IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 30, 2020, accepted July 22, 2020, date of publication August 10, 2020, date of current version August 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3015242

Transportation Mode Recognition With Deep Forest Based on GPS Data

MAOZU GUO[1,2], SHUTONG LIANG[1,2], LINGLING ZHAO[3], AND PENGYUE WANG[1,2]
[1]School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China
[2]Beijing Key Laboratory of Intelligent Processing for Building Big Data, Beijing 100044, China
[3]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

Corresponding author: Lingling Zhao (zhaoll@hit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61871020 and Grant 61305013, in part by the Science and Technology Program of Beijing Municipal Education Commission (key program) under Grant KZ201810016019, and in part by the Beijing Municipal Universities High-Level Innovation Team Construction Project under Grant IDHT20190506.

ABSTRACT Transportation mode recognition (TMR) is a common but critical task in the human behavior research field, which provides decision support for urban traffic planning, public facility arrangement, travel route recommendations, etc. The rapid development of urban information technology, mobile sensors and artificial intelligence has generated solutions for TMR; however, they rely on extra sensors and Geographic Information System (GIS) information, which are not always available. Recognition is usually simplified by disregarding the trajectories among transportation mode change points. In this paper, we proposed an ensemble learning-based approach to automatically recognize transportation modes (including a hybrid mode) using only Global Positioning System (GPS) data. A total of 72 features were extracted to better distinguish different transportation modes. Furthermore, we exploited a deep forest to combine various types of classification models, which facilitates robust learning with different trajectory samples and modes. The experimental results for the Geolife dataset show the efficiency of our approach, and the improved deep forest model achieved the best performance among all experiments that we conducted with 88.6% accuracy.

INDEX TERMS Transportation mode recognition, ensemble learning, deep forest, hybrid transportation mode.

I. INTRODUCTION

The motion behaviors of residents has a certain regularity according to a certain time cycle, and the hidden patterns and trends are crucial for urban development and governance. Transportation mode recognition (TMR) can help reveal these patterns by showing how individuals migrate among points of interests (POIs). At the urban level, TMR aids in decision-making for traffic system management, regional function divisions, public facilities layouts etc. For individuals, by inferring their life patterns and preferences from their past trajectories, TMR can provide support for many applications, such as travel route recommendations.

For the past few years, the methods employed for acquiring transportation mode data information were questionnaires or telephone interviews. However, the information collected in these traditional ways was not universal for all urban residents. The data quality is strongly affected by respondents' memories, which are commonly inaccurate and incomplete, especially when the transportation mode changes frequently in a trajectory [1]. The rapid development of urban informatization, mobile sensors and artificial intelligence has generated solutions for TMR; however, they rely on extra sensors and Geographic Information System (GIS) information, which are not always available. In recent years, vast amounts of Global Positioning System (GPS) data have been produced due to the popularity and refinement of GPS devices [2]. GPS data have a high sampling rate and accuracy and can describe individual travel behaviors more completely with greater detail.

Two primary issues in the research of machine learning-based TMR have been addressed: feature extraction from raw GPS data and selection of a classification model. Various kinematics or statistical features have been extensively

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

applied in the representation of TMR, which can be divided into two categories: point features, including velocity, acceleration, turning angle, and sinuosity; and segment features, such as mean velocity or maximum acceleration. For the different combinations of these features, decision trees [3]–[8], k-nearest neighbors [5], SVMs [3], [5], [9], [10], fuzzy systems [11], [12], ensemble learning [5], [9], [13], [14], and deep learning methods [15]–[17] have been applied to enhance the identification capability of the TMR task. However, the following problems still exist: recognition of the trajectory segmentation at transportation mode change points is usually disregarded in the training process, which simplifies the recognition task but causes errors when the transportation model changes. Additionally, in previous studies, Zheng [4] employed the maximum velocity and acceleration and the mean, variance, and expectation of the velocity as input features. Based on these features, researchers further selected more statistical measures of the velocity and acceleration, which have been applied to better explain the differences between different modes but introduce interference from human experience [18]. The increase in the number of features may also increase the complexity of the model and the computational costs. The performance of the classification model remains unsatisfactory in some real situations due to the existence of noise and outliers. Therefore, improvement in the generalization ability and robustness of the models is needed.

More recently, a decision tree-based ensemble approach, which is referred to as the deep forest [19], has been proposed. The approach contains fewer hyperparameters and lower computational costs than deep neural networks (DNNs) and provides competitive performance. Inspired by its promising capacity in classification tasks, we propose an improved deep forest method to automatically recognize transportation modes, including a hybrid mode, with only GPS data. In this framework, we employ 72 global trajectory features extracted by using statistical methods to distinguish transportation modes. Our research contributes to the field in the following ways:

1) In addition to a few general transportation modes, including walking, riding a bike, taking a bus, driving a car, and taking a subway or train, our approach is able to identify whether a segment belongs to a hybrid mode, which makes it better able to determine when and where an individual is likely to change his/her transportation mode.
2) We develop a deep forest method that combines various types of classification models, including the random forest (RF), completely RF (CRF), SVM and XGBoost, to facilitate robust learning with different trajectory samples and modes that does not require substantial effort to tune the hyperparameters compared to the effort needed for DNNs. However, the deep forest method still performs similarly or better than these models.

The remainder of our paper is organized as follows: Section 2 reviews relevant references. Section 3 presents our model, and the experimental results and discussion are shown in section 4. The conclusions are presented in section 5.

II. LITERATURE REVIEW

Over the last decade, in the field of TMR, the GeoLife dataset has become well known for its quality and quantity [3], [4], [20]. Some researchers have either employed data released by governments or independently collected data. Biljecki *et al* [11] use a dataset with 17 million GPS points from the Netherlands and elsewhere in Europe. Bantis *et al* [21] collected GPS data via a customized smartphone application (app).

