Transportation Research Part A 136 (2020) 282–292

Contents lists available at ScienceDirect

Transportation Research Part A

journal homepage: www.elsevier.com/locate/tra

Coupled application of generative adversarial networks and conventional neural networks for travel mode detection using GPS data

Linchao Li[a], Jiasong Zhu[a,*], Hailong Zhang[b,*], Huachun Tan[b], Bowen Du[c], Bin Ran[d]

[a] College of Civil and Transportation Engineering Shenzhen University, Shenzhen 518061, China
[b] School of Transportation Southeast University, Nanjing, Jiangsu 211189, China
[c] State Key Laboratory of Software Development Environment Beihang University, Beijing 100191, China
[d] Department of Civil and Environmental Engineering, University of Wisconsin-Madison, Madison 53711, United States

ARTICLE INFO

Keywords:
Deep learning
Trajectory data
Classification
Transportation planning

ABSTRACT

Inferring travel modes of travelers in the city is important to transportation planning and infrastructure design. Based on the distribution of travel modes, transportation engineers could provide some proper strategies to reduce traffic congestion and air pollution. With advanced sensing techniques, it is possible to collect high-resolution GPS trajectory data of travelers and we can infer travel modes using some popular machine learning methods. One of the difficult tasks facing the application of machine learning especially deep learning in travel mode detection is the lack of large, labeled dataset, because to label the trajectory data is expensive and time-consuming. Moreover, samples of different travel modes are always unbalanced. Accordingly, in this paper, we take advantage of the generative model and the Convolutional Neural Networks (CNN) to develop a hybrid travel modes detection model using less labeled trajectory data. Our key contribution is the utilization of a generative adversarial network (GAN) to artificially create some training samples in such a way that it not only increases the required sample size but balances the dataset to improve the accuracy of the detection model. Furthermore, CNN is applied to extract deep features of trajectory data, and then to classify the travel modes. The results show that the highest accuracy (86.70%) can be achieved by the proposed model. In particular, the proposed method can improve the detection accuracy of bus and driving modes because it can solve the small sample size problem. Moreover, the large sample size can provide an opportunity to develop some advanced deep learning models in future studies.

1. Introduction

Travel mode detection is important to transportation planning. Based on the distribution of travel modes, transportation engineers could generate some proper strategies to reduce traffic congestion and air pollution. For example, a new bus route could be designed based on the current travel mode distribution of different traffic analysis zones to transfer private transport ridership to public transport ridership. The information from travelers is the base to infer their travel modes and there are many methods to collect the travel information. Traditional methods to obtain the travel mode choice are household surveys and telephone interviews (Stopher

* Corresponding authors.
E-mail addresses: zhujiasong@gmail.com (J. Zhu), zhlheisenberg@163.com (H. Zhang), tanhc@seu.edu.cn (H. Tan), dubowen@buaa.edu.cn (B. Du), bran@uwc.edu (B. Ran).

https://doi.org/10.1016/j.tra.2020.04.005
Received 25 October 2019; Received in revised form 25 March 2020; Accepted 8 April 2020
Available online 27 April 2020
0965-8564/

and Greaves, 2007; Gadziński, 2018; Dabiri et al., 2019). However, there are some limitations of traditional surveys such as low accuracy (Stopher and Greaves, 2007; Li et al., 2018), time-consuming (Gadziński, 2018), and expensive. These limitations call for new data sources automatically collected to infer travel modes. In recent years, smartphone data, smart card data, and social media data have been widely used (Zhong et al., 2018; Huang et al., 2019; Dabiri and Heaslip, 2019). The main advantage of automatic technologies is its massive data collection on spatial and temporal trajectories from which we can easily infer the travel origin-destination and the travel time. The most popular technology applied in previous studies and projects to collect behavior of travelers was the smartphone with internal GPS. With the smartphone, it is possible to obtain detailed data of travelers over many days. The device can collect precise geography and time of many points on the travel route, and then it is possible to obtain extremely accurate speed data. Also, the acceleration or deceleration can be calculated based on the speed data, which are also important to infer the travel mode (Stopher and Greaves, 2007). With speed and the acceleration or deceleration, many derived variables are artificially selected as the input of detection models. Most models have been developed based on supervised learning methods, and therefore some samples should be labeled by travelers.

The same as traditional surveys, one limitation of surveys by smartphones is its expense. Some fees have to be paid to travelers to label their trips. As introduced in (Stopher and Greaves, 2007), each device should cost 750$ to achieve one week of data collection. Thus, it is expensive to collect a large number of labeled samples. Another limitation is that the survey data is unbalanced that is the number of labeled travel modes, such as the car, the bus, and the train in the dataset is unequal. In this study, our main objective is to improve the application of the smartphone to detect travel modes. A method, containing three steps: data preparation, sample generation, and mode detection, is proposed. The key contributions of the study are summarized as follows:

(1) **Applying the GAN to augment the sample size and balance the dataset.** A deep generative model is used to capture the distribution of different travel modes and to generate some fake samples. The unsupervised model is an attempt to reduce the cost of the survey by smartphones. Moreover, the combination of fake samples and real samples can provide a balanced dataset for the detection model.
(2) **Building a balanced CNN to detect travel modes.** The popular CNN is built using the generated balanced dataset to automatically extract features from raw trajectory data and to achieve the detection task.
(3) **Conducting a set of experiments to evaluate the performance of the proposed framework.** The results show that the proposed framework outperforms some state-of-the-art detection models. Moreover, the results demonstrate the superiority of the generative model.

The rest of the paper is organized as follows. After the introduction, we review the previous literature and summarize the existing problems in Section 2. In Section 3, we present the overview of the proposed framework including data preparation, sample generation, and mode detection. The data used to evaluate our proposed model is introduced and preprocessed in Section 4. Then, in Section 5, we show the comparison of the proposed model with some traditional models. Finally, the findings in the experiment are concluded in Section 5.

2. Literature review

In previous studies, many data sources collected by smartphones, such as GPS data and GSM data, have been used to travel mode detection. The travel mode detection was always defined as a classification task and then solved by some machine learning models. Before the development of detection models, the feature selection step should be conducted. In this section, we will review the literature about feature selection and mode detection methods.

