TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 02/15 pp403–416
DOI: 10.26599/TST.2020.9010014
Volume 26, Number 4, August 2021

Transportation Mode Identification with GPS Trajectory Data and GIS Information

Ji Li, Xin Pei, Xuejiao Wang, Danya Yao*, Yi Zhang, and Yun Yue

Abstract: Global Positioning System (GPS) trajectory data can be used to infer transportation modes at certain times and locations. Such data have important applications in many transportation research fields, for instance, to detect the movement mode of travelers, calculate traffic flow in an area, and predict the traffic flow at a certain time in the future. In this paper, we propose a novel method to infer transportation modes from GPS trajectory data and Geographic Information System (GIS) information. This method is based on feature extraction and machine learning classification algorithms. While using GIS information to improve inference accuracy, we ensure that the algorithm is simple and easy to use on mobile devices. Applied to GeoLife GPS trajectory dataset, our method achieves 91.1% accuracy while inferring transportation modes, such as walking, bike, bus, car, and subway, with random forest classification algorithm. GIS features in our method improved the overall accuracy by 2.5% while raising the recall of the bus and subway transportation mode categories by 3.4% and 18.5%. We believe that many algorithms used in detecting the transportation modes from GPS trajectory data that do not utilize GIS information can improve their inference accuracy by using our GIS features, with a slight increase in the consumption of data storage and computing resources.

Key words: transportation mode; Global Positioning System (GPS); Geographic Information System (GIS); random forest

1 Introduction

The popularity of smart portable devices facilitates the ability to collect Global Positioning System (GPS) trajectories of travelers. These trajectories are time-series data with each point constituting longitude, latitude, and timestamp value. Using these data, we can determine the location of a traveler at a certain time without additional calculation. Using simple statistical calculations, we can also know whether a traveler has visited a certain area, the number of visits, and the duration of the visit. However, if we want to obtain in-depth information, such as information on traffic flow or congestion in a certain area, we need to comprehend the data deeply. In this procedure, inferring transportation modes from GPS trajectories is particularly important, because it adds semantic information to the original data. Such semantic information enables further data analysis, for example, to establish the traveler's movement modes[1–3], estimate traffic demand in a certain area, estimate the congestion degree of a road (by calculating car speed), perform accurate trajectory prediction, and estimate the calorie consumption of the traveler or the carbon dioxide emission of the vehicle caused by traveler's movement. The results of these calculations can provide important basic data for intelligent transportation systems.

• Ji Li, Xin Pei, Danya Yao, Yi Zhang, and Yun Yue are with the Department of Automation, Tsinghua University, and also with National Laboratory for Information Science and Technology (TNList), Tsinghua University, Beijing 100084, China. E-mail: liji16@mails.tsinghua.edu.cn; peixin@tsinghua.edu.cn; yaody@tsinghua.edu.cn; zhyi@mail.tsinghua.edu.cn; ziyueyunxiao@163.com.
• Xuejiao Wang is with the National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data (NEL-PSRPC), Beijing 100041, China. E-mail: xjwang20180717@163.com.
* To whom correspondence should be addressed.
Manuscript received: 2019-10-16; accepted: 2020-04-06

Many previous studies have focused on inferring transportation modes from time-series sensor data, and the algorithms can be divided into four categories: (1) The first category[4–10] only uses GPS trajectory data and no other information, such as Geographic Information System (GIS) data or travelers' personal information to infer transportation modes. With the effect of traffic congestion and bad weather on GPS trajectory data, as well the resulting similarity of the GPS data in different transportation modes, the accuracy of these methods can be improved by using external information, such as GIS data and travelers' personal information. (2) The second category[11–13] considers travelers' personal information when inferring their transportation modes from GPS trajectory data. This type of research method can usually obtain higher inference accuracy than methods in the first category, but they involve dependence on travelers' personal private data, which cannot always be obtained. (3) The third category[14–19] uses GPS trajectory data in combination with large volumes of GIS data to infer travelers' transportation modes. These GIS data include those on road networks, subway networks, railway networks, and real-time bus locations. These methods can obtain highly accurate inference results, but they usually lead to reliance on large amounts of GIS data and complex calculations, which make them difficult to use in fully automatic processing mode. (4) The fourth category[20–28] uses a combination of time-series data produced by sensors of different types, such as GPS sensors, acceleration sensors, gyroscopes, and altitude sensors. These methods usually grant high inference accuracy while introducing reliance on multiple amounts of sensors, and difficulty in matching transportation mode data to geographic locations occurs when GPS sensors are not used. Unlike the aforementioned four categories of algorithms, our method uses GPS trajectory data and limited GIS information to infer travelers' transportation modes. While utilizing GIS information to improve inference accuracy, our method ensures simplicity of the algorithm and easy automatic computing on mobile devices. Moreover, our method does not depend on travelers' personal data or other sensors' data besides GPS sensors.

In this study, we propose a novel method to infer transportation modes from travelers' GPS trajectory data. This method uses GPS trajectory data and some GIS information to distinguish between five transportation modes, namely, walking, bike, bus, car (taxi and private car), and subway train (referring to subway sections above the ground to ensure good GPS signals). We extract 15 features from each trajectory and then use machine learning algorithm to classify feature vectors to determine transportation modes. Our feature extraction method differs from previous methods and has the following advantages. First, the nine statistical features of velocity, acceleration, and heading change speed we extract from each trajectory can reflect the distribution of velocity, acceleration, and heading change speed value in each trajectory; they also have a good restraining effect on relatively large GPS data error of a few trajectory points. Second, the GIS features we extract can effectively differentiate between bus, car, and subway modes, which are generally considered difficult to distinguish. The third advantage of our method is full account of easy algorithm usability and low computing and storage resource consumption when using GIS information, which allows our algorithm to run automatically on mobile terminal devices in offline mode. The machine learning classification algorithms we investigated include Decision Tree (DT), Random Forest (RF), AdaBoost, XGboost, LightGBM, and Artificial Neural Network (ANN). Experimental use of our method on the GeoLife dataset shows that the RF classification algorithm achieves the highest accuracy.

The rest of this paper is organized in the following order. Section 2 is a literature review, which introduces previous research findings on this subject. Section 3 is a detailed introduction to our method, including feature extraction method and machine learning classification algorithms. Section 4 describes our experiment and findings. The last section provides the conclusion.

2 Literature Review

Many studies have explored machine learning methods to infer transportation modes from GPS trajectory data. Some of them[4–10] use GPS trajectory data solely and do not depend on any other external information. Zheng et al.[4,5] divided GPS trajectory into segments and extract nine features from each segment, including five velocity features, one acceleration feature, one heading change-rate feature, one stop-rate feature, and one total travel distance feature, then used the DT algorithm to obtain a preliminary inference result. Thereafter the transition probability matrix is utilized between different transportation modes to modify the preliminary inference result to obtain the final inference result, which has 75% accuracy. Wang et al.[6] extracted 11 features related to velocity, acceleration, and heading

change rate from each trajectory, and then use box plot algorithm to remove outlier trajectories. Then, the remaining trajectories' feature vectors are classified using the LightGBM algorithm, and a classification accuracy of 90.6% is achieved. Liu and Lee[7] used Recurrent Neural Network–Bidirectional Long Short-Term Memory (RNN-BLSTM) to infer transportation modes from GPS trajectories with end-to-end work mode. They used 3D time-series data of each trajectory, which consists of (Δlongitude1, Δlatitude1, Δt1), (Δlongitude2, Δlatitude2, Δt2), and so on, as input data for neural networks, and achieved an Area Under the Curve (AUC) value of 0.946 for transportation mode classification results. Considering the effects of traffic congestion, bad weather, and GPS error on GPS trajectory data, we believe that the aforementioned methods would achieve higher inference accuracy if appropriate external information (such as GIS data) is used.