Based on the collected GPS dataset, TMR can be considered a multicategory classification using a machine learning scheme; thus, two types of research efforts have been commonly employed: extraction of trajectory features and classification of TMR using these features.

A. EXTRACTING FEATURES BASED ON GPS DATA

Many researchers have extracted the statistical values of a trajectory segment as global features. Zheng *et al* [3] employed common statistical features, including the mean velocity, expected velocity, top three velocities and top three accelerations, to identify four different transportation modes (bike, bus, car, and walking). Based on their previous results, Zheng [4] then selected three features that are more advanced: the Head Change Rate (HCR), the Stop Rate (SR), and the Velocity Change Rate (VCR). The results show the capability of these features for improving the robustness of TMR models. Dodge [22] extracted a total of 58 features, including global features and local features, via a statistical method and profile decomposition. Xiao *et al* [14] further increased the number of features to 111.

B. SELECTION OF CLASSIFICATION MODELS FOR SPECIFIC DATASETS

In addition to traditional machine learning methods, the deep learning method that has developed rapidly in recent years is considered a new solution for TMR. Endo *et al* [23] first employed a DNN in the TMR task with time information only; that is, no kinematic characteristics were considered. Song *et al* [24] employed heterogeneous data in a city to build a long short-term memory (LSTM) network to simulate and predict the movement of personnel in the whole city. To address the problem of human bias when creating efficient features in traditional machine learning methods, Dabiri *et al* [25] proposed a convolutional neural network (CNN) architecture with 84.8% accuracy. Wang *et al* [15] further presented the CNN-BiGRU, which combines a CNN with a bidirectional gated recurrent unit (Bi-GRU), to better mine the timing characteristics of a trajectory. Additionally, Toan H. Vu *et al* [17] proposed an improved recurrent neural network (RNN) model that is referred to as the Control Gate-based Recurrent Neural Network (CGRNN), which is an end-to-end model that works directly with raw signals from an embedded accelerometer.

III. METHODOLOGY

A. FORMULATION

The GPS point can be denoted by three parameters with $p_i = (l_i, \; g_i, \; t_i)$, where l_i, g_i and t_i denote the latitude, longitude, and timestamp, respectively, of point p_i. The trajectory segment S_i of length n is a set of pairs of points and the corresponding transportation mode. $S_i = \{(p_1, m_1), (p_2, m_2), \ldots, (p_n, m_n)\}$, where $m_i \in \{Walk, Bike, Bus, Car, Subway, Train\}$, and $i = 1, \ldots, n$. The mode M_s of segment S_i remains the same as that of each point when all points fall into a single category. Formally, if $\forall i \in \{1, \ldots, n\}, \; j \in \{1, \ldots, n\}, \; \exists m_i \neq m_j$, then the transportation mode of the segment is defined as "Hybrid"; that is, $M_s =$ "*Hybrid*", otherwise $M_s = m_i = m_j$. The trajectory, which consists of L segments, can be formulated as $T = \{(S_1, M_1), (S_2, M_2), \ldots, (S_L, M_L)\}$. Therefore, the TMR addresses how to determine the transportation mode M_i of each segment S_i based on its corresponding GPS point sequence $[p_1, p_2, \ldots, p_n]$.

B. THE FRAMEWOR OF OUR METHOD

In this paper, we propose a deep forest-based method to automatically recognize seven transportation modes: walking, bicycle, bus, car, subway, train, and hybrid. The flowchart of our method (as shown in Figure 1) includes three modules: data processing, feature extraction and classification & evaluation. In the data processing module, the cleaned GPS data are segmented into a number of trajectory segments. For a single segment, we first extract the kinematic parameters, including the velocity, acceleration, turning angle and sinuosity, which are then applied to calculate a variety of statistical measures to serve as trajectory features. For the 72 extracted features, a deep forest model is adopted to classify different transportation modes in the classification & evaluation module.

C. DATA PROCESSING

In our study, only surface transportation modes are included; thus, the trajectories with the "airplane" and "boat" labels are pruned, and the "taxi" mode is merged into "car" due to their similarity.

For each trajectory, three preprocessing steps were conducted. First, if a GPS point has a time interval that exceeds 15 min compared to the previous point, then it is treated as the start of a new trajectory. These trajectories are divided into segments with a fixed length of m = 300. Segments with fewer than 15 GPS points are removed. The segment set is utilized as the input for the following blocks.

D. FEATURE EXTRACTION

Based on a previous study [22], the features are categorized into global features and local features. The global features represent the descriptive statistics of the entire trajectory, while the local features reveal more details about the movement behavior. However, global features have been determined to be more important for trajectory segments after theoretical analysis, which was also shown in [14] via feature importance ranking; therefore, our study selects 68 global features in terms of 17 statistical measures of the velocity v_t, the acceleration a_t, the turning angle θ_t (difference between the azimuth angles of two consecutive points) and the sinuosity s_t (winding path divided by distance). To obtain the previously mentioned 68 features, first, v_t, a_t, θ_t and s_t were calculated for each GPS point. The following 17 statistical measures were then respectively estimated for the four previously mentioned kinematic parameters for all GPS points included in a segment.

1) Mean: The mean value reflects the general value of the data in a trajectory segment.

2) Standard deviation: The standard deviation reflects the degree of dispersion in a data set.

3) Mode: The mode is the most frequently occurring value in the statistical distribution.

4) – 9) Three max and three min values: These parameters aim to reduce the impacts of abnormal points with positional errors.

10) Range: The maximum value minus the minimum value.

11) – 12) Percentiles: Measures of the positions of the data, which provide information about how the dataset is distributed between the minimum value and the maximum value. In this paper, we selected the 25th percentile (lower quartile) and the 75th percentile (upper quartile).

13) Interquartile range: The difference between the upper quartile and the lower quartile.