2.1. Feature selection

As mentioned in (Dabiri et al., 2019; Dabiri and Heaslip, 2019), feature selection was one of the major tasks in the studies about travel mode detection using GPS data. In Yang et al., features to describe the GPS trajectories were summarized into two types: geometric variables and kinetic variables. The geometric variables included straightness, straightness index, tortuosity, and fractal dimension which could describe the shape and pattern of the whole GPS trajectory. In a city, different travel modes often shared the road network, so the geometric variables were rarely used (Yang et al.). The kinetic variables included speed, acceleration or deceleration, turning angle, and the distance. Moreover, to further capture the difference between different travel modes, some variables derived from the above kinetic variables were also used such as their standard deviation, median value, skewness, and frequency.

In (Gonzalez et al., 2010), different combinations of variables were used as input of the neural network to select the best one. The result showed the maximum speed, maximum acceleration, total distance, and average distance between critical points could contribute to the accuracy. Moreover, to reduce the negative influence of the noise data, a critical point selection method was proposed. In (Nitsche et al., 2014), the Kalman filter, in which the acceleration was assumed to obey a Gaussian distribution with zero mean, was applied to smooth the trajectory. Then, a total of 77 features, such as the 5th, 50th, and 95th percentile of speed, deceleration or acceleration, and bearing rate, were selected as the input of the detection model. In (Lari, 2015), the importance of variables was ranked by mean decrease Gini of a random forest. The results showed that the most influential factor was the speed, and then the accuracy of tracking GPS data. The other four variables: delta bearing, delta speed, delta acceleration, and acceleration had a similar influence.

It can be concluded that many variables have been defined and artificially selected in previous studies (Wu). A small number of studies has been made to automatically extract features from the raw GPS trajectories. However, artificially feature selection is a time-consuming task and it is difficult to select all possible features which will be useful for the model. Moreover, engineers working on traffic planning might build up bias for some features because of the lack of experience in trajectory data (Li et al., 2019b, 2019a). With the emergence of the deep learning theory, it is possible to extract high-level representations for detection models (Dabiri et al., 2019). Thus, in this paper, we only simply convert the raw trajectories into four widely used variables and then applying a deep learning model to automatically extract features.

2.2. Mode detection models

The travel modes detection can be defined as a classification task in the machine learning area. After the selection of the important features, some supervised models have been applied to achieve the classification task. The neural network is one of the most popular machine learning models to detect travel modes. In (Yang et al., 2015), the neural network was applied to detect the travel mode of each trip segment. The results showed that the accuracy was more than 86%. However, the neural network was easy to obtain a local optimum (Wu; Li et al., 2016). In (Byon et al., 2014), the neural network was applied using traditional GPS data and data collected by smartphones. It could be found that it was very important to extract the deep features from the raw GPS trajectory. A shallow neural network might be insufficient to handle this task. Another popular machine learning model is the random forest. In (Lari, 2015), the random forest with different parameters was tested which can not only achieve the classification task but also rank the importance of the variables. In (Stenneth et al., 2011), five classification models including Bayesian network, decision tree, random forest, naïve Bayesian, and multilayer perceptron were applied. The results showed the accuracy of the random forest was the highest. However, the number of samples is insufficient because of the difficulty to collect GPS data. The support vector machine was also applied to detect travel modes. In the study, it could be found that support vector regression outperformed other models for specific modes. Moreover, the result showed no significant improvement could be obtained by increasing more features while the improvement could be obtained by increasing the training samples (Jahangiri and Rakha, 2015). Similar to the above machine learning models, the Bayesian network was applied in (Xiao et al., 2015).

Although a large number of machine learning models have been applied in previous studies, they were criticized by their accuracy. Moreover, in traditional machine learning models, most of the input features should be identified by domain experts to reduce the complexity of GPS data and make the pattern of the GPS data visible to learning models to work. Deep learning models can automatically extract features from the data and therefore we need not worry about feature engineering. Moreover, deep learning models are good at handling complex problems. Because of the above advantages, deep learning models also were widely used to detect travel modes. In (Dabiri et al., 2019), a conventional autoencoder architecture is proposed to extract the relevant features from GPS data. Furthermore, the model can also exploit important information from unlabeled samples. In (Zhang et al., 2019), a deep multi-scale learning model was proposed to learn features in different time and space scales. It's worth noting that in this paper the trajectories were mapped into grid data and converted into images. In (Endo et al., 2016), a fully-connected deep neural network was applied to extract deep features.

It proved that deep learning models were effective and efficient to extract features contributing to detection models. We still face two difficulties when deep learning models were applied. The first one is the sample size. Due to the high cost of the survey by smartphone, the sample size of the collected dataset is always small. Generally, it pointed out a small training sample size could not result in an effective and reasonable approximation between input variables of trajectory data and output travel modes in previous studies. The second difficulty is unbalanced dataset. For example, in (Bolbol et al., 2012) the number of samples by walk (1130) was much more than the number of samples by train (104). When the training dataset is unbalanced, the deep learning models are easily fallen into suboptimum because sometimes the minority samples are considered as noise (Haixiang et al.). Thus, we apply the GAN to solve the above two difficulties.

3. Methods

Here our hybrid deep learning framework which is specially tailored to mine the large, low-cost, noisy GPS trajectory data from the mobile phone for travel mode detection is introduced. The general working of the developed framework is shown in Fig. 1. In the first step, we need to preprocess the raw data and construct the input samples for the model. After that, the GAN is applied to augment the sample size and balance the dataset. In the next step, the CNN classifier is trained using the balanced dataset and the method is evaluated.

3.1. Data preparation

The smartphone's GPS can receive signals from the satellites to triangulate where a traveler is and what time it is. Each GPS point can be represented by a tuple of latitude, longitude, and timestamp $p_i = [lat_i, lon_i, t_i]$. During a trip, a sequence of GPS points $t = [p_1, \cdots, p_N]$ can be automatically collected. However, to build a detection model based on the supervised learning theory, travel modes should be artificially marked by travelers for their trips. Moreover, the samples input to the detection model should have the same size. To generate samples to train the detection model, the following method is used.

