

## TRAVEL MODE RECOGNITION FROM GPS DATA BASED ON LSTM

Shaowu ZHU, Haichun SUN\*, Yongcheng DUAN, Xiang DAI

*College of Police Information Technology and Cyber Security*

*People's Public Security University of China*

*Beijing, China*

*e-mail: shaowuzhu@163.com, sunhaichun@ppsuc.edu.cn,  
{443130851, 991836225}@qq.com*

Sangeet SAHA

*School of Computer Science and Electronic Engineering*

*University of Essex*

*Colchester, UK*

*e-mail: sangeet.saha@essex.ac.uk*

**Abstract.** A large amount of GPS data contains valuable hidden information. With GPS trajectory data, a Long Short-Term Memory model (LSTM) is used to identify passengers' travel modes, i.e., walking, riding buses, or driving cars. Moreover, the Quantum Genetic Algorithm (QGA) is used to optimize the LSTM model parameters, and the optimized model is used to identify the travel mode. Compared with the state-of-the-art studies, the contributions are: 1. We designed a method of data processing. We process the GPS data by pixelating, get grayscale images, and import them into the LSTM model. Finally, we use the QGA to optimize four parameters of the model, including the number of neurons and the number of hidden layers, the learning rate, and the number of iterations. LSTM is used as the classification method where QGA is adopted to optimize the parameters of the model. 2. Experimental results show that the proposed approach has higher accuracy than BP Neural Network, Random Forest and Convolutional Neural Networks (CNN), and the QGA parameter optimization method can further improve the recognition accuracy.

---

\* Corresponding author

**Keywords:** GPS, LSTM, QGA, deep learning, travel mode

**Mathematics Subject Classification 2010:** 68-T10

## 1 INTRODUCTION

With the increasing usage of mobile communication devices, more and more personal location data are generated by GPS. These data could turn out to be useful information if utilized properly. Various types of applications are based on location data analysis. With the aid of GPS data, we can identify the individual's travel mode. It has great advantages, for example, we can display resident traveling information, guide people to choose a better travel mode, and help urban traffic planning and management. It thus can bring many benefits such as decreasing traffic congestion and environmental pollution. Along with this, through accurate recognition of the travel mode, we can analyze individual travel characteristics to improve urban transport efficiency. It may also be useful in recommending relevant services. Such as launching customized advertisements and launching sneaker advertisements for pedestrians.

Travel mode recognition has been studied for many years. From the beginning of paper-and-pencil interviews to computer telephone interviews and computer-assisted self-interviews, the way to collect information on residents' travel mode cannot achieve highly accurate results [1]. These methods are complicated.

In comparison, the information obtained utilizing GPS trajectory data is more comprehensive, detailed and accurate. There exist some researches on this topic. Dabiri et al. [2], Liang et al. [3] used GPS data to recognize the travel mode. However, such methods to recognize the travel mode using GPS trajectory data need to be improved to increase accuracy. People can simply use some rules to judge the way of travel, such as speed, directions, etc., however, the obtained results are not very effective. The same speed may correspond to different modes of transportation in different scenarios. Therefore, the extraction of more features is crucial when judging the travel mode. In the feature extraction process, it is also necessary to consider the redundancy of features and the complex relationship among the features [4]. Moreover, different data samples, the method for cutting the GPS trajectory, and classification models will affect the accuracy of the mode detection.

The use of GPS trajectory data and other sensors to infer traffic modes is constantly evolving, and many classification algorithms have been used in previous studies. On the one hand, individual GPS historical data is small, and the conventional machine and deep learning algorithm rely heavily on the amount of data. LSTM has a good effect on the timing prediction class. Thus, we have employed the LSTM as the GPS travel pattern recognition method. In this paper, the accuracy of the LSTM model is compared with the existing algorithms. The results show that the accuracy of the LSTM model is higher than that of the random forest and BP

neural network. Moreover, the QGA is used to optimize the LSTM model parameters, and the optimized model is used to identify the travel mode. The results reveal that the model recognition accuracy is higher after using QGA optimization.

The rest of the paper is organized as follows. Section 2 presents the literature review. Section 3 depicts the design of the algorithm in detail. Section 4 designs the preprocessing method and QGA. Section 5 evaluates the effectiveness of the proposed approach by comparing their results with classical machine learning algorithms and QGA. Finally, Section 6 concludes this work and discusses the future direction.

## 2 LITERATURE REVIEW

In recent years, research on the use of GPS trajectory data and other sensor data to infer travel modes has become more and more popular. In order to study different travel mode recognition, different classification algorithms have been employed and these techniques have a great influence on the accuracy of the final recognition result. Typical travel pattern recognition algorithms include neural networks, random forests, support vector machines. Specifically, Zheng et al. [5] evaluated four classification algorithms, i.e., decision tree, Bayesian network, support vector machine (SVM) and conditional random field (CRF), and employed a segmentation method to cut GPS trajectory data. The classification algorithm shows that the results based on decision trees are more accurate. Zhu et al. [6] designed the trajectory segmentation algorithm using a logic hypothesis, through random forest classifier on the classification method they achieved the final recognition degree of 82.85%. Stenneth et al. [7] proposed a method combining the GPS sensor with the basic traffic network knowledge for the confusion of the past inferred motor vehicle model. The random forest algorithm was used as the main model, which improves detection accuracy by 17%. Guvensan et al. [8] propose a novel post-processing algorithm, healing algorithm, to correct the classification error generated by the machine model algorithm. The segment-based treatment algorithm improves the average accuracy of travel mode detection by 11.7%. Reddy et al. [9] use a classification system combining decision trees with first-order discrete hidden Markov models and use correlation feature-based feature selection (CFS) feature sets to eliminate irrelevant and redundant attributes. The final accuracy goes as high up to 93.6%. Bolbol et al. [10] use a framework-based reasoning model based on support vector machine (SVM) classification and tests with coarse-grained GPS data, resulting in an accuracy of 88%. Brunauer et al. [11] propose a fully data-driven classification method using a feedforward multilayer perceptron (MLP) to select the most beneficial feature subsets through evolutionary features and compare them with a logical model tree and a C4.5 tree. The achieved overall accuracy is 92.24%, which is higher than the other two methods, which are 92.09% and 84.48%, respectively. Xia et al. [12] conduct an in-depth analysis of the stationary state during the travel mode, and divide the speed into zero and the pause and wait for modes. The significance of the pause during transportation is evaluated. The support vec-