14) Skewness: Skewness is the digital characteristic of the degree of asymmetry of a statistical data distribution, which measures the direction and degree of the data distribution deviation. Skewness is defined as follows:

$$Skewness = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] \tag{1}$$

where μ represents the mean value of the data, and σ represents the standard deviation of the data.

15) Kurtosis: The kurtosis measures the flatness of a data distribution. If the kurtosis is steeper than a normal distribution, the value is greater than 0; otherwise, it is less than 0. It is defined as follows:

$$Kurtosis = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] - 3 \tag{2}$$

where μ denotes the mean value of the data and σ denotes the standard deviation of the data.

16) Coefficient of variation: The coefficient of variation reflects the degree of the data dispersion, which is similar to the standard deviation, and eliminates the influence of the measurement scale and dimension, whereas the standard deviation does not eliminate this influence. The coefficient is

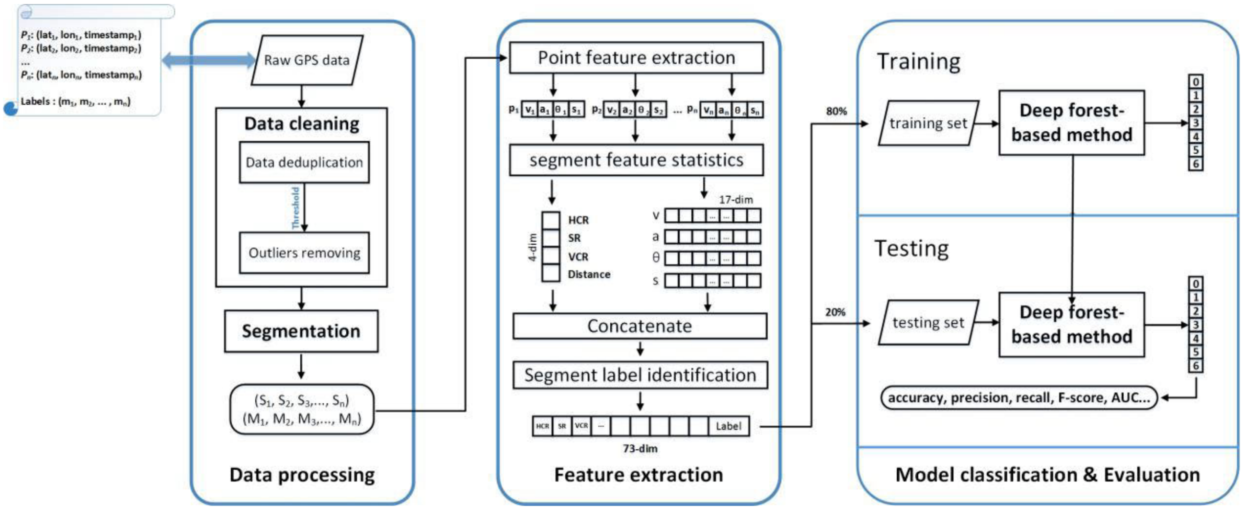

FIGURE 1. Flow chart of the methodology of transportation mode recognition.

defined as:

$$CV = \frac{\sigma}{\mu} \tag{3}$$

where μ represents the mean value of the data, and σ represents the standard deviation of the data.

17) Autocorrelation coefficient: Autocorrelation is a mathematical representation of the degree of similarity between a given time series and a lagged version over successive time intervals.

The coefficient of autocovariance is formulated as follows:

$$c_k = \frac{1}{N} \sum_{i=1}^{N-k} (x_i - \mu)(x_{i+k} - \mu) \tag{4}$$

The autocorrelation coefficient is defined as:

$$AC = \frac{c_1}{c_0} \tag{5}$$

where μ represents the mean value of the data.

We further employed three advanced features proposed by Zheng [4], the HCR, SR, and VCR, to further assess the robustness of the classification model. The thresholds for the HCR, SR, and VCR were defined as H_t, V_s, and V_c, whose values were obtained according to the accuracy changes when the HCR, SR and VCR were selected for classification. H_t, V_s, and V_c were set as 19°, 3.4 m/s, and 0.26 m/s, respectively.

1) HCR: The HCR can be regarded as the frequency with which individuals change their direction, which exceeded H_t. The HCR can better distinguish motorized and nonmotorized transportation modes.

$$HCR = |Pc| / Distance \tag{6}$$

where $p_c = \{p_i \,|p_i \in P,\ p_i.H > H_t\}$.

2) SR: By setting the threshold V_s, the SR represents how often individuals stop in their trajectories, which effectively distinguishes walking and other modes, as well as the VCR.

$$SR = |Ps| / Distance \tag{7}$$

where $p_s = \{p_i \,|p_i \in P,\ p_i.V < V_s\}$.

3) VCR: Similar to the SR, the VCR calculates the frequency with which individuals change their velocity over a certain threshold V_c.

$$P1.VCRate = |V2 - V1| / V1 \tag{8}$$

$$VCR = |Pv| / Distance \tag{9}$$

where $p_v = \{p_i \,|p_i \in P,\ p_i.VCRate > V_c\}$.

In addition, the length of each segment is involved in the feature set. Thus, a feature set including 72 global features is constructed and available for the downstream classification task.

E. DEEP FOREST-BASED CLASSIFICATION MODEL

On top of the feature layer, we apply a deep forest as the classifier due to its strong and robust prediction performance. The deep forest [19], which is also known as the Multi-Grained Cascade Forest (gcForest), is a kind of ensemble learning method that is based on a decision tree. Inspired by deep learning, the gcForest employs sliding windows of different sizes to achieve representation learning and inputs the obtained transformed features into cascade forests. The deep forest does not require high computational costs such as DNNs but achieves comparable prediction performance with other deep learning methods. In addition, it shows promising performance, even for small scale datasets.