Some studies[11–13] use travelers' personal characteristics when inferring transportation modes from GPS trajectory data. Stopher et al.[11] considered whether a traveler owns a bike. Bantis and Haworth[12] investigated the GPS data of two special travelers: one who uses crutches (female, 40–59 years old, full-time worker) and another who uses a wheelchair (male, 22–39 years old, full-time worker), and identified the differences between them in terms of velocity distribution while using the same modes of transportation (walking, static, bus, or rail) and in terms of the two travelers' probability to transition between different transportation modes. Such differences are used in detecting transportation modes and have helped improve the final inference accuracy. Obviously, travelers' personal information is immensely helpful in inferring their modes of transportation, but in many cases, such personal information is difficult to obtain and some of them involves the privacy of travelers. Therefore, inferring transportation modes from GPS trajectory data without using travelers' personal information is an important and worthwhile topic to consider.

Some articles[11,14–19] use GIS information when detecting transportation modes from GPS trajectory data. Chung and Shalaby[14] matched every GPS point to a specific road and then inferred transportation modes based on the trajectory speed characteristics and traffic rules. The traffic rules include whether the road sections covered by the trajectory overlap with bus routes, whether the trajectory is along expressway sections (if so, walking and cycling can be excluded), and whether the trajectory goes in the wrong direction on any one-way road (if so, cars and buses can be excluded). Gong et al.[15] also matched each GPS point to a road, and then calculated average speed, speed at 85th percentile, acceleration at 95th percentile, total duration of each trajectory, and distance between the starting and ending points of each trajectory from the nearest bus and subway stations. Finally, the transportation mode was inferred with rules with manually set thresholds of each feature. The inference accuracy in the study is 86%. Stenneth et al.[18] obtained bus station location, railway line data, and buses' real-time location data in one city, and calculated the distances between each point in the trajectory from the nearest bus, bus station, and railway line. Then, the researchers classified the GPS trajectory's transportation modes into stationary, walking, bike, bus, car, and aboveground train with RF algorithm while considering the following trajectory features: average speed, average acceleration, average rail-line closeness, average bus closeness, and candidate bus closeness. An inference accuracy rate of 93.5% is obtained (three GIS features improve the accuracy by 17%). These algorithms can obtain higher inference accuracy than that of inference methods using GPS trajectory data alone. However, these algorithms generally consume a great amount of computing resources and some of them are difficult to process in a fully automatic calculation mode. Methods that use real-time location of all the buses in one city require connecting the calculating devices with all buses in the city when they are running in real-time inference mode, which exerts high requirements on the communication and computing resources of algorithm-calculating devices and onboard devices of the buses.

Some studies[20–23] use a combination of data from GPS and acceleration sensors to achieve transportation mode detection. Compared with calculating the average acceleration of each interval from GPS sensor data, the acceleration sensor provides instantaneous acceleration value, which makes it more effective in reflecting the dynamic characteristics of various transportation modes and thereby providing effective information in differentiating between transportation modes. Therefore, compared with relying solely on GPS sensor data, this type of approach can often obtain higher inference accuracy. Other studies[24–28] do not use GPS sensors, but rather use combinations of other sensor data from smartphones to infer transportation modes. For instance, Su et al.[24] and Su[25] used data from accelerometers,

gravity sensors, gyroscopes, magnetometers, and barometers to infer transportation modes. This approach, while in most cases is able to obtain higher inference accuracy, involves dependence on more types of sensors compared with methods using GPS sensor data alone. Some of the methods that belong to this approach do not use GPS sensor data, which means that their final results cannot indicate how transportation modes correspond to geographic locations, thereby causing difficulty for them to be used in traffic-related application scenarios.

Compared with the aforementioned methods, our method uses GPS sensor data combined with limited GIS information to infer transportation modes. While obtaining relatively high inference accuracy by using GIS data, our method ensures that the entire algorithm consumes limited storage and computing resources of the calculating devices while performing normally when the calculating devices are offline.

3 Research Method

In this section, we introduce the method of inferring transportation modes from GPS trajectory data. First, we explain our feature extraction method. Then, we introduce algorithms used in this study. The features we extract from each trajectory include nine statistical features related to velocity, acceleration and heading change speed, two global features of trajectory, and four GIS features. The algorithms we explain include the fast calculating method of two GIS features and six machine learning classification algorithms for generating transportation mode from feature vector of GPS trajectory.

3.1 Classification feature selection

A traveler may use multiple transportation modes in a GPS trajectory, such as walking→car→walking or walking→subway→walking. When a GPS trajectory segmentation method is used, a trajectory can be divided into multiple segment trajectories, with each segment trajectory traveled via a single transportation mode. The GPS trajectory segmentation methods proposed by Zheng et al.[4,5] and Stenneth et al.[29] are effective in this task. They considered that a walking segment often exists during the transition from one transportation mode to another. Thus, they detected the walking segment trajectories by setting threshold values of speed, acceleration, heading change speed, and time duration, and then divided the trajectory into several sub-trajectories with the starting and ending points of the walking segment trajectories to obtain trajectories with a single transportation mode. Thereafter, some features can be extracted from each segment trajectory to form feature vectors, which are then used for classification into different transportation modes. We extract 15 features for each trajectory, specifically, nine statistical features related to velocity, acceleration, and heading change speed, two global features of the trajectory, and four GIS features.

3.1.1 Velocity features

For a GPS trajectory containing n data points, calculating the distance and time interval between each two adjacent data points would deliver distance Δs_i, time length Δt_i, and average velocity v_i of $n-1$ intervals, with $i \in [2, n]$. Then, we can calculate the following three features.

Feature 1. Average speed: Its value is equal to the total traveled distance of the trajectory divided by the total time,

$$V_{\text{avg}} = \frac{\sum_{i=2}^{n} \Delta s_i}{\sum \Delta t_i} \tag{1}$$

Feature 2. Speed variance: This feature can be calculated by statistical sample variance formula,

$$V_{\text{variance}} = \frac{\sum_{i=2}^{n} (v_i - V_{\text{avg}})^2}{n-2} \tag{2}$$

Feature 3. 85th percentile of speed: The average speed of each interval in the trajectory, namely, v_2, v_3, …, v_n, is ordered from small to large to form a new serial data v'_2, v'_3, …, v'_n. Then we select the 85th percentile value as feature value,

$$V_{85\text{th}} = v'_{[(n-1)\times 0.85]+1} \tag{3}$$

3.1.2 Acceleration features

For a trajectory containing n data points, the average acceleration a_i of $n-2$ intervals can be obtained. The formula is as follows:

$$a_i = \frac{(v_i - v_{i-1})}{\Delta t_i} \tag{4}$$

where $i \in [3, n]$. Then, the following three features can be calculated from $a_3, a_4, \ldots, a_n$.

