGORITHMS

A hidden Markov model (HMM) is a statistical model that is very suitable for processing time series data. It describes

the process of randomly generating random sequences from an invisible Markoff chain and generating an observation from each state to generate random sequences [16]. Hidden Markov models are commonly used to predict trajectory points with time series characteristics. The shortest-distance matching algorithm and the map-matching algorithm based on HMMs are the two most commonly used map-matching algorithms. However, there are two shortcomings in existing map-matching methods. First, taxi trajectories are low-frequency data, so it is not very accurate to use the shortest-distance matching algorithm to match taxi trajectories. Second, although the improved HMM can effectively improve the accuracy of map matching, it takes a long time [36]. In addition, HMM-based approaches suffer from latency, so their online map-matching effect is not very satisfactory [37], [38].

Based on these two points, we propose a shortest-distance matching algorithm based on the transition probability of the hidden Markov model. It has the characteristics of fast speed and high accuracy.

OSM [39] is often referred to as the Wikipedia map of the world. OSM was launched in 2004 with the aim of creating an editable map of the whole world and was released with an open content licence (http://wiki.openstreetmap.org/wiki/About). OSM provides researchers with datasets on a global scale and is maintained by a very large collaborative network of volunteers. When processing OSM maps online, there are restrictions on the size of the selected area and the number of roads, so the road network information is downloaded from the OSM map. The downloaded data are stored in XML format, not in vector format. As shown in Fig. 4, the roads in the download data are represented by an ordered sequence of coordinate points.

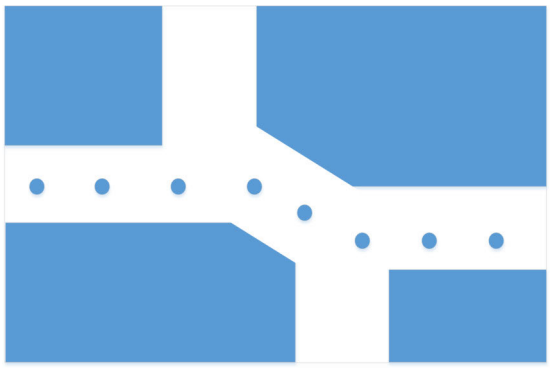


FIGURE 4. Simulation diagram of OSM format data. The points in the figure are the GPS points recorded by the vehicle GPS recorder.

Because the multipath effect and urban canyons are common in taxi trajectories in Shanghai, we need to process GPS data to eliminate their impact. There are currently some techniques for dealing with errors caused by multipath effects and urban canyons. S. Syed and M.E. Cannon used fuzzy logic to solve this problem [40]. Andrew Rae used a Kalman filter to fuse data from a GPS receiver and a machine vision system for map matching [41]. In this paper, the ultimate goal of map matching is to calculate the average road speed, so small errors do not affect the final result. To reduce the complexity of the entire prediction system, only GPS signals with large errors are corrected before map matching. A GPS signal with a distance of more than 20 m from the nearest road is regarded as a GPS point with a large error. Since the exact position of the GPS point cannot be known, it must be maintained while being corrected. The GPS point is corrected to a position 20 m from the nearest road along the vertical line from the GPS point to the nearest road.

These points are connected by two adjacent points. As shown in Fig. 5, each adjacent point is regarded as a state, and a road consists of several states. Each state has its own unique numerical ID. From west to east and from north to south, the IDs of the states on the same road are arranged from smallest to largest, and the absolute value of the ID difference between two adjacent states is equal to 1.

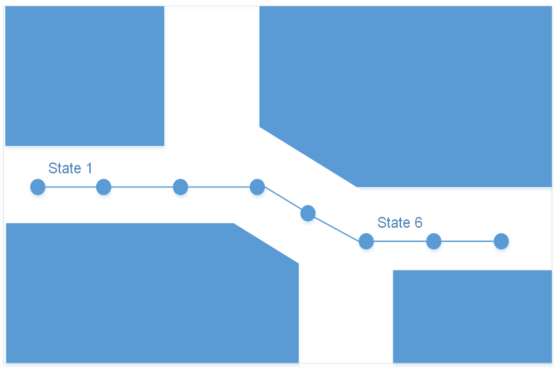


FIGURE 5. Multiple states make up a road. Connect two adjacent GPS points to form a state. Connect the adjacent state end to end to form a road.

The accuracy of GPS ranges from several metres to tens of metres in civil use [42]. Here, we set 20 m as the search area. For a given GPS point, the state of 20 m around the GPS coordinate can be regarded as the candidate state of the GPS point, as shown in Fig. 6.

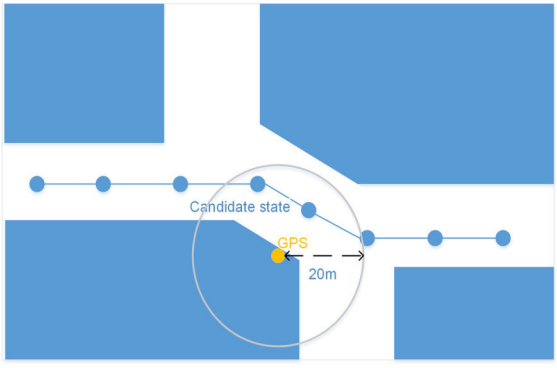


FIGURE 6. Candidate states. States within 20 meters of the GPS point are candidate states.

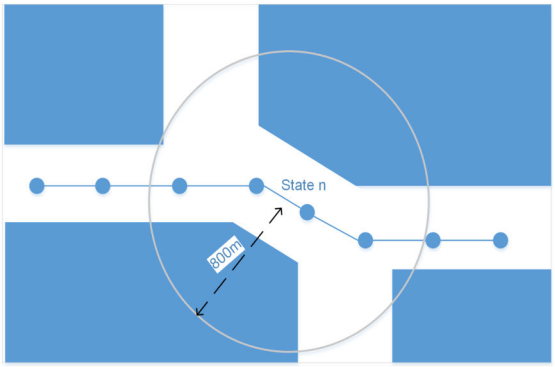


FIGURE 7. Transition probability. The state within 800m from state n has a transition probability with it.

The transition probability of the hidden Markov model is used to indicate the probability of a taxi moving from state A to state B.

The transition probability is related to the distance between the states. To set a maximum range, only the two states within the maximum range distance have a transition probability. As shown in Fig. 7, a circle is drawn with state n as the centre and a radius of 800 m. Only the state in the circle has a transition probability with state n. The transition probability is affected by the distance between states. Taking into account the average speed of the taxi and the road conditions in Shanghai, the taxi can travel 800 min one minute, so the maximum distance is set to 800 m. The transition probability between states is as in (4).

$$P_t(r, r') = \frac{1}{\beta} exp^{-\frac{d_t}{\beta}} \tag{4}$$

where d_t represents the distance between two states r and r', and β is a tuneable parameter, whose value is 0.01 in this paper.

