


## ORIGINAL RESEARCH PAPER

# Trip destination prediction based on a deep integration network by fusing multiple features from taxi trajectories

Jinjun Tang<sup>1</sup>  | Jian Liang<sup>1</sup> | Tianjian Yu<sup>1</sup> | Yong Xiong<sup>2</sup> | Guoliang Zeng<sup>2</sup>

<sup>1</sup> Smart Transportation Key Laboratory of Hunan Province, School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China

<sup>2</sup> Department of Traffic and Transportation, Hunan Lianzhi Technology Co., Ltd, Changsha 410200, China

## Correspondence

Tianjian Yu, Smart Transportation Key Laboratory of Hunan Province, School of Traffic and Transportation Engineering, Central South University, Changsha, 410075, China  
Email: yutianjian@csu.edu.cn

## Funding information

Natural Science Foundation of Hunan Province, Grant/Award Number: 2020JJ4752; Innovation-Driven Project of Central South University, Grant/Award Number: 2020CX041; National Key R&D Program of China, Grant/Award Number: 2020YFB1600400; Foundation of Central South University, Grant/Award Number: 502045002

## Abstract

Trip destination prediction plays an important role in exploring urban travel patterns. Accurate prediction can improve the efficiency of traffic management and the quality of location-based services. Here, a deep learning structure that contains three components: travel information extraction, classification learning mechanism, and output module is proposed. Three types of information (the partial trajectory of on-going trips, historical trajectories, and related external information) are extracted in the first component. Then, the classification learning mechanism chooses different methods (i.e. Long Short-Term Memory network and Embedding technology) according to the characteristic of variables. Finally, an output layer that integrates the prior information about destinations is constructed. Two open-source trajectory datasets are used to validate the effectiveness of the proposed model. Results show that the proposed model outperforms benchmark models using only part of the information or using all of the information but ignore the classification learning mechanism. The performance of the proposed model under different call types and travel durations is further explored. The result of this study will help understand travel behaviour in urban cities.

## 1 | INTRODUCTION

With the concentration of populations in large cities, traffic demand has markedly increased in recent years. This rapid growth also places high pressure on urban transportation systems. The effective management of urban travel demand can optimize travel patterns, relieve traffic congestion, and reduce traffic pollution. To realize intelligently manage of travel demand, it is necessary to explore the regularity of citizens' daily travel behaviour based on large-scale trajectory data. With the widespread application of Global Positioning System-enabled in-vehicle systems and mobile devices, massive trajectory data collected from vehicles on road networks can be used to study urban travel mobility. The accumulation of trajectory data provides the opportunity for many possible applications that rely on vehicle location information, in which travel destination analysis and prediction are critical. The Enabling Advanced Traveler Information Services (Enable ATIS) project supported by the

U.S. Federal Highway Administration (FHWA) investigates travel destination prediction because the highly accurate prediction of travel destinations has several effects on transportation management and control, can help to rationally allocate road network resources and improve traffic control schemes. Accordingly, the Traffic Management Center (TMC) can provide reasonable route suggestions to travellers based on current traffic congestion on a road network. Additionally, the prediction of destinations will also benefit many location-based services, such as providing more personalized navigation services, recommending sightseeing places, and mobile advertisements.

Trip destination prediction aims to estimate the destination of the current trip based on the trajectory prefix and known information related to the current trip. For example, a child goes from home to the bus stop and then takes the bus to school. The trajectory from home to the bus stop is regarded as the trajectory prefix and the departure time, departure location, and the child's historical travel information are regarded as the known

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. *IET Intelligent Transport Systems* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

information related to the current trip. We regard the trajectory prefix as partial information and the known information as historical information and external information. Specifically, historical information describes the information of historical trajectories related to on-going trips, and external information includes the date and time, driver characteristics, and land-use information related to on-going trips.

People often travel regularly. For example, the child mentioned above may often leave home to school at the same time during the working day. Inspired by this phenomenon, probabilistic methods are widely used in destination prediction. The probabilistic method matches the current trip with historical trajectories based on the ID of the traveller, departure time, and departure location, and then determines the destination as the most likely location that is matched [1–5]. Because probabilistic methods are highly dependent on the existence of historical trajectories related to the on-going trip, this type of method does not work when the on-going trip cannot be matched to any historical trajectories. To overcome this problem, researchers apply deep learning methods to predict destinations based on the trajectory prefix of the on-going trip [6–9]. This type of method treats the trajectory as a time series, and predicts the destination based on the autocorrelation of the trajectory. Since it does not rely on historical trajectories, deep learning-based methods avoid the problem of data sparsity.

However, when predicting the destination, the existing deep learning methods are only based on partial information [7, 8] or partial information and external information [6, 9], and historical information is not fully utilized. Moreover, information processing should be further refined instead of simply splicing different features together and inputting them into the prediction model. This problem has not been noted in many existing deep learning-based studies in the field of destination prediction. For example, as a time series, trajectory data should be processed by a specific model (e.g., recurrent neural network) instead of being directly sent to a multi-layer perceptron [6]. It is also inappropriate to use a long short-term memory network, which is suitable for processing sequence data, to manage external information composed of discrete values [9].

In this paper, we propose a neural network-based model called Classification Learning Neural Network (CLNN) for destination prediction. We first extract the information related to the on-going trip from historical trajectories and use it as the input of CLNN together with trajectory prefix and external information. Then, a classification learning mechanism is designed to improve the model's ability to extract knowledge from the data. The classification learning mechanism chooses LSTM to model time series (i.e. trajectory prefix), combines embedding and MLP to model discrete variables (i.e. external information), and uses MLP to model continuous variables (i.e. historical information). Finally, a prediction module that incorporates prior information about the destination is developed to complete the prediction. The primary contributions of this study can be summarized as follows:

- We integrate three types of information (partial, historical, and external) to improve the performance and stability of destination prediction.
- Based on the format and characteristics of variables in the different types of information, a classification learning mechanism is designed to extract hidden features that affect travel behaviour.
- To explore the effect of the temporal distribution of trajectories on travel behaviour and destination prediction, trajectories are divided into three groups based on departure time.

The remainder of this paper is organized as follows. Section 2 summarizes the related work of destination prediction. In Section 3, we introduce the model framework and details of the proposed model. Section 4 presents an analysis of the data and the experimental results. The impact analysis of call type and travel duration on destination prediction is shown in Section 5. Conclusions and future research directions are then summarized in the last section.