tor machine (SVM) and the ant colony optimization are used to reduce the feature dimension, thus achieving 96.31% detection accuracy. However, Zong et al. [13] use the support vector (SVC) classification model, with the genetic algorithm (GA) for optimization purposes in the SVC model and the accuracy reaches 92.2%. Liang et al. [14] use CNN for identification and add some filtering algorithms to smooth the data (can reduce the fluctuation of data), and its highest recognition accuracy is about 94%. Xiao et al. [4] use the Bayesian network and the K2 algorithm, and the resulting walking recognition rate is over 97%. Besides, the researchers have proposed many new recognition algorithms. Martin et al. [15] develop a new classification algorithm and compare it with k-Nearest Neighbor (KNN) and random forest. Experiments show that the recognition accuracy combined with the random forest algorithm is 94%. Zhu et al. [16] propose a travel mode selection model and a directed graph-guided fusion Lasso method, which reduces the time complexity of the algorithm.

Compared with other algorithms, the LSTM recognition algorithm combines the input and output states of the previous data, so it is more sensitive to time series and it also has a memory function for long-term data. Hence, it can produce good effects on travel mode recognition. Dai et al. [18] introduce shortcut connections between the inputs and the outputs of two consecutive LSTM layers to handle gradient vanishment. The result shows that the new model has a higher trajectory predicting accuracy. Yuan et al. [17] proposed a supervised LSTM (SLSTM) network to learn quality-relevant hidden dynamics for soft sensor applications, which are more relevant and useful for quality prediction. Using the LSTM model, the GPS data is analyzed and processed, the relevant parameters of the LSTM model are optimized by QGA, and the travel mode of the GPS data is finally identified.

### 3 OVERVIEW OF LSTM

The LSTM is obtained by improving the recurrent neural network (RNN). The RNN is difficult to deal with long-distance dependence, and gradient disappearance or gradient explosion are prone to occur. The LSTM introduces the cell state and adds the input and output of the previous time to the current time processing through the gating unit and linear connection. Hence, it can deal with the long-distance dependency problem.

#### 3.1 Forward Computing

In forward computing, the LSTM saves the long-term memory of the model in the cell state. The current information, the cell state, and the output of the previous moment are the input. It controls the input information of different gated units and gets output. The flowchart is shown in Figure 1.

By adding a cell state for saving the long-term state of input data, the problem that the RNN hidden layer is sensitive to short-term input is solved. The key of the

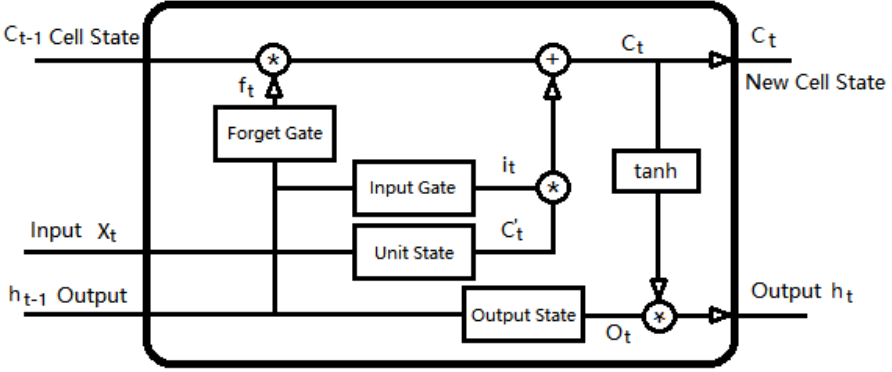


Figure 1. Forward computing structure diagram

LSTM is to control the long-term state. Here, the cell state is controlled by four control switches: input gate, output gate, forgetting gate, and unit state.

$$A * B = (A * B)_{ij} = a_{ij}b_{ij}. \quad (1)$$

Formula (1) represents the Hadamard product of A and B, which multiplies the corresponding elements in the matrix to form a new matrix. Formulas (2), (3), (4), (5), (6) and (7) describe each process of the LSTM forward computing.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \quad (2)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \quad (3)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \quad (4)$$

$$C'_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \quad (5)$$

$$C_t = f_t * C_{t-1} + i_t * C'_t, \quad (6)$$

$$h_t = o_t * \tanh(C_t). \quad (7)$$

The  $x_t$  represents the processed data of the input, and after combining with the output  $h_{t-1}$ , the gate state  $f_t$  is formed by the function to control the output at the previous moment.

$\sigma$  denotes a sigmoid function in (8), and  $\tanh$  denotes hyperbolic tangent function in (9).