Feature 4. Average acceleration: The value of this feature is equal to the arithmetic average of the acceleration in each interval,

$$A_{\text{avg}} = \frac{\sum_{i=3}^{n} \Delta a_i}{n-2} \tag{5}$$

Feature 5. Acceleration variance: This feature value can be calculated by the statistical sample variance formula,

$$A_{\text{variance}} = \frac{\sum_{i=3}^{n} (a_i - A_{\text{avg}})^2}{n-3} \tag{6}$$

Feature 6. 85th percentile of acceleration: The acceleration value of each interval, namely, $a_3, a_4, \ldots, a_n$, is ordered from small to large to form a new data series $a'_3, a'_4, \ldots, a'_n$. Then, we select the 85th percentile value as feature value,

$$A_{85\text{th}} = a'_{[(n-2)\times 0.85]+1} \tag{7}$$

3.1.3 Heading change speed features

For a trajectory containing n GPS points, $n-2$ average heading change speed, namely, hcs_i, can be calculated, where $i \in [3, n]$. The calculation steps are as follows:

(1) The equivalent displacement of each point relative to the previous point can be represented by r_i, with formation $(\Delta\text{long}_i, \Delta\text{lat}_i)$, which can be calculated as

$$\Delta\text{long}_i = (\text{long}_i - \text{long}_{i-1}) \cdot \cos(\text{lat}_i) \tag{8}$$

$$\Delta\text{lat}_i = \text{lat}_i - \text{lat}_{i-1} \tag{9}$$

where long_i and lat_i represent longitude and latitude value of the i-th GPS point in the trajectoy, respectively.

(2) The heading direction change angle of each point relative to the previous point (starting from the third GPS point) is calculated as

$$\text{hc}_i = \arccos\left(\frac{r_i \cdot r_{i-1}}{\sqrt{r_i \cdot r_i}\sqrt{r_{i-1} \cdot r_{i-1}}}\right) \tag{10}$$

(3) The heading change speed of each point (starting from the third GPS point) is calculated as

$$\text{hcs}_i = \frac{\text{hc}_i}{\Delta t_i} \tag{11}$$

Feature 7. Average heading change speed: The value of this feature is equal to the arithmetic average of the heading change speed in each interval,

$$\text{HCS}_{\text{avg}} = \frac{\sum_{i=3}^{n} \text{hcs}_i}{n-2} \tag{12}$$

Feature 8. Heading change speed variance: This feature value can be calculated by the statistical sample variance formula,

$$\text{HCS}_{\text{variance}} = \frac{\sum_{i=3}^{n} (\text{hcs}_i - \text{HCS}_{\text{avg}})^2}{n-3} \tag{13}$$

Feature 9. 85th percentile of heading change speed: The heading change speed value of each interval, namely, $\text{hcs}_3, \text{hcs}_4, \ldots, \text{hcs}_n$, is ordered from small to large to form a new data series $\text{hcs}'_3, \text{hcs}'_4, \ldots, \text{hcs}'_n$. Then, we select the 85th percentile value as feature value,

$$\text{HCS}_{85\text{th}} = \text{hcs}'_{[(n-2)\times 0.85]+1} \tag{14}$$

3.1.4 Global features

Feature 10. Total distance: The formula is

$$s_{\text{total}} = \sum_{i=2}^{n} \Delta s_i \tag{15}$$

Feature 11. Stop rate: This is the proportion of intervals whose speed is equal to or less than 0.6 m/s in the trajectory,

$$\text{lsrate} = \frac{\text{count}_{v_i \leqslant 0.6\,\text{m/s}}}{n-1} \tag{16}$$

3.1.5 GIS features

Considering the various degrees of curvature of the trajectory at different transportation modes, for instance, walking trajectory often features more turning, subway trajectory features less, and bus trajectory features less turning than the car trajectory, we introduce two features to describe the curvature degree of each trajectory.

Feature 12. Original and Destination (OD) points distance: This feature represents distance between the original point and the destination point of a GPS trajectory. The formula is

$$s_{\text{od}} = \text{distance}(\text{point}_{\text{origin}}, \text{point}_{\text{destination}}) \tag{17}$$

Feature 13. Straight rate: The formula is

$$\text{srate} = \frac{s_{\text{od}}}{s_{\text{total}}} \tag{18}$$

Features 14 and 15. Ratio of low-speed points near bus and subway stations: Considering that buses and subway trains stop at specific stations where passengers embark and disembark, we calculate the distances between low-speed points in the trajectory and the nearest bus and subway stations. Then, we extract two features from them to differentiate between bus, subway, and other transportation modes.

Figures 1 and 2 show a bus and subway GPS trajectory from the GeoLife dataset, respectively. The blue dots in Figs. 1 and 2 indicate trajectory points with speed greater than 1 m/s, the red dots are trajectory points with speed equal to or less than 1 m/s, and the green squares are bus and subway stations. Figures 1 and 2 easily show that the points where the speed is equal to or less than 1 m/s in the bus trajectory are very close to bus stations, and the points where the speed is equal to or less than 1 m/s in the subway trajectory are very close to subway stations. Thus, we define two GIS features, one is the ratio of points within 80 m from the nearest bus station among all points whose speed is equal to or less than

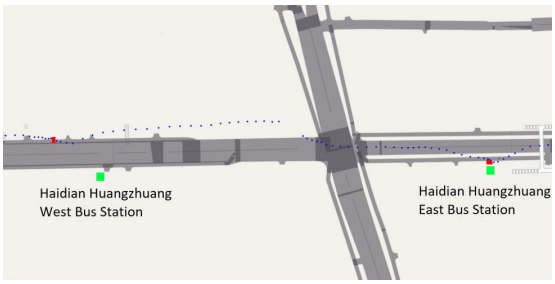


Fig. 1 A bus GPS trajectory segment obtained from GeoLife dataset.

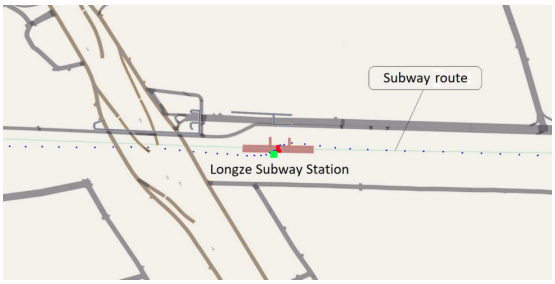


Fig. 2 A subway GPS trajectory segment obtained from GeoLife dataset.

1 m/s in the trajectory, named ratio_lsp_nearbs, another is the ratio of points within 500 m from the nearest subway station among all points whose speed is equal to or less than 1 m/s in the trajectory named ratio_lsp_nearss.

The specific calculation steps of these two features are as follows:

(1) All intervals with velocity equal to or less than 1 m/s in the trajectory are identified. The total number of such intervals is represented as count_total.

(2) If no interval exists whose velocity is equal to or less than 1 m/s in the trajectory, the values of these two features are both set as -1.

$$\text{ratio\_lsp\_nearbs} = -1 \tag{19}$$

$$\text{ratio\_lsp\_nearss} = -1 \tag{20}$$

(3) For the trajectory with interval(s) whose velocity is equal to or less than 1 m/s, the distance from the ending point of the interval to the nearest bus station, d_{bstation}, and the distance from the ending point of the interval to the nearest subway station, d_{sstation}, are calculated. Considering possible error of GPS, we count the number of trajectory points that $d_{\text{bstation}} \leqslant 80$ m as count_b and the number of gps points that $d_{\text{sstation}} \leqslant 500$ m as count_s, and then we calculate the two feature values as

$$\text{ratio\_lsp\_nearbs} = \frac{\text{count\_b}}{\text{count\_total}} \tag{21}$$

$$\text{ratio\_lsp\_nearss} = \frac{\text{count\_s}}{\text{count\_total}} \tag{22}$$

3.2 Algorithm

3.2.1 GIS feature extraction method

In this section, we present two methods for calculating distances from low-speed points in the trajectory to the nearest bus and subway stations.