## 2 | LITERATURE REVIEW

Prior studies of destination prediction were primarily based on trip matching, which requires the existence of historical trajectories related to the current trip. Among these studies, the Markov model [1–3] and the Bayesian model [4, 5] have been widely used. Specifically, Simmons et al. [10] combined GPS data and a map database to capture the driving habits of private drivers and then performed real-time prediction based on a Hidden Markov Model (HMM). Cho et al. [11] developed a location recognition and prediction system based on an HMM in the context of smartphones. All these studies require a map database to link location coordinates to the real road network. Alvarez-Garcia et al. [12] developed an HMM to use historical trajectories to predict future travel destinations without a street-map dataset. These methods are usually affected by the problem of data sparsity, which means that the trajectory to be predicted cannot be matched to any historical trajectory in some cases. To solve this problem, the Sub-Trajectory Synthesis (SubSyn) model is proposed to increase the number of query trajectories by nearly 10 times [13, 14]. Also, trajectory similarity analysis can be used to mitigate data sparsity issues [15–17]. Chen et al. [18] classified trajectories into different clusters and predicted trip destinations based on trajectory similarities in the same cluster. Besse et al. [19] clustered historical trajectories and then used the internal trajectories of these clusters to make a final prediction. Terada et al. [20] incorporated trip purpose estimation into destination prediction. Miyashita et al. [21] designed a map-matching algorithm for estimating destinations in an artificial navigation system. Even with various improvement, probabilistic methods still face the challenge of trajectories data sparsity when predicting trip destinations.

Another commonly used method is to predict the destination based on the trajectory prefix. Krumm et al. [22, 23] proposed a model called “predestination” to predict the next location of a trip using historical destinations and information on driving behaviour. Ziebart et al. [24] utilized a probabilistic reasoning method to predict destinations based on partial information about driving routes. External information, such as

the date and time of trajectories, can be used to improve the accuracy of destination prediction [25, 26]. In the 2015 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD) Challenge [27], Brébisson et al. [6] proposed a highly automated model based on multi-layer perceptrons that fused multi-source information related to trajectories, including partial trajectory data, spatiotemporal features of trips and driver characteristics to make a final prediction. Endo et al. [7] converted a map into a two-dimensional grid, and each location point in the trajectory was represented by a one-hot vector based on its position in the grid. Then, a Recurrent Neural Network (RNN) was used to predict destination. Lv et al. [8] converted trajectories into images, and then extracted vehicle movement features using Convolutional Neural Network (CNN) to predict travel destinations. Zhang et al. [28] introduced an image frequency domain processing method to reduce noise and sparsity to improve prediction. Because there are few meaningful pixels in the trajectory image, this type of study will yield many invalid calculations and high computational demands.

Current research has focused on improving prediction structures with advanced models and integrating different data sources related to travel. Krause et al. [29] combined the trip purpose with the hierarchical Markov model to predict the destination. Rossi et al. [9] combined the Long Short-Term Memory network (LSTM) with attention mechanisms to predict travel destinations. Zhang et al. [30] introduced a Circular Fuzzy Embedding (CFE) technique in the feature representation of trips, and then applied an ensemble learning approach based on Support Vector Regression (SVR) and Deep Belief Network (DBN) to finish destination prediction. However, few studies have fully explored the useful information that is hidden in different data sources. In this paper, we propose a Classification Learning Neural Network to fully utilize the potential of knowledge hidden in the raw data.

### 3 | METHODOLOGY

#### 3.1 | Framework

As mentioned earlier, all three types of information are essential to destination prediction. Hence, we propose a model called CLNN to fully extract and integrate all three types of information based on the format and characteristics (e.g. discrete or continuous, time series or non-time series) of information itself. Figure 1 shows the network architecture of the proposed CLNN, which consists of three parts. The first part introduces how to extract three types of information from the original trajectory data, including historical information searching, external information integration, and partial information obtaining. Then, the second part proposes a classification learning mechanism, which uses different technologies (e.g. LSTM, Embedding, and MLP) to deal with different information in a targeted manner. Finally, the third part fuses the prediction results of different modules into the model output.

#### 3.2 | Feature extraction

##### 3.2.1 | History information searching

In this section, we present an algorithm to extract related historical information. Most drivers have a fixed driving route. For example, commuting between their workplace and home or delivering children between school and home in the morning or evening. Therefore, capturing these similarities among large historical trajectories can improve prediction accuracy. We design a historical information retrieval algorithm to quickly search for historical information related to on-going trips. This algorithm searches for historical trajectories that are associated with the current trip based on fixed factors. These fixed factors include driver ID, call type, departure area, and departure time, which are denoted as  $f_1$ ,  $f_2$ ,  $f_3$ , and  $f_4$ , respectively. Because the similarity between destinations is primarily based on the driver's own travel habits, driver ID (i.e.  $f_1$ ) will be given a higher priority. The specific steps for the historical information retrieval algorithm are described as follows:

##### 3.2.2 | External information integration

External information includes temporal information, driver characteristics, and POIs. Specifically, temporal information consists of day of week, week of year, and quarter-hour of day. The driver characteristics consist of taxi ID and call type. These five features are represented by discrete values, and the POI information is represented by a 10D vector.

Traditional method uses one-hot vectors to represent discrete values. It usually encounters two problems: many redundant sparse matrices will be generated, and the relationship between the discrete values cannot be estimated. To overcome these limitations, we adopt an Embedding technique to encode discrete values [31]. Each discrete value is considered to be an input word and then another value from the nearby is selected as the output word. The input-output word pairs are then used as data to train a network. The weights of the hidden layer in the trained network are the dense representation of the corresponding discrete values. The embedding technique transforms discrete values into vectors of fixed size, while maintaining the relationship between discrete values in the vector space. For day of week, week of year, quarter-hour of day, taxi ID, and call type, we send each one into an embedding layer to generate a 10D vector. These vectors are fused with a 10D vector representing POI information, and fed to a Dense layer to obtain a deep representation of the external information.

##### 3.2.3 | Partial information obtaining

Partial information is obtained by randomly cutting the original complete trajectory. We use the last location in a complete trajectory as the trip destination, and then randomly cut the