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (8)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (9)$$

### 3.2 Back-Propagation Algorithm

The LSTM uses the back-propagation algorithm to calculate the weight gradient of each cell through the error term and uses the gradient descent method to get the training model.

The error is propagated back in time to control the weight matrix of the gating unit at each time and adjust the model.

$$\delta_k^T = \prod_{j=k}^{t-1} (\delta_{oj}^T W_{oh} + \delta_{fj}^T W_{fh} + \delta_{ij}^T W_{ih} + \delta_{cj}^T W_{ch}). \quad (10)$$

Formula (10) shows how the error term is transmitted forward to any time  $k$ . Among them,  $\delta$  is the error term of different elements at different times, and  $W$  is the weight matrix of different elements at different times.

## 4 OPTIMIZATION OF LSTM PARAMETERS USING QGA

In order to evaluate the travel mode from the GPS data, firstly, a segment of GPS data is processed through filtering, eliminating dirty data. Secondly, the model is constructed by calculating the relevant features based on pure data. Then, the parameters of the LSTM model are optimized. Finally, the data are trained and identified to obtain the accuracy of model recognition. The overall framework of the whole method is shown in Figure 2.

At each point of the data, we adopt three characteristics, including the stopping point, velocity, and acceleration. Each data point contains these three characteristics. Each sample contains 300 data, and each sample label is marked as a travel mode.

In the output part, we select four modes, including walk, bike, bus, and car. So the output dimension is set as (4, 1), i.e., the number of neurons in the output layer is 4.

In the LSTM model, input data is dimensionally reduced to form  $(n, 300)$  data groups, where  $n$  is the number of travel records, and each travel record corresponds to a motion mode label. During each training, a set of data of batch\_size parameter number in the LSTM model is read, and a set of current time output is calculated by using the weight coefficient of the current time. The error between the current

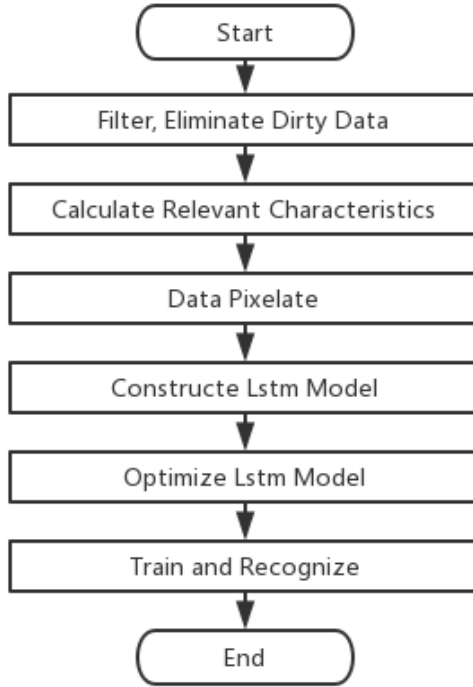


Figure 2. Framework for travel mode research and judgment

time output and the actual label is compared, and the error items are transmitted back to the end in time order. Finally, according to the error items of each time, the weight coefficients of the current time are updated to form the corresponding recognition model.

## 4.1 Data Process

### 4.1.1 Gaussian Filtering

Huang et al. [19], Sun [20], Liang [14] have used filtering methods to smooth the data. Because of the mechanical error, all the GPS positioning data cannot locate the collected position accurately. Thus the collected data will fluctuate in a certain range and produce unnecessary noise, and thus needs to be smoothed by employing Gaussian filtering algorithm, data are processed and the white noise of the data is filtered out.

$$F'_i = (F_{i-1} + F_i + F_{i+1})/3 \quad (11)$$

where  $F'_i$  is the filtered longitude and latitude of point  $i$ ,  $F_{i-1}$  is the original GPS data longitude and latitude of point  $i-1$ ,  $F_i$  is the original GPS data longitude and

latitude of  $i$ ,  $F_{i+1}$  is the original GPS data longitude and latitude of  $i+1$ . Replacing the original value with the average value of a point and its adjacent points can reduce the sudden change of data, makes the trajectory smoother, more continuous and more realistic.

#### 4.1.2 Characteristic

The original data provides the longitude, latitude, time, height difference and other information of the acquisition point. We use Geo Life project data [21, 22, 23] and introduce it in Section 5. In this paper, the stopping point, velocity, and acceleration parameters are selected as the experimental characteristics. Experiments show that the longitude and latitude data hamper model recognition as the absolute position of longitude and latitude is not closely related to the motion pattern and thus the recognition accuracy will be reduced in recognition. The elevation data are omitted because of the small variation and large error of the elevation measured in the experiment.

When the speed is below the threshold for a while, the stopping point is determined. The stopping point is according to the movement of motor vehicles under the condition of urban roads, so it is more useful to distinguish between motor vehicles and non-motor vehicles.

$$S_t = \left[ \text{sgn} \left( 10 - \sum_{i=1}^{10} v_{t-i} \right) + 1 \right] / 2. \quad (12)$$

Formula (12) shows the calculation of the stopping point, which means that when the velocity is less than 1 m/s in the first ten seconds of time  $t$ , the stopping point is denoted as 1, and 0 otherwise.

However, it is meaningful to select acceleration as the input variable of the LSTM. Different vehicles have different characteristics of acceleration and deceleration, and travel pattern recognition can be realized by using acceleration. Traditional GPS devices do not provide velocity  $V_t$  and acceleration  $A_t$ , so they can acquire the acquired GPS data by post-processing.

$$V_t = \frac{\text{Vincenty}(P_{t-1}, P_t)}{\Delta t}. \quad (13)$$

In (13), we use Vincenty Formula [24] to process GPS data and obtain the distance between two points on the sphere.  $P_{t-1}$  represents the longitude and latitude of  $t-1$  time, and  $P_t$  represents the longitude and latitude of  $t$  time.