Apart from the training strategies, the performance of the deep forest is strongly affected by the selection of its component learner. However, the original deep forest proposed by Zhou [19] uses only the RF and the CRF as component learners. Although the CRF can increase the generalization ability of the model, it cannot effectively avoid the interference of noise and outliers in the dataset. Considering the scale and dimension of the selected feature vector, we employ the RF, CRF, SVM and XGBoost as the component learners. Among these component learners, the RF, CRF and XGBoost are ensemble learning models based on decision trees. Their

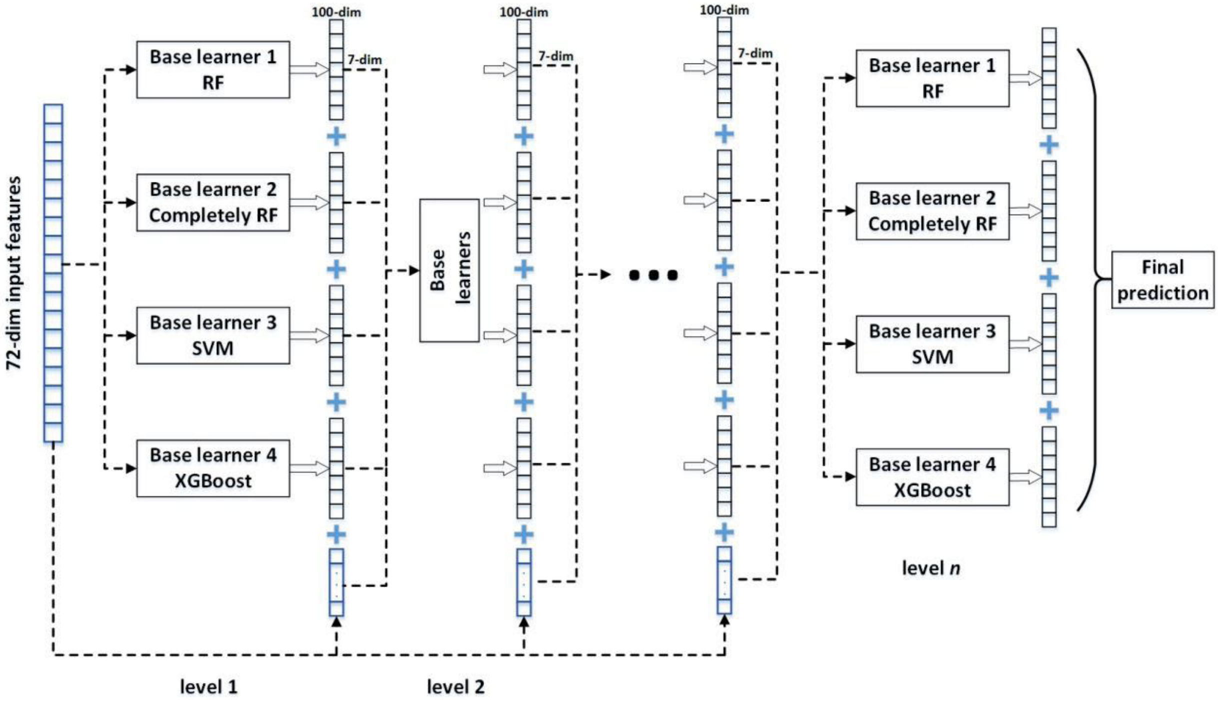

FIGURE 2. The architecture of the deep forest model employed in this paper using the RF, the CRF, the SVM, and XGBoost as the component learners.

difference is that XGBoost focuses on reducing the deviation while the RF concentrates more on reducing the variance. Compared with the RF, XGBoost is suitable for sparse data and is less likely to overfit since it adds a regularization term to control the complexity of the model. Compared to the regular RF, the CRF helps by improving the generalization ability of the deep forest since it randomly selects a single feature from the full feature space when nodes are split. The SVM is a traditional classifier, that is similar to the RF, and can work well in high dimensions; however, it usually does not perform well for large-scale datasets. Therefore, the joint utilization of these four learning models can facilitate better generalization and robustness of the proposed deep forest in various cases of available data.

The architecture of the proposed deep forest is shown in Figure 2. The 72 global features serve as the input of the cascade forest structure, which consists of n levels of component learners. Each level is composed of an SVM, an XGBoost, an RF and a completely RF to ensure the diversity of the component learners. The output vectors of the component learners in the same level are concatenated with the raw feature vector and then put on the next level as the feature representation of each learner. As a result, the layer-by-layer processing of the features is executed until there is no significant performance gain, and an average strategy is adopted to accomplish the classification at the last level.

1) RF/CRF

The RF was first proposed in 2001 [26]. As a representative bagging method, which is shown in Algorithm 1, the RF uses a decision tree as the component learner and considers the majority votes as the final result. The training procedure of the RF can be briefly described as follows:

1) Sample m samples each time by using bootstrap sampling to construct n sample subsets.
2) *n* decision trees are constructed with n subsets of samples, and each tree grows without pruning.
3) Compared with a regular decision tree, instead of using all features, each node utilizes a feature subset when splitting. The nodes of the trees in the CRF use a single randomly selected feature from all features when splitting.

```
Algorithm 1 Process of Bagging Methods
Input: Training set: D={(x_1,y_1),(x_2,y_2)...,(x_m, y_m)};
DT Algorithm:Z; Training rounds: T
Output: $H(x) = \underset{y\in\gamma}{argmax} \sum_{t=1}^{T} I(h_t(x) = y)$, where $I(x)$ represents the indicator function.
    1: for t = 1,2...,T do:
    2: $h_t = Z(D, D_{bs})$ ($D_{bs}$: sample distribution through bootstrap sampling)
    3. end for
```

2) SVM

A SVM [27] aims to find a hyperplane that can linearly divide the samples in the original sample space or a high-dimensional feature space. The model can be

TABLE 1. Common kernel functions.