Method 1. Fast calculation algorithm using local data. As the ratio of low-speed points near the bus and subway station features in our algorithm involve calculating whether the distance from low-speed GPS points to the nearest bus station and subway station is equal to or less than 80 m and 500 m, we divide a large area (such as a city) into squared grids, each with a length and width of 500 m. Then, we name each grid with an ID, and store the location information of the bus station(s) and subway station(s) in each grid in the storage space corresponding to the grid ID. When dividing a not so large area into grids according to longitude and latitude, we can calculate the displacements of Δlat and Δlong with the following formulas, which use the maximum absolute value of latitude $\text{lat}_{\max}$ in this area:

$$\Delta Y_{\Delta\text{lat}} = R_{\text{radius}} \cdot \left(\frac{\Delta\text{lat}}{\pi}\right) \tag{23}$$

$$\Delta X_{\Delta\text{long}} = \cos\left(\frac{\text{lat}_{\max}}{\pi}\right) \cdot R_{\text{radius}} \cdot \left(\frac{\Delta\text{long}}{\pi}\right) \tag{24}$$

where R_{radius} is the radius of the earth, and $\text{lat}_{\max}$ is the highest latitude value of the area.

Then, the following steps can be performed to calculate the distance from a low-speed GPS point to the nearest bus and subway stations:

(1) The ID of the grid where the GPS point is located is calculated according to the point's latitude and longitude values.

(2) The grid IDs of the eight grids surrounding the grid found in Step 1 are chosen. These nine grids constitute the candidate zone, which has a size of 1500 m $\times$ 1500 m.

(3) All the bus and subway stations in the candidate zone are found. In most cities, the number of bus stations in a candidate zone should not be more than five, and the number of subway stations should not be more than two.

(4) All the bus stations and subway stations in the candidate zone are iterated to obtain the smallest distance from the GPS point to the bus and subway stations.

(5) If the storage space is large enough, the set of candidate bus stations and subway stations corresponding to each grid that GPS point located can be calculated and stored in advance, which can save time spent in Steps 2 and 3.

Method 2. Algorithm using Point-Of-Interest (POI) query service provided by online mapping company. Some online mapping companies provide free POI query service[30]. We can use them to inquire about bus and subway station information around designated GPS points. The query results include the names of the bus and subway stations around the designated GPS points and the distances from these stations to the GPS point.

3.2.2 Classification algorithm

To find the most suitable classification algorithm for this task, we select six machine learning classification algorithms to classify the feature vectors of GPS trajectories and compare their results. The classification algorithms are DT[31], RF[32], AdaBoost[33], XGboost[34], LightGBM[35], and ANN[36]. They are described as follows:

(1) DT

The DT we select is Classification And Regression Trees (CART), which is a binary tree that uses Gini index when searching the best split point. The formula for calculating the Gini index is

$$\text{Gini}(D) = 1 - \sum_{i=1}^{n} p_i^2 \tag{25}$$

where p_i represents the proportion of the i-th category sample in the current tree node.

Each time a tree splits at a feature's value, the new Gini index is calculated as

$$\text{Gini}(D)_{\text{new}} = \frac{M_1}{M} \cdot \text{Gini}(D_1) + \frac{M_2}{M} \cdot \text{Gini}(D_2) \tag{26}$$

where M is the number of samples in the current tree node before splitting, and M_1 and M_2 are the number of samples in the left and right child tree nodes of current tree node after splitting.

When the tree splits, the candidate split points are the average value of each two adjacent feature values of every feature. After the new Gini index value for each candidate split point is calculated, the smallest one is selected to split the tree. CART continues splitting until the Gini index value is zero, or the tree reaches the restricted conditions, which are set for countering the overfitting effect. In our experiment, the maximum depth of the tree is restricted to seven to prevent an overfitting effect of the algorithm.

(2) RF

The RF algorithm uses a number of DTs to vote for the final classification result. When each DT is created, a part of the training samples is randomly selected. Not all the features are considered each time the split point is found when a tree splits. Only some candidate features chosen randomly are involved to find the best split point. In our experiment, we use the RF composed of 120 CART DTs. The number of candidate features used to calculate the best split point in a tree is three.

(3) AdaBoost

The AdaBoost algorithm uses a number of DTs to vote for the final classification result. The algorithm assigns each training sample an opportunity value to be selected when creating a DT, and then selects a portion of the samples based on those opportunity values to construct each tree. At the beginning of this algorithm, the selected opportunity of every sample is set with the same value. After each decision tree is created, the tree is used to classify all training samples immediately. Then, the selected opportunity values of the incorrect classified samples are increased and the selected opportunity values of the correct classified samples are decreased before the next decision tree is created. After all the decision trees have been created, the error rate of each tree to the train samples is recorded. When trees vote for the sample category, each tree holds a voting weight of $\log((1-\text{error\_rate})/\text{error\_rate})$ with the error_rate value of itself. In our experiment, 500 CART decision trees are used to construct an AdaBoost classifier, and the maximum depth of each tree is set to 16.

(4) XGboost

The XGboost algorithm is composed of many CART regression trees. For each sample, the algorithm sums the result of each tree to obtain the final regression value, and then calculates the category information. The loss function contains regularization items to counter overfitting. The formulas are

$$\Omega_j = \gamma T + \frac{1}{2}\lambda \sum_{s=1}^{T} w_s^2 \tag{27}$$

$$\text{obj}(\theta) = \sum_{i=1}^{n} l(y_{o-i}, y_i) + \sum_{j=1}^{K} \Omega_j \tag{28}$$

where Ω represents the regular term in the objective function of a regression tree, T represents the number of leaf nodes of a tree, w_s represents the value of the s-th leaf node of a tree, obj() represents the optimized target function of the XGboost algorithm, l represents the error function of a sample, and y_i and y_{o-i} represent true value and regression value of the sample i, respectively.

The preceding object function is expanded by the second-order Taylor formula. With some deductions

and merges, the minimum object value of each tree can be calculated as

$$\text{obj}_{\min} = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T \tag{29}$$

and leaf's regression value is

$$w_j = -\frac{G_j}{H_j+\lambda} \tag{30}$$

where G_j and H_j represent the sum of first and second partial derivatives value of function l respect to variable w_j of all samples which is located at j-th leaf of the decision tree.

When each tree is created, the best structure is searched to reach the minimum object value. In our experiment, the XGboost algorithm includes 550 trees, the maximum depth of each tree is 5, and the values of parameters γ and λ is 0 and 1.

(5) LightGBM

This algorithm is a gradient boosting DT algorithm that uses some strategies to speed up the calculation and at the same time helps counter overfitting, thereby improving the generalization ability. When calculating the feature splitting point of the CART tree, LightGBM uses the gradient-based one-side sampling strategy to narrow down the selection range. The algorithm sorts the samples according to their gradient values, selects the top samples with ratio a, and then randomly selects samples from the remaining data with ratio b, multiplies the coefficient of $(1-a)/b$ to amplify their influence. This algorithm also uses the exclusive feature bundling strategy to bind some mutually exclusive features into a bundle to further improve the calculation speed. In our experiment, the values of parameters a and b are 0.2 and 0.1, the maximum depth of each tree is 10, and the minimum data count in a tree leaf is 30.