By matching raw trajectory dataset and label dataset, we can obtain **trajectory segments (TS)** with labels for each traveler. If the time interval of two consecutive points in one TS is more than 1 min, one TS will be divided into two **sub-TS.** Then, all sub-TS with an

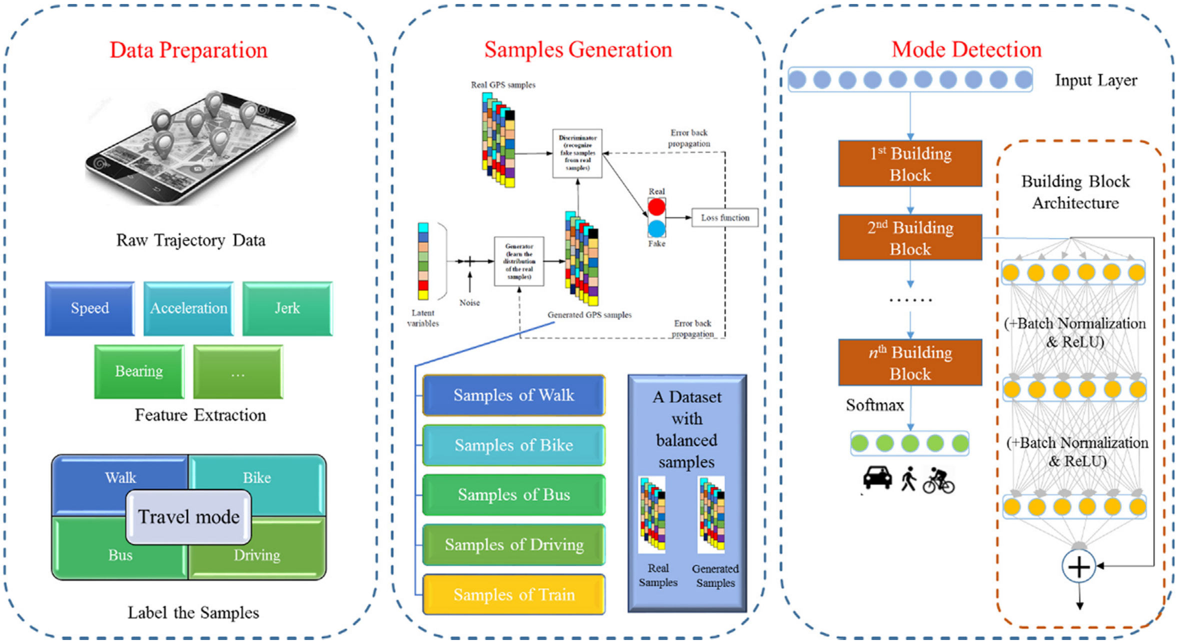

Fig. 1. Overview of the proposed framework for inferring travel mode using GPS trajectories (details of sample generation part can be found in Fig. 2).

identical transportation mode are then concatenated together, and a **whole segment (WS)** can be obtained. Finally, the WS is divided into **samples** using a fixed mutual exclusive window. The number of GPS points in the remaining last one segment is less than the time window and was directly deleted. After that, each segment is defined as a sample and motion variables for each GPS point can be computed using the temporal and spatial information.

According to some previous studies, four variables, the speed, acceleration or deceleration, jerk, and bearing rate, are selected to represent the characteristics of the motion on each segment. The first variable, speedS_{p_i}, can be easily calculated by the distance and time interval between two consecutive points using the following equation:

$$S_{p_i} = \frac{V(p_i, p_{i+1})}{t_{i+1} - t_i} \tag{1}$$

where V is a function to calculate the geodesic distance between p_1 and p_2. In this study, Vincenty's Formulae is applied. Then, based on the speed, the acceleration or deceleration and jerk can be calculated according to the following equations:

$$A_{p_i} = \frac{S_{p_{i+1}} - S_{p_i}}{t_{i+1} - t_i} \tag{2}$$

$$J_{p_i} = \frac{A_{p_{i+1}} - A_{p_i}}{t_{i+1} - t_i} \tag{3}$$

where A_{p_i} is the accelerationor deceleration of point p_i and J_{p_i} is the jerk of point p_i. As demonstrated in (Bagdadi and Várhelyi, 2013), the acceleration or deceleration and the jerk are two critical variables that affect the behaviors of the travelers and thus contribute to the development of the detection model. The last variable is the bearing rate which represents the change rate of the driving direction. According to the conclusions in (Zheng et al., 2010), a traveler by bike or walk modes changes his direction more frequently than a traveler by bus or car. The bearing rate br can be calculated by the Eqs. (4)–(7).

$$y = sin[p_{i+1}(lon) - p_i(lon)] \times cos[p_{i+1}(lat)] \tag{4}$$

$$x = cos[p_i(lat)] \times \sin[p_{i+1}(lat)] - sin[p_i(lat)] \times \cos[p_{i+1}(lat)] \times \cos[p_{i+1}(lon) - p_i(lon)] \tag{5}$$

$$B_{p_1} = arctan(y, x) \tag{6}$$

$$br_{p_1} = |B_{p_1} - B_{p_2}| \tag{7}$$

where sin, cos, and $arctan$ are three trigonometric functions; $p_i(lon)$ and $p_i(lat)$ are latitude and longitude of the GPS point p_i.

By the above method to generate samples, it can be found one sample could contain GPS points from different **sub-TS**. The calculation of variables for the joint points are different from that of other points. For example, if a sample contains GPS points from two **sub-TS,** and two joint points are jp_i, jp_{i+1}, four variables (S_{jp_i}, A_{jp_i}, J_{jp_i}, br_{jp_1}) are replaced by the mean value of other points in this sample.

After calculating the variables of each segment, input samples of the detection model should be reconstructed. For each GPS point, four variables can be obtained. Then, each sample is a $4 \times N$ matrix where$N$ is the number of points in each segment. In the following sections, the matrix will be used as inputs of the generative model and the detection models.

3.2. Samples generation

The commonly applied GAN contains two parts: a generator $G(\mathbf{z}; \theta_g)$ which is used to generate new samples $G(\mathbf{z}) \in \mathbb{R}^d$ from a random prior $\mathbf{z} \in \mathbb{R}^r$ and a discriminator $D(\mathbf{x}; \theta_d)$ which is used to recognize whether a newly generated sample is real or fake. The aim is to train a generative model G which can maximize the probability of the discriminative model D misclassifying real sample and generated sample (Goodfellow et al., 2014). As demonstrated in (Goodfellow et al., 2014), the framework of the GAN can be abstracted as a simple minimax two-player game which will coverage when Nash equilibrium is satisfied. Thus, the objective of GAN is to minimax the following objective function:

$$\min_G \max_D V(G, D) = \mathbb{E}_{\mathbf{x}\ p_{data}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\ p_{\mathbf{z}}}[\log(1 - D(G(\mathbf{z})))] \tag{8}$$

where p_{data} and $p_{\mathbf{z}}$ represent the distribution of the real sample and random prior distribution such as Gaussian distribution, respectively. During the training process, the parameters of G and D are updated as following two equations:

$$\theta_d \leftarrow \theta_d + \alpha \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} (\log D(\mathrm{x}_i) + \log(1 - D(G(\mathrm{z}_i)))) \tag{9}$$

$$\theta_g \leftarrow \theta_g - \alpha \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(\mathrm{z}_i))) \tag{10}$$

where m represents the number of training samples and α is the step size. As demonstrated in (Goodfellow et al., 2014), the parameters of generator G are optimized by maximizing $\log(D(G(\mathbf{z})))$ to fast the training of the GAN. Thus, the **Equation (10)** can be rewritten as:

$$\theta_g \leftarrow \theta_g + \alpha \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D(G(\mathrm{z}_i))) \tag{11}$$