The main travel modes of bus, car, bike, and walk are selected as the output of the LSTM.

#### 4.1.3 Normalization and Data Pixelate

Through normalization, the input characteristics are changed to decimal numbers between 0 and 1, which makes data processing more convenient and faster. And



in sigmoid function, the output tends to be zero. In the calculation of gradient descent, it can improve the convergence speed of the model and make the training identification process faster.

For each data, it needs to be pixelated and the float data normalized by one characteristic becomes eight-bit data from 0 to 255. The set of data represents eight pixels, that is, the data dimension is expanded to (100, 24). Specifically, all the data are normalized, and then each data are encoded with eight-bit binary, and finally form eight pixels data. In general, the input is three characteristics, the stopping point, velocity, and acceleration. In the grayscale image, there are 24 columns. One to eight columns are the stopping point, nine to sixteen columns are the velocity, seventeen to twenty-four columns are the acceleration. And every row is a one-time point, every one hundred rows is a group.

Each integer type data represents one pixel, and the value of 0–255 represents the gray value of each pixel. Each sample contains 2400 pixels to form a grayscale image, which is input into the model. The grayscale image is shown in Figure 3. Pixelization of the data is effective, makes the model easier to recognize, and enhances the interpretability of the model.

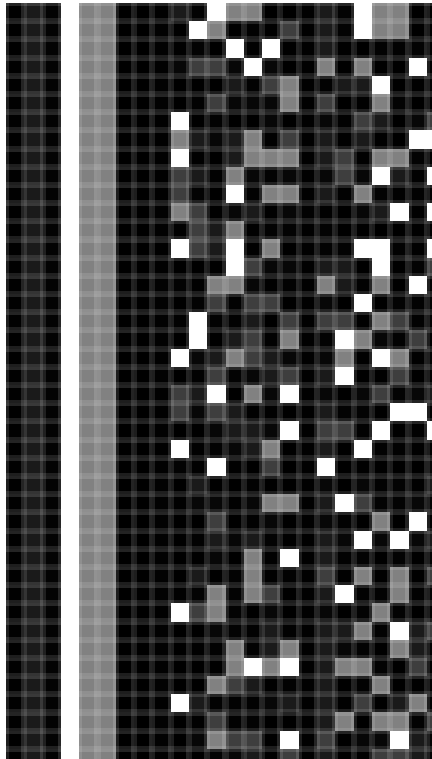


Figure 3. Grayscale images sample

## 4.2 Model Optimization Using the QGA

The QGA uses the qubit encoding method and uses the probability amplitude to represent the superposition states of 0 and 1. QGA imitates the situation of cross-variation of chromosomes in genetics, combines individuals with diversity, selects the best individuals of the population in the generational inheritance, and also generates the optimal solution of the parameters sought.

QGA [25] is an intelligent optimization algorithm combining the quantum algorithm and genetic algorithm. It solves the problem when traditional genetic algorithms fall into a locally optimal solution to a certain extent and have higher global search ability and convergence speed [26]. QGA replaces the binary coding method of chromosomes in traditional genetic algorithms by qubit and quantum superposition hence it increases the range of chromosome values and replaces the original chromosome crossover method by quantum full interference crossover method. It also improves chromosomes by the quantum revolving algorithm. The mutation method ensures the convergence of the algorithm.

Here, we optimize the number of neurons, the number of hidden layers, the learning rate, and the number of iterations. So the chromosome number is 4. In the experiment, the QGA parameters are initialized by combining the common parameter setting experience. The chromosome length is 20.

And in the QGA experiment, the iterations number is 100. As Figure 4 shows, with the addition of the iterations number, the accuracy increases. When the number is close to 100, there is hard to find more accurate parameters.

The flow chart of the QGA mechanism in order to complete the optimization is shown in Figure 5.

- Step 1:** Parameter initialization: This step determines population size, number of iterations, number of chromosomes, and chromosome length.
- Step 2:** Population initialization: This step randomly generates a population particle  $Q_i(a, b, c, d)$ , where  $a$  represents the number of neurons,  $b$  represents the number of hidden layers, and  $c$  represents the learning rate of LSTM,  $d$  represents the number of iterations of the LSTM.
- Step 3:** Measurement: The observed state  $P_i$  is obtained by measuring each individual in the population  $Q_i$ .
- Step 4:** Evaluation of fitness: The fitness function set in this paper is the accuracy of the LSTM model. This function is used to evaluate each individual in the observed state  $P(t)$ .
- Step 5:** Optimization: Using the most retained individual strategy, record the individuals with the greatest fitness in the observed state  $P(t)$ .
- Step 6:** Terminate condition: If the expected optimal result is obtained for the whole result, the algorithm ends; otherwise, return to Step 7.
- Step 7:** Update: Update the population with a quantum revolving door, return to Step 3.