(6) ANN

The ANN we use includes one input layer, three hidden layers, and one output layer. Neurons in adjacent layers are fully connected. The input layer has 15 nodes, which correspond to 15 features of each trajectory. Each one of the hidden layers contains 90 neurons. The output layer has five nodes that correspond to the probability of each category. We use the batch train and dropout strategies to counter the overfitting effects of this algorithm. The batch size value and dropout value we select are 2000 and 0.8.

We use sigmoid function as the activation function. Then, we calculate the softmax value of the output layers, and figure out its cross-entropy value with the real category of the sample as the loss value of this algorithm. Their formulas are

$$\text{sigmoid}(x) = \frac{1}{1+\mathrm{e}^{-x}} \tag{31}$$

$$y_{i-\text{softmax}} = \frac{\mathrm{e}^{y_i}}{\sum_{j=1}^{5}\mathrm{e}^{y_j}} \tag{32}$$

$$\text{error} = -\sum_{i=1}^{5} y_{i-\text{groundtruth}} \log_2^{y_{i-\text{softmax}}} \tag{33}$$

where $y_{i-\text{softmax}}$ represents the output value of i-th node in the softmax layer, y_i represents the i-th node value of the output layer, error represents the optimized target value of Back Propagation (BP) neural network algorithm, and $y_{i-\text{groundtruth}}$ represents the i-th element value in the vector of a sample.

Then, we use Adam gradient descent algorithm to find the minimum loss value of the train samples in one train batch. The formulas for calculating the step length of the Adam gradient descent algorithm are

$$\boldsymbol{g}_{t+1} = \frac{1}{m}\cdot\nabla\mathbf{error}_{t+1} \tag{34}$$

$$\boldsymbol{s}_{t+1} = \rho_1\boldsymbol{s}_t + (1-\rho_1)\boldsymbol{g}_{t+1} \tag{35}$$

$$\boldsymbol{r}_{t+1} = \rho_2\boldsymbol{r}_t + (1-\rho_2)\boldsymbol{g}_{t+1}\odot\boldsymbol{g}_{t+1} \tag{36}$$

$$\Delta\boldsymbol{\theta}_{t+1} = -\varepsilon\cdot\frac{\dfrac{\boldsymbol{s}_{t+1}}{1-\rho_1^{t+1}}}{\sqrt{\dfrac{\boldsymbol{r}_{t+1}}{1-\rho_2^{t+1}}}+\delta} \tag{37}$$

where $\boldsymbol{g}_{t+1}$ represents the gradient vector of the error function during the $t+1$ round of training, $\boldsymbol{s}$ and $\boldsymbol{r}$ are the intermediate variables with the value of $\boldsymbol{s}_0$ and $\boldsymbol{r}_0$ are zero, $\odot$ operator generates a vector with the same shape of the two operand vectors and each element in the result vector is equal to the multiplication result of the two corresponding elements in left and right operand vectors, $\Delta\boldsymbol{\theta}_{t+1}$ is the moving step of Adam gradient descent strategy in the $t+1$ round of training.

4 Experiment and Result

4.1 Data description

The open-source GPS trajectory dataset GeoLife provided by Microsoft Research Asia is used to verify the method proposed in this paper. This dataset was collected by 182 volunteers over a five-year period from April 2007 to August 2012. The dataset contains 24 178 078 data points, each composed of volunteer ID,

time and date, longitude, latitude, and altitude. The dataset also includes 14 718 transportation mode data provided by volunteers, with each data piece composed of volunteer ID, starting time, ending time, and mode of transportation used during the period. Matching these travel mode data with the GPS trajectory data, we found 9592 GPS trajectories with transportation mode labels. Then, the following standards are implemented to select eligible data for our experiment:

(1) The data capture frequency of GPS trajectory points is equal to or larger than 0.033 Hz, i.e., the time interval between adjacent GPS points does not exceed 30 s.

(2) At least 30 valid time intervals should exist in a trajectory.

We calculate speed, acceleration, and heading change speed of each interval in each trajectory. Then, we delete the data on intervals longer than 30 s and the next pieces immediately following them. Thereafter, the number of valid interval data in each trajectory is counted and trajectories with less than 30 valid interval data are deleted. The preceding steps left us with a total of 8308 trajectories of transportation modes in walking, bike, bus, car (taxi and private car), and subway. The statistical data are shown in Figs. 3–5.

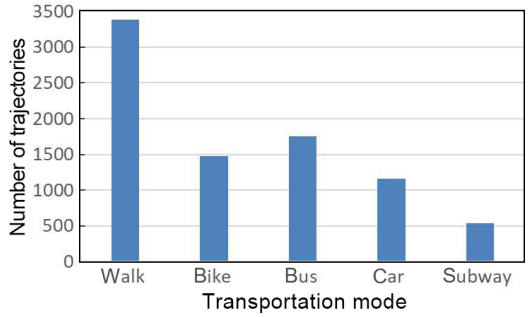


Fig. 3 Number of trajectories of five transportation modes.

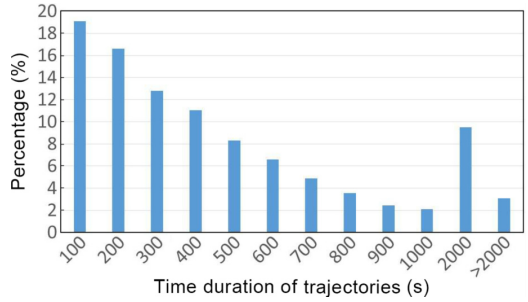


Fig. 4 Distribution of time duration of all trajectories.

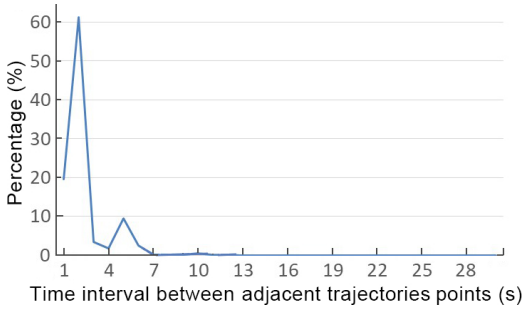


Fig. 5 Distribution of duration of all intervals between every two adjacent data points.

4.2 Feature extraction

The feature extraction method described in the Section 3 is adopted to extract features from the experiment data. Due to lack of location information on bus and subway stations, we use the POI query service[30] provided by Baidu to obtain the distances between low-speed points in trajectories and the nearest bus and subway stations. Then, we calculate the ratio of low-speed points near the bus station and the ratio of low-speed points near the subway station.

Figures 6–16 show the distributions of some feature data in our experiment.

Figures 17 and 18 show the ratio of low-speed points within 80 m from the nearest bus station among all

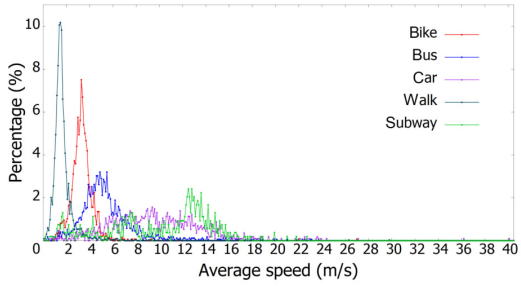


Fig. 6 Distribution of average speed.

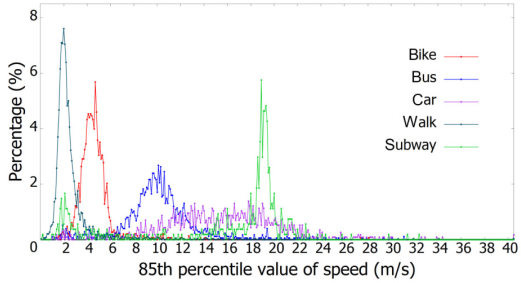


Fig. 7 Distribution of 85th percentile of speed.