In this study, an alternative training method is applied which contains two steps. In the first step, the generator G is fixed and the discriminator D is optimized by maximizing its accuracy. In the second step, the discriminator D is fixed and the generator G is optimized by minimizing the accuracy of discriminator$D$. When$p_{data} = p_{\mathbf{z}}$, the training process is stopped. The architecture of a GAN is shown in Fig. 2 in which the two models can be any type of multilayer perception. In this study, two full connected neural networks are applied as generator and discriminator, respectively.

3.3. Mode detection

After the generation of samples for different travel modes, a balanced dataset can be obtained. Then, the detection model based on CNN is applied to finish the mode detection task. The used CNN contains four main parts: input part, travel mode feature extraction part, flatten part, and mode output part.

The first part is the input layer. As demonstrated in the above subsection, the input sample is a $4 \times N$ matrix which can be written as:

$$x^t = [S_t, A_t, J_t, br_t], t \in [1, \cdots, T] \tag{12}$$

where t is the sample index, T is the total number of samples, and S_t, A_t, J_t, br_t are vectors whose elements represent speed, acceleration or deceleration, jerk and bearing rate of a GPS point on the tth segement.

The second part of the CNN is to extract the features of travel modes from the input data achieved by alternating and stacking the conventional layers and the pooling operations. The conventional layer receives input from the previous layer and convolves it by multiple learnable filters. The subsequent pooling layer extracts the most important features with equal length vectors by pooling operators. In this study, max pooling that selects the maximum value of the mapped region as the most important feature is applied. Denote the input and output of the lth layer by x_l^k ando_l^k, where k is the index of the channel in the conventional layer. The output of the lth layer can be obtained by:

$$o_l^k = maxpool\left[\varphi\left(\sum_{j=1}^{c_{l-1}} (W_l^k x_l^j + b_l^k)\right)\right], k \in [1, c_l] \tag{13}$$

where $\varphi(\bullet)$ is the activation function and c_l is the number of filters in the lth conventional layer.

The flatten part is easy. After finishing the previous two parts, we can obtain the extracted features in a matrix or a tensor. As the name of this step implies, the features are concatenated as a vector which can be written as:

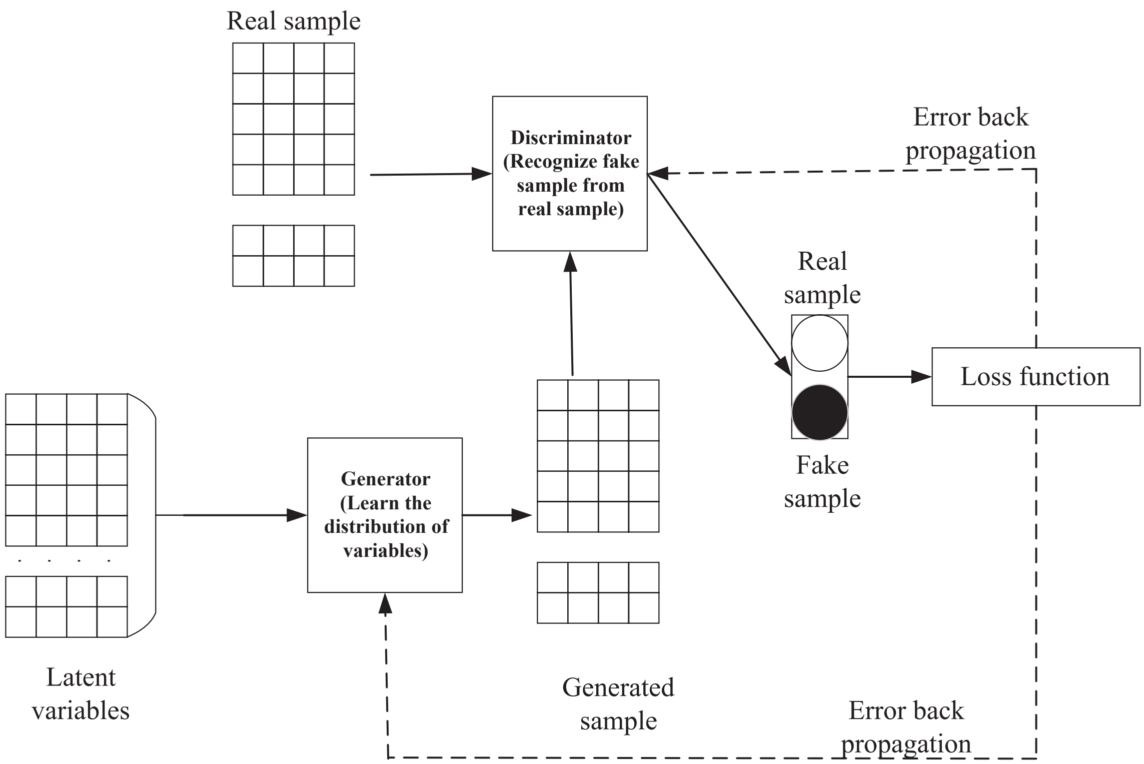

Fig. 2. Framework of the proposed GAN-based method.

$$o^{flatten} = flatten([o_l^1, \cdots, o_l^k]), \ k = c_l \tag{14}$$

The final part of CNN is to output the travel mode that is to finish a classification task using the vector of the flatten part as the input. Fully connected layers are used in this study. The same as the traditional multilayer perception, nodes in a fully connected layer are connected to all nodes in its previous layer. Between the output layer and the last fully connected layer, the SoftMax activation function is used to calculate the probability distribution of all travel modes. Moreover, to avoid the overfitting problem which always appears in CNN, regularization is also applied when we train the detection model. Dropout is an effective technique widely applied to solve the overfitting problem in CNN. The parameter selection and the training procedure of CNN in this study are according to the tutorial in (Dabiri and Heaslip, 2018; Ma et al., 2017).