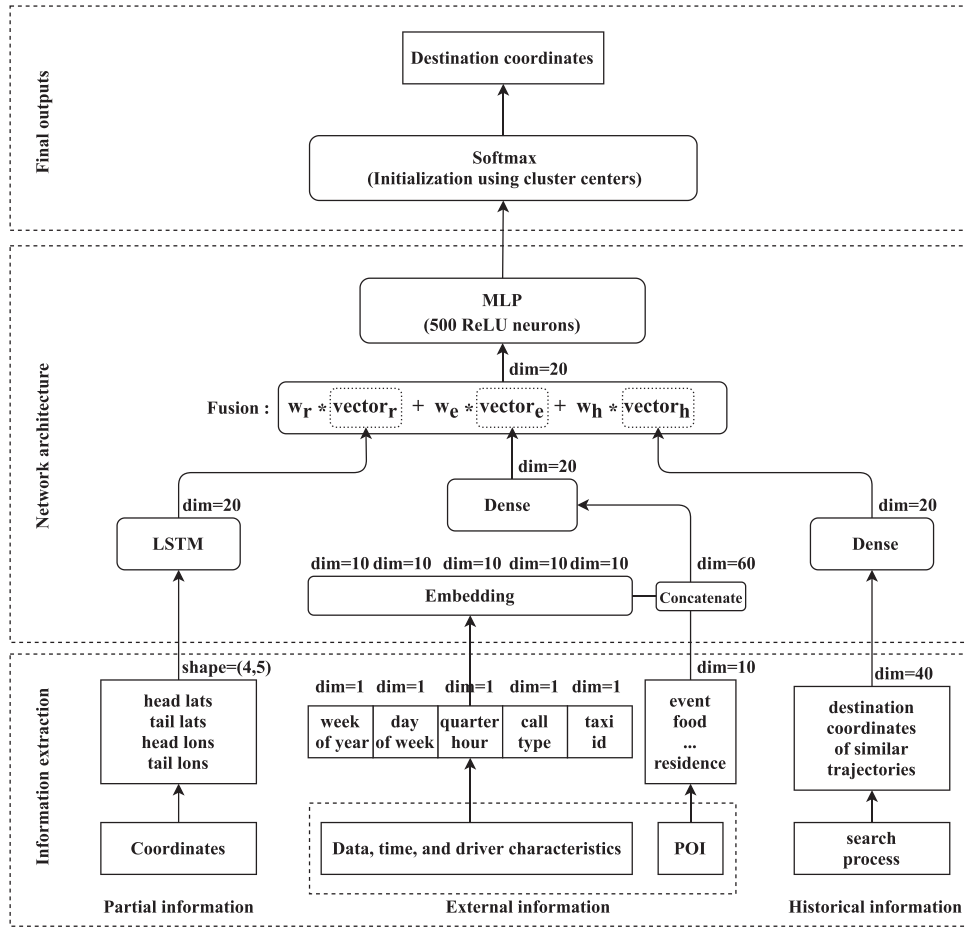


FIGURE 1 The network architecture of CLNN

trajectory without destinations to generate trajectory prefix. The deep learning network requires a fixed-length input, while the length of the trajectory prefix is inconsistent due to the random cutting process. It is necessary to further process the trajectory prefix to make the length of partial information consistent. Similar to [6], the first  $L$  GPS location points and the last  $L$  points of the partial trajectory are selected to form partial information, which consists of four parts: the longitudes of first and last  $L$  locations, and the latitudes of the first and last  $L$  locations. We set the length of the trajectory prefix in the cut process to be longer than  $L$ , and then stack the four parts into a matrix with a size of  $4 \times L$ . Because the trajectory prefixes are available after drivers begin their trips, they are considered partial information.

### 3.3 | Deep learning network structure

#### 3.3.1 | MLP

Multi-Layer Perceptron (MLP) [32] is a feed-forward neural network, which consists of an input layer, several hidden layers, and an output layer. In MLP, each neuron in each layer is connected to all neurons in the next layer, while the neurons

in the same layer are not connected. The MLP accepts fixed size vectors as input and processes them through one or several hidden layers, which compute a higher-level representation of the input. Finally, one or more neurons in the output layer return the predicted results. For a single neuron, MLP weights and sums the output value of all neurons connected to it in the previous layer, and obtain its output through an activation function.

$$y^l = \sigma \left( \sum_{i=1}^n w_i y_i^{l-1} + b \right) \quad (1)$$

where  $\sigma$  is the activation function,  $w$  is the weight,  $b$  is the bias,  $y_i^{l-1}$  is the output of the  $i_{th}$  neuron in the  $l-1$  layer, and  $n$  denotes the number of neurons in the current layer.

In MLP, weights determine the final output. Weights are first manually assigned an initial value, and then are continuously updated by calculating the gradients of the loss function. Finally, the optimal weights can be obtained by minimizing the difference between the predicted and observed values. Specifically, when the MLP contains only one network layer, it can be defined as Dense.



### 3.3.2 | LSTM

Unlike a simple feed-forward neural network, the neurons in the same hidden layer of the RNN are interconnected. A single neuron thus receives the output of the neurons in the previous layer and the output of the neurons in the same layer. Due to gradient disappearance and explosion, ordinary RNNs have difficulty in capturing long-term dependence. Accordingly, LSTM was designed to solve this long-term dependence issue [33]. By improving the normal RNN, the memory cell is added to each neuron in the hidden layer to provide a path for the gradient to maintain long-term flow. In LSTM, the input gate, forgetting gate, and output gate are designed to make valid information last longer. The calculation process is shown as follows:

$$f_t = \sigma_g (W_f x_t + U_f b_{t-1} + b_f) \quad (2)$$

$$i_t = \sigma_g (W_i x_t + U_i b_{t-1} + b_i) \quad (3)$$

$$o_t = \sigma_g (W_o x_t + U_o b_{t-1} + b_o) \quad (4)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c (W_c x_t + U_c b_{t-1} + b_c) \quad (5)$$

$$b_t = o_t \circ \sigma_b (c_t) \quad (6)$$

where  $x_t$  represents the input vector to the LSTM unit,  $b_t$  indicates the hidden state and will be sent to the next neuron in the same layer,  $b_{t-1}$  indicates the hidden state passed from the previous layer,  $o_t$  is the output gate's activation vector;  $c_t$  is the cell state vector,  $f_t$  and  $i_t$  are variables that appear in the calculation process,  $\sigma_g$  represents the sigmoid function,  $\sigma_c$  and  $\sigma_b$  represent the hyperbolic tangent functions,  $W$  and  $U$  are weight matrices, and  $b$  is bias matrix, and the operator  $\circ$  denotes the Hadamard product.

### 3.3.3 | Fusion

In this experiment, inputs for destination prediction include partial information, historical information, and external information. Referring to the setting of [6], when performing random cutting, the hyperparameter  $L$  used to control the length of the partial information is set to 5. Therefore, the dimension of  $vector_R$  used to represent partial information is 20 ( $5 \times 4$ ). To be consistent with the partial information, the value of  $K$  used to control the length of historical information is set to 10, resulting in a 20-dimensional  $vector_H$  ( $10 \times 2$ ). The construction of the  $vector_E$  used to represent external information requires two steps. The discrete values are first sent to the embedding layer to generate the embedded vector, and then the generated embedded vector is sent to the Dense layer together with the POI information. Referring to the setting of embedding in [9], the dimension of the embedding vector is set to ten. Then, to main-

tain the dimensions of different information consistent, the hidden unit number in the Dense layer is set to 20.