Name	Functional expression
Linear kernel	$\kappa(x_i,x_j) = x_i^T x_j$
Polynomial kernel	$\kappa(x_i,x_j) = (x_i^T x_j)^d, d \geq 1$
Gaussian kernel	$\kappa(x_i,x_j) = \exp(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}), \sigma > 0$
Laplace kernel	$\kappa(x_i,x_j) = \exp(-\frac{\left\|x_i - x_j\right\|}{\sigma}), \sigma > 0$
Sigmoid kernel	$\kappa(x_i,x_j) = tanh(\beta a_i^T x_j + \theta), \beta > 0, \theta < 0$

described as

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i \kappa(x, x_i) + b \tag{10}$$

where α_i is the Lagrange multiplier, b is the displacement term that determines the distance between the hyperplane and the origin, and $\kappa(x, x_i)$ is the kernel function. The commonly employed kernel functions are listed in Table 1.

3) XGBOOST

The Classification And Regression Tree (CART) is a type of decision tree that uses the Gini index to select partition attributes. By combining a certain number of CARTs, which give a prediction score for each leaf, XGBoost [28] calculates the final score by summing each individual tree's prediction score as

$$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{11}$$

where K is the number of trees f is a function in the functional space F, and F is the set of all possible CARTs. The objective function to be optimized is given as follows:

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{12}$$

IV. RESULT AND DISCUSSION

A. DATASET

The GPS trajectory dataset employed in this paper was collected by the GeoLife project (Microsoft Research Asia) by 182 users in a period of more than three years (from April 2007 to August 2012) [3], [4], [29], [30]. The GPS trajectories in this dataset are represented by sequences of time-stamped points, each of which contains the latitude, longitude and altitude. According to the statistical results, the whole dataset contains 17,621 trajectories with a total distance of approximately 1.2 million kilometers and a total duration of more than 48,000 hours. These trajectories were recorded by GPS devices or smart phones and have a variety of sampling rates, most of which (91%) are densely logged, e.g., every 1-5 seconds or every 5-10 meters per point.

TABLE 2. Thresholds for each transportation mode.

Transportation Mode	Max Velocity (m/s)	Max Acceleration (m/s^2)
Walking	7	3
Bike	12	3
Bus	34	2
Car/Taxi	50	10
Subway	25	5
Train	80	3
Hybrid mode	/	/

TABLE 3. Number of segments for each transportation mode.

Transportation Mode	Number of segments	Proportion
Walking	4373	22.7%
Bike	3056	15.9%
Bus	3405	17.7%
Car/Taxi	2530	13.1%
Subway	822	4.3%
Train	1828	9.5%
Hybrid mode	3232	16.8%
Overall	19246	100%

In general, raw GPS data, to some extent, contains noise, outliers and gaps. Thus, we cleaned the raw dataset by removing the duplicate data that have the same timestamp information. Second, we empirically set the velocity and acceleration thresholds for each transportation mode (as shown in Table 2).

With the previous conditions, Table 3 shows the number of segments for each transportation mode.

B. MODEL EVALUATION

To better compare the performance of different classification models, we use the precision, recall, F-score, confusion matrix, receiver operating characteristic (ROC) Curve and area under the curve (AUC) as the evaluation metrics. For the multiple classification tasks, by matching the labels with the prediction results, all samples in the test dataset fall into four categories: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Among these criteria, precision indicates how many samples are actually positive compared with those predicted to be positive by the model. Recall represents how many samples are successfully identified among all the positive samples. The precision and recall are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{13}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{14}$$

The precision and recall evaluate the model from two aspects. However, in general, they are contradictory. The F-score is extensively employed due to its comprehensive consideration. The closer the F-score is to 1, the better is the model's performance.

$$\mathrm{F} = \frac{(\alpha^2 + 1)\mathrm{P} * \mathrm{R}}{(\alpha^2 * \mathrm{P}) + \mathrm{R}} \tag{15}$$

Especially, when α =1, P and R have the same weight:

$$\mathrm{F1} = \frac{2 * \mathrm{P} * \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{16}$$

The graph for the ROC curve uses the False Positive Rate (FPR) and True Positive Rate (TPR) as the horizontal axis and vertical axis, respectively, and draws the curve by traversing all thresholds. The graph works well when addressing class-imbalance problems. We calculate the AUC to quantify the performance, which is referred to as the AUC value.

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{17}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{18}$$

To assess the performance of the proposed method, we compare our method with a set of machine learning algorithms according to the previously mentioned evaluation metrics. Additionally, in our experiment, all classification models were implemented in Python.

C. EXPERIMENTAL RESULTS AND DISCUSSION

To better evaluate the effectiveness of our study, we constructed a set of comparative experiments, including RF, XGBoost, CNN and regular deep forest. The RF was employed because it was commonly employed in the TMR field and performed well in most experimental conditions. The XGBoost and CNN were conducted based on the studies carried out by [14] and [25]. These two methods were chosen because they achieved the best results as representatives of ensemble learning methods and deep learning methods. In addition, the regular deep forest is employed to verify our improvements in the ensemble structure. Based on these reasons, to ensure an unbiased and consistent test, we trained 5 models on the same preprocessed dataset, and tuned the parameters, respectively.

The dataset is randomly divided into a training set that contains 80% of the segments and a testing set with the remaining 20% of the segments. A 5-fold cross-validation method was employed. The details of different models can be obtained in Table 4, Table 5, and Table 6.

TABLE 4. Parameters of the best XGBoost model.

	Parameters	Value		Parameters	Value
1	booster	'gbtree'	7	max_depth	6
2	objective	'multi:s oftmax'	8	lambda	1
3	scale_pos_ weight	5	9	min_child_ weight	1
4	n_estimator	500	10	eta	0.05
5	gamma	0	11	nthread	-1
6	num_class	7			

TABLE 5. Structure of the optimal CNN.