For a GPS point, the GPS positioning is very accurate, and the distance from other candidate states is large if the distance between the GPS point and the nearest state is less than 1.5 m, or the difference between the minimum distance and the next smallest distance is greater than 2 m. The state closest to the GPS point is the matching state.

For a GPS point, if the distance between the GPS point and the nearest state is less than 5 m, or the difference between the minimum distance and the next smallest distance is greater than 5 m, the GPS positioning error is small. $Pos_{l,a}$ is calculated, which indicates the possibility of matching the candidate state a as the next state based on the last matching state l. $Pos_{l,a}$ is calculated as in (5).

$$Pos_{l,a} = \frac{P_t(l, a)}{|l - a|} \tag{5}$$

where l and a are the IDs of state l and state a, respectively. The candidate state with the maximal Pos is chosen to be the next matching state. As shown in Fig. 8, assume that state 1 is the state matched to the previous point G1, and the distance between the next point to be matched (G2) to state 5 and state 9 is the same. Obviously, state 5 is the correct candidate

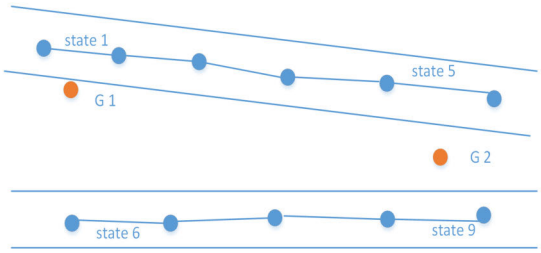


FIGURE 8. Application of state ID in road matching. The state that matches G2 correctly is state 5, even though the distance from G2 to state 5 and state 9 is the same.

state. The absolute value of the difference between the ID of state 5 and the ID of state 1 is 4, and the absolute value of the difference between the ID of state 9 and the ID of state 1 is 8. The calculation between state IDs can prevent points on one road from being matched to other roads. Since the absolute value of the difference between state IDs is small and the matching probability is large, we use the absolute value of the difference between state IDS in the denominator.

When the above conditions are not satisfied, the positioning error of the GPS point is large; flagx and flagy are used to further calculate the correlation between candidate states and GPS points. We compare the longitude and latitude of the GPS point with the longitude and latitude range of each candidate state. If the longitude of the GPS point is within the candidate-state longitude range, then flagx is 1; otherwise, it is set to 0, and if the latitude of the GPS point is within the candidate-state latitude range, then flagy is 1. At this point, the calculation of $Pos_{l,a}$ is as in (6).

$$Pos_{l,a} = \frac{AP_t(l, a)}{|l - a|} + B(flagx + flagy) \tag{6}$$

where A and B are parameters; it is found by experiment that A is 0.9 and B is 0.1.

When the GPS point of the taxi appears on a certain road for the first time, the time is recorded. When one of the remaining GPS points in the sequence is not on the road for the first time, the time of the GPS point is recorded, and the time interval is calculated. The length of the road is the sum of the lengths of the states that make up the road. The average speed of the road is calculated as in (7).

$$\bar{v}_i = \frac{\sum_{j=1}^{n} \frac{L_i}{t_b^j - t_a^j}}{n} \tag{7}$$

where L_i is the length of road i, t_b^j is the time that taxi j leaves road i, t_a^j is the time taxi j enters road i, and n is the number of taxis on road i.

The detailed flow chart of the algorithm is shown in Fig. 9.

D. PREDICT TRAFFIC CONGESTION USING Dual_XGBoost

XGBoost is a scalable tree boosting machine learning system that is useful for the vast majority of regression and classification problems [43]. It is also an optimized distributed gradient enhancement library designed to be efficient, flexible and portable.

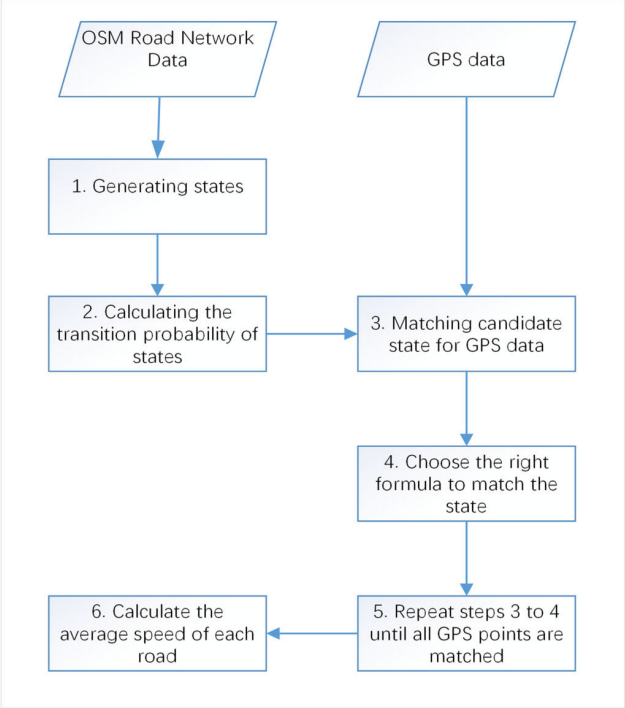


FIGURE 9. The process of average speed calculation. The calculation of road average speed can be divided into six steps.

Because a single decision tree model is prone to overfitting and cannot be effectively applied in practice, XGBoost is a learner that combines multiple CART trees. We have no precedent for reference to determine the number of CART trees in an XGBoost, the depth of each CART tree and the learning rate of XGBoost. Training these parameters one by one takes much time. Therefore, we use a random search to find the optimal values of these parameters. We set the number of CART trees to [1,100], the depth range of each CART tree to [1,10] and the range of the XGBoost learning rate to [0.01,0.1]. The random search algorithm inputs these values into the model and trains them one by one. Then, the optimal parameters of the model are found and saved.

In our method, the workday indicator, time, angle and area clustering variables are input into the XGBoost model to predict the average road speed, and then the results are combined with the angle, area clustering, time and workday indicator to be used as the final data for road congestion prediction. Finally, the final data are input into a BP network, SVM or XGBoost model to analyse road congestion and to select the best prediction method.

Experiments show that Dual_XGBoost has the best prediction effect.

The prediction model is shown in Fig. 10.

E. EVALUATION METHOD

The evaluation measures we use are precision, recall, f1-score.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{8}$$

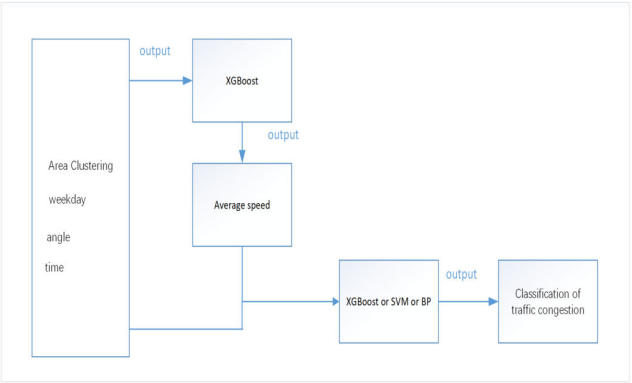


FIGURE 10. The prediction model. The first XGBoost predicts average road speed, and the second XGBoost predicts congestion.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{9}$$

$$f1 - scor = \frac{2 \cdot Precision \cdot Recall}{Precision \cdot Recall} \tag{10}$$

We also calculated the proportion for the wrong classification.

$$W_r = \frac{X_Y}{Y} \tag{11}$$

where Y is the number of samples in class Y and Y_x represents the number of samples that should have been classified as Y but were incorrectly classified as X.

$$W_p = \frac{X_Y}{X} \tag{12}$$

where X is the number of samples in class X.

IV. EXPERIMENT

A. DATA

In this paper, Shanghai taxi GPS data and road condition data are used. The Shanghai taxi GPS data are from December 28, 2014, to January 10, 2015. The data include approximately 50000 taxis, and each vehicle is recorded every 1 minute.

The data format is shown in Table 1.

TABLE 1. Data format.

Index	Field name	Meaning
1	ID	Serial number (recorded number, negligible)
2	LONGITUDE	Longitude (Unit: degree)
3	LATITUDE	Latitude (unit: degree)
4	COMPANY	Company (taxi company number)
5	CARID	Taxi ID
6	DATE_TIME	Time (accurate to the second)
7	SPEED	Speed (unit: km/h)
8	DIRECTION	Direction (value range 0-360, 0: north, 90: east, 180: south, 270: west)
9	STATUS	Empty car logo (0 is empty car, 1 means carrying passengers)
10	COLUMN1	Complement (meaningless)
11	COLUMN2	Complement (meaningless)

B. DATA PREPROCESSING

The data collected from the actual traffic situation in Shanghai have many outliers and redundant data caused by traffic and human factors and cannot be directly used for trajectory data analysis and mining. Therefore, the taxi trajectory data should be pre-processed. Because Spark is a fast and universal cluster computing platform, it is suitable for processing and analysing big data. This paper uses Spark to pre-process the big traffic data. The data pre-processing steps are as follows.

- Taxi trajectories are time series data, and the extraction of OD pairs needs to be performed in an ordered taxi trajectory. The recorded GPS data are disordered, so it is necessary to sort the data from the car ID and time.
- This paper takes Shanghai as the research object, and the key research area is within several districts of Shanghai. An origin point or a destination point of the taxi trajectory not in Shanghai is removed. The taxi trajectories that leave Shanghai during the journey need to be removed.
- Li Bin *et al.* found that taxi drivers mainly used two strategies when finding passengers, namely, hunting and waiting [44]. In some cases, drivers will leave their taxis, such as having lunch and going to the supermarket to buy cigarettes. The taxi trajectories shows that many taxis stay in one place for a long time without moving. It is considered that there is no driver in the car while the GPS recorder is still working, or the driver is waiting for passengers. We set the time threshold for cars staying in the same GPS location to 30 minutes. GPS records that exceed the time threshold and do not move do not contributed to congestion prediction and should be removed.
- Not all parts of the data are useful, so useless data should be removed, which can reduce the dimensions of the data. Company, column 1, and column 2 are removed.
- OD pairs and intermediate trajectories need to be processed separately, so OD pairs should be extracted from the data.
- This paper uses multi-period hotspot clustering, so the data should be divided into multiple groups by time interval. Each time interval is 15 minutes.
- As the buildings in the city will block GPS signals or the GPS recorders may be defective, some abnormal data are recorded. These data should be removed.

The processed OD point data are shown in Fig. 11.

C. CLUSTERING OF URBAN TRAFFIC HOTSPOTS

The map of Shanghai is divided into multiple polygons known as ''grid cells''. The difference between the longitude and latitude of each cell is 0.02°. The range of longitude can be divided into 68 intervals. The range of latitudes can be divided into 60 intervals, for a total of 4080 zones. The longitude and latitude coordinates of the four vertices for each cell are recorded. This is the process of map gridding. The effect of map gridding is shown in Fig. 12.

longitude	latitude	car ID	time	speed	angle	flag
121.43162	31.306597	00053f5e11d1fe4e49a221165b39abc9	20141228205953	0	270	1
121.43164	31.306578	00053f5e11d1fe4e49a221165b39abc9	20141228210053	0	270	0
121.444405	31.310308	00053f5e11d1fe4e49a221165b39abc9	20141228210533	45	360	1
121.42491	31.368666	00053f5e11d1fe4e49a221165b39abc9	20141228210753	0	270	0
121.436104	31.31858	00053f5e11d1fe4e49a221165b39abc9	20141228210833	31	180	1
121.41173	31.221647	00053f5e11d1fe4e49a221165b39abc9	20141228211053	0	90	0
121.4134	31.220469	00053f5e11d1fe4e49a221165b39abc9	20141228214033	28	90	1
121.378975	31.132845	00053f5e11d1fe4e49a221165b39abc9	20141228215133	0	270	0
121.42997	31.237026	00053f5e11d1fe4e49a221165b39abc9	20141228220813	0	90	1
121.41267	31.220121	00053f5e11d1fe4e49a221165b39abc9	20141228223213	0	225	0
121.42246	31.219168	00053f5e11d1fe4e49a221165b39abc9	20141228224253	45	90	1
121.42685	31.23779	00053f5e11d1fe4e49a221165b39abc9	20141228231033	0	45	0
121.42911	31.236841	00053f5e11d1fe4e49a221165b39abc9	20141228182845	19	270	1
121.43959	31.227737	00053f5e11d1fe4e49a221165b39abc9	20141228191627	0	225	0
121.43007	31.236973	00053f5e11d1fe4e49a221165b39abc9	20141228194047	0	270	1
121.31651	31.195145	00053f5e11d1fe4e49a221165b39abc9	20141228195707	15	270	0

FIGURE 11. The OD pairs. In the figure, when flag is 1, this record is the origin point, and when flag is 0, it is the destination point.

FIGURE 12. Map gridding. The whole map is divided into several traffic grids.

For partitioned grids, OD points are clustered according to the map grids. If O points (D points) are in a quadrilateral formed by four latitude and longitude coordinates, these O points (D points) are clustered together. All OD points are clustered into the corresponding map grids. Each grid is treated as a cluster to calculate the attractiveness of each grid to human beings according to the number of OD points in the cluster. As shown in Fig. 13, the black solid dots in the graph represent the O points in the grid, and the other dots are the D points in the grid. Fig. 14 shows the intermediate trajectories of taxis.

The PageRank-K algorithm is used to calculate and cluster the attractiveness of each grid. Fig. 15 shows the certain period distribution of hotspots after clustering. Points of different shapes in the graph represent different clusters. Each point is a square with side lengths of 2.23288888 km. It can be seen that the location of the city centre is a cluster. Each point in the cluster has the greatest regional attractiveness and the greatest potential for congestion. The attractiveness of the area gradually decreases from the city centre to the suburbs. However, some areas far from the city centre also

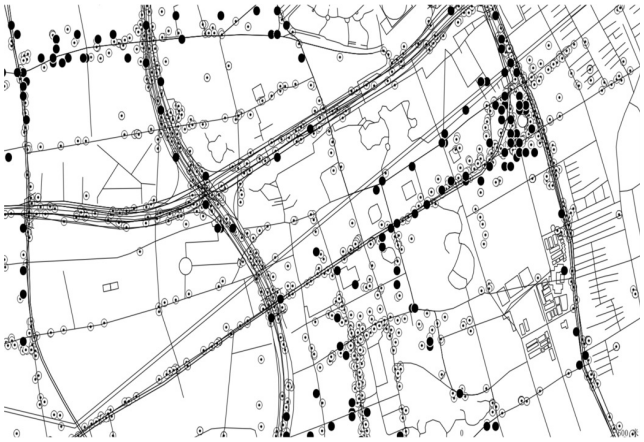


FIGURE 13. OD points in the map. The black solid dots in the graph represent the O points in the grid, and the others are the D points in the grid.

FIGURE 14. Intermediate points in the map. The dots in the graph represent the intermediate points of vehicles.

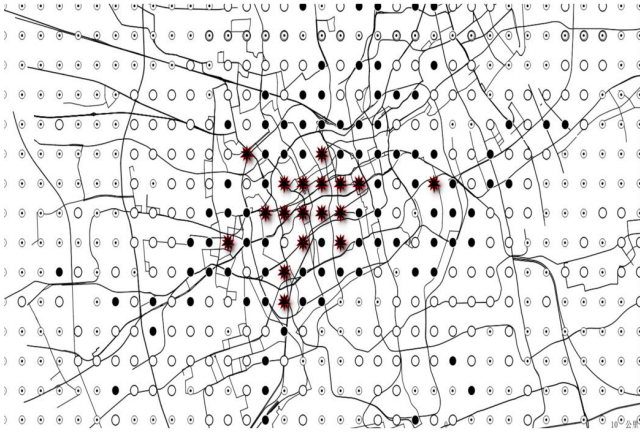


FIGURE 15. The cluster of areas. The point in different shape represents different hot area clustering.

have higher attractiveness. Most of these areas are airports and large markets. The attractiveness values of the outermost areas are the smallest. The distribution of clusters in hotspots on the map conforms to the functional distribution and spatial characteristics of cities.

D. TRAFFIC CONGESTION PREDICTION

XGBoost_SVM, XGBoost_BP and Dual_XGBoost are used to predict road congestion. Traffic congestion is closely related to the average road speed and is one of the conditions for judging traffic congestion. Table 2 shows the relationship between traffic congestion and the average road speed.

TABLE 2. The congestion classification standard.

Congestion level	speed range(km/h)
Unknown(0)	Unknown
Smooth(1)	25 or more
Slightly blocked(2)	[10,25)
Congestion(3)	[0,10)

Because the average speed of each time period is changing, but it is basically within a range, the traffic congestion is judged according to the range of speed, so it is not necessary to accurately predict the speed value. This paper divides the speed values into multiple speed ranges, which can improve the prediction accuracy and ensure the accuracy of the data. Regression learning is transformed into classification learning. The speed range is shown in Table 3. The speed threshold of traffic congestion is 10 km/h. When the speed is less than 10 km/h, the traffic situation is considered congested.

TABLE 3. The speed classification.

Speed classification	speed range(km/h)	Speed classification	speed range(km/h)
0	[0,5)	5	[5,10)
10	[10,15)	15	[15,20)
20	[20,25)	25	[25,30)
30	[30,35)	35	[35,40)
40	[40,45)	45	45 or more

The XGBoost model achieves good speed prediction, and the accuracy rate reaches 93.506%. The prediction results are shown in Fig. 16. Fig. 16 is a confusion matrix. The number on the main diagonal of the matrix is recall rate, and the number not on the main diagonal is W_r. The sum of the values of each row in the matrix is 1.

The predicted speed classification is input into the SVM, BP network, XGBoost models along with data such as the workday indicator, time, and angle. Fig. 17, Fig. 18 and Fig. 19 show the prediction effects on all test data of the XGBoost_SVM, XGBoost_BP neural network, and Dual_XGBoost models. They are all confusion matrices like Fig. 16. The accuracies of the prediction results are shown in Table 4.

TABLE 4. The accuracy of each model.

	XGBoost_SVM	XGBoost_BP	Dual_XGBoost
Accuracy	75.324675%	75.324675%	94.1558%

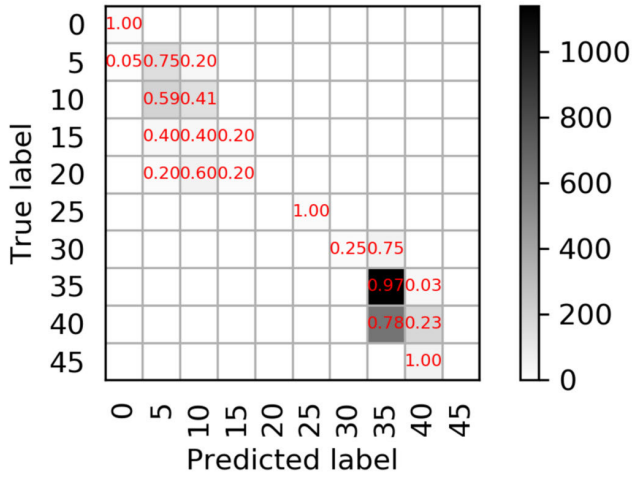


FIGURE 16. Confusion matrix. Prediction effect with XGBoost.

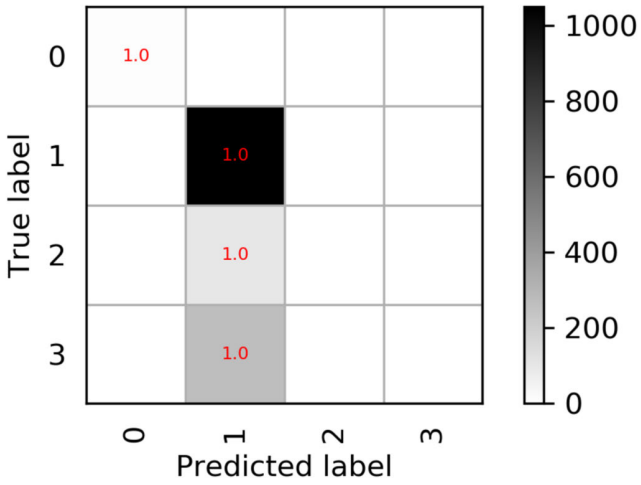


FIGURE 17. Confusion matrix. Prediction effect with XGBoost_SVM.

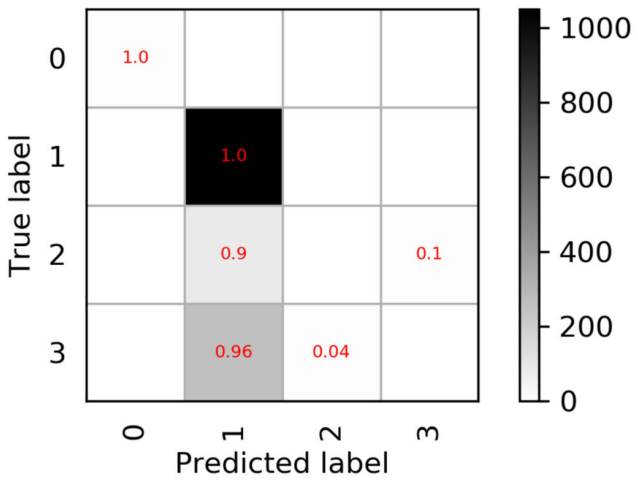


FIGURE 18. Confusion matrix. Prediction effect with XGBoost_BP.

It can be seen from the data that the accuracy of the Dual_XGBoost model is the highest, and the accuracies of the XGBoost_SVM and XGBoost_BP neural networks are all just 75.3246%, which can hardly distinguish congestion. The results indicate that the roads are not congested for most of the day, and road congestion occurs only on certain roads during a certain period of time, but the percentage of congestion is too small. Therefore, there are few samples labelled as congestion in the training sample set, and the distribution of each label in the sample is not uniform, which causes the SVM and BP neural network models to be underfit for traffic congestion learning and overfit for the learning of smooth roads. However, the Dual_XGBoost model still achieves a high accuracy in the case of sparse data or an uneven sample distribution.

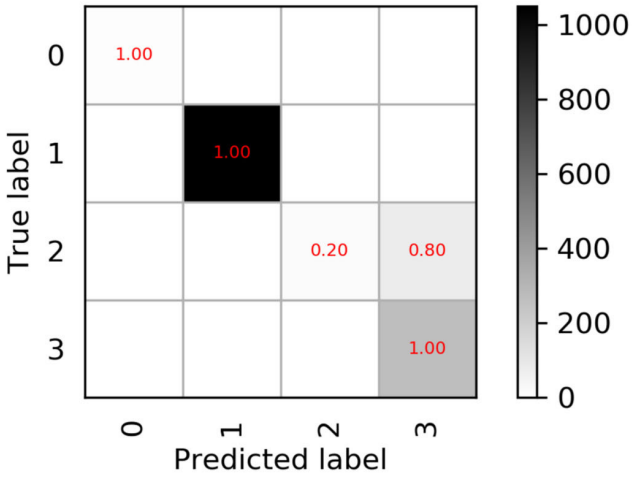


FIGURE 19. Confusion matrix. Prediction effect with Dual_XGBoost.

V. RESULT AND DISCUSSION

Finally, we compared our method with other existing methods.

In the methods of Shuming Sun *et al.*, the CNN learns the spatio-temporal characteristics better than the LSTM model, and the LSTM model has a better learning effect on time series data. We chose the CNN and LSTM neural network of Shuming Sun *et al.* and the random forest model mentioned by Elfar *et al.* [45] to compare traffic congestion prediction results with those attained by the method proposed in this paper. The LSTM neural network and our method achieve high accuracy in a large number of road traffic congestion predictions, while our method has a faster training speed than the LSTM neural network. For the prediction of one road, the training time of our model is 0.01 minutes, and the training time of the LSTM model is affected by the number of training epochs (in this paper, we set it 100 and the training time is 0.38 minutes). The random forest model also has a better prediction effect when the training speed is faster. However, the CNN's learning and congestion prediction effect is not as good as expected. For road 1, the congestion situation is always the same (not congested), and the CNN's prediction accuracy for it is not 100%, while the prediction accuracy of the other methods can reach 100%. For road 3, which

occasionally has congestion, the prediction effect is not as good as that of other prediction methods. The reason may be that some road sections that are far away from the predicted road section and are not closely related to the predicted road section will change dramatically in a short period of time, and these changes will not affect the tested road sections, but the CNN mistakenly learns their relationship. It is also possible that the predicted road section is slightly related to the surrounding road sections, and there is no inevitable connection between the changes in the other road sections and the predicted road section, so the CNN cannot learn from them. The above two cases may be related to the lack of an attention mechanism in the CNN. Of course, as Shuming Sun said, some prediction values may have small gaps with real values, but they are classified into different congestion levels. In addition, the CNN requires 10,000 training epochs, which takes nearly two minutes to train the model for one road. Our method can predict traffic congestion at any time of the day after sample training, while the LSTM and CNN models can only predict traffic congestion according to the time interval of the training data. Moreover, our method outperforms other models on some specific roads.

Table 5 shows the prediction results of the Dual_XGBoost, CNN, random forest and LSTM models on road 1, road 2, and road 3.

TABLE 5. The accuracy of different method in different road.

	road 1	road 2	road 3
LSTM	100%	76.47%	97.16%
Dual_XGBoost	100%	82.35%	97.16%
CNN	96.44%	73.52%	90.75%
Random Forest	100%	79.41%	94.06%

The area used to test the model needs to be specifically selected, and the selected area needs to have different road conditions. On road 1 and road 3, traffic congestion is not sensitive to whether it is a weekend or weekday, and the traffic situations change slightly over a week. Road 2 is a road sensitive to whether it is a weekend or weekday, and the traffic congestion situations on workdays and non-workdays are very different. Fig. 20, Fig. 21, Fig. 22 and Fig. 23 show the prediction effect of the LSTM, Dual_XGBoost, CNN and random forest on road 2 during workdays. In the confusion matrix, some squares have two numbers. In the squares on the main diagonal, the first number is recall rate, and the second is precision rate. Numbers that are not on the main diagonal, the first number is W_r, the second number is W_p. The LSTM neural network is good at learning the features of short-term time series, but when more features affect the results, its performance is unsatisfactory. The CNN model is able to learn spatio-temporal features, but it also cannot take many features into account. Both neural networks are not very good at learning whether it is a workday, which is a long-term feature. This finding is the reason why the two neural networks are mostly used in short-term prediction. The performance of the random forest model is better than that of the two neural networks, but

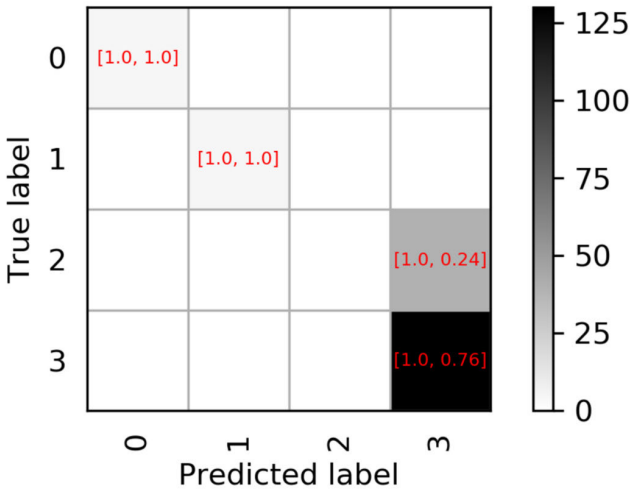


FIGURE 20. Confusion matrix. Prediction effect with LSTM.

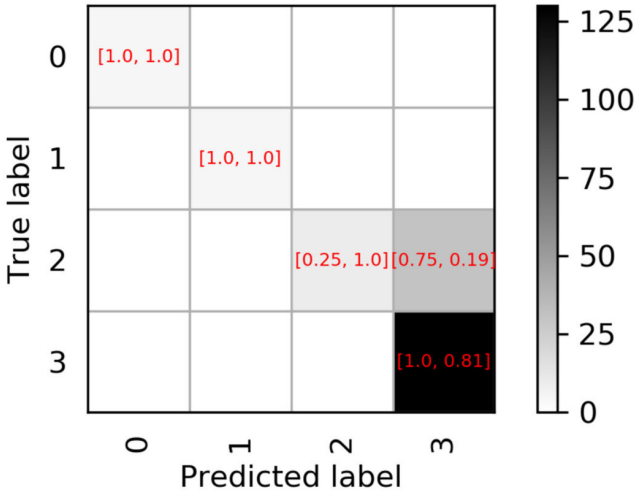


FIGURE 21. Confusion matrix. Prediction effect with Dual_XGBoost.

it does not consider as many features as our method. The lack of features such as multi-period hotspot area clustering and whether it is a workday that makes it is not as good as our method. The proposed traffic congestion algorithm based on clustering of hotspots takes more features into account, so it is more sensitive to some potential features that affect traffic congestion than other methods. For the long-term feature of whether it is a workday or not, our proposed algorithm has a better effect than the other models.

Table 6 shows the f1-score of the Dual_XGBoost, CNN, random forest and LSTM models on selected road section. Our method has the highest values of f1_macro and f1_micro.

Traffic congestion prediction using Shanghai as a case study can also be used to analyse Shanghai's traffic situation. The selected roads are all roads in the city, so large vehicles are unlikely to travel on these roads. Road 1 and road 3 are not major roads in terms of traffic, and not many vehicles travel on it every day. They are not prone to congestion. Even

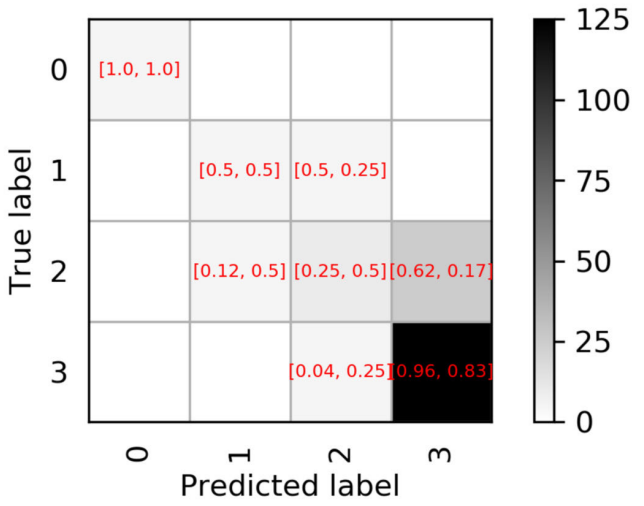


FIGURE 22. Confusion matrix. Prediction effect with CNN.

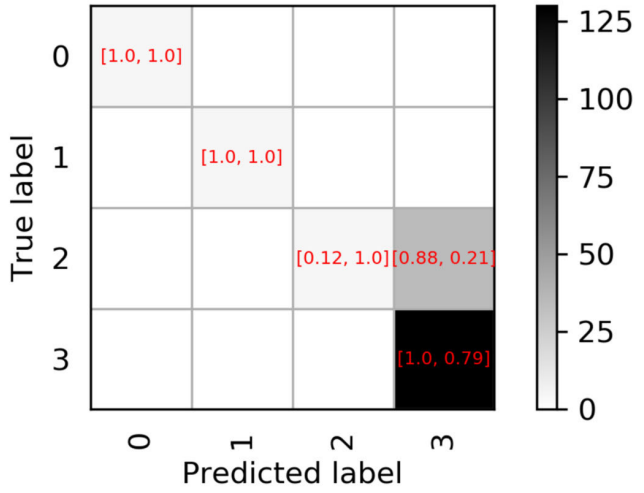


FIGURE 23. Confusion matrix. Prediction effect with Random Forest.

TABLE 6. The f1-score of the Dual_XGBoost, CNN, random forest and LSTM models.

	Unknown	Smooth	Slightly blocked	Congestion	f1_macro	f1_micro
LSTM	1	1	0	0.86	0.716	0.777
CNN	1	0.5	0.33	0.89	0.681	0.783
Random Forest	1	1	0.214	0.86	0.775	0.805
Dual_ XGBoost	1	1	0.4	0.89	0.824	0.833

if the congestion is a chain reaction due to the congestion of the main roads nearby, the congestion time on these roads is short, and the range is small. Road 2 is a busy road, and the surrounding residents will travel on it to and from work. High traffic flow also increases the possibility of traffic accidents.

In the selected sections, most of the traffic congestion occurs during the commute time. Very few cases meet the second traffic congestion judgement condition.

The greatest cause of traffic congestion in Shanghai is peak hours. During daily commuting hours, there will be a large number of people moving between the work area and the residential area. The recommended strategies for this cause of congestion are to promote the use of public transport to reduce the number of private cars on the road during peak times and take diversion traffic control to avoid lane occupation as much as possible. The Shanghai Municipal Government has adopted the method of constructing underground passages or overpasses for areas where crowds of people are likely to cause car congestion to alleviate the pressure of traffic jams caused by crowds. Public transportation, such as light rail and subways, is actively being planned and developed.

Another cause of traffic congestion is illegal driving by drivers, which often causes traffic accidents. For this cause of congestion, the recommended measures are to strengthen the intensity and measures of traffic supervision.

VI. CONCLUSION

According to R.L. Bertini's definition of traffic congestion and Michael A.P. Taylor's description of traffic congestion and combining Shanghai's traffic situations, two conditions are used to determine whether roads are congested in our work. One condition is an average road speed less than or equal to the speed threshold, and the other condition is that two or more cars on a certain section of the road have a large deviation in the driving angle, and these cars stay on the road for more than 15 minutes.

In this paper, a traffic congestion prediction algorithm based on multi-period hotspot clustering is proposed. Our contributions are as follows. First, we improve the PageRank algorithm to make it more suitable for hotspot area mining. In equations (1) and (2), considering the driving mode of the driver and the working condition of the GPS recorder, we used the damping factor in PageRank. In equation (3), equal weights are used to combine the PR scores to highlight that transportation hubs and hotspots have the same effect on traffic congestion. The improved PageRank algorithm can determine the traffic hotspot areas and transportation hubs. Second, we introduce the state transition probability of the HMM to improve the road-matching method based on the shortest distance. The improved method is faster and more accurate than the original. Third, we combine the random search and Dual_XGBoost algorithms to predict traffic congestion. Finally, taking Shanghai transportation as a case study, our method is compared with some existing methods. The result shows that our method is faster in model training, more flexible in predicting time and more sensitive to changes in the long-term features. However, our method also has shortcomings. As Liu, L. *et al.* discovered, taxi drivers have their own driving modes [46]. Drivers, based on their own experience, will avoid sections of roads that are often congested. Therefore, some road congestion cannot be discovered. Our method also has this shortcoming. In addition, in Shanghai, many elevated roads overlap with roads. It is difficult to distinguish whether a vehicle is driving on an elevated road or a road based on GPS signals alone, which will cause errors in calculating the average road speed.

In the future, we will try to predict traffic congestion with a spatio-temporal mining algorithm combined with GCNs, GRU networks, and gated linear units.

REFERENCES

[1] P. S. Castro, D. Zhang, C. Chen, S. Li, and G. Pan, "From taxi GPS traces to social and community dynamics: A survey," *ACM Comput. Surv.*, vol. 46, no. 2, pp. 1–34, 2013.

[2] W. Chen, F. Guo, and F.-Y. Wang, "A survey of traffic data visualization," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 2970–2984, Dec. 2015.

[3] F. Mao, M. Ji, and T. Liu, "Mining spatiotemporal patterns of urban dwellers from taxi trajectory data," *Frontiers Earth Sci.*, vol. 10, no. 2, pp. 205–221, Jun. 2016.

[4] G. Qi, X. Li, S. Li, G. Pan, Z. Wang, and D. Zhang, "Measuring social functions of city regions from large-scale taxi behaviors," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PERCOM Workshops)*, Mar. 2011, pp. 384–388.

[5] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong, "Discovering urban functional zones using latent activity trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 712–725, Mar. 2015.

[6] Z. Abbas, A. Al-Shishtawy, S. Girdzijauskas, and V. Vlassov, "Short-term traffic prediction using long short-term memory neural networks," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, Jul. 2018, pp. 57–65.

[7] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Acad. Annu. Conf. Chin. Assoc. Automat. (YAC)*, Nov. 2016, pp. 324–328.

[8] R. Toncharoen and M. Piantanakulchai, "Traffic state prediction using convolutional neural network," in *Proc. 15th Int. Joint Conf. Comput. Sci. Softw. Eng. (JCSSE)*, Jul. 2018, pp. 1–6.

[9] C. Song, H. Lee, C. Kang, W. Lee, Y. B. Kim, and S. W. Cha, "Traffic speed prediction under weekday using convolutional neural networks concepts," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1293–1298.

[10] T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, and P. Hellinckx, "A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data," *Transp. Res. C, Emerg. Technol.*, vol. 112, pp. 62–77, Mar. 2020.

[11] L. Zheng, X. Zhao, Z. Jiang, J. Deng, D. Xia, and W. Liu, "Mining uban attractive areas using taxi trajectory data," *Comput. Appl. Softw.*, vol. 35, p. 1, Jan. 2018.

[12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, vol. 96, no. 34, pp. 226–231.

[13] Z. Gui and H. Yu, "Mining traffic hot spots from massive taxi trace," *J. Comput. Inf. Syst.*, vol. 10, no. 7, pp. 2751–2760, 2014.

[14] Y. Shen, L. Zhao, and J. Fan, "Analysis and visualization for hot spot based route recommendation using short-dated taxi GPS traces," *Information*, vol. 6, no. 2, pp. 134–151, 2015.

[15] Z. Xu, Z. Lin, C. Zhou, and C. Huang, "Detecting traffic hot spots using vehicle tracking data," in *Proc. 2nd ISPRS Int. Conf. Comput. Vis. Remote Sens. (CVRS)*, vol. 9901, 2016, Art. no. 99010Y.

[16] Z. Xia, Q. Lei, Y. Yang, H. Zhang, Y. He, W. Wang, and M. Huang, "Vision-based hand gesture recognition for human-robot collaboration: A survey," in *Proc. 5th Int. Conf. Control, Automat. Robot. (ICCAR)*, Apr. 2019, pp. 198–205.