Through the above processes, we obtain three vectors of the same size, including  $vector_R$ ,  $vector_H$ , and  $vector_E$ . These vectors are deep representations of the partial information, historical information, and external information. Finally, three vectors are fused by a simple weighted summation.  $W_R$ ,  $W_H$ , and  $W_E$  represent the weights for three vectors, which are controlled by the contributions of partial information, historical information, and external information to the results, respectively. Their optimal values will be determined in the model training process described in the following section.

### 3.4 | Model output

To improve prediction performance, prior information about the destination is integrated into the proposed model. First, cluster the destinations of all training trajectories based on the mean-shift algorithm [34] to obtain  $C$  cluster centres. Then, a linear layer with softmax as the activation function is adopted to introduce the cluster centres into the destination prediction. The number of neurons in the linear layer is set to  $C$ , and the coordinates of the cluster centres are used to initialize the weights of the linear layer. The calculation process is shown as follows:

$$P_i = \frac{\exp(e_i)}{\sum_{i=1}^C \exp(e_i)} \quad (7)$$

$$\tilde{y} = \sum_{i=1}^C P_i \times c_i \quad (8)$$

where  $P_i$  represents the output of the softmax function,  $c_i$  represents the coordinates of the cluster center,  $e_i$  is the output of a single neuron in the previous network layer, and  $\tilde{y}$  are the final predicted latitude and longitude coordinates.

## 4 | DATA SOURCE DESCRIPTION AND RESULTS ANALYSIS

### 4.1 | Data source description

Taxi trajectory data collected in Porto City, Portugal [27, 28] were used to verify the proposed method destination in this study. It contains nearly 1.7 million records from 2013-07-01 to 2014-06-30 (a total 12 months) from 442 taxi vehicles, as shown in Table 1. Each record is composed of a complete taxi trajectory and associated external information. The trajectory contains a list of GPS locations every 15 s and the last location represents the trip destinations. External information includes the taxi ID, the departure time of the trip, and the call type. The call type identifies the way passengers called taxi services. If the trip was dispatched from the central, the call type is denoted by

**TABLE 1** The historical information retrieval algorithm**Algorithm Historical information retrieval**

**Input:** The trajectory  $T$  to be queried, The historical trajectory database  $D$ . The length of historical information  $K$ .

**Output:** The historical information  $H$  related to  $T$ .

1.  $k = 0, H = []$ ;
2. Extract the four factors corresponding to  $T$ , denoted as  $f^T = [f_1^T, f_2^T, f_3^T, f_4^T]$ ;
3. Extract the four factors corresponding to  $D$ , denoted as  $F = [F_1, F_2, F_3, F_4]$ .  $F^i = [F_1^i, F_2^i, F_3^i, F_4^i]$  refers to the  $i$ -th trajectory in  $D$ ;
4. Condition library =  $[\xi_1, \xi_2, \xi_3, \xi_4, \xi_5]$ . Among it,  $\xi_1$  is  $F^i = f^T$ ,  $\xi_2$  is  $[F_a^i, F_b^i, F_c^i] = [f_a^T, f_b^T, f_c^T]$ ,  $(a, b, c) = (1, 2, 3)$  or  $(1, 2, 4)$  or  $(1, 3, 4)$ ,  $\xi_3$  is  $[F_a^i, F_b^i] = [f_a^T, f_b^T]$ ,  $(a, b) = (1, 2)$  or  $(1, 3)$  or  $(1, 4)$ ,  $\xi_4$  is  $[F_a^i, F_b^i, F_c^i] = [f_a^T, f_b^T, f_c^T]$ ,  $(a, b, c) = (2, 3, 4)$ ,  $\xi_5$  is  $[F_a^i, F_b^i] = [f_a^T, f_b^T]$ ,  $(a, b) = (3, 4)$ ;
5. **For** condition in condition library, **do**;
6. Find all indexes satisfying condition in  $F$ ;
7. Stack the destination of the historical trajectories corresponding to these indexes into  $H$ ;
8. Set  $k$  as the length of  $H$ ;
9. **IF**  $k > K$ :
10. **Break For**
11. **Return**  $H$ .

'A'. If a passenger calls taxi services at a specific stand, the call type is denoted by 'B'. The call type is denoted by 'C' when a trip starts randomly on a street.

We integrate Point of Interest (POI) data into the proposed deep learning model. POI is an entity of interest with a defined location. Typical POIs are restaurants, hotels, tourist sites etc. POI data in this study were obtained from the Foursquare. Foursquare provides ten different macro categories to describe location and activity, including Arts and Entertainment, College and University, Event, Food, Nightlife Spot, Outdoors and Recreation, Professional and Other Places, Residence, Shop and Service, Travel and Transport. In this study, we divided Porto City into many grids of the same size, and then counted the number of different categories of POIs in each grid square. Thus, we obtain a 10D vector for each grid, and the specific value in the vector represents the number of entities belonging to each category. Based on the corresponding relationship between the origin of the trajectory and the grid, we assign a POI vector to each trajectory to describe the level of facilities around the trajectory pick up point. Details of the trajectory data are shown in Table 2.

**TABLE 2** Attribute structure table of taxi trajectory data in Porto

Trip ID	Call type	Taxi ID	Departure time	Trajectory	POI
1376853078620000367	C	20000367	2013/08/18 19:11:18	[(-8.600,41.1823), ...]	[0,0,0,0,0,1,0,0,2]
1382825517620000173	A	20000173	2013/10/26 22:11:57	[(-8.583,41.181), ...]	[0,0,0,1,0,0,0,0,0]
1400715767620000066	C	20000066	2014/05/21 22:11:57	[(-8.612,41.167), ...]	[0,0,0,0,1,0,0,0,0]
...	...	...	...	...	...

In addition, the public dataset Geolife is used to verify the effectiveness of the proposed model in different cities. Geolife records 18,670 trajectories by 182 users over 5 years. This dataset involves various outdoor activities of users, including daily routines, diversions, and sports.