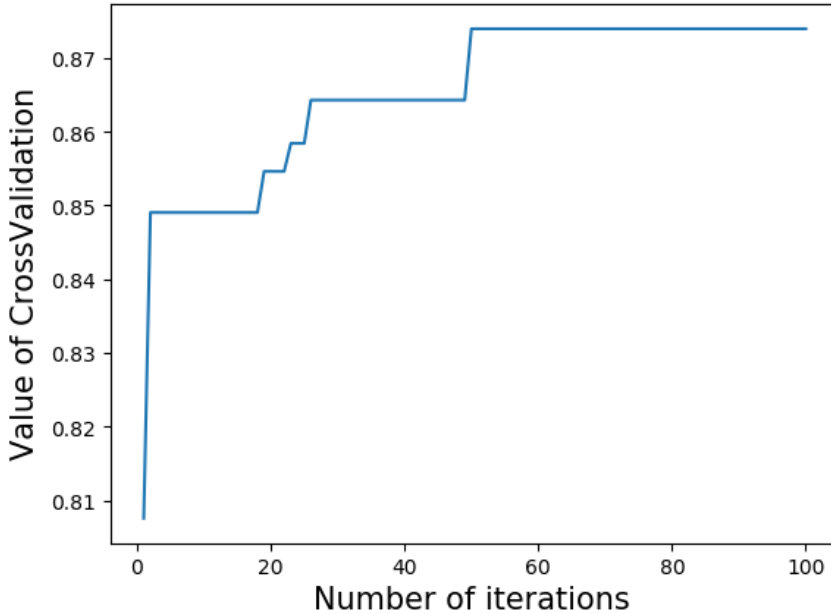


Figure 4. The effect of the number of iterations

## 5 EXPERIMENTAL RESULTS

### 5.1 Test Conditions and Data Sets

The operating environment used in the experiment is as follows: the motherboard: ASUS X455LJ, CPU: Intel® Core™ i5-5200U CPU @ 2.20 GHz (2 195 MHz), memory: 12.00 GB (1 600 MHz), main hard disk: Samsung SSD 860 EVO 250 GB; software environment: Microsoft Windows 10 (64 bit), and Pycharm integrated development environment.

As the collection methods of data sets used in each article are different, the amount of inconsistent data in each data set will vary and it may lead to different recognition accuracy. Hence, it is necessary to compare the data sets with large gaps separately. The collected data sets are usually divided into self-collection data sets and engineering data sets. Data collected independently is usually realized by the built-in function of mobile application which collects the movement of individuals for a while. Engineering data sets are collected and measured by a professional project, including Geo Life Project [21, 22, 23] and some other projects used in most studies. For example, Zhu et al. [16], Zhu et al. [6], and Zhu et al. [27] all use Geo Life's data. Geo Life Project is also used in this paper. Compared to the data collected by the application, the data collected by the project is more

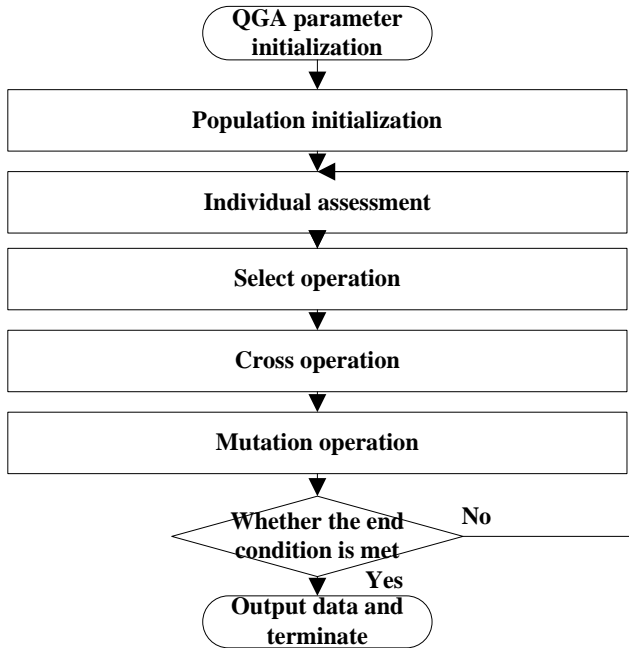


Figure 5. QGA flow chart

standard, accurate and stable. In reality, data will inevitably have inconsistency, and thus using data closer to the real environment will help to improve the value of application in real-life.

### 5.2 Comparing Results of Travel Mode Recognition Test

In this work, the LSTM model is used to identify the GPS data, and the three characteristics of stop point, speed and acceleration are added to recognize the travel pattern. This algorithm is compared with the existing algorithms in terms of recognition accuracy.

In the experiments, to ensure that the proportion of training set and test set samples is similar, 20% of the data is used as a test set through a random selection.

#### 5.2.1 Comparison with the Different Models

During testing, it is found that the accuracy of the algorithm varies slightly with each training recognition. To reduce the fluctuation of test results, the test is repeated 10 times. The average accuracy of 10 times is taken as the final accuracy of the improved algorithm.

Besides, the standard deviation of 10 recognition accuracy is calculated, and the stability of this algorithm and the existing algorithm is analyzed, as shown in Tables 1 and 2. By performing ten repeated tests, the fluctuation of the accuracy in one test is avoided, and the contingency of the experimental result is reduced to some extent.