4. Data

In this study, the proposed model was evaluated using a real-world GPS trajectory dataset collected by 178 users in the GeoLife project (Zheng et al., 2010; Zheng et al., 2008; Zheng). The trajectories, represented by a sequence of points with timestamps, were recorded by GPS-installed smartphones. The locations of the users were updated in a dense representation with latitude and longitude. In the dataset, 69 users manually labeled their transportation modes including walk, bike, car, taxi, bus, train and airplane which can be used in our study. According to the study in (Dabiri and Heaslip, 2018), the raw data could be preprocessed using the following three steps.

(1) Firstly, we deleted the data labeled with the airplane, because the main purpose of the study is to detect the land transportation. The car and taxi were combined as driving because they had the same characteristics. Therefore, the transportation modes could be defined as five classes: walking, bike, driving, bus, and train.
(2) Then, labels marked by travelers were matched to the corresponding GPS trajectories. Each trip of a traveler was divided into segments with the same number of GPS points in the second step. According to the suggestion in (Dabiri and Heaslip, 2018), each segment contained 200 GPS points. After the division, a total of 25,955 segments were obtained. The window size is set as 200, because the experiment shows the accuracy does not increase much when the window size is more than 200 (Fig. 3).
(3) Finally, the speed, acceleration or deceleration, jerk, and bearing rate were calculated for each point. Different from the method in (Dabiri and Heaslip, 2018), we did not remove outliers and GPS points containing random errors because this study aimed to to build a robust infer model. The final obtained dataset was called as the raw dataset.

In the raw dataset, it could be found that the sample sizes of five types of travel modes were not equal that is dataset was

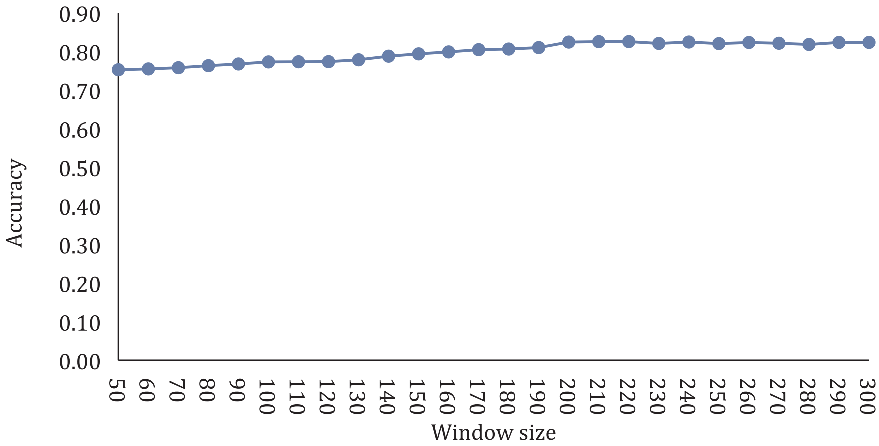

Fig. 3. Accuracy of different window size.

unbalanced. To solve this problem, this paper proposed a GAN-based method and the samples are artificially generated. The generated samples constituted a dataset which was called as the generated dataset. The sample sizes of each travel mode in the raw dataset and the generated dataset are shown in Table 1. The total dataset, in which the ratio of sample sizes for five travel modes approximates 1:1:1:1:1, was a combination of the raw dataset and the generated dataset. In addition, average values of the four variables were calculated and shown in Table 1. It can be seen the values of five travel modes are different which indicates the effectiveness of the selected variables. In contrast, it can be found the values of the raw dataset and the generated dataset are close which indicates the GAN-based method can generate some similar samples.

5. Results

All methods in this study have been conducted using the Python programming language. The CNN and GANs have been built using Keras on Tensorflow beckend. The ratio of training samples and test samples was 4:1. Two widely used classification metrics: accuracy (A) and F-measure (F) have been calculated to compare the performance of detection models.

$$A = \frac{\text{Number of true classified samples}}{\text{Number of all samples}} \tag{15}$$

$$F = \frac{2 \bullet \text{precision} \bullet \text{recall}}{\text{precision} + \text{recall}} \tag{16}$$

5.1. A comparison of generative samples with raw samples

In the first experiment, the GAN was applied whose parameters were shown in Table 2. As shown in Table 1, the generated datasets for different travel modes have different sample sizes. The average values of the four variables for different travel modes

Table 1
Description of the Raw Dataset and Generated Dataset.

Mode		Walk	Bike	Bus	Driving	Train
Sample size	Raw dataset	8260	4415	5923	3588	3769
	Generated dataset	1800	5600	4100	6500	6200
	Total dataset	10,060	10,015	10,023	10,088	9969
Average speed	Raw dataset	0.63	1.55	3.19	5.66	9.5
	Generated dataset	0.75	1.57	4.01	6.24	11.4
	Total dataset	0.66	1.56	3.53	6.04	10.7
Average acceleration or deceleration	Raw dataset	0.04	0.08	0.1	0.11	0.11
	Generated dataset	0.03	0.07	0.17	0.18	0.24
	Total dataset	0.04	0.08	0.13	0.16	0.19
Average bearing rate	Raw dataset	−0.01	−0.03	−0.01	−0.02	−0.02
	Generated dataset	−0.01	−0.02	−0.02	−0.04	−0.12
	Total dataset	−0.01	−0.03	−0.01	−0.03	−0.08
Average jerk	Raw dataset	1.76	0.04	0.82	2.35	1.31
	Generated dataset	1.65	0.02	0.76	2.17	1.45
	Total dataset	1.74	0.03	0.80	2.23	1.40

Table 2
Architecture of the GAN.

The generator model	Layer number	Layer	Kernel size	Stride	Activation function
	1	FC layer	–	–	ReLU
	2	Decon layer	10*4	2	ReLU
	3	Decon layer	10*4	2	ReLU
	4	Decon layer	10*4	2	ReLU
	5	Decon layer	10*4	2	ReLU
The discriminator model	1	Con layer	10*4	2	LReLU
	2	Con layer	10*4	2	LReLU
	3	Con layer	10*4	2	LReLU
	4	Con layer	10*4	2	LReLU
	5	FC layer	–	–	Sigmoid

were also presented. Furthermore, to compare samples of the raw dataset and samples of the generated dataset, we have calculated the correlations between variable of the generated dataset and corresponding variable of the raw dataset and the results are shown in following Table 3. It can be seen all correlations are more than 0.66 partly indicating there are few spurious examples.

The result indicates the proposed GAN can increase the sample size while keeping the characteristics of the variables. Moreover, it can be found the distributions of four variables in the raw dataset and the generated dataset have some differences. The difference suggests that the GAN can increase the diversity of the samples. As demonstrated in (Tang et al., 2006), diversity can improve the accuracy of the machine learning models.