Layer	Nodes	Kernel size	Stride	Activation Function
Convolution 1-2	32	13	11	ReLU
Max pooling		12	12	
Convolution 3-4	64	13	11	ReLU
Max pooling		12	12	
Convolution 5-6	128	13	11	ReLU
Max pooling		12	12	
Fully Connected	7			Softmax

TABLE 6. Parameter settings of the best deep forest model.

Component learners	Parameter settings
RF	n_estimators=300,max_features='sqrt',min_samples_leaf=10, min_samples_split=50, max_depth=6
CRF	n_estimators=300, max_features=1, max_depth=5
XGBoost	n_estimators=500,max_depth=8,objective='multi:softprob', nthread= -1, learning_rate=0.1
SVM	kernel='rbf', c=1, gamma =0.01, probability=True

The performance of the proposed models and baselines are evaluated in terms of the previously mentioned metrics. From the confusion matrixes shown in Table 7, Table 8, Table 9, and Table 10, we can visually determine that our deep forest achieves the highest accuracy at 88.6%, which is 4.6%, 0.8%, and 14.6% higher than the RF, XGBoost and CNN. For the precision rate, our deep forest ranks first for "Bicycle", "Car", "Train" and "Hybrid" and ranks second for "Walk" and "Bus". In terms of the recall rate, our model performs better than the precision rate. The improved deep forest ranks first in six categories— "Walk", "Bicycle", "Bus", "Car", "Subway" and "Train"—and ranks second for the "Hybrid" mode with a slight numerical difference. Although the recall rate of our model is slightly lower than that of the XGBoost, it achieves a higher precision rate, which yields a higher value of AUC. These results demonstrate a better performance of our method in recognizing "Hybrid" trajectory segments.

As a representative method of deep learning, the CNN had been shown to be able to perform well in the TMR task [25]. However, it can be obviously seen from the confusion matrix in Table 9 and the ROC curve in Figure 6 that the performance of the CNN is roundly lagging behind that of the RF, XGBoost and our deep forest. The overall accuracy of CNN was only 74.2%. The poor performance of the CNN may be attributed to the finding that the performance of the CNN is positively correlated with the scale of the training set. However, the segment length in our method (300) is longer than that of Dabiri [25](200), which produces a 40% reduction in the scale of the dataset.

Considering the condition of class imbalance in the dataset, the unweighted average precision (UAP) and unweighted

TABLE 7. Confusion matrix of the RF model.

RF		**Prediction**							**Recall**
		Walking	Bicycle	Bus	Car	Subway	Train	Hybrid	
Ground Truth	Walking	843	10	0	0	0	0	0	0.99
	Bicycle	45	514	8	3	0	0	8	0.89
	Bus	31	8	553	48	0	15	34	0.80
	Car	22	5	37	**381**	2	13	70	0.72
	Subway	26	3	5	18	88	6	30	0.50
	Train	0	1	5	8	0	341	2	0.96
	Hybrid	22	24	67	30	8	2	514	0.77
Precision		0.85	0.91	0.82	0.78	0.90	0.90	0.78	

TABLE 8. Confusion matrix of the XGBoost model.

XGBoost		**Prediction**							**Recall**
		Walking	Bicycle	Bus	Car	Subway	Train	Hybrid	
Ground Truth	Walking	893	0	1	1	1	10	0	0.99
	Bicycle	34	539	6	1	0	0	4	0.92
	Bus	23	7	572	37	0	4	15	0.87
	Car	20	5	34	384	8	7	56	0.75
	Subway	19	0	7	8	112	5	24	0.64
	Train	0	0	5	6	0	365	9	0.95
	Hybrid	25	32	3	29	5	19	515	0.82
Precision		0.88	0.92	0.91	0.82	0.89	0.89	0.83	

TABLE 9. Confusion matrix of CNN.

CNN		**Prediction**							**Recall**
		Walking	Bicycle	Bus	Car	Subway	Train	Hybrid	
Ground Truth	Walking	1035	46	27	24	12	0	31	0.88
	Bicycle	85	685	19	12	2	2	22	0.83
	Bus	104	21	509	150	10	6	86	0.57
	Car	51	8	75	528	5	14	35	0.74
	Subway	57	2	10	14	118	4	12	0.54
	Train	2	1	8	13	3	441	3	0.94
	Hybrid	63	19	66	65	32	9	193	0.43
Precision		0.74	0.88	0.71	0.66	0.65	0.93	0.51	

average recall (UAR) were calculated, as shown in Figure 7. The results indicate that the improved deep forest is more robust to the class-imbalance data. For example, compared with the RF, for the smallest category (subway, $\sim 4\%$), although the precision rate of the deep forest is not better than that of the RF, it has a higher recall rate, which yields a larger AUC. This finding illustrates that the deep forest tends to be more sensitive to the positive samples in the class-imbalance cases.

Based on the experimental results, we intuitively discover that:

1) Most misclassification occurs between "walking" and other modes, which may be caused by the similarity between "walking" and "driving" in the case of heavy traffic or transfer at a bus station.

2) Misclassification commonly occurs between the "hybrid" mode and other single transportation modes because the differences in the kinematic characteristics

TABLE 10. Confusion matrix of the improved deep forest model.