## 4.2 | Evaluation indicator

The evaluation indicator helps compare the performance of different methods. When choosing an indicator, we should consider the data used in the study and the requirements of the primary task. The evaluation indicator used in this study is the Mean Haversine Distance (MHD) error, which is widely used to evaluate the accuracy of destination prediction [29–32]. The Haversine distance measures the distance between two points on a sphere based on latitude and longitude, which is appropriate to calculate the distance between the predicted and actual destinations. The Haversine distance (km) between the two positions can be calculated as follows:

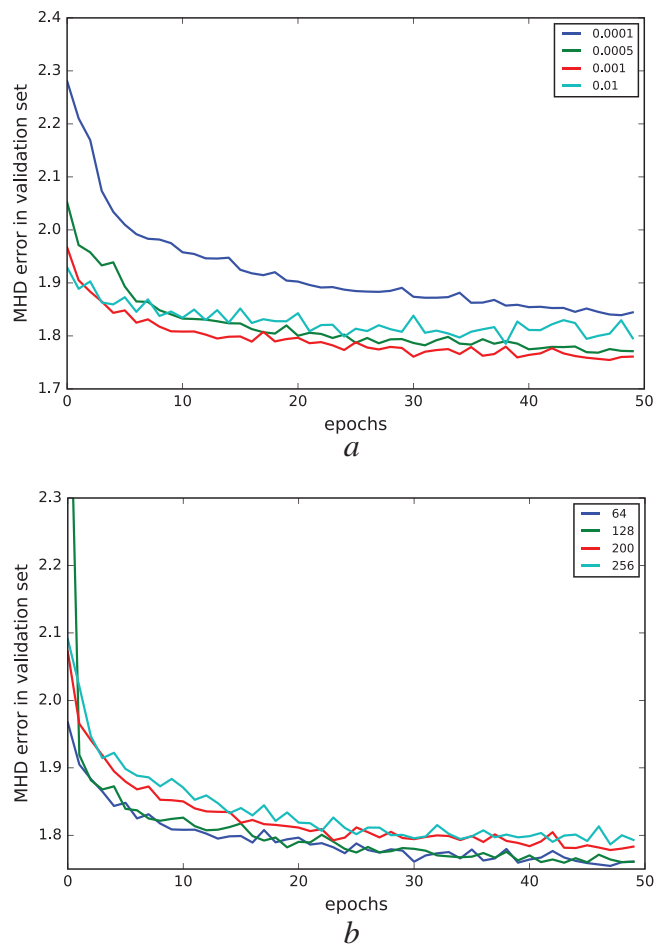
$$a = \sin^2 \left( \frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right) \quad (9)$$

$$d_{\text{haversine}} = 2 \times r \times \arctan \left( \sqrt{\frac{a}{1-a}} \right) \quad (10)$$

where  $\phi$  is the latitude,  $\lambda$  is the longitude and  $r$  is the radius of the Earth.

## 4.3 | Optimization of hyperparameters

The optimization of hyperparameters is an important part of model training and also has a significant effect on model performance. The hyperparameters that must be determined in this study include the number of epochs, the learning rate, and the batch size. To accurately estimate the generalization error of the model, the original dataset is divided into four parts: history library, training set, validation set, and test set. The method of dataset division is to randomly extract 10% of the trajectories from the original dataset as a training set, 1% of the trajectories as a validation set, 1% of the trajectories as a test set. The remaining trajectories are used as a history library for searching historical information. A set of hyperparameters with the



**FIGURE 2** The variation of the MHD error on the validation set with epochs under different learning rate and batch size (a) learning rate, (b) batch size

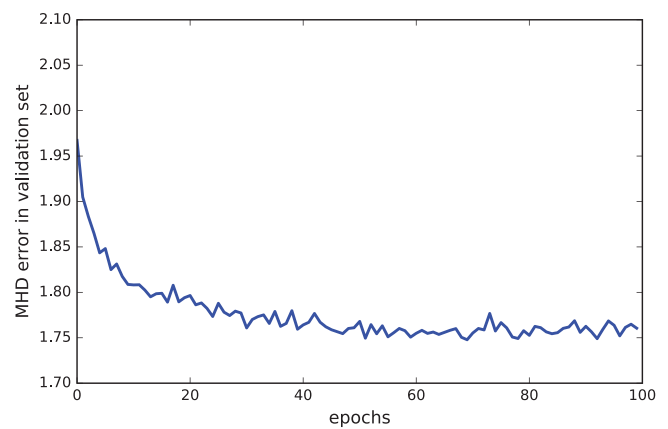
smallest error on the validation set is selected to determine the final model. These final models will be compared under the same test set. The optimal learning rate and batch size are determined based on the error on the validation set.

Figure 2 shows the variation of the MHD error on the validation set with epochs under the different learning rates and batch sizes. The learning rate and batch size have a significant impact on the model performance. When the learning rate is 0.001, the MHD error on the validation set is the lowest. The model with a batch size of 64 yields the smallest MHD error.

For the third parameter, the optimal number of epochs is selected when the MHD error value becomes stable. As shown from Figure 3, the MHD error gradually becomes small when the number of epochs reaches 40. Thus, the optimal number of epochs is set as 40.

#### 4.4 | Baselines and time division

In this study, the following models are selected as benchmark methods for comparison:



**FIGURE 3** The MHD error varies with different epochs on the validation set

**MLP:** The winning approach of the ECML/PKDD challenge in 2015 [29]. It consists of a hidden layer with 500 ReLU neurons and an output layer that is initialized with cluster coordinates. The network is trained with a stochastic gradient descent algorithm, a batch size of 200, a learning rate of 0.01, and a momentum term of 0.9.

**LSTM:** LSTM is an improved recurrent neural network that is widely used in speech recognition and machine translation. It is also widely used in traffic flow prediction and traffic speed prediction [35, 36].

**Lasso:** Lasso regression uses the L1 regularization term based on ordinary linear regression, is a simple and practical regression model, and is typically used as a baseline in traffic prediction tasks [37, 38].

**Random Forest:** Random Forest (RF) is an ensemble learning method based on a decision tree. Random forest has strong generalization ability and is suitable for processing high-dimensional data. It is widely used in many traffic prediction tasks, such as traffic flow prediction [39] and crash duration prediction [40].

**Bayesian regression:** Bayesian regression is a traditional method in the field of destination prediction [4, 5]. It can introduce prior information and display uncertainty in the estimation procedure.

**T-CONV:** T-CONV is a state-of-the-art approach in the field of destination prediction. T-CONV treats trajectories as images and uses a convolutional neural network to achieve precise prediction [8].

**SVR:** Support Vector Regression is a classic machine learning model that has been successfully applied in various traffic prediction tasks, e.g., traffic flow prediction [41–43].

**CLNN<sup>no-partial</sup>, CLNN<sup>no-external</sup>, and CLNN<sup>no-historical</sup>:** Those are variants of the proposed CLNN, which correspond to the models with partial information, external information, and historical information missing.

**TABLE 3** Division of time period in a day

Time period	Peak hours	Off-peak hours	Evening hours
Hours	8:00–9:00 17:00–19:00	6:00–8:00 9:00–17:00 19:00–21:00	21:00–6:00

For a fair comparison, all models are trained on the same training set, hyperparameters are determined on the same validation set, and prediction errors are estimated on the same test set. In Bayesian regression, only historical information is used as input. The input variables of the MLP do not include historical information and POI information, but it uses a deep information extraction mechanism. The LASSO, RF, and LSTM use all three types of information as inputs, but do not apply a classification learning mechanism. The traffic states exhibit fluctuations and variations at different times of day. To further consider the influence of temporal factors on travel behaviour and destination prediction, we divide the original dataset into three sub-datasets based on the different travel behaviours at peak hours, off-peak hours, and evening hours, as shown in Table 3.