5.2. A comparison of detection models using balanced dataset and raw dataset

In the second experiment (Fig. 4), the proposed detection model (CNN) was evaluated using the raw dataset and the balanced dataset. To ensure a fair and valid comparison, the same samples were selected to test the models. The remaining samples in the raw dataset and the balanced dataset are used to optimize the hyper-parameters of CNN. The CNN trained using the raw dataset was named as R-CNN while the CNN trained using the balanced dataset was named as B-CNN. The architectures of CNNs (Fig. 5) are developed according to the method in (Dabiri and Heaslip, 2018). Moreover, an ensemble B-CNN is also implemented to improve the accuracy and robustness of the detection. In ensemble B-CNN, ten CNNs with the architecture of B-CNN in Fig. 5 are trained but using training datasets randomly selected from the balanced dataset.

The confusion matrix of R-CNN and ensemble B-CNN is shown in Table 4. The elements without brackets are results of R-CNN while the elements with brackets are results of ensemble B-CNN. On the lead diagonal of Table 4, it can be seen the values with brackets are greater than the values without brackets which indicates ensemble B-CNN can correctly classify more samples than R-CNN. Using **Eq. (15)**, the accuracy of the two models can be calculated. The accuracy of B-CNN (86.70%) is higher than the accuracy of R-CNN (80.38%). In summary, ensemble B-CNN outperforms R-CNN for all five travel modes. The results indicate the balance of the dataset can improve the performance of the detection model. In particular, R-CNN can improve the accuracy of the bus (1058 > 1015) and the driving (595 > 544) which are difficult to classify because of their similar characteristics of the movement.

5.3. Comparison with some benchmark models

In the third experiment, some traditional machine learning models including artificial neural network (ANN), support vector machine (SVM), and random forest (RF) were also implemented. Ten-fold cross-validation and grid search methods are used to select the best hyper-parameters of benchmark models. When the number of hidden layers and the number of nodes in each layer are set as 1 and 6, the ANN model can reach the highest accuracy. For SVM, gamma and soft margin are set as 0.0125 and 4, respectively. The number of trees in the random forest is set as 90. To ensure a fair comparison, we used hand-crafted features introduced in previous studies (Zheng et al., 2010; Dabiri and Heaslip, 2018). The evaluation criteria of ANN, SVM, RF, and our proposed model are shown in Fig. 6. It can be seen the proposed model can obtain the highest accuracy and average F-measure. Furthermore, the proposed model can improve accuracy by 30.33%, 23.41%, and 10.84% compared with ANN, SVM, and RF. The proposed model can improve the average F-measure by 33.49%, 22.56%, and 11.05% compared with ANN, SVM, and RF. In summary, the proposed model outperforms the traditional machine learning models. It can infer the deep neural network (CNN) can effectively mine the hidden

Table 3
Correlation of variable of the generated dataset and corresponding variable of the raw dataset.

	Speed	Acceleration/Deceleration	Bearing rate	Jerk
Walk	0.81	0.72	0.80	0.96
Bike	0.78	0.78	0.74	0.72
Bus	0.93	0.76	0.94	0.66
Driving	0.95	0.70	0.70	0.84
Train	0.71	0.77	0.72	0.85

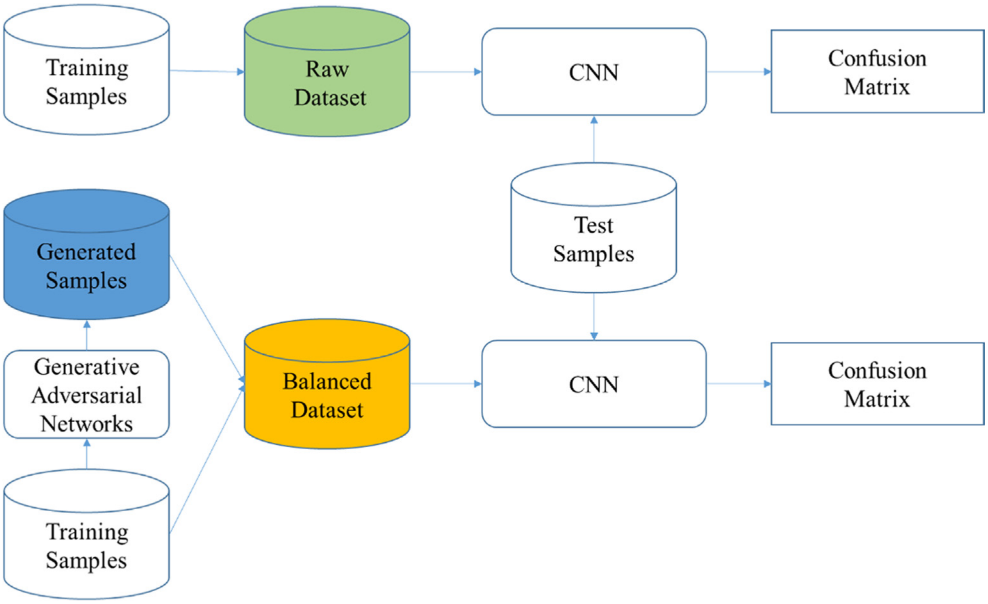

Fig. 4. Experiment 1: effective of generative adversarial networks.

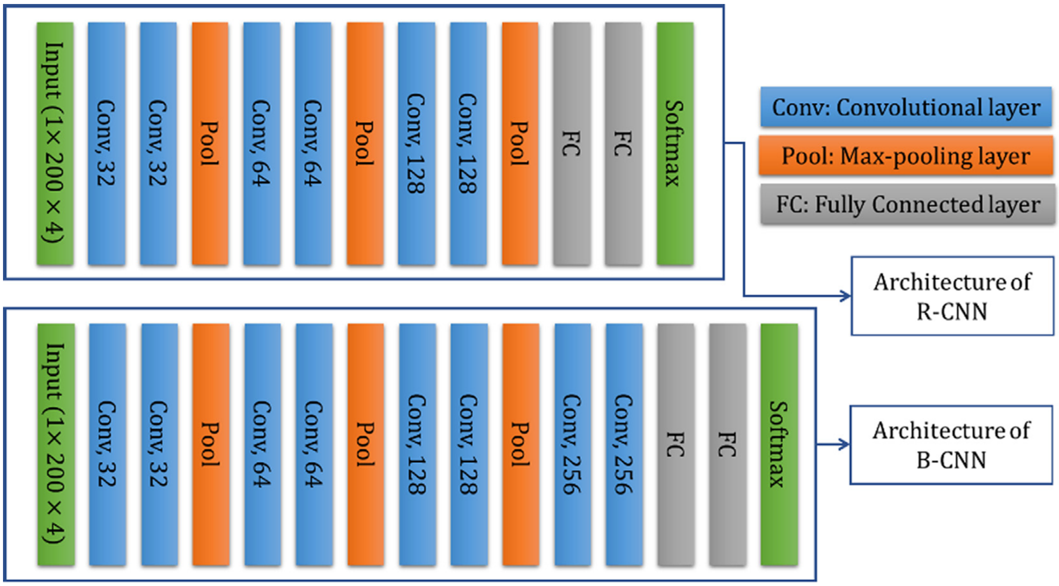

Fig. 5. Architecture of CNN.