[17] M. Francia, E. Gallinucci, and F. Vitali, "Map-matching on big data: A distributed and efficient algorithm with a hidden Markov model," in *Proc. 42nd Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2019, pp. 1238–1243.

[18] H. Qi, X. Di, and J. Li, "Map-matching algorithm based on the junction decision domain and the hidden Markov model," *PLoS ONE*, vol. 14, no. 5, 2019, Art. no. e0216476.

[19] M. Quddus and S. Washington, "Shortest path and vehicle trajectory aided map-matching for low frequency GPS data," *Transp. Res. C, Emerg. Technol.*, vol. 55, pp. 328–339, Jun. 2015.

[20] Z. Yong-Chuan, Z. Xiao-Qing, Z. Li-Ting, and C. Zhen-Ting, "Traffic congestion detection based on GPS floating-car data," *Procedia Eng.*, vol. 15, pp. 5541–5546, Jan. 2011.

[21] S. Sun, J. Chen, and J. Sun, "Traffic congestion prediction based on GPS trajectory data," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 5, pp. 1–18, May 2019.

[22] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 166–180, May 2018.

[23] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Jun. 2017.

[24] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*. [Online]. Available: http://arxiv.org/abs/1709.04875

[25] D. Zang, J. Ling, Z. Wei, K. Tang, and J. Cheng, "Long-term traffic speed prediction based on multiscale spatio-temporal feature learning network," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3700–3709, Oct. 2019.

[26] Y. Gu, W. Lu, L. Qin, M. Li, and Z. Shao, "Short-term prediction of lane-level traffic speeds: A fusion deep learning model," *Transp. Res. C, Emerg. Technol.*, vol. 106, pp. 1–16, Sep. 2019.

[27] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong, "Traffic speed prediction and congestion source exploration: A deep learning method," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 499–508.

[28] L. N. N. Do, H. L. Vu, B. Q. Vo, Z. Liu, and D. Phung, "An effective spatial-temporal attention based neural network for traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 108, pp. 12–28, Nov. 2019.

[29] H. Xiong, A. Vahedian, X. Zhou, Y. Li, and J. Luo, "Predicting traffic congestion propagation patterns: A propagation graph approach," in *Proc. 11th ACM SIGSPATIAL Int. Workshop Comput. Transp. Sci.*, 2018, pp. 60–69.

[30] A. V. Khezerlou, X. Zhou, L. Li, Z. Shafiq, A. X. Liu, and F. Zhang, "A traffic flow approach to early detection of gathering events," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2016, pp. 1–10.

[31] A. V. Khezerlou, X. Zhou, L. Li, Z. Shafiq, A. X. Liu, and F. Zhang, "A traffic flow approach to early detection of gathering events: Comprehensive results," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 6, pp. 1–24, Jul. 2017.

[32] R. L. Bertini, "You are the traffic jam: An examination of congestion measures," in *Proc. 85th Annu. Meeting Transp. Res. Board*, 2006, pp. 1–17.

[33] M. A. P. Taylor, J. E. Woolley, and R. Zito, "Integration of the global positioning system and geographical information systems for traffic congestion studies," *Transp. Res. C, Emerg. Technol.*, vol. 8, nos. 1–6, pp. 257–285, Feb. 2000.

[34] M. Taylor, "An extended family of traffic network equilibria and its implications for land use and transport policies," in *Proc. 8th World Conf. Transp. Res. World Conf. Transp. Res. Soc.*, vol. 4, 1999, pp. 29–42.

[35] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the Web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. SIDL-WP-1999-0120, 1999.

[36] M. Srivatsa, R. Ganti, J. Wang, and V. Kolar, "Map matching: Facts and myths," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.* New York, NY, USA: Association for Computing Machinery, 2013, pp. 484–487, doi: 10.1145/2525314.2525466.

[37] M. Hashemi and H. A. Karimi, "A critical review of real-time map-matching algorithms: Current issues and future directions," *Comput., Environ. Urban Syst.*, vol. 48, pp. 153–165, Nov. 2014.

[38] S. Taguchi, S. Koide, and T. Yoshimura, "Online map matching with route prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 338–347, Jan. 2019.

[39] J. J. Arsanjani, A. Zipf, P. Mooney, and M. Helbich, "An introduction to OpenStreetMap in geographic information science: Experiences, research, and applications," in *OpenStreetMap in GIScience*. Cham, Switzerland: Springer, 2015, pp. 1–15.

[40] S. Syed and M. E. Cannon, "Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons," in *Proc. ION Nat. Tech. Meeting*, San Diego, CA, USA, vol. 1, 2004, pp. 26–28.

[41] A. Rae and O. Basir, "Reducing multipath effects in vehicle localization by fusing GPS with machine vision," in *Proc. 12th Int. Conf. Inf. Fusion*, 2009, pp. 2099–2106.

[42] S. Zhi-Tao, "Basic principles of GPS and its application prospects," *J. Zhuzhou Inst. Technol.*, vol. 15, no. 5, pp. 44–47, 2001.

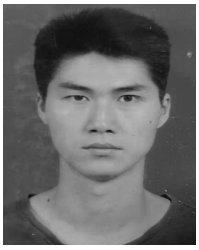
[43] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.

[44] B. Li, D. Zhang, L. Sun, C. Chen, S. Li, G. Qi, and Q. Yang, "Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PERCOM Workshops)*, Mar. 2011, pp. 63–68.

[45] A. Elfar, A. Talebpour, and H. S. Mahmassani, "Machine learning approach to short-term traffic congestion prediction in a connected environment," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2672, no. 45, pp. 185–195, Dec. 2018.
[46] L. Liu, C. Andris, A. Biderman, and C. Ratti, "Revealing taxi driver's mobility intelligence through his trace," in *Movement-Aware Applications for Sustainable Mobility: Technologies and Approaches*. Hershey, PA, USA: IGI Global, 2010, pp. 105–120.

CHUNHUI XU is currently pursuing the degree with the College of Computer Science and Technology, Shanghai University of Electric Power. His main research interests include data mining, machine learning, and traffic control.

ANQIN ZHANG received the Ph.D. degree from Fudan University. She is currently an Associate Professor with the College of Computer Science and Technology, Shanghai University of Electric Power. Her main research interests include social computing and pervasive computing.



YU CHEN is currently pursuing the degree with the College of Computer Science and Technology, Shanghai University of Electric Power. His main research interests include data mining and machine learning.