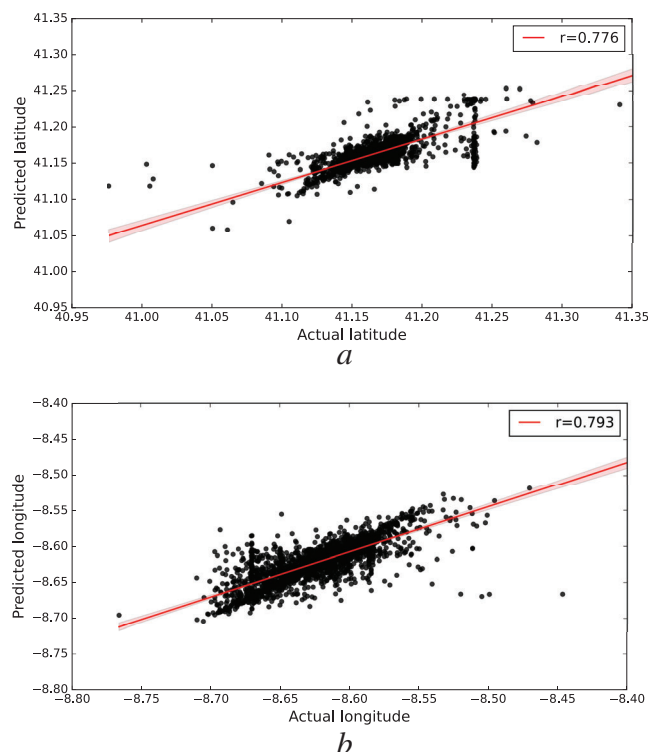
#### 4.5 | Experimental constraints

This section summarizes some experimental constraints of this paper, as follows:

- When searching for historical information, we assume that the trajectory in the dataset covers all available history records of travellers.
- The main purpose of this paper is to predict the destination of private users to benefit some location-based applications, but it is difficult to obtain the trajectory of large-scale private cars. Two data sets are used, one consists of small-scale individual user trajectories, and the other is composed of large-scale taxi trajectories. We ignore the impact of taxis as a means of public transportation.
- We calculate the POI distribution in each grid based on grid division, and match the departure location of the trajectory with the grid to describe the level of facilities around the trajectory pick-up point. Here, we equate the grid with the vicinity of the departure location.

#### 4.6 | Results comparison

Table 4 shows the MHD error of the models participating in comparison on the test set. The proposed CLNN outperforms the other models for different time periods. More specifically, the MHD error of CLNN is lower than that of Lasso, LSTM, SVR, and RF, which indicates that the classification extraction mechanism is effective. The MHD error of MLP and Bayesian regression is higher than that of CLNN, which shows that the fusion of the three types of information can improve predic-

**FIGURE 4** Comparison between prediction values and ground truth. (a) Latitude, (b) longitude

tion accuracy. Also, the MHD error of LSTM is higher than that of the traditional models, such as Random Forest and SVR, which shows that it is unbecoming to simply mixing all types of information and employ an inappropriate model to process. T-CONV considers the spatial correlations between trajectories, and outperforms the traditional machine models. However, due to the lack of historical information and external information, the MHD error of T-CONV is higher than that of CLNN. By comparing CLNN and its variants, we find that fusion of the three types of information can help improve the accuracy of destination prediction, and partial information is the most helpful. The MLP considers only two types of information as inputs but produces a smaller MHD error than the RF which considers all three types of information. This result indicates that extracting multiple features with reasonable methods could effectively improve prediction performance compared to simply adding new features in the input layer.

By comparing the MHD errors of all models at different time periods, we find that the prediction performance of all models in the evening is inferior to the other two time periods. The reason for this result may be that taxi drivers usually have a fixed travel pattern during the daytime, but tend to wander around the city at night to find passengers. Searching for customers may increase the randomness of driving behaviour and affect prediction performance in the evening. Compared to the other models, CLNN achieves better prediction performance by properly fusing three types of information in the input layer.

In addition, Figure 4 shows the correlation of predicted and observed destinations using CLNN under longitude and latitude



**TABLE 4** Comparison of MHD errors under different time periods for different models (km)

Time period	Porto dataset				Geolife dataset			
	Peak hours	Off-peak hours	Evening hours	All-time periods	Peak hours	Off-peak hours	Evening hours	All-time periods
LASSO	2.0735	2.0794	2.3086	2.1402	6.4488	5.3903	4.2966	5.2186
MLP	1.7761	1.7929	1.9258	1.8141	5.3697	4.6068	3.0902	4.2899
RF	1.9931	1.9385	2.1302	1.9994	5.5716	4.7507	3.8625	4.6069
LSTM	2.9404	3.0126	3.3638	3.1017	7.6101	6.2634	4.1068	5.8421
Bayesian Regression	3.3924	3.4211	3.7934	3.5060	8.3905	6.6675	3.5967	6.0459
SVR	2.0126	2.0347	2.1869	2.0763	6.7009	5.2878	3.5050	4.9731
T-CONV	1.8366	1.8598	1.9914	1.8394	5.3250	4.5228	3.2740	4.2814
CLNN <sup>no-partial</sup>	3.3251	3.3720	3.7472	3.4704	5.6712	4.8408	3.1060	4.4736
CLNN <sup>no-external</sup>	1.6864	1.7211	1.8505	1.7525	5.4639	4.6636	2.9401	4.2961
CLNN <sup>no-historical</sup>	1.7373	1.7704	1.8776	1.7958	5.5071	4.7190	3.0275	4.3286
CLNN	1.7104	1.7234	1.9307	1.7486	4.9166	4.2808	2.9544	4.0001

dimensions. A total of 3000 samples in the test set are shown in a scatterplot. For most samples, the proposed CLNN model is shown to accurately track variations in the ground truth.