Table 4
Confusion matrix for raw dataset and total dataset.

CNN with raw dataset (CNN with total dataset)		Predicted class				
		Walk	Bike	Bus	Driving	Train
Actual class	Walk	1980 (2185)	64 (22)	54 (22)	8 (3)	6 (5)
	Bike	213 (104)	894 (966)	38 (28)	5 (8)	3 (4)
	Bus	138 (54)	33 (19)	1015 (1058)	145 (138)	38 (26)
	Driving	75 (51)	33 (14)	206 (172)	544 (595)	44 (58)
	Train	85 (43)	21 (18)	21 (19)	43 (15)	783 (822)

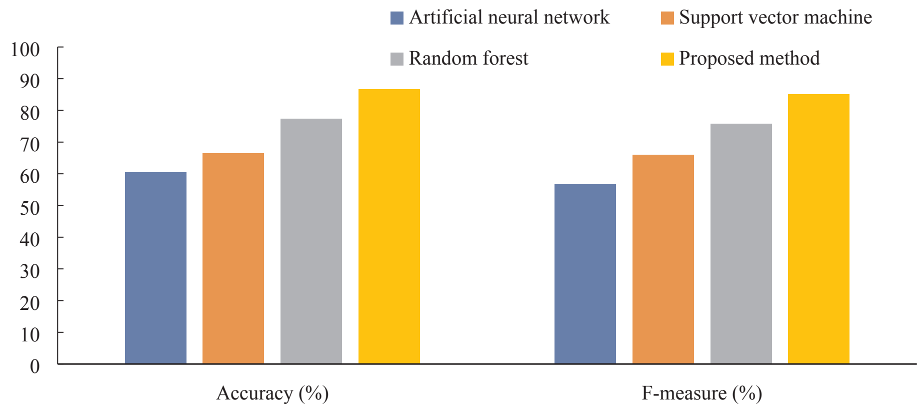

Fig. 6. Comparison with some traditional models.

relationships of the GPS trajectory data.

The results of the proposed model were also compared with some previous studies using the dataset of the GeoLife project. The first study we compared is (Zheng et al., 2008) in which the change point-based method combined a decision tree was implemented. The second study is (Dabiri and Heaslip, 2018) in which the CNN is applied. The third one is (Endo et al., 2016) in which a deep neural network was applied. The final one is (Zhu et al., 2016) in which the feature engineering method was applied. It can be found the proposed model can achieve the best performance from Fig. 7.

Moreover, to evaluate the effectiveness of the GAN, two methods, random undersampling and random oversampling, are also applied in this study. The mechanic of random undersampling follows naturally from its description by removing samples from the original dataset. In particular, we randomly select 4672, 827, 2355, and 181 samples from the walk, the bike, the bus, and the train class, respectively, and then remove these samples from the dataset. Consequently, a balanced dataset can be obtained. While random undersampling removes samples from the original dataset, random oversampling appends samples to the original dataset (He and Garcia, 2009). The accuracies of CNN models, trained by these two balanced datasets, are 80.35% and 80.94%. It can be found the accuracy of B-CNN is higher indicating the GAN can improve the performance of travel mode detection.

6. Conclusions

To reduce the cost of the travel survey, a method containing data preparation, sample generation, and mode detection is proposed in this paper to detect travel modes using trajectory data from smartphones. In the first step, the trajectory of each traveler is divided into some segments with an equal number of GPS points. For each segment, four widely used kinematic variables are calculated and constructed as the input of generative and detection models. Since a large number of samples and a balanced dataset are important to train machine learning models. Then, the GAN is implemented to augment the sample size and balance the dataset. Finally, CNN is configured and trained using the combination of raw samples and generated samples. Using the GeoLife dataset, the proposed model is compared with some traditional machine learning models, including SVM, ANN, and RF. It shows the proposed model can obtain the best results. Furthermore, the accuracy of the proposed model is higher than some reported accuracy in previous studies. It worth noticing that the augment of the sample size, the balance of the dataset, and the effective feature extraction can contribute to the travel mode detection. The GAN can reduce the number of labeled samples, and then decrease the cost and time of the survey.

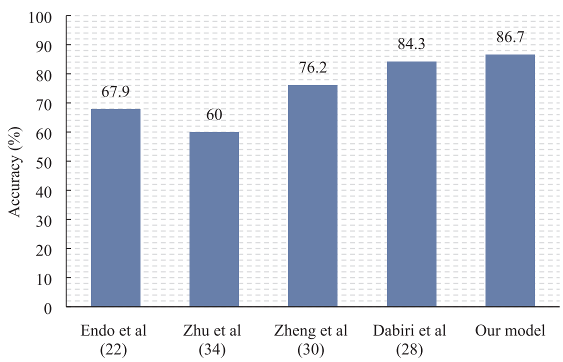

Fig. 7. Comparison with some related studies.

Acknowledgement

We acknowledge the the editors and anonymous reviewer for their valuable support and comments. The project presented in this article is supported by the National Natural Science Foundation of China (No. 51822802; 51778033; U1811463).

References

Bagdadi, O., Várhelyi, A., 2013. Development of a method for detecting Jerks in safety critical events. Accid. Anal. Prev. https://doi.org/10.1016/j.aap.2012.03.032.

Bolbol, A., Cheng, T., Tsapakis, I., Haworth, J., 2012. Inferring hybrid transportation modes from sparse GPS data using a moving window SVM Classification. Comput. Environ. Urban Syst. https://doi.org/10.1016/j.compenvurbsys.2012.06.001.

Byon, Y.J., Liang, S., 2014. Real-Time Transportation Mode Detection Using Smartphones and Artificial Neural Networks: Performance Comparisons between Smartphones and Conventional Global Positioning System Sensors. J. Intell. Transport. Syst.: Technol., Plann., Oper., 2014. https://doi.org/10.1080/15472450.2013.824762.