Improved Deep Forest		Prediction							Recall
		Walking	Bicycle	Bus	Car	Subway	Train	Hybrid	
Ground Truth	Walking	854	5	0	1	2	0	1	0.99
	Bicycle	31	589	11	0	0	0	2	0.93
	Bus	31	6	586	27	3	3	16	0.87
	Car	29	6	22	374	4	3	49	0.77
	Subway	17	4	5	7	130	4	10	0.73
	Train	0	0	4	1	2	343	5	0.97
	Hybrid	23	18	39	32	10	4	537	0.81
Precision		0.87	0.94	0.88	0.85	0.86	0.96	0.87	

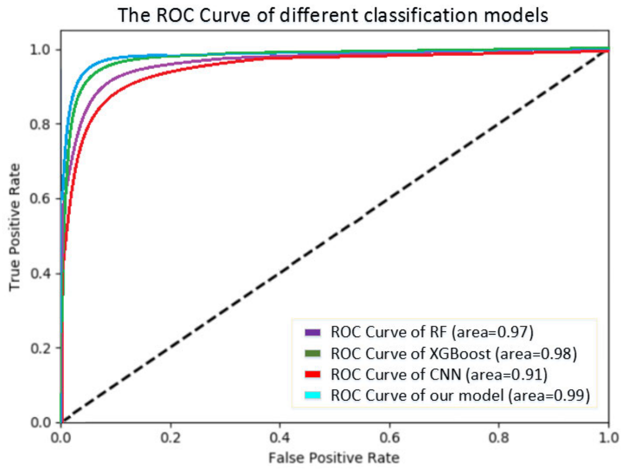

FIGURE 3. ROC Curve of different classification models.

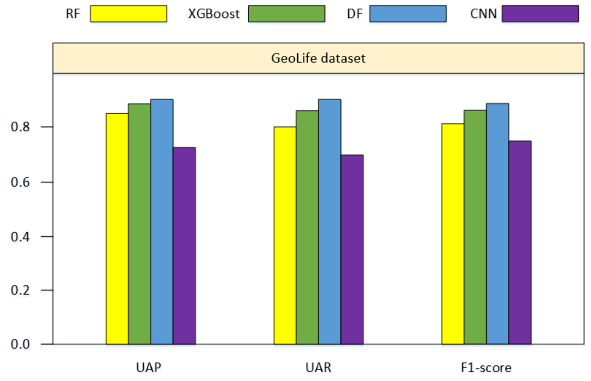

FIGURE 4. Comparison of overall performance of four models on the GeoLife dataset. The unweighted average precision, recall and F1-score for all 7 transportation mode classes were calculated.

between different transportation modes will be diluted or amplified according to their weight of the distance.

TABLE 11. Overall accuracy of selected models.

Model	Overall accuracy
CNN	74.0%
RF	84.0%
Regular Deep Forest	84.6%
XGBoost	87.8%
Our deep forest	88.6%

3) There is a high correlation between the classification performance and the number of samples of a specific mode. For example, the large number of samples and unique characteristics of the "walking" mode make its recall rate stable at approximately 99% with the RF, XGBoost and improved deep forest. However, the recall rate of the "subway" mode with the least number of samples fluctuated greatly and the value was low: only 73% for the improved deep forest, although it is more than the RF of 50%.

In terms of running time, CNN has the longest running time while RF has the shortest, which is consistent with the complexity of the model. As for XGBoost and deep forest, deep forest spends longer time than XGBoost, since XGBoost is one of the component learners of deep forest.

In the case of the same travel distance, a higher sampling rate means larger dataset size and more refined data description, both of which generally have a positive impact on the feature representation and model performance, leading to the better classification. However, the influence will be limited when the sampling rate reaches a certain level.

The overall accuracies are shown in Table 11, and the improved deep forest that we proposed are 14.6%, 4.6%, 4%, 0.8% higher than the CNN, RF, regular deep forest, and XGBoost, which demonstrates the effectiveness of our method.

V. CONCLUSION

In this paper, we presented a deep forest and trajectory global feature-based TMR model using only raw GPS data to recognize the transportation modes. In this ensemble learning framework, the SVM and XGBoost are employed as the component classifiers in addition to the RF and CRF models to enhance the diversity of component classifiers. A total of 7 transportation modes of individuals can be determined by the proposed model, including walking, bicycle, bus, car, train, and subway and hybrid modes. In particular, our method can effectively identify hybrid pattern recognition, which helps to infer an individual's transfer time and location and can improve the accuracy of travel behavior recognition. In the evaluation on the GeoLife dataset, we compared our model to the state-of-the-art baseline of conventional and deep learning methods: XGBoost and CNN as well as the commonly employed method, the RF. The evaluation results showed that our model achieves the highest accuracy of 88.6%. The combination of ensemble learning and deep learning shows the potential to be regarded as a new solution for prediction tasks in other applications.

REFERENCES

[1] X. Yang, K. Stewart, L. Tang, Z. Xie, and Q. Li, "A review of GPS trajectories classification based on transportation mode," *Sensors*, vol. 18, no. 11, p. 3741, Nov. 2018, doi: 10.3390/s18113741.

[2] E. Murakami and D. P. Wagner, "Can using global positioning system (GPS) improve trip reporting?" *Transp. Res. C, Emerg. Technol.*, vol. 7, nos. 2–3, pp. 149–165, Apr./Jun. 1999, doi: 10.1016/S0968-090X(99)00017-0.

[3] Y. Zheng, L. Liu, L. Wang, and X. Xie, "Learning transportation mode from raw GPS data for geographic applications on the Web," in *Proc. 17th Int. Conf. World Wide Web (WWW)*, Beijing, China, Apr. 2008, pp. 247–256.

[4] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on GPS data," in *Proc. 10th Int. Conf. Ubiquitous Comput. (UbiComp)*, Seoul, South Korea, Sep. 2008, pp. 312–321.

[5] A. Jahangiri and H. A. Rakha, "Applying machine learning techniques to transportation mode recognition using mobile phone sensor data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2406–2417, Oct. 2015, doi: 10.1109/TITS.2015.2405759.

[6] M. Elhoushi, J. Georgy, A. Noureldin, and M. Korenberg, "Online motion mode recognition for portable navigation using low-cost sensors," *Navigation*, vol. 62, no. 4, pp. 273–290, Dec. 2015, doi: 10.1002/navi.120.