LSTM	BP Neural Network	Random Forest	CNN
81.0 %	80.4 %	79.4 %	81.0 %
81.7 %	78.0 %	77.9 %	69.0 %
80.2 %	77.4 %	80.2 %	84.9 %
82.5 %	78.6 %	79.8 %	83.3 %
81.7 %	78.6 %	79.9 %	63.5 %
83.3 %	76.2 %	78.9 %	74.6 %
81.7 %	76.8 %	79.5 %	84.1 %
84.1 %	83.3 %	78.7 %	74.6 %
83.3 %	78.6 %	79.5 %	81.7 %
82.5 %	80.4 %	81.7 %	81.7 %

Table 1. Comparison of 10 recognition accuracy of different models

	LSTM	BP Neural Network	Random Forest	CNN
Accuracy	82.22 %	79.13 %	79.55 %	77.86 %
Average Standard Deviation	0.01132	0.00966	0.01970	0.071814

Table 2. Comparison of recognition of different models

Long-term memory is stored in the LSTM model. Long-term memory and short-term memory are used to fit different patterns, which is more suitable for identifying problems with continuous characteristics. In different modes, their travel modes are different. For example, the acceleration time of the walk mode is short, the speed is stable and low; the acceleration time of bus and car mode is longer and the speed is larger. Therefore, LSTM can distinguish between different modes more accurately. Compared to the existing algorithms, the average accuracy of the proposed algorithm is improved, and the fluctuation of recognition accuracy is stable.

### 5.2.2 Accuracy Analysis of Mode Recognition

In Figure 6, the horizontal axis is the data amount, which is represented by training set/test set, and the vertical axis is the recognition accuracy. It is evident from Figure 5 that as the number of data increases, the recognition accuracy of the algorithm also increases, and gradually stabilizes. The selection of more data could increase the recognition accuracy of the model, but it may also increase the time complexity of recognition. In the proposed experiment, the existing data has shown that the LSTM is better than other algorithms in travel mode recognition and can

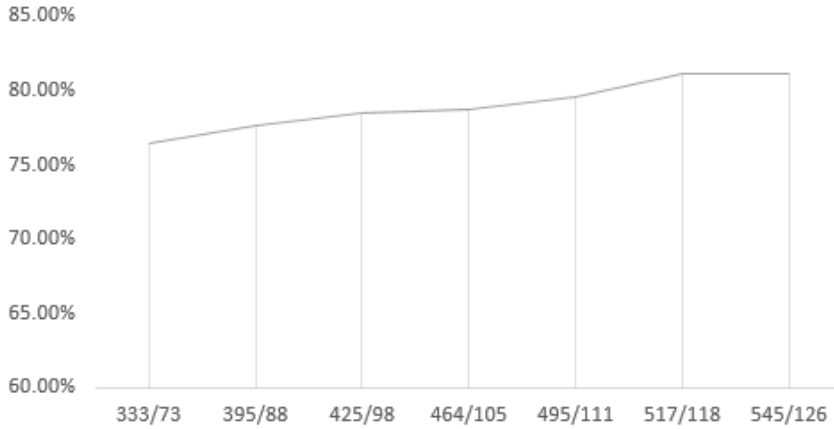


Figure 6. Algorithm identification accuracy analysis chart

have higher recognition accuracy. Therefore, although the experimental results are limited by the number of data sets, they can still make a good classification of the travel mode.

In the deep learning algorithm, the amount of data is important for the improvement of recognition accuracy. Data set covers the common modes of travel in people’s daily life, and the proportion of different travel modes in the data set is also close to reality. Among the 671 travel data, we selected 80% of the data as the training set and the remaining 20% as the test set. All the data was split by absolutely random and ensured one data does not appear in both training and test sets. Among the 10 identifications, this paper chooses the best model to be used as the test travel mode recognition obfuscation matrix, as shown in Table 3. In the case of optimal recognition, the natural error can be sufficiently reduced, the influence of actual data and algorithms on the recognition accuracy can be better reflected, which facilitates the analysis of characteristics and algorithms.

Actual \ Prediction	Prediction				Total	Accuracy
	Walk	Bike	Bus	Car		
Walk	36	4	0	0	40	90.0 %
Bike	3	18	2	1	24	75.0 %
Bus	0	3	7	6	16	43.8 %
Car	0	0	4	59	63	93.7 %
Total	39	25	13	66	143	83.9 %

Table 3. Travel mode recognition confusion matrix

In all our conducted experiments, the accuracy of bus mode recognition is significantly lower than that of other modes. According to the analysis provided in Table 3, due to the limitation of the number of data, there are fewer bus modes in the data set, so the characteristics of bus mode cannot be fitted well in the training process thus it reduces the recognition rate.

Velocity, acceleration and stop point are selected as features to be input into the model. In characteristics analysis, in the case of motor vehicles and non-motor vehicles, the polarization of features is more obvious – usually, the speed and acceleration of motor vehicles are greater, and there are more stop points. There will not be any major errors in model recognition. For the accurate recognition of motor vehicle and non-motor vehicle modes, there will be errors. For example, in Table 3, the left-lower and right-upper parts of the confusion matrix are 0, while the left-upper and right-lower parts are confusion.

### 5.2.3 Parameter Optimization Using the QGA

	Before Optimization	After Optimization
Accuracy	82.22 %	83.75 %
Error	17.78 %	16.25 %

Table 4. Comparison of the effects of different parameter optimization algorithms on classification results

To verify that the proposed QGA method has an impact on the identification results of the LSTM model, Table 4 shows the test results after optimization and without optimization. The results show that the LSTM model with optimization parameters has higher recognition accuracy and better classification effect than that of without optimization parameters.

## 6 CONCLUSION AND FUTURE WORK

In this paper, the travel mode of GPS data is extracted from a section of GPS data. Compared to previous studies, this paper converts data to the grayscale image and uses the LSTM model to identify the travel mode, and uses the QGA to optimize the parameters of the model. The identification accuracy of the model can be improved by using the QGA parameter optimization algorithm. Comprehensive analysis shows that the LSTM model and parameter optimization have a better effect on improving the accuracy of travel modes recognition algorithm, can effectively improve the accuracy of travel modes recognition for GPS data, and has higher application value for the problem of travel modes recognition for GPS data.

The adopted technique has greatly improved the overall recognition accuracy, but due to the limitation of data sets, the data used for training in this paper is

less, and the recognition problem is not enough to make accurate judgments, which should be considered in our future work.

## 7 DATA AVAILABILITY

The Geo-life project GPS data supporting this meta-analysis are from previously reported studies and datasets, which have been cited. The processed data are available from the corresponding author upon request. Data can be downloaded through the following address: <https://www.microsoft.com/en-us/download/details.aspx?id=52367>.

## Acknowledgment

This work was supported by the National Key R&D Program of China (Grant No. 2017YFC0803700), the Beijing Natural Science Foundation Program (Grant No. 4184099), National Natural Science Foundation of China (Grant No. 41971367) and Construction and Development of Key Laboratory of the Ministry of Public Security.

## REFERENCES

- [1] WANG, B.—GAO, L.—JUAN, Z.: Travel Mode Detection Using GPS Data and Socioeconomic Attributes Based on a Random Forest Classifier. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, 2018, No. 5, pp. 1547–1558, doi: 10.1109/TITS.2017.2723523.
- [2] DABIRI, S.—HEASLIP, K.: Inferring Transportation Modes from GPS Trajectories Using a Convolutional Neural Network. *Transportation Research Part C: Emerging Technologies*, Vol. 86, 2018, No. 1, pp. 360–371, doi: 10.1016/j.trc.2017.11.021.
- [3] LIANG, J.—ZHU, Q.—ZHU, M. et al.: An Enhanced Transportation Mode Detection Method Based on GPS Data. In: Zou, B., Li, M., Wang, H., Song, X., Xie, W., Lu, Z. (Eds.): *Data Science (ICPCSEE 2017)*. Springer, Singapore, Communications in Computer and Information Science, Vol. 727, 2017, pp. 605–620, doi: 10.1007/978-981-10-6385-5\_51.
- [4] XIAO, G.—JUAN, Z.—ZHANG, C.: Travel Mode Detection Based on GPS Track Data and Bayesian Networks. *Computers, Environment and Urban Systems*, Vol. 54, 2015, pp. 14–22, doi: 10.1016/j.compenvurbsys.2015.05.005.
- [5] ZHENG, Y.—LIU, L.—WANG, L.—XIE, X.: Learning Transportation Mode from Raw GPS Data for Geographic Applications on the Web. *Proceedings of the 17<sup>th</sup> International Conference on World Wide Web (WWW 2008)*, 2008, pp. 247–256, doi: 10.1145/1367497.1367532.
- [6] ZHU, Q.—ZHU, M.—LI, M.—FU, M.—HUANG, Z.—GAN, Q.—ZHOU, Z.: Transportation Modes Behaviour Analysis Based on Raw GPS Dataset. *International*



- Journal of Embedded Systems (IJES), Vol. 10, 2018, No. 2, pp. 126–136, doi: 10.1504/IJES.2018.090569.
- [7] STENNETH, L.—WOLFSON, O.—YU, P. S.—XU, B.: Transportation Mode Detection Using Mobile Phones and GIS Information. Proceedings of the 19<sup>th</sup> ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '11), 2011, pp. 54–63, doi: 10.1145/2093973.2093982.
- [8] GUVENSAN, M. A.—DUSUN, B.—CAN, B.—TURKMEN, H. I.: A Novel Segment-Based Approach for Improving Classification Performance of Transport Mode Detection. Sensors (Basel), Vol. 18, 2018, No. 1, Art.No. 87, 19 pp., doi: 10.3390/s18010087.
- [9] REDDY, S.—MUN, M.—BURKE, J.—ESTRIN, D.—HANSEN, M.—SRIVASTAVA, M.: Using Mobile Phones to Determine Transportation Modes. ACM Transactions on Sensor Networks, Vol. 6, 2010, No. 2, Art.No. 13, 27 pp., doi: 10.1145/1689239.1689243.
- [10] BOLBOL, A.—CHENG, T.—TSAPAKIS, I.—HAWORTH, J.: Inferring Hybrid Transportation Modes from Sparse GPS Data Using a Moving Window SVM Classification. Computers, Environment and Urban Systems, Vol. 36, 2012, No. 6, pp. 526–537, doi: 10.1016/j.compenvurbsys.2012.06.001.
- [11] BRUNAUER, R.—HUFNAGL, M.—REHRL, K.—WAGNER, A.: Motion Pattern Analysis Enabling Accurate Travel Mode Detection from GPS Data Only. 16<sup>th</sup> International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), 2013, pp. 404–411, doi: 10.1109/ITSC.2013.6728265.
- [12] XIA, H.—QIAO, Y.—JIAN, J.—CHANG, Y.: Using Smart Phone Sensors to Detect Transportation Modes. Sensors (Basel), Vol. 14, 2014, No. 11, pp. 20843–20865, doi: 10.3390/s141120843.
- [13] ZONG, F.—BAI, Y.—WANG, X.—YUAN, Y.—HE, Y.: Identifying Travel Mode with GPS Data Using Support Vector Machines and Genetic Algorithm. Information, Vol. 6, 2015, No. 2, pp. 212–227, doi: 10.3390/info6020212.
- [14] LIANG, X.—WANG, G.: A Convolutional Neural Network for Transportation Mode Detection Based on Smartphone Platform. 2017 IEEE 14<sup>th</sup> International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2017, pp. 338–342, doi: 10.1109/MASS.2017.81.
- [15] MARTIN, B. D.—ADDONA, V.—WOLFSON, J.—ADOMAVICIUS, G.—FAN, Y.: Methods for Real-Time Prediction of the Mode of Travel Using Smartphone-Based GPS and Accelerometer Data. Sensors (Basel), Vol. 17, 2017, No. 9, Art.No. 2058, 20 pp., doi: 10.3390/s17092058.
- [16] ZHU, X.—LI, J.—LIU, Z.—YANG, F.: Learning Transportation Mode Choice for Context-Aware Services with Directed-Graph-Guided Fused Lasso from GPS Trajectory Data. 2017 IEEE International Conference on Web Services, 2017, pp. 692–699, doi: 10.1109/ICWS.2017.83.
- [17] YUAN, X.—LI, L.—WANG, Y.: Nonlinear Dynamic Soft Sensor Modeling with Supervised Long Short-Term Memory Network. IEEE Transactions on Industrial Informatics, Vol. 16, 2020, No. 5, pp. 3168–3176, doi: 10.1109/TII.2019.2902129.