Dabiri, S., Heaslip, K., 2018. Inferring Transportation Modes from GPS Trajectories Using a Convolutional Neural Network. Transport. Res. Part C: Emerg. Technol. 86, 360–371. https://doi.org/10.1016/j.trc.2017.11.021.

Dabiri, S., Heaslip, K., 2019. Developing a Twitter-based traffic event detection model using deep learning architectures. Exp. Syst. Appl. 118, 425–439.

Dabiri, S., Lu, C.-T., Heaslip, K., Reddy, C.K., 2019. Semi-supervised deep learning approach for transportation mode identification using GPS trajectory data. IEEE Trans. Knowl. Data Eng. https://doi.org/10.1109/tkde.2019.2896985.

Endo, Y., Toda, H., Nishida, K., Kawanobe, A., 2016. Deep Feature Extraction from Trajectories for Transportation Mode Estimation. 2016.

Endo, Y., Toda, H., Nishida, K., Kawanobe, A., 2016. Deep Feature Extraction from Trajectories for Transportation Mode Estimation.

Gadziński, J., 2018. Perspectives of the Use of Smartphones in Travel Behaviour Studies: Findings from a Literature Review and a Pilot Study. Transport. Res. Part C: Emerg. Technol. https://doi.org/10.1016/j.trc.2018.01.011.

Gonzalez, P.A., Weinstein, J.S., Barbeau, S.J., Labrador, M.A., Winters, P.L., Georggi, N.L., Perez, R., 2010. Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks. IET Intel. Transport Syst. https://doi.org/10.1049/iet-its.2009.0029.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative Adversarial Nets.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G. Learning from class-imbalanced data: review of methods and applications. Exp. Syst. Appl.

He, H., Garcia, E.A., 2009. Learning from Imbalanced Data. IEEE Trans. Knowl. Data Eng. https://doi.org/10.1109/TKDE.2008.239.

Huang, H., Cheng, Y., Weibel, R., 2019. Transport mode detection based on mobile phone network data: a systematic review. Transport. Res. Part C: Emerg. Technol.

Jahangiri, A., Rakha, H.A., 2015. Applying machine learning techniques to transportation mode recognition using mobile phone sensor data. IEEE Trans. Intell. Transp. Syst. https://doi.org/10.1109/TITS.2015.2405759.

Lari, A., 2015. Automated Transportation Mode Detection Using Smart Phone Applications via Machine Learning: Case Study Mega City of Tehran. In: Transportation Research Board 94th Annual Meeting, 2015.

Li, L., He, S., Zhang, J., Ran, B., 2016. Short-term highway traffic flow prediction based on a hybrid strategy considering temporal-spatialInformation. J. Adv. Transport. 50 (8), pp. https://doi.org/10.1002/atr.1443.

Li, L., Zhang, J., Wang, Y., Ran, B., 2018. Missing value imputation for traffic-related time series data based on a multi-view learning method. IEEE Trans. Intell. Transp. Syst.

Li, L., Qin, L., Qu, X., Zhang, J., Wang, Y., Ran, B., 2019a. Day-optimized by the multi-objective particle swarm ahead traffic flow forecasting based on a deep belief network algorithm. Knowl.-Based Syst. 172, 1–14.

Li, L., Qu, X., Zhang, J., Wang, Y., Ran, B., 2019b. Traffic speed prediction for intelligent transportation system based on a deep feature fusion model. J. Intell. Transport. Syst. 1–12.

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., Wang, Y., 2017. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors (Switzerland). https://doi.org/10.3390/s17040818.

Nitsche, P., Widhalm, P., Breuss, S., Brändle, N., Maurer, P., 2014. Supporting large-scale travel surveys with smartphones - a practical approach. Transport. Res. Part C: Emerg. Technol. https://doi.org/10.1016/j.trc.2013.11.005.

Stenneth, L., Wolfson, O., Yu, P.S., Xu, B., 2011. Transportation Mode Detection Using Mobile Phones and GIS Information.

Stopher, P.R., Greaves, S.P., 2007. Household travel surveys: where are we going? Transport. Res. Part A: Policy Pract. 41 (5), 367–381.

Tang, E.K., Suganthan, P.N., Yao, X., 2006. An analysis of diversity measures. Mach. Learn. 65 (1), 247–271. https://doi.org/10.1007/s10994-006-9449-2.

Wu, L., Yang, B., Jing, P. Travel Mode Detection Based on GPS Raw Data Collected by Smartphones: A Systematic Review of the Existing Methodologies. Information (Switzerland).

Xiao, G., Juan, Z., Zhang, C., 2015. Travel Mode Detection Based on GPS Track Data and Bayesian Networks. Computers, Environment and Urban Systems, 2015. https://doi.org/10.1016/j.compenvurbsys.2015.05.005.

Yang, X., Stewart, K., Tang, L., Xie, Z., Li, Q. A Review of GPS Trajectories Classification Based on Transportation Mode. Sensors (Switzerland).

Yang, F., Yao, Z., Jin, P.J., 2015. GPS and Acceleration Data in Multimode Trip Data Recognition Based on Wavelet Transform Modulus Maximum Algorithm. Transport. Res. Rec.: J. Transport. Res. Board. https://doi.org/10.3141/2526-10.

Zhang, R., Xie, P., Wang, C., Liu, G., Wan, S., 2019. Classifying transportation mode and speed from trajectory data via deep multi-scale learning. Comput. Netw., 106861.

Zheng, Y., Liu, L., Wang, L., Xie, X., 2008. Learning Transportation Mode from Raw Gps Data for Geographic Applications on the Web.

Zheng, Y., Chen, Y., Li, Q., Xie, X., Ma, W.-Y., 2010. Understanding Transportation Modes Based on GPS Data for Web Applications. ACM Trans. Web. https://doi.org/10.1145/1658373.1658374.

Zheng, Y., Fu, H. Geolife GPS Trajectory Dataset - User Guide. Geolife GPS trajectories 1.1.

Zhong, G., Zhang, J., Li, L., Chen, X., Yang, F., Ran, B., 2018. Analyzing Passenger Travel Demand Related to the Transportation Hub inside a City Area Using Mobile Phone Data. Transp. Res. Rec. 2672 (50), 23–34.

Zhu, Q., Zhu, M. Li, M., Fu, M., Huang, Z., Gan, Q., Zhou, Z., 2016. Identifying Transportation Modes from Raw GPS Data. 2016.