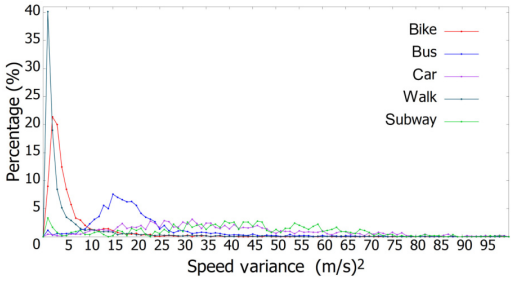


Fig. 8 Distribution of speed variance.

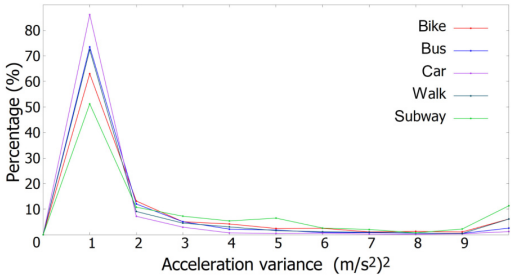


Fig. 9 Distribution of acceleration variance.

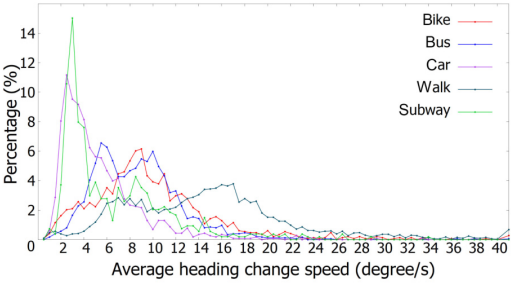


Fig. 10 Distribution of average heading change speed.

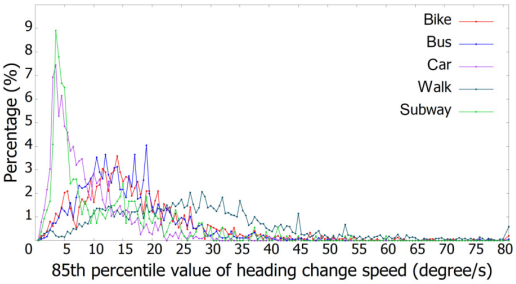


Fig. 11 Distribution of 85th percentile of heading change speed.

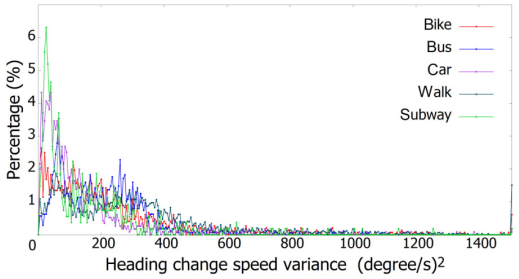


Fig. 12 Distribution of heading change speed variance.

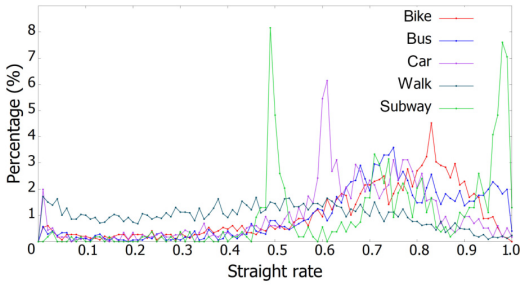


Fig. 13 Distribution of straight rate.

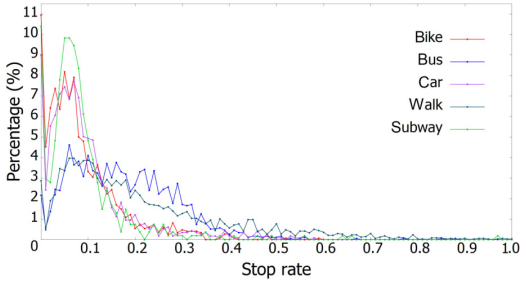


Fig. 14 Distribution of stop rate.

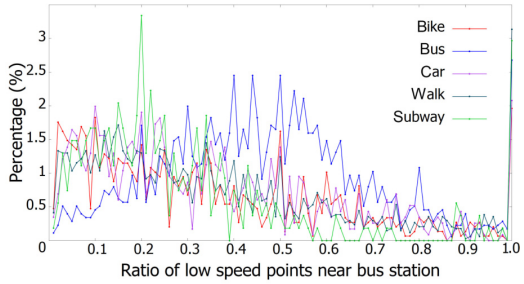


Fig. 15 Ratio of low-speed points near bus station.

low-speed points and the ratio of low-speed points within 500 m from the nearest subway station among all low-speed points for different transportation modes in trajectories that contain low-speed points ($v \leqslant 1$ m/s).

4.3 Result

For the trajectories of each transportation mode, 80% were taken as training samples and the remaining 20% as testing samples, which offered 6645 pieces of training sample trajectories and 1663 pieces of testing sample trajectories. DT, RF, AdaBoost, XGboost, LightGBN, and ANN classification algorithms were used on the training data to build classifiers. Then, they were applied to the testing and training data to calculate the accuracy. Table 1 shows the results with all of the 15 features of

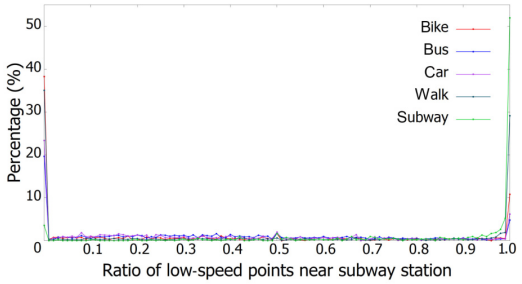


Fig. 16 Distribution of ratio of low-speed points near subway station.

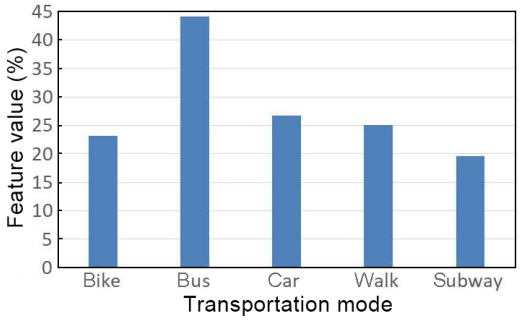


Fig. 17 Average value of Feature 14 of each transportation mode of trajectories that contain low-speed points.

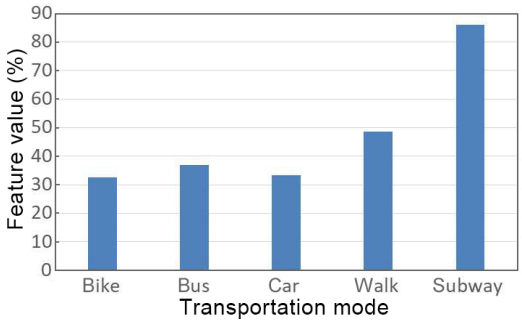


Fig. 18 Average value of Feature 15 of each transportation mode of trajectories that contain low-speed points.

Table 1 Classification accuracy of six algorithms. (%)

Classification algorithm	Train samples accuracy	Test samples accuracy
DT	88.9	86.5
RF	100.0	91.1
AdaBoost	100.0	90.5
XGboost	89.3	88.5
LightGBM	100.0	90.0
ANN	93.3	88.3

each trajectory.