- [18] DAI, S.—LI, L.—LI, Z.: Modeling Vehicle Interactions via Modified LSTM Models for Trajectory Prediction. *IEEE Access*, Vol. 7, 2019, pp. 38287–38296, doi: 10.1109/ACCESS.2019.2907000.
- [19] HUANG, R.—TIAN, F.—TIAN, W.: Motion Pattern Recognition Using Acceleration Transducer. *Computer Engineering and Applications*, Vol. 51, 2015, No. 6, pp. 235–239.
- [20] SUN, B.—LÜ, W.—LI, W.: Activity Recognition Based on Smartphone Sensors and SC-HMM Algorithm. *Journal of Jilin University (Science Edition)*, Vol. 51, 2013, No. 6, pp. 1128–1132.
- [21] ZHENG, Y.—ZHANG, L.—XIE, X.—MA, W. Y.: Mining Interesting Locations and Travel Sequences from GPS Trajectories. *Proceedings of International Conference on World Wide Web (WWW 2009)*, Madrid, Spain. ACM Press, 2009, pp. 791–800, doi: 10.1145/1526709.1526816.
- [22] ZHENG, Y.—LI, Q.—CHEN, Y.—XIE, X.—MA, W. Y.: Understanding Mobility Based on GPS Data. *Proceedings of the 10<sup>th</sup> ACM Conference on Ubiquitous Computing (UbiComp 2008)*, Seoul, Korea. ACM Press, 2008, pp. 312–321, doi: 10.1145/1409635.1409677.
- [23] ZHENG, Y.—XIE, X.—MA, W. Y.: GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. *Invited Paper. IEEE Data(base) Engineering Bulletin*, Vol. 33, 2010, No. 2, pp. 32–39.
- [24] VINCENTY, T.: Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations. *Survey Review*, Vol. 23, 1975, No. 176, pp. 88–93, doi: 10.1179/sre.1975.23.176.88.
- [25] ZHANG, H.—GUAN, B.—REN, T.: Inverse Synchronization with Heterogeneous Structure of Chaotic System Based on Quantum Genetic Algorithm. *Chinese Journal of Quantum Electronics*, Vol. 36, 2019, No. 1, pp. 75–81.
- [26] CHEN, T.—GUESTRIN, C.: XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22<sup>nd</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
- [27] ZHU, Q.—ZHU, M.—LI, M.—FU, M.—HUANG, Z.—GAN, Q.—ZHOU, Z.: Identifying Transportation Modes from Raw GPS Data. In: Che, W. et al. (Eds.): *Social Computing (ICYCSEE 2016)*. Springer, Singapore, Communications in Computer and Information Science, Vol. 623, 2016, pp. 395–409, doi: 10.1007/978-981-10-2053-7\_35.



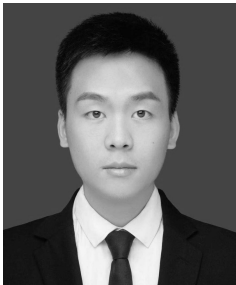
**Shaowu ZHU** is Postgraduate at the College of Information Technology and Cyber Security, People's Public Security University of China, Beijing, China. His current research interests include machine learning and information service.



**Haichun SUN** received her Ph.D. degree in computer software and theory from the Tongji University, Shanghai, China, in 2015. She is currently Assistant Professor at the College of Information Technology and Network Security, People's Public Security University of China, Beijing, China. She is a member of Professional Committee of Internet Information Service of Chinese Association of Automation. Her current research interests include information service, Petri nets, and service-oriented computing. She has published over 10 papers in journals and conferences such as the IEEE Transactions on Systems, Man, and Cybernetics, WISE 2014.



**Yongcheng DUAN** is Postgraduate at the College of Information Technology and Cyber Security, People's Public Security University of China, Beijing, China. He is working on situational awareness.



**Xiang DAI** is Postgraduate at the College of Information Technology and Cyber Security, People's Public Security University of China, Beijing, China. His current research interests include natural language processing.



**Sangeet SAHA** received his B.Tech. degree in information technology in 2011 and M.Tech. degree in computer science and engineering in 2013 from University of Calcutta, Kolkata in West Bengal, India. He completed his Ph.D. from the same institute as a Tata Consultancy Services (TCS) research fellow in 2018. After completing his Ph.D. in May 2018, he was appointed Senior Research Officer in the Embedded and Intelligent Systems (EIS) Research Group at the University of Essex in Colchester. His current research interests include real-time scheduling, scheduling for reconfigurable computers, real-time and fault-tolerant

embedded systems, cloud computing.