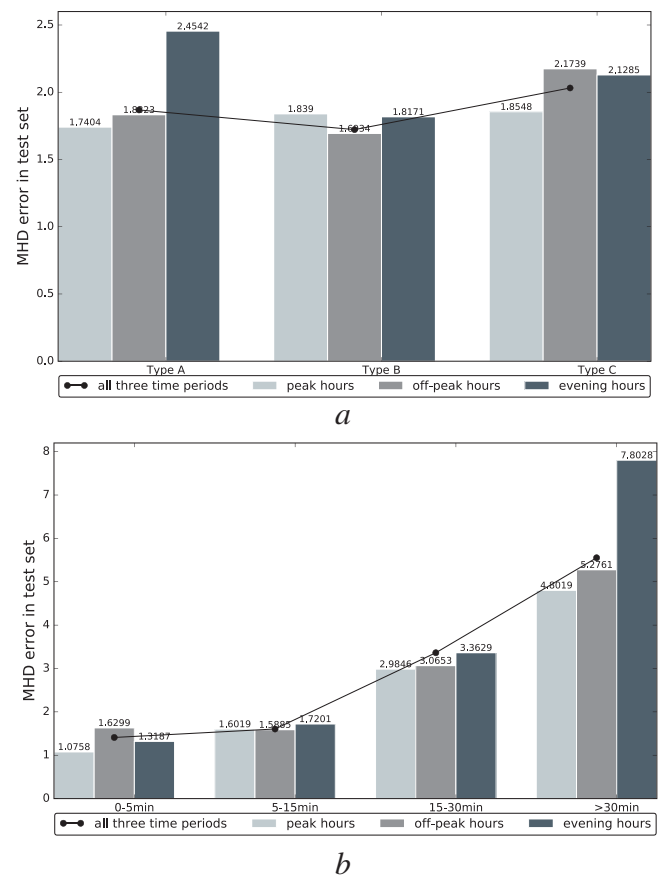
## 5 | RESULTS DISCUSSION

This section explores the effect of two key variables on the prediction results: the call type and the travel duration. These two variables can describe travel behaviour from two perspectives: before and after trips. The reason why these two factors are investigated is that they have strong explanatory meanings and have significant effects on travel behaviour.

### 5.1 | Call type

There are three ways for passengers to request taxi services: from the calling centre (Type A), on a specific stand (Type B), and randomly on a street (Type C). To describe the impact of different call types on destination prediction, we first divide the test set into three classes based on the call type, and then continue to subdivide the dataset based on the time period, yielding nine sub-datasets. The trained CLNN model is then used to calculate the MHD errors of the nine sub-datasets.

The results are shown in Figure 5(a). Because taxis can randomly pick up customers on the road, the randomness of the pick-up point of the trajectory increases, which affects the prediction accuracy. Thus, the destination of the trajectory identified as type C is much more difficult to predict than the other two call types. For type B, there are only 64 taxi stands in Porto City, and the randomness of the starting point of travel is relatively low. Thus, the accuracy of destination prediction by calling the taxi on a specific stand (type B) is the highest among the three call types. The prediction accuracy of the trip destination identified as type A is lower than that in type B. Customers who



**FIGURE 5** The MHD error for different call type and duration on the test set (a) call type, (b) duration

book taxis from the central system typically have relatively stable travel habits and behaviours.

Also, the destination prediction error for type A in the evening hours is much higher than that in peak and off-peak

hours. This may be caused by the small number of samples at night and insufficient samples for model training.

## 5.2 | Travel duration

Travel duration can be divided into four categories: 0–5 min, 5–15 min, 15–30 min, and more than 30 min. The test set is thus divided into 12 sub-datasets. Figure 5(b) shows the MHD error of models using different sub-datasets. As the travel duration increases, the prediction error grows gradually. When the travel duration is more than 15 min, the error increases exponentially. When the duration is less than 15 min, the model can produce high prediction accuracy.

Moreover, compared with other durations, the prediction error in the evening is much higher for a trip that lasts longer than 30 min. Since the number of trips lasting more than 30 min in the evening is very small, and the samples used for training is insufficient, resulting in this result.

## 6 | CONCLUSION

In this paper, we developed a deep learning framework for destination prediction by fusing different types of information. Two public trajectory datasets were used to verify the proposed model, and MHD was used as an indicator to evaluate prediction performance in different time periods. In model comparisons, benchmark methods were divided into four categories: Machine learning models with all information simply combined as input variables, MLP with an appropriate information extraction mechanism using only some information, traditional Bayesian regression that used only historical information, and variants of the proposed model. The contributions of this research can be summarized as follows. First, the proposed CLNN achieves the lowest MHD error over different time periods, demonstrating the effectiveness of the classification learning mechanism and the fusion of three types of information. Second, some internal factors affecting the performance of destination prediction were revealed. For example, for the trips with a short duration from the central or at a specific stand, the proposed model yielded higher predictive accuracy. In contrast, trips that occur at night and last longer than 30 min were difficult to predict.

This study has some limitations that required further investigation. First, different cities have different road network structures, traffic policies, and management measures. Different modes of transportation also have different traffic patterns. In the future, fusing different data sources should be considered to determine the impact of different transportation modes, road network structure, and transportation policies on travel destination prediction, so as to provide a more effective strategy for traffic management and control. Second, for trajectories that last longer than 15 min and trajectories generated from taxis that wander around the city, prediction error was relatively high. Future research should focus on improving the prediction performance for trips with long travel durations and random origins.

## ACKNOWLEDGEMENT

This research was funded in part by the Natural Science Foundation of Hunan Province (No. 2020JJ4752), Innovation-Driven Project of Central South University (No. 2020CX041), National Key R&D Program of China (No. 2020YFB1600400), Foundation of Central South University (No. 502045002).

## ORCID

Jinjun Tang  <https://orcid.org/0000-0002-5172-387X>

## REFERENCES

- Ashbrook, D., Starner, T.: Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5), 275–286 (2003)
- Dantas Nobre Neto, F., de S. Baptista, C., Campelo, C.E.C.: Combining markov model and prediction by partial matching compression technique for route and destination prediction. *Knowledge-Based Systems*, 154, 81–92 (2018)
- Tang, J., et al.: Estimating hotspots using a Gaussian mixture model from large-scale taxi GPS trace data. *Transportation Safety and Environment*, 1(2), 145–153 (2019)
- Patterson, D.J., et al.: Inferring high-level behavior from low-level sensors. In: *UbiComp 2003: Ubiquitous Computing*, A. K. Dey, A. Schmidt, J. F. McCarthy, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 73–89 (2003)
- Liao, L., et al.: Learning and inferring transportation routines. *Artif. Intell.*, 171(5–6), 311–331 (2007)
- De Brébisson, A., et al.: Artificial neural networks applied to taxi destination prediction. in *Proceedings of the 2015th International Conference on ECML PKDD Discovery Challenge - Volume 1526*, Porto, Portugal, pp. 40–51 (2015)
- Endo, Y., et al.: Predicting destinations from partial trajectories using recurrent neural network. In: *Advances in Knowledge Discovery and Data Mining*, 10234, J. Kim, K. Shim, L. Cao, J.-G. Lee, X. Lin, Y.-S. Moon, Eds. Springer International Publishing, Cham, pp. 160–172 (2017)
- Ly, J., et al.: T-CONV: A convolutional neural network for multi-scale taxi trajectory prediction. In: *2018 IEEE International Conference on Big Data and Smart Computing (bigcomp)*, Shanghai, China, pp. 82–89 (2018)
- Rossi, A., et al.: Modeling taxi drivers' behaviour for the next destination prediction. *IEEE Trans. Intell. Transport. Syst.*, 21(7), 2980–2989 (2019)
- Simmons, R., et al.: Learning to Predict Driver Route and Destination Intent. In: *2006 IEEE Intelligent Transportation Systems Conference*, Toronto, ON, Canada, pp. 127–132 (2006)
- Cho, S.-B.: Exploiting machine learning techniques for location recognition and prediction with smartphone logs. *Neurocomputing*, 176, 98–106 (2016)
- Alvarez-Garcia, J.A., Ortega, J.A., Gonzalez-Abril, L., Velasco, F.: Trip destination prediction based on past GPS log using a Hidden Markov Model. *Expert Syst. Appl.*, 37(12), 8166–8171 (2010)
- Xue, A.Y., et al.: Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. In: *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, Brisbane, QLD, pp. 254–265 (2013)
- Xue, A.Y., et al.: Solving the data sparsity problem in destination prediction. *The VLDB Journal* 24(2), 219–243 (2015)
- Hung, C.-C., Peng, W.-C., Lee, W.-C.: Clustering and aggregating clues of trajectories for mining trajectory patterns and routes. *The VLDB Journal*, 24(2), 169–192 (2015)
- Dodge, S., Laube, P., Weibel, R.: Movement similarity assessment using symbolic representation of trajectories. *International Journal of Geographical Information Science* 26(9), 1563–1588 (2012)
- Furtado, A.S., et al.: Unveiling movement uncertainty for robust trajectory similarity analysis. *International Journal of Geographical Information Science*, 32(1), 140–168 (2018)

18. Chen, M., Liu, Y., Yu, X.: Predicting next locations with object clustering and trajectory clustering. In: *Advances in Knowledge Discovery and Data Mining*, T. Cao, E.-P. Lim, Z.-H. Zhou, T.-B. Ho, D. Cheung, H. Motoda, Eds. Springer International Publishing, Cham, pp. 344–356 (2015)
19. Besse Philippe C., Guillouet Brendan, Loubes Jean-Michel, Royer Francois Destination Prediction by Trajectory Distribution-Based Model. *IEEE Transactions on Intelligent Transportation Systems* 19(8), 2470–2481 (2018). <https://doi.org/10.1109/its.2017.2749413>
20. Terada, T., et al.: Design of a car navigation system that predicts user destination. In: *7th International Conference on Mobile Data Management (MDM'06)*, Nara, Japan, pp. 145–145 (2006)
21. Miyashita, K., Terada, T., Nishio, S.: A map matching algorithm for car navigation systems that predict user destination. In: *22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)*, Gino-wan, Okinawa, Japan, pp. 1551–1556 (2008)
22. Krumm, J., Horvitz, E.: Predestination: Inferring destinations from partial trajectories. In: *UbiComp 2006: Ubiquitous Computing*, 4206, P. Dourish, A. Friday, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 243–260 (2006)
23. Krumm, J., Horvitz, E.: Predestination: Where do you want to go today?. *Computer* 40(4), 105–107 (2007)
24. Ziebart, B.D., et al.: Navigate like a cabbie: probabilistic reasoning from observed context-aware behavior. In: *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*, Seoul, Korea, p. 322 (2008)
25. Kostov, V., et al.: Travel destination prediction using frequent crossing pattern from driving history. In: *Proceedings. 2005 IEEE Intelligent Transportation Systems*, 2005, Vienna, Austria, pp. 970–977 (2005)
26. Vibhav, G., et al.: Modeling transportation routines using Hybrid Dynamic Mixed Networks. In: *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI'05)*, AUAI Press, Arlington, Virginia, USA, pp. 217–224 (2015)
27. ECML/PKDD 15: Taxi Trajectory Prediction (I). <https://kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i>. Accessed 03 November 2019.
28. Zhang, L., et al.: Multi-features taxi destination prediction with frequency domain processing. *PLoS One*, 13(3), e0194629 (2018)
29. Krause, C.M., Zhang, L.: Short-term travel behavior prediction with GPS, land use, and point of interest data. *Transp. Res. Part B: Methodological* 123, 349–361 (2019)
30. Zhang, X., et al.: Prediction of taxi destinations using a novel data embedding method and ensemble learning. *IEEE Trans. Intell. Transport. Syst.* PP(99) 1–11 (2019)
31. Bengio, Y., Ducharme, R., Vincent, P.: A neural probabilistic language model. *Journal of Machine Learning Research*, 3, 932–938 (2000)
32. Pal, S.K., Mitra, S.: Multilayer perceptron, fuzzy sets, and classification. *IEEE Trans. Neural Networks*, 3(5), 683 (1992)
33. Graves, A.: Long short-term memory. *Neural Comput.*, 9(8), 1735–1780, (1997)
34. Cheng, Y.: Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 17(8), 790–799 (1995)
35. Tian, Y., et al.: LSTM-based traffic flow prediction with missing data. *Neurocomputing*, 318, 297–305 (2018)
36. Zhao, J., et al.: Traffic speed prediction under non-recurrent congestion: Based on LSTM method and BeiDou navigation satellite system data. *IEEE Intell. Transp. Syst. Mag.*, 11(2), 70–81 (2019)
37. Sun, S., Huang, R., Gao, Y.: Network-scale traffic modeling and forecasting with graphical lasso and neural networks. *J. Transp. Eng.*, 138, 1358–1367 (2012)
38. Kamarianakis, Y., Shen, W., Wynter, L.: Real-time road traffic forecasting using regime-switching space-time models and adaptive LASSO. *Appl. Stochastic Models Bus. Ind.*, 28, 297–315 (2012)
39. Zhang, L., et al.: A hybrid forecasting framework based on support vector regression with a modified genetic algorithm and a random forest for traffic flow prediction”, *Tsinghua Sci. Technol.*, 23, 479–492 (2018)
40. Tang, J., et al.: Statistical and machine-learning methods for clearance time prediction of road incidents: A methodology review. *Analytic Methods in Accident Research*, 27, 100123 (2020)
41. Castro-Neto, M., et al.: Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Syst. Appl.*, 36, 6164–6173 (2009)
42. Hong, W.C.: Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm. *Neurocomputing*, 74, 2096–2107 (2011)
43. Hu, W., Yan, L., Liu, K., Wang, H.: A short-term traffic flow forecasting method based on the hybrid PSO-SVR. *Neural Process. Lett.* 43, 155–172 (2016)

**How to cite this article:** Tang, J., Liang, J., Yu, T., Xiong, Y., Zeng, G.: Trip destination prediction based on a deep integration network by fusing multiple features from taxi trajectories. *IET Intell. Transp. Syst.* 15, 1131–1141 (2021). <https://doi.org/10.1049/itr2.12075>