The RF classification algorithm delivered the highest accuracy rate, which is 91.1% on the testing samples. Its confusion matrix is shown in Table 2.

Table 2 Confusion matrix of RF algorithm on test samples.

		Inference					Recall (%)
		Bike	Bus	Car	Walking	Subway	
Ground truth	Bike	**264**	4	0	28	0	89.2
	Bus	3	**310**	28	10	0	88.3
	Car	5	32	**190**	2	3	81.9
	Walking	13	7	1	**654**	1	96.7
	Subway	1	2	3	5	**97**	89.8
Precision (%)		92.3	87.3	85.6	93.6	96.0	–

To evaluate the contribution of GIS features, we calculated the classification accuracy of the RF classification algorithm on the conditions without using four GIS features (12–15) and without using two GIS features (14 and 15). Tables 3–5 show the accuracies and confusion matrices of the two scenarios.

These results show that using the four GIS features markedly improved the accuracy of the algorithm, especially for the bus and subway transportation modes.

5 Conclusion

In this study, we proposed a novel method of GPS trajectory transportation mode detection. We attempt to

Table 3 Classification accuracy with different features using RF algorithm. (%)

Feature	Train samples accuracy	Test samples accuracy
1–11	100	88.5
1–13	100	89.0
1–15	100	91.1

Table 4 Confusion matrix of RF algorithm with Features 1–11.

		Inference					Recall (%)
		Bike	Bus	Car	Walking	Subway	
Ground truth	Bike	**263**	5	0	28	0	88.9
	Bus	2	**298**	41	10	0	84.9
	Car	3	41	**177**	4	7	76.3
	Walking	11	5	1	**658**	1	97.3
	Subway	1	3	17	10	**77**	71.3
Precision (%)		93.9	84.7	75.0	92.7	90.6	–

Table 5 Confusion matrix of RF algorithm with Features 1–13.

		Inference					Recall (%)
		Bike	Bus	Car	Walking	Subway	
Ground truth	Bike	**266**	4	0	26	0	89.9
	Bus	2	**300**	38	11	0	85.5
	Car	5	37	**179**	3	8	77.2
	Walking	14	7	0	**654**	1	96.7
	Subway	1	3	17	6	**81**	75.0
Precision (%)		92.4	85.5	76.5	93.4	90.0	–

improve the accuracy of the algorithm by using statistical features of speed, acceleration, heading change speed, and GIS features that are easy to calculate. Experiments on open-source GPS trajectory dataset GeoLife show that compared with some traditional methods, ours can obtain higher inference accuracy.

The statistical functions of speed, acceleration, and heading change speed (average value, variance, and 85th percentile value) of each interval in the trajectory are used as nine features of the trajectory. These statistical function values can comprehensively describe the distribution of these random variables and also have good countermeasure against the relatively large GPS error of few points. Experiments on the GeoLife dataset show that these features are effective in differentiating between various transportation modes. With the nine statistical features and the other two global trajectory features, the RF classification algorithm achieved 88.6% accuracy.

We used GIS information to improve the inference accuracy of the algorithm. An advantage of our study over similar ones is full consideration of the algorithm's low consumption of storage and computing resources, as well as the ability to perform well without online data. Experiments on the GeoLife dataset show that these GIS features in our algorithm can effectively improve the inference accuracy, especially for the bus and subway modes, whose recall values increased from 84.9% and 71.3% to 88.3% and 89.8%, respectively, when the four GIS features were added.

Our method achieved relatively high inference accuracy on the GeoLife dataset shows that this algorithm can obtain good results when applied to the GPS trajectory data with low capture frequency, which could reduce the GPS data capture frequency requirement for recognizing transportation modes from GPS trajectory data. Thus, the power consumption of mobile device consumed by GPS data collection can be reduced.

We believe that many algorithms for detecting the transportation modes from the GPS trajectory data that do not utilize GIS data can improve their inference accuracy by using our GIS features, with slightly increased consumption of data storage resources and computing resources of their algorithms. The proposed algorithm can be extended to the inference of additional new transportation modes. For instance, trains can be included by adding a feature that describes the distance between low-speed points in the trajectory and the nearest railway station, and the calculation method of the new feature would be similar to the computing method of Features 14 and 15 of this algorithm. Our method does not perform well in distinguishing cars from buses. The RF classification algorithm applied to the testing samples from the GeoLife dataset resulted in a recall of only 81.9% for the car category. A ratio of 13.7% of the car mode sample was mistakenly recognized as bus. To address the defect, we can adopt the approach mentioned in Ref. [18], i.e., obtaining the real-time location of each bus in the city and calculating the distance from the GPS point to the nearest bus as a feature to raise the classification accuracy for the bus and car categories.

Acknowledgment

This research was supported in part by the National Key Basic Research and Development Program of China (No. 2017YFC0820502), the Director Foundation Project of National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data (PSRPC), and the National Natural Science Foundation of China (No. 61673233).

References

[1] E. Murakami, D. P. Wagner, and D. M. Neumeister, Using global positioning systems and personal digital assistants for personal travel surveys in the United States, in *Proc. of International Conference on Transport Survey Quality and Innovation*, doi:10.1109/CCC.2009.17.

[2] E. Murakami and D. P. Wagner, Can using global positioning system (GPS) improve trip reporting? *Transportation Research Part C*: *Merging Technologies*, vol. 7, nos. 2&3, pp. 149–165, 1999.

[3] D. Ashbrook and T. Starner, Using GPS to learn significant locations and predict movement across multiple users, *Personal and Ubiquitous Computing*, vol. 7, no. 5, pp. 275–286, 2003.

[4] Y. Zheng, L. Liu, L. Wang, and X. Xie, Learning transportation mode from raw GPS data for geographic applications on the web, in *Proceedings of the 17th International Conference on World Wide Web*, Beijing, China, 2008, pp. 247–256.

[5] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W. Y. Ma, Understanding transportation modes based on GPS data for web applications, *ACM Transactions on the Web* , vol. 4, no. 1, pp. 1–36, 2010.

[6] B. Wang, Y. Wang, K. Qin, and Q. Xia, Detecting transportation modes based on LightGBM classifier from GPS trajectory data, in *Proceedings of 26th International Conference on Geoinformatics*, Kunming, China, 2018, pp. 1–7.

[7] H. Liu and I. Lee, End-to-end trajectory transportation mode classification using Bi-LSTM recurrent neural network, in *Proceedings of 12th International Conference on Intelligent*

Systems and Knowledge Engineering, Nanjing, China, 2017, pp. 1–5.

[8] G. Xiao, Z. Juan, and C. Zhang, Travel mode detection based on GPS track data and Bayesian networks, *Computers, Environment and Urban Systems*, vol. 54, pp. 14–22, 2015.

[9] R. Brunauer, M. Hufnagl, K. Rehrl, and A. Wagner, Motion pattern analysis enabling accurate travel mode detection from GPS data only, in *Proceedings of 16th International IEEE Conference on Intelligent Transportation Systems*, The Hague, the Netherlands, 2013, pp. 404–411.

[10] C. Xu, M. Ji, W. Chen, and Z. Zhang, Identifying travel mode from GPS trajectories through fuzzy pattern recognition, in *Proceedings of Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, Yantai, China, 2010, pp. 889–893.

[11] P. Stopher, C. FitzGerald, and J. Zhang, Search for a global positioning system device to measure person travel, *Transportation Research Part C*: *Emerging Technologies*, vol. 16, no. 3, pp. 350–369, 2008.

[12] T. Bantis and J. Haworth, Who you are is how you travel: A framework for transportation mode detection using individual and environmental characteristics, *Transportation Research Part C*: *Emerging Technologies*, vol. 80, pp. 286–309, 2017.

[13] H. F. Li and W. Wang, The option model of traffic mode based on neural network, *Journal of Highway and Transportation Research and Development*, vol. 24, no. 7, pp. 132–136, 2007.

[14] E. H. Chung and A. Shalaby, A trip reconstruction tool for GPS-based personal travel surveys, *Transportation Planning and Technology*, vol. 28, no. 5, pp. 381–401, 2005.

[15] H. Gong, C. Chen, E. Bialostozky, and C. T. Lawson, A GPS/GIS method for travel mode detection in New York City, *Computers, Environment and Urban Systems*, vol. 36, no. 2, pp. 131–139, 2012.

[16] W. Bohte and K. Maat, Deriving and validating trip purposes and travel modes for multi-day GPS-based travel surveys: A large-scale application in the Netherlands, *Transportation Research Part C*: *Emerging Technologies*, vol. 17, no. 3, pp. 285–297, 2009.

[17] S. Y. A. Tsui and A. S. Shalaby, Enhanced system for link and mode identification for personal travel surveys based on global positioning systems, *Transportation Research Record*, vol. 1972, no. 1, pp. 38–45, 2006.

[18] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, Transportation mode detection using mobile phones and GIS information, in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, New York, NY, USA, 2011, pp. 54–63.

[19] F. Biljecki, H. Ledoux, and P. Van Oosterom, Transportation mode-based segmentation and classification of movement trajectories, *International Journal of Geographical Information Science*, vol. 27, no. 2, pp. 385–407, 2013.

[20] T. Feng and H. J. Timmermans, Transportation mode recognition using GPS and accelerometer data, *Transportation Research Part C*: *Emerging Technologies*, vol. 37, pp. 118–130, 2013.

[21] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, Using mobile phones to determine transportation modes, *ACM Transactions on Sensor Networks*, vol. 6, no. 1, pp. 1–27, 2010.

[22] S. Reddy, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, Determining transportation mode on mobile phones, in *Proceedings of 12th IEEE International Symposium on Wearable Computers*, Los Alamitos, CA, USA, 2008, pp. 25–28.

[23] L. Bedogni, M. Di Felice, and L. Bononi, Context-aware Android applications through transportation mode detection techniques, *Wireless Communications and Mobile Computing*, vol. 16, no. 16, pp. 2523–2541, 2016.

[24] X. Su, H. Caceres, H. Tong, and Q. He, Online travel mode identification using smartphones with battery saving considerations, *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2921–2934, 2016.

[25] X. Su. Travel mode identification with smartphone sensors, PhD dissertation, The City University of New York, New York, NY, USA, 2017.

[26] S. H. Fang, H. H. Liao, Y. X. Fei, K. H. Chen, J. W. Huang, Y. D. Lu, and Y. Tsao, Transportation modes classification using sensors on smartphones, *Sensors*, vol. 16, no. 8, pp. 1324–1339, 2016.

[27] J. V. Jeyakumar, E. S. Lee, Z. Xia, S. S. Sandha, N. Tausik, and M. Srivastava, Deep convolutional bidirectional LSTM based transportation mode recognition, in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, Singapore, 2018, pp. 1606–1615.

[28] M. C. Yu, T. Yu, S. C. Wang, C. J. Lin, and E. Y. Chang, Big data small footprint: The design of a low-power classifier for detecting transportation modes, *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1429–1440, 2014.

[29] L. Stenneth, K. Thompson, W. Stone, and J. Alowibdi, Automated transportation transfer detection using GPS enabled smartphones, in *Proceedings of 2012 15th International IEEE Conference on Intelligent Transportation Systems*, Anchorage, AK, USA, 2012, pp. 802–807.

[30] Web query interface for POI information in the area around a certain point in China, http://api.map.baidu.com/place/v2/search?query=%E5%9C%B0%E9%93%81%E7%AB%99&location=39.915,116.404&radius=2000&scope=2&output=xml&ak=Dd7v5zhGt1ISG8ClY4DSXPjgBos82rT7, 2020.

[31] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Boston, MA, USA: Wadsworth International Group, 1984.

[32] L. Breiman, Random forests, *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[33] Y. Freund, R. Schapire, and N. Abe, A short introduction to boosting, *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999.

[34] T. Chen and C. Guestrin, XGboost: A scalable tree boosting system, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.

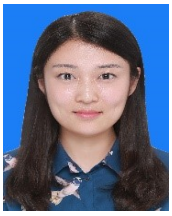
[35] G. Ke, Q. Meng, T. W. Finley, T. Wang, W. Chen, W.

Ma, Q. Ye, and T. Liu, LightGBM: A highly efficient gradient boosting decision tree, in *Proceedings of Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 3146–3154.

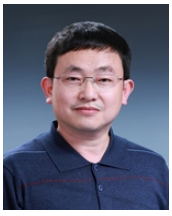
[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

Ji Li received the BS degree from Huazhong University of Science and Technology, China in 2006, and the MS degree from Peking University, China in 2014. He is currently a PhD candidate at the Department of Automation, Tsinghua University, China. He has participated in several research projects granted from MOST, NSFC, etc. His research interests include Intelligent Transportation System (ITS) and data-driven traffic safety analysis.

Xin Pei received the BEng and MEng degrees from Tsinghua University, China in 2005 and 2007, respectively, and the PhD degree from the University of Hong Kong in 2011. She is currently an associate professor at the Department of Automation, Tsinghua University. Her current research interests include road safety evaluation and driving behavior analysis. She is the Principal Investigator (PI)/co-PI of 3 road safety related NSFC projects. She has published more than 50 SCI/EI indexed papers, especially, there are 7 papers published on the *Journal of Accident Analysis and Prevention*, which is a top journal for road safety analysis.



Xuejiao Wang received the PhD degree from the Tsinghua University in 2018. She is currently an engineer in the National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data (NEL-PSRPC). Her research interests include deep learning, computational intelligence, and design of system engineering



Danya Yao received the BEng, MEng, and PhD degrees from Tsinghua Unversity in 1988, 1990, and 1994, respectively. He is currently a professor at Tsinghua University. His active research areas include vehicle infrastructure cooperation systems, advanced detection technology, and systems engineering. He has published more than 100 papers. He is the PI/co-PI of more than 5 national research programs, including 863 program, 973 program, and NSFC project. He was ever the chief expert of the National High-Tech Research and Development Program Research of China on Key Technologies of Intelligent Vehicle-Infrastructure Co-operation System.

Yi Zhang received the BEng and MEng degrees from Tsinghua University, China in 1986 and 1988, respectively, and the PhD degree from the University of Strathclyde, UK in 1995. He is currently a professor in control science and engineering at Tsinghua University. His main research interests include intelligent transportation systems, intelligent vehicle-infrastructure cooperative systems, analysis of urban transportation systems, urban road network management, traffic data fusion and dissemination, urban traffic control and management, advanced control theory and applications, advanced detection and measurement, and systems engineering.

Yun Yue received the BS degree from ChangAn University, China in 2004, and the MS and PhD degrees from Tsinghua University, China in 2011 and 2019, respectively. She is currently a postdoctor in the Department of Automation, Tsinghua University. Her research interests include road safety evaluation, traffic safety simulation, and travel behavior analysis.