[7] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, "Transportation mode detection using mobile phones and GIS information," in *Proc. 19th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst. (GIS)*, Chicago, IL, USA, Nov. 2011, pp. 54–63.

[8] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Trans. Sensor Netw.*, vol. 6, no. 2, pp. 1–27, Feb. 2010, doi: 10.1145/1689239.1689243.

[9] A. Jahangiri and H. Rakha, "Developing a support vector machine (SVM) classifier for transportation mode identification by using mobile phone sensor data," in *Proc. Transp. Res. Board 93rd Annu. Meeting*, Washington, DC, USA, Jan. 2014, pp. 1–14.

[10] A. Bolbol, T. Cheng, I. Tsapakis, and J. Haworth, "Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification," *Comput., Environ. Urban Syst.*, vol. 36, no. 6, pp. 526–537, Nov. 2012, doi: 10.1016/j.compenvurbsys.2012.06.001.

[11] F. Biljecki, H. Ledoux, and P. van Oosterom, "Transportation mode-based segmentation and classification of movement trajectories," *Int. J. Geograph. Inf. Sci.*, vol. 27, no. 2, pp. 385–407, Feb. 2013, doi: 10.1080/13658816.2012.692791.

[12] R. Das and S. Winter, "Detecting urban transport modes using a hybrid knowledge driven framework from GPS trajectory," *ISPRS Int. J. Geo-Inf.*, vol. 5, no. 11, p. 207, Nov. 2016, doi: 10.3390/ijgi5110207.

[13] Q. Zhu, M. Zhu, M. Li, M. Fu, Z. Huang, Q. Gan, and Z. Zhou, "Identifying transportation modes from raw GPS data," in *Proc. Int. Conf. Young Comput.*, Singapore, Aug. 2016, pp. 395–409.

[14] Z. Xiao, Y. Wang, K. Fu, and F. Wu, "Identifying different transportation modes from trajectory data using tree-based ensemble classifiers," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 2, p. 57, Feb. 2017, doi: 10.3390/ijgi6020057.

[15] M. Guo, P. Wang, and L. Zhao, "Research on recognition method of transportation modes based on deep learning," *J. Harbin Inst. Technol.*, vol. 51, no. 11, pp. 1–7, Feb. 2019, doi: 10.11918/j.issn.0367-6234.201902039.

[16] X. Liang and G. Wang, "A convolutional neural network for transportation mode detection based on smartphone platform," in *Proc. IEEE 14th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Oct. 2017, pp. 338–342, doi: 10.1109/MASS.2017.81.

[17] T. H. Vu, L. Dung, and J.-C. Wang, "Transportation mode detection on mobile devices using recurrent nets," in *Proc. ACM Multimedia Conf. (MM)*, Oct. 2016, pp. 392–396, doi: 10.1145/2964284.2967249.

[18] W. Dong, J. Li, R. Yao, C. Li, T. Yuan, and L. Wang, "Characterizing driving styles with deep learning," 2016, *arXiv:1607.03611*. [Online]. Available: http://arxiv.org/abs/1607.03611

[19] Z. H. Zhou and J. Feng, "Deep forest: Towards an alternative to deep neural networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Feb. 2017, pp. 1–7.

[20] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, "Understanding transportation modes based on GPS data for Web applications," *ACM Trans. Web*, vol. 4, no. 1, pp. 1–36, Jan. 2010, doi: 10.1145/1658373.1658374.

[21] T. Bantis and J. Haworth, "Who you are is how you travel: A framework for transportation mode detection using individual and environmental characteristics," *Transp. Res. C, Emerg. Technol.*, vol. 80, pp. 286–309, Jul. 2017, doi: 10.1016/j.trc.2017.05.003.

[22] S. Dodge, R. Weibel, and E. Forootan, "Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects," *Comput., Environ. Urban Syst.*, vol. 33, no. 6, pp. 419–434, Nov. 2009, doi: 10.1016/j.compenvurbsys.2009.07.008.

[23] Y. Endo, H. Toda, K. Nishida, and A. Kawanobe, "Deep feature extraction from trajectories for transportation mode estimation," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, Berlin, Germany, Apr. 2016, pp. 54–66, doi: 10.1007/978-3-319-31750-2_5.

[24] X. Song, H. Kanasugi, and R. Shibasaki, "Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, San Francisco, CA, USA, Jul. 2016, pp. 2618–2624.

[25] S. Dabiri and K. Heaslip, "Inferring transportation modes from GPS trajectories using a convolutional neural network," *Transp. Res. C, Emerg. Technol.*, vol. 86, pp. 360–371, Jan. 2018, doi: 10.1016/j.trc.2017.11.021.

[26] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[27] C. Cortes and V. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[28] T. Q. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf.*, Aug. 2016, pp. 758–794, doi: 10.1145/2939672.2939785.

[29] Y. Zheng, X. Xie, and W.-Y. Ma, "GeoLife: A collaborative social networking service among user, location and trajectory," *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, Jun. 2010.

[30] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proc. 18th Int. Conf. World Wide Web (WWW)*, Madrid, Spain, Apr. 2009, pp. 791–800, doi: 10.1145/1526709.1526816.

MAOZU GUO received the Ph.D. degree from the College of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. He is currently a Professor with the School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China. His current research interests include machine learning, bioinformatics, and image processing.

SHUTONG LIANG received the B.S. degree from the College of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China, in 2017, where he is currently pursuing the M.S. degree. His research interests include machine learning and urban computing.

LINGLING ZHAO received the Ph.D. degree in computer application technology, in 2011. She is currently a Lecturer with the College of Computer Science and Technology, Harbin Institute of Technology. Her research interests include machine learning and urban computing.

PENGYUE WANG received the bachelor's degree from the School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China, where she is currently pursuing the Ph.D. degree with the School of Architecture. Her current research interests include machine learning and urban computing.

• • •