



# A parallel grid-search-based SVM optimization algorithm on Spark for passenger hotspot prediction

Dawen Xia<sup>1</sup> · Yongling Zheng<sup>1</sup> · Yu Bai<sup>1</sup> · Xiaobo Yan<sup>1</sup> · Yang Hu<sup>2</sup> · Yantao Li<sup>3</sup> · Huaqing Li<sup>4</sup>

Received: 28 January 2021 / Revised: 31 May 2021 / Accepted: 4 January 2022 /  
Published online: 28 March 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Predicting passenger hotspots helps drivers quickly pick up travelers, reduces cruise expenses, and maximizes revenue per unit time in intelligent transportation systems. To improve the accuracy and robustness of passenger hotspot prediction (PHP), this paper proposes a parallel Grid-Search-based Support Vector Machine (GS-SVM) optimization algorithm on Spark, which provides an efficient methodology to search for passengers in a complex urban traffic network quickly. Specifically, to effectively locate passenger hotspots, an urban road network is gridded on the Spark parallel distributed computing platform. Moreover, to enhance the accuracy of PHP, the grid search (GS) approach is employed to optimize the radial basis function (RBF) of the support vector machine (SVM), and the cross-validation method is utilized to find out the global optimal parameter combination. Finally, the SVM optimization algorithm is implemented on Spark to improve the robustness of PHP. In particular, the proposed GS-SVM algorithm is applied to successfully predict passenger hotspots. By analyzing seven groups of data sets and comparing with several state-of-the-art algorithms including autoregressive integrated moving average (ARIMA), support vector regression (SVR), long short-term memory (LSTM), and convolutional neural network (CNN), the results of an empirical study indicate that the MAPE value of our GS-SVM algorithm is lower than that of comparative algorithms at least 78.4%.

---

✉ Dawen Xia  
dwxia@gzmu.edu.cn

✉ Huaqing Li  
huaqingli@swu.edu.cn

<sup>1</sup> College of Data Science and Information Engineering, Guizhou Minzu University, Guiyang, 550025, China

<sup>2</sup> Department of Automotive Engineering, Guizhou Traffic Technician and Transportation College, Guiyang, 550008, China

<sup>3</sup> College of Computer Science, Chongqing University, Chongqing, 400044, China

<sup>4</sup> College of Electronic and Information Engineering, Southwest University, Chongqing, 400715, China

**Keywords** Visual analytics · Data visualization · Passenger hotspot prediction · Parallel SVM optimization algorithm · Spark

## 1 Introduction

It is generally noticed that taxicab imposes positive influences on the travel of residents. However, it is difficult to pick up passengers timely for drivers owing to the greater flexibility of route and time [20]. Experienced taxi drivers can pick up passengers continuously, while those lacking experience cannot find the fastest route, leading to problems including low taxi usage, huge income gap among drivers, aggravation of traffic congestion, and waste of energy in the process of locating passengers. Consequently, it is important to construct an effective model to help taxis quickly find passengers by analyzing historical trajectory data. With the popularity of the global positioning system (GPS), taxis in most cities are equipped with GPS sensors sending real-time geographic information and operating conditions of taxis, which ensures the authenticity and reliability of the data. For this reason, many researchers have paid extensive attention to the analysis, mining, and application of taxi GPS trajectory data (e.g., urban planning [2, 5, 24], urban visualization [3, 36], residents' travel characteristics [8, 40], and traffic flow forecasting [1, 4, 16, 29, 41, 47]).

When it comes to searching passengers, experienced taxi drivers quickly and accurately arrive at the destination rather than locate passengers blindly. That is why they have higher incomes than inexperienced ones. However, inexperienced ones cannot look for passengers timely, which causes high cost per unit time and low profitability. To address the problems mentioned above and achieve accurate prediction, the historical taxi GPS trajectory data are employed to model creation and data analysis, which have significant practical applications. On the one hand, accurate and effective passenger location can not only reduce the time and cost of seeking passengers and maximize benefits but also improve taxi service quality according to the needs of passengers. On the other hand, accurate positioning lays a solid foundation for reducing energy consumption and environmental pollution, providing greater accessibility through more direct traffic routes, and improving urban road networks.

Recently, existing studies on passenger location prediction have mainly focused on: (1) Linear models, such as ARIMA and Kalman filtering (KF) [10, 15, 31]; (2) Nonlinear models, such as CNN, SVM, and LSTM [12, 13, 18, 22, 27, 44]. Most of the aforementioned investigations are implemented in the traditional stand-alone environment. However, dealing with the exponential growth of the taxi trajectory data set can easily cause problems, such as time-consuming data transmission and low computational performance. Therefore, we present a parallel GS-SVM algorithm on the Spark distributed computing platform to predict passenger hotspots in this paper. Specifically, our algorithm employs the GS approach to optimize the  $C$  (penalty factor) and  $\gamma$  (core parameter) of RBF, and particularly the memory overflow, memory consumption, and other problems that are compensated in the Spark-based parallel distributed environment. Meanwhile, the speed of data transmission and the computing performance are improved in the proposed algorithm.

The main contributions of this paper are summarized as follows:

- A gridding method of the road network is proposed on the Spark parallel computing platform to locate the passenger position in a complex transportation network efficiently and accurately.
- A parallel GS-SVM algorithm on Spark is presented to predict passenger hotspots accurately. This proposed algorithm uses the GS approach to optimize the RBF function of

SVM and employs the leave-one-out cross-validation method to verify the algorithm's global optimal parameters, so that the optimized SVM algorithm has a better generalization ability and higher prediction accuracy than the traditional SVM algorithm in passenger hotspot prediction.

- With massive taxi GPS trajectory data, the parallel GS-SVM algorithm is applied to predict passenger hotspots successfully. Results demonstrate that the accuracy of GS-SVM is significantly superior to that of ARIMA, SVR, LSTM, and CNN.

The remainder of this article is organized as follows. Section 2 reviews existing works on passenger hotspot prediction. The GS-SVM algorithm and its implementation on Spark are illustrated in Section 3. Section 4 reports the extensive experiments and the result analysis, and Section 5 describes the conclusion and the future work.

## 2 Related work

In this section, we briefly introduce the related works that predict passenger hotspots and then analyze existing problems.

Finding passengers has always been one of the hot topics in intelligent transportation systems, and it has also attracted mounting researchers in recent years. Li et al. presented an optimized calculation approach based on a watershed algorithm to divide hotspots in cities and to predict the spatiotemporal variation of passenger hotspots based on the improved ARIMA model [15]. Jamil et al. utilized the automatic ARIMA model to conduct time-series analysis and forecast the hotspot areas for passengers with the spatiotemporal taxi data [10]. Qu et al. developed an effective taxi route recommendation approach, adaptive shortest expected cruising route (ASER), to forecast pick-up probability and capacity of each location using KF [31]. Niu et al. put forward a new neural network, L-CNN (combining with CNN and LSTM), and developed a robust model to predict potential passengers in real time [27]. Yang et al. proposed a novel Wave-LSTM model based on LSTM and Wavelet to predict the passenger flow [44]. Kuang et al. conducted feature embedding by attention-based LSTM and captured the correlations between pick-up and drop-off of taxis using 3D ResNet [13]. Li et al. considered the historical order demand, travel time rate, the demand of neighboring regions, day-of-week, time-of-day, weather, and point of interests, and a combined model based on WT-FCBF-LSTM (Wavelet Transform, Fast Correlation-based Filter, and Long Short-term Memory) was proposed to predict the passenger demand in different regions for different time intervals [18]. Kim et al. aimed to systematically design an explainable deep learning model being capable of assessing the quota system and balancing the demand volumes between two modes. A two-stage interpretable machine learning modeling framework was developed by a linear regression (LR) model, coupled with a neural network, and layered by long short-term memory (LSTM) [12]. Luo et al. introduced a multi-task deep learning (MTDL) model to predict short-term taxi demand at a multi-zone level [22]. To promote efficiency while reducing the emission of hybrid electric buses (HEB), Li et al. proposed a novel predictive energy management strategy with passenger prediction and exhaust emission optimization [17]. Mridha et al. proposed a two-step method predicting the taxi pickup hotspots, during various road closure incidents, by utilizing data of past taxi pickup trends [25]. Huang et al. developed a robust model based on hypertext-induced topic search (HITS), which flawlessly described the relationships of hotspots and drivers' experience [9]. Ou et al. put forward a novel deep learning framework,

STP-TrellisNets, which for the first time augmented the newly-emerged temporal convolutional framework (TrellisNet) for spatial-temporal prediction to accurately predict metro station passenger (MSP) flows [28]. Liu et al. put forward a new density peaks clustering (DPC) approach to finding demand hotspots from a low-frequency and low-quality taxi dataset [19]. Based on an improved DBSCAN algorithm, Mu et al. introduced a taxi pickup recommendation system that analyzed taxis in hotspot areas according to distinct factors [26]. Ke et al. developed a new deep learning (DL) method, fusion convolutional LSTM network (FCL-Net), to deal with these three dependencies within an end-to-end learning framework [11]. Hao et al. put forward an end-to-end DL architecture that can make multi-step predictions for all stations in a large-scale underground system [7]. Markou et al. built a multi-task learning component, and the prediction of getting in a taxi and taxi arrivals were regarded as two related events to extract the characteristics of space and time [23].

Furthermore, Zhang et al. adopted a method to obtain primary positive spatial relationships between POIs from DOPs to represent and model spatial dependencies using multivariate point pattern analysis and presented the MKDE method to predict the service range from the perspective of taxi drivers because temperature sometimes plays an essential role [46]. Saadallah et al. proposed an approach named BRIGHT, and it is a supervised learning framework with the function of predicting demand [32]. The goal of BRIGHT is to offer precise predictions for short-term demand via an innovative time series analysis method, which can address distinct types of concept drift. Zhou et al. put forward an ST-Attn model based on a DL method with a spatiotemporal attention mechanism for MsCPDP, which followed the general encoder-decoder structure for serial data modeling, yet adopted a multiple-output policy without RNN units [49]. Yu et al. developed a deep-learning framework combining an altered DBSCAN and a conditional GAN model, and the spatiotemporal and external dependencies were considered [45]. Li et al. focused on the potential passenger demand prediction of last trains from public traffic data (e.g., FCD data from the taxi and GPS/smart data from the bus) [14]. Xu et al. presented a sequence learning model to predict taxi demands based on the latest demand and correlated information [42]. Liu et al. analyzed the correlation among demand, grid probability of passengers in hotspots, and taxi drivers' different income levels. After analyzing the taxi GPS data and the income level classification of taxi drivers, the pickup spots were filtrated and matched with all these grids [21]. Qin et al. developed two new hybrids mean integrating seasonal-trend decomposition procedures based on loess (STL) with ESN improved by GOA and Adaboost structure presented to predict monthly passenger flow [30].

To sum up, the simple structure of the linear models is one of the advantages. However, the prediction accuracy of these models is relatively low, and poor performance could be perceived when they are used to deal with nonlinear problems. By contrast, the nonlinear model can make up for the shortcoming of the linear model and improve the accuracy of passenger hotspot prediction, but it has some disadvantages, such as complex structure and time-consuming calculation. Meanwhile, the nonlinear model reaches local optimality due to the selection of parameters. It is undoubted that there are still some drawbacks, including high I/O and memory cost, time-consuming data processing, and low computing performance. Moreover, to effectively compensate for the limitations of the traditional single-machine environment in processing large-scale data, such as low effectiveness and poor scalability, several works have migrated the SVM algorithm to the Spark distributed computing platform. For example, Yan et al. put forward a Microblog sentiment classification approach with a parallel Spark-based SVM [43], and Shen et al. adopted the fusion effect of SVM on Spark for speech data mining [35]. Gong et al. developed a parallel

one-to-many SVM optimization algorithm on Spark [6]. Wang et al. proposed a parallel Spark-based SVM for intrusion detection [39], and Sai et al. presented a budgeted parallel primal gradient descent Kernel SVM on Spark [33].

To this end, this paper employs a Spark parallel distributed computing platform to remedy the weakness mentioned above effectively and aims to enhance the effectiveness and scalability of prediction by optimizing the SVM algorithm. With historical data, passenger hotspot prediction is mainly used to predict the number of passengers in hotspots, which is a nonlinear problem. It is well known that SVM and deep learning algorithms can effectively cope with nonlinear problems, especially in the field of the visual analysis [34, 38, 48]. However, these methods are rarely applied to predict passenger hotspots. Therefore, this paper utilizes the SVM optimization algorithm to predict passenger hotspots. In particular, parameters of RBF built in the SVM algorithm are not the globally optimal ones. Therefore, we develop the GS approach to optimize parameters of RBF of the SVM algorithm to find the global optimal parameter combinations, which is beneficial for accurately predicting the passenger hotspots.

### 3 GS-SVM algorithm

In this section, we propose a parallel GS-SVM algorithm to improve the accuracy, effectiveness, and scalability of passenger hotspot prediction on the distributed computing platform with Spark and then implement the parallelization of the GS-SVM algorithm for predicting passenger hotspots under the Spark framework.

#### 3.1 Algorithm overview

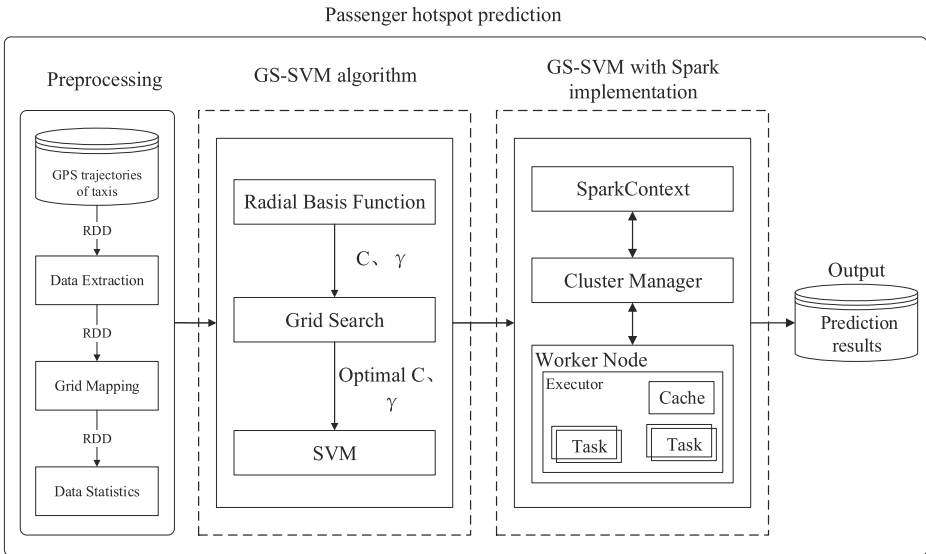
As shown in Fig. 1, the passenger hotspot prediction with the parallel GS-SVM algorithm on Spark mainly includes three steps, i.e., data preprocessing, data modeling, and algorithm implementation. In data preprocessing, there are four phases for handling GPS data based on the Spark framework consisting of data extraction, data sorting, grid mapping, and data statistics. Then, in the data modeling, the RBF kernel function parameters of SVM are optimized by the GS approach. Finally, in the algorithm implementation, to improve the efficiency and scalability of passenger hotspot prediction, this work uses RDD under the Spark framework to implement the parallelization of GS-SVM.

#### 3.2 Data preprocessing

Taxi GPS information errors occur in some situations, including GPS equipment failure, error operation of taxi drivers, and signal delay. Specifically, the signal is weak when taxis pass through specific places, such as tunnels, leading to the delay in sending messages and the deviation of the taxi's longitude and latitude. Also, some drivers deliberately set GPS operation status to occupied status to avoid being disturbed during the rest period. To enhance the reliability and accuracy of the algorithm, data preprocessing is needed to eliminate invalid data and incorrect data. The process of data preprocessing is composed of four steps shown in Fig. 2.

- **Step 1:** Data extraction.

Trajectory data in the HDFS file are read, and they are converted into RDD elastic distribution data set on Spark. Then, RDD is divided for filtering out the invalid data,



**Fig. 1** The overview of the proposed GS-SVM algorithm

and the required characteristics (taxi ID, operating condition, time, longitude, and latitude) are extracted from GPS trajectories of taxicabs. These fields are sorted by taxi ID (see Algorithm 1).

---

**Algorithm 1** Data extraction.

---

**Input:** GPS Trajectory Data of Taxicabs

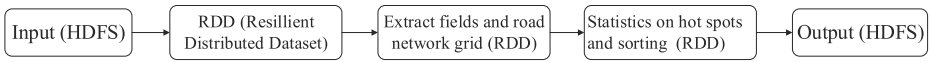
- 1:
- 2: **if**  $GPSStatus = 1$  &  $Direction < 250$  &  $Speed < 360$  **then**
- 3:     **if**  $OperatingCondition = 0$ || $1$  **then**
- 4:         **Put in** linesRDD
- 5:     **end if**
- 6: **end if**
- 7: **For** each linesRDD
- 8: Extract key: taxi ID, time; value: required field
- 9: Sort by taxi ID in ascending order

**Output:** Sort\_idRDD

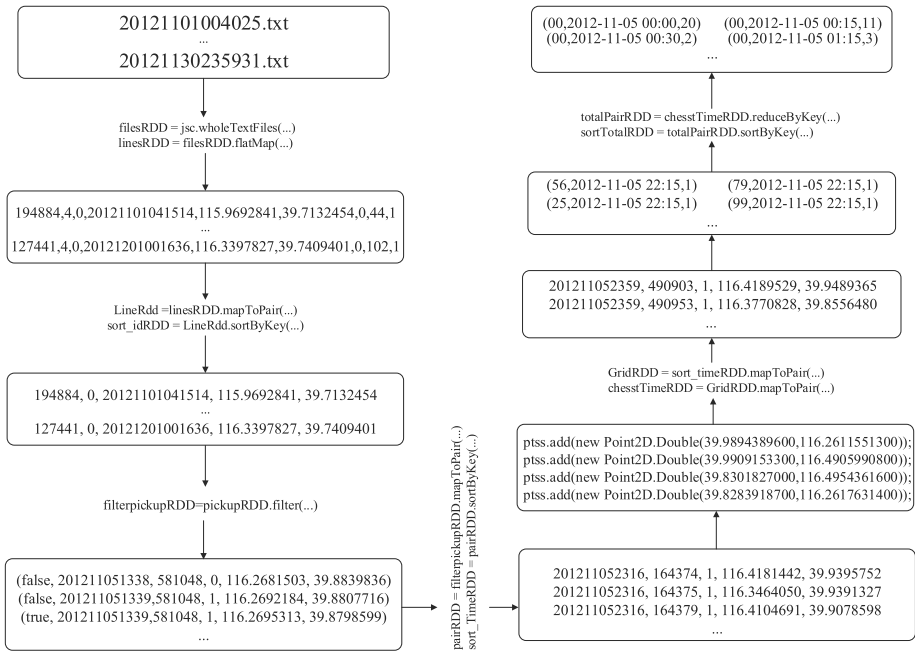
---

• **Step 2:** Data sorting.

According to the results obtained in the previous step, other taxi IDs with the same operation status can be found. If the operation status of the same taxi ID is 0, it is evident that there is no passenger in the taxi, while 1 represents that the taxi is occupied. When the second operation status of taxi ID is equal to 1, the data characteristics, including ID, operation status, time, longitude, and latitude, are saved and ordered by time (see Algorithm 2).



(a)



(b)

**Fig. 2** Data preprocessing. (a) The flow chart of data preprocessing on Spark, and (b) the flow chart of data on Spark

**Algorithm 2** Data sorting.

**Input:** Sort\_idRDD

- 1:
- 2: **if** ID fixed **then**
- 3:     Find a sequence with operating condition 011
- 4:     **if** Connect three conditions = 011 **then**
- 5:         As a pick-up point
- 6:     **end if**
- 7: **end if**
- 8: **put in** pickupRDD
- 9: Data with operating condition 1
- 10: Sort by time in ascending order

**Output:** Sort\_timeRDD

- **Step 3:** Grid mapping.

The longitude and latitude ranges are [39.8283918700-39.9909153300] and [116.2611551300-116.4954361600] in this work, respectively. The results first obtained in Step 2 are filtered out in this range. Then, the latitude and longitude of the filtered data are gridded, where the filtered data are divided into 10\*10 grids (see Algorithm 3 and Fig. 3).

---

**Algorithm 3** Grid mapping.
 

---

**Input:** Sort\_timeRDD

- 1:
- 2: Find the maximum and minimum values of longitude and latitude
- 3: Determine the rectangle by maximum and minimum
- 4:
- 5: Mesh it according to the existing rectangle
- 6:
- 7: For each sort\_timeRDD
- 8:
- 9: Determine the grid of the latitude and longitude and then mark it (GridRDD)
- 10:

**Output:** GridRDD

---

- **Step 4:** Data statistics.

The time in the data obtained in Step 3 is one day, and it is divided into data with intervals of fifteen minutes. Next, the number of pick-up hotspots in the same grid at 15-minute intervals is counted (see Algorithm 4).

---

**Algorithm 4** Data statistics.
 

---

**Input:** GridRDD

- 1:
- 2: Divide the time into 15 minutes
- 3: Mark the number of hotspots in each grid
- 4:
- 5: **put in** chesstTimeRDD
- 6:
- 7: For each chesstTimeRDD
- 8:
- 9: Count the number of hotspots with 15-minute intervals on the same grid
- 10:
- 11: The same grid is sorted by time
- 12:

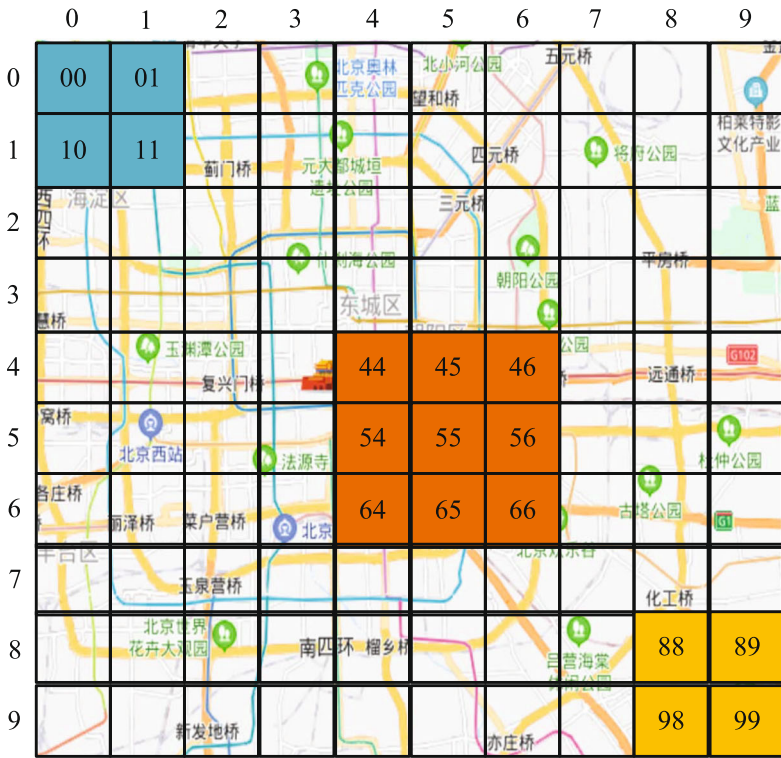
**Output:** Pick-up hotspots RDD

---

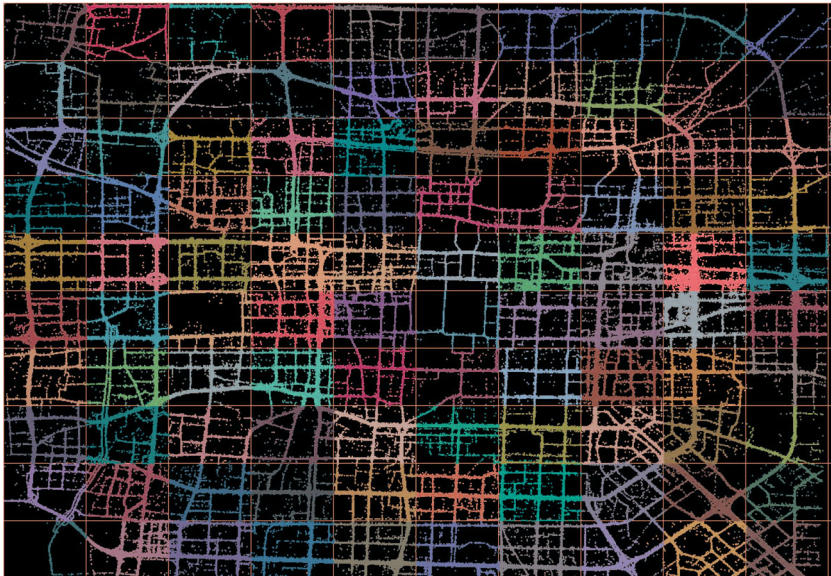
### 3.3 Algorithm design

The core idea of SVM is to construct a suitable hyperplane that is used to classify samples of the same category for addressing the linear inseparability problem. In real-world applications, there are many linearly inseparable problems. To solve them, inseparable data in





(a)



(b)

Fig. 3 Road network grid. (a) 10 × 10 grid, and (b) road network 10 × 10 grid

the low-dimensional space are usually mapped into a high dimensional space, which makes the data linearly separable. This kind of non-linear mapping is mainly realized by kernel function, and the SVM algorithm mainly uses the following four kernel functions:

(1) Linear kernel function.

$$K(X_i, X_j) = X_i^T X_j, \tag{1}$$

(2) Polynomial kernel function.

$$K(X_i, X_j) = (\gamma X_i^T X_j + r)^d, \gamma > 0, \tag{2}$$

(3) Gaussian kernel function (RBF).

$$K(X_i, X_j) = \exp(-\gamma \|X_i - X_j\|^2), \gamma > 0, \tag{3}$$

(4) Sigmoid kernel function.

$$K(X_i, X_j) = \tanh(\gamma X_i^T X_j + r), \tag{4}$$

where  $\gamma$ ,  $r$ , and  $d$  are the kernel parameters.  $d$  is used to set the highest degree of the polynomial kernel function,  $\gamma$  is utilized to set the gamma parameter in the kernel function, and  $r$  is employed to set the  $\text{coef}_0$  in the kernel.

When the SVM algorithm is employed for regression prediction, the Gaussian kernel function (RBF) is mainly utilized to project low-dimensional linear data into high-dimensional space. However, SVM is easily affected by the parameters of the built-in radial basis function (RBF), which makes the solution of the equation fall into local optimum. Therefore, this paper uses the grid search (GS) approach to find the optimal parameter combination of the kernel function in SVM, and the optimized SVM algorithm has better generalization ability than the traditional SVM algorithm.

The linear regression function of SVM optimized by the Grid Search approach is given as follows.

$$f(t) = w\phi(x) + b, \tag{5}$$

where  $\phi(x)$  is the nonlinear mapping function.

Linear insensitive loss function  $\varepsilon$  could be defined as follows.

$$L(f(x), y, \varepsilon) = \begin{cases} 0 & |y - f(x)| \leq \varepsilon \\ |y - f(x)| - \varepsilon & |y - f(x)| > \varepsilon \end{cases}, \tag{6}$$

where  $f(x)$  represents the prediction value returned by the regression function and  $y$  denotes the real value.

To obtain  $w$  and  $b$  values, slack variables  $\xi_i$  and  $\xi_i^*$  are introduced as follows.

$$\begin{cases} \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ \text{s. t. } \begin{cases} y_i - w\Phi(x_i) - b \leq \varepsilon + \xi_i \\ -y_i + w\Phi(x_i) - b \leq \varepsilon + \xi_i^* \\ \xi_i \geq 0, \xi_i^* \geq 0 \end{cases} & i = 1, 2, \dots, l \end{cases}, \tag{7}$$

where  $C$  is the punishment factor. The bigger the value of  $C$ , the greater the sample punishment is when training errors are greater than  $\varepsilon$ .  $\varepsilon$  stipulates the error requirement of the regression function. The lower the value of  $\varepsilon$ , the smaller the error of the regression function.

The Lagrangian function is introduced and transformed into a dual solution as follows.

$$\left\{ \begin{array}{l} \max \left[ \begin{array}{l} -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i^*)(\alpha_i - \alpha_i^*)K(x_i, x_j) - \\ \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l (\alpha_i - \alpha_i^*)y_i \end{array} \right], \\ \text{s. t. } \left\{ \begin{array}{l} \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ 0 \leq \alpha_i, \alpha_i^* \leq C \end{array} \right. \end{array} \right. \quad (8)$$

where  $K(x_i, x_j) = \phi(x_i)\phi(x_j)$  is the RBF kernel function. The SVM algorithm is very susceptible to the influence of the RBF kernel function parameters  $C$  (penalty factor) and  $\gamma$  (kernel parameters), and thus we adopt the GS method to find the optimal parameter combination of the RBF kernel function  $C$  and  $\gamma$ . The optimal parameter combination obtained by GS is substituted into (8) to obtain the optimal solution of  $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_l]$  as  $\alpha^* = [\alpha_1^*, \alpha_2^*, \dots, \alpha_l^*]$ , and then there are:

$$w^* = \sum_{i=1}^l (\alpha_i - \alpha_i^*)\Phi(x_i), \quad (9)$$

$$b^* = \frac{1}{N_{sv}} \left\{ \sum_{0 < \alpha_i < C} [y_i - \sum_{x_j \in SV} (\alpha_i - \alpha_i^*)K(x_i, x_j) - \varepsilon] + \sum_{0 < \alpha_j < C} [y_j - \sum_{x_i \in SV} (\alpha_j - \alpha_j^*)K(x_i, x_j) + \varepsilon] \right\}, \quad (10)$$

where  $N_{sv}$  denotes the number of support vectors. We substitute the optimal solution obtained by (9) and (10) into the linear regression function of the SVM algorithm, and the optimized GS-SVM algorithm is achieved as follows.

$$\begin{aligned} f(x) &= w^*\Phi(x) + b^* = \sum_{i=1}^l (\alpha_i - \alpha_i^*)\Phi(x_i)\Phi(x) + b^* \\ &= \sum_{i=1}^l (\alpha_i - \alpha_i^*)K(x_i, x) + b^*. \end{aligned} \quad (11)$$

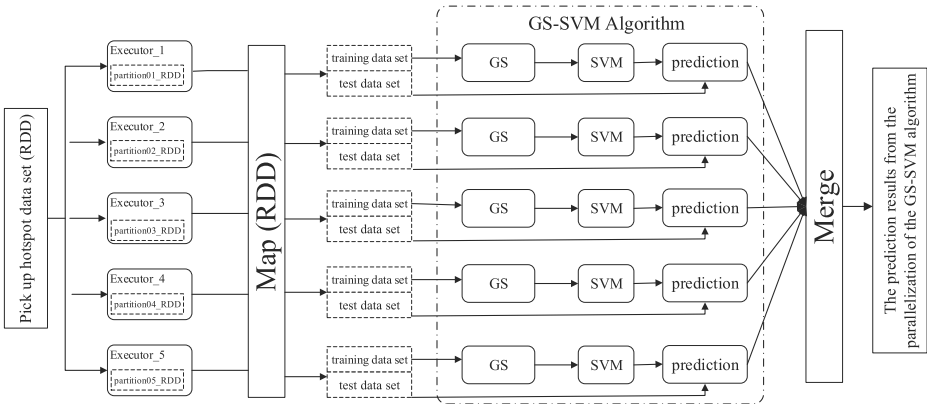
### 3.4 Algorithm implementation

To improve the efficiency and scalability of predicting passenger hotspots, we implement the parallelization of the GS-SVM algorithm under the Spark framework. The core of Spark is RDD, and the GS-SVM algorithm implemented on Spark with RDD is described in Fig. 4 in detail.

As illustrated in Fig. 4, the process of implementing GS-SVM on Spark is mainly composed of three steps.

- **Step 1:** Optimizing GS

Through a large number of experiments, the parameter range of the RBF kernel function is selected to be within  $C = [100, 300, 500, 700, 900]$  and  $\gamma = [0.001, 0.003, 0.005, 0.007, 0.009]$ , and the optimal parameter combination is chosen



**Fig. 4** The implementation of GS-SVM on Spark

as  $C = 900$  and  $\gamma = 0.001$ . The leave-one-out cross-validation (LOO-CV) method is used to test the superiority of the parameter combination.

- **Step 2:** Verifying optimal parameters

We obtain the optimal parameter combination of the GS algorithm by Step 1  $C = 900$  and  $\gamma = 0.001$ . With the application of the RBF function, the GS-SVM algorithm is experimentally verified via data sets with different sizes, and a parallel GS-SVM algorithm based on Spark is obtained.

- **Step 3:** Predicting passenger hotspots

A parallel Spark-based GS-SVM algorithm is utilized to capture nonlinear information, which predicts the number of passenger hotspots in the same grid within 15 minutes, and it is compared with other algorithms under different data sets. It is concluded that the prediction accuracy of GS-SVM is better than that of other algorithms in this paper (see Algorithm 5).

---

**Algorithm 5** Passenger hotspot prediction.

---

**Input:** Pick-up hotspotsRDD

- 1:
- 2: GS
- 3:
- 4: Search optimal  $C = 900$  and  $\gamma = 0.001$
- 5:
- 6: Optimal RBF
- 7:
- 8: SVM train
- 9:
- 10: Save mode
- 11:
- 12: Test predict.
- 13:

**Output:** PredictRDD

---

## 4 Experiments

In this section, compared with several state-of-the-art algorithms, the prediction performance of the proposed GS-SVM algorithm for passenger hotspots is validated using a real-world taxi trajectory data set, and then experimental results are analyzed in detail.

### 4.1 Experimental setup

This case study is based on a Hadoop distributed computing platform built by a Spark parallel processing framework. Furthermore, all experiments are performed on Ubuntu 18.04 OS using Hadoop 3.1.1 with Java, DL4J, and Spark 2.4.3. Our workstation consists of an Intel Xeon i7-3550 CPU and ECC DDR3 8.0 GB Memory.

In addition, we compare the GS-SVM algorithm with cutting-edge algorithms, including ARIMA, SVR, LSTM, and CNN.

### 4.2 Data description

The experimental data come from a real-world GPS trajectory data set (about 50 GB) generated by 12,000 taxis of Beijing in November 2012. The data set has more than 900 million GPS trajectory records. The data records of the data set are illustrated in Fig. 5.

In particular, for the extensive comparisons, the aforementioned data set is divided into 7 groups (i.e., 1 day: Nov. 1, 5 days: Nov. 1–Nov. 5, 10 days: Nov. 1–Nov. 10, 15 days: Nov. 1–Nov. 15, 20 days: Nov. 1–Nov. 20, 25 days: Nov. 1–Nov. 25, 30 days: Nov. 1–Nov. 30).

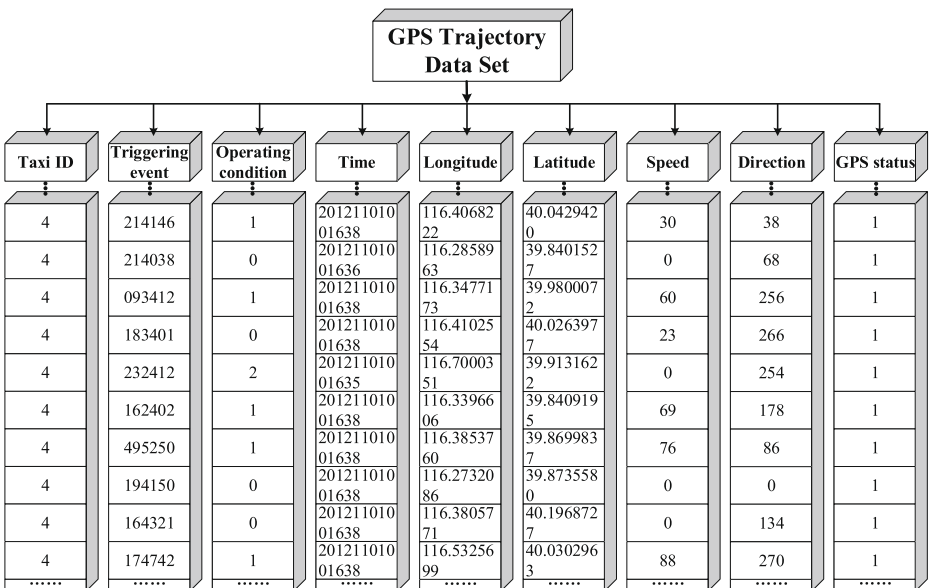


Fig. 5 GPS trajectory data of taxi

Furthermore, after data preprocessing, we select passenger hotspot data divided into blocks of 15 minutes of 1 day, 5 days, 10 days, 15 days, 20 days, 25 days, and 30 days in the first grid for prediction. For each group of data sets, this work takes 70% of the data set as the training set and 30% as the test set.

### 4.3 Evaluation metrics

To validate the measures of effectiveness (MOEs) of the algorithm, we take MAPE, RMSE, MAE, and ME as four evaluation metrics in this experiment [41], which are defined as follows.

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \frac{|X_t - \hat{X}_t|}{X_t} \times 100\%, \quad (12)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (X_t - \hat{X}_t)^2}, \quad (13)$$

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |X_t - \hat{X}_t|, \quad (14)$$

$$\text{ME} = \max_{t=1, \dots, n} |X_t - \hat{X}_t|, \quad (15)$$

where  $X_t$  denotes the real value of passenger hotspots at the time interval  $t$ ,  $\hat{X}_t$  represents the prediction value of passenger hotspots, and  $n$  is the total number of samples in the provided time. The accuracy of prediction mainly depends on MAPE. The lower the value of MAPE, the higher the accuracy of the algorithm.

### 4.4 Experimental results

Based on a large number of experiments,  $C = [100, 300, 500, 700, 900]$  and  $\gamma = [0.001, 0.003, 0.005, 0.007, 0.009]$  are defined in this paper. We also calculate the  $C$  value on the same data set, change the  $\gamma$  value, and traverse all grids to verify the performance of the parameters, which are illustrated in Table 1 and Fig. 6.

As shown in Table 1 and Fig. 6, when the parameter combination is  $C = 100$ , and  $\gamma = 0.001$ , the accuracy of the training set and the test set are 1, and the MAPE value of the GS-SVM algorithm is 0.003, and thus the algorithm performance is the best. However, the cross-validation score is poor. When  $\gamma = 0.001$  and the values are 300, 500, 700, and 900, the correct rates of the training set and the test set are equal to 1, and the cross-validation scores are also 0.94, 0.95, 0.95, and 0.96, respectively. At this time, the MAPE value of the GS-SVM algorithm is 0.002, and the algorithm performance is the best. Experimental results indicate that when the parameter combination is  $C = 900$  and  $\gamma = 0.001$ , both the correct rate of the training set and the test set are 1. Meanwhile, the cross-validation score reaches the peak, demonstrating that the optimal parameter combination found by the GS approach is the global optimal rather than local optimal.

To further evaluate the superiority of this parameter combination, this paper validates the performance of the GS-SVM algorithm when  $C = 900$  and  $\gamma = 0.001$  under different data sets, which are illustrated in Table 2.

**Table 1** Comparisons of parameters on the same data set

Data set on 1 day	GS-SVM	MAPE	MAE	RMSE	ME	Training set score	Test set score	Best cross-validation score
$C = 100$	$\gamma = 0.001$	0.003	0.084	0.105	0.224	0.999	1.000	0.790
	$\gamma = 0.003$	0.002	0.082	0.092	0.191	1.000	1.000	0.330
	$\gamma = 0.005$	0.003	0.104	0.11	0.212	1.000	1.000	-0.040
	$\gamma = 0.007$	0.003	0.098	0.109	0.229	1.000	1.000	-0.460
	$\gamma = 0.009$	0.003	0.108	0.128	0.365	1.000	1.000	-0.780
$C = 300$	$\gamma = 0.001$	0.002	0.076	0.081	0.159	1.000	1.000	0.940
	$\gamma = 0.003$	0.003	0.093	0.106	0.258	1.000	1.000	0.550
	$\gamma = 0.005$	0.003	0.099	0.106	0.206	1.000	1.000	0.050
	$\gamma = 0.007$	0.003	0.108	0.126	0.351	1.000	1.000	-0.410
	$\gamma = 0.009$	0.003	0.107	0.115	0.245	1.000	1.000	-0.760
$C = 500$	$\gamma = 0.001$	0.002	0.069	0.076	0.104	1.000	1.000	0.950
	$\gamma = 0.003$	0.002	0.071	0.076	0.101	1.000	1.000	0.590
	$\gamma = 0.005$	0.003	0.101	0.111	0.243	1.000	1.000	0.060
	$\gamma = 0.007$	0.003	0.108	0.125	0.341	1.000	1.000	-0.410
	$\gamma = 0.009$	0.003	0.106	0.112	0.204	1.000	1.000	-0.760
$C = 700$	$\gamma = 0.001$	0.002	0.068	0.075	0.104	1.000	1.000	0.950
	$\gamma = 0.003$	0.002	0.071	0.076	0.101	1.000	1.000	0.590
	$\gamma = 0.005$	0.003	0.102	0.114	0.262	1.000	1.000	0.060
	$\gamma = 0.007$	0.003	0.107	0.125	0.343	1.000	1.000	-0.410
	$\gamma = 0.009$	0.003	0.106	0.113	0.21	1.000	1.000	-0.760
$C = 900$	$\gamma = 0.001$	0.002	0.068	0.075	0.103	1.000	1.000	0.960
	$\gamma = 0.003$	0.002	0.071	0.076	0.101	1.000	1.000	0.590
	$\gamma = 0.005$	0.003	0.102	0.114	0.263	1.000	1.000	0.060
	$\gamma = 0.007$	0.003	0.108	0.119	0.291	1.000	1.000	-0.410
	$\gamma = 0.009$	0.003	0.099	0.102	0.147	1.000	1.000	-0.760

It is verified in Table 2 that, when the parameter combination is  $C = 900$  and  $\gamma = 0.001$ , the correct rates of the training set and the test set under different data sets both tend to 1, and the cross-validation score also tends to 1. In particular, the value of MAPE is relatively low. The results mentioned above show that the prediction accuracy of GS-SVM is better with the above parameter combination. Therefore, the GS approach is feasible to find the optimal parameter combination of the SVM algorithm.

To validate the effectiveness and accuracy of the GS-SVM algorithm, we divide the passenger boarding data set on the same grid into 1 day, 5 days, 10 days, 15 days, 20 days, 25 days, and 30 days for prediction. Thus, 70% of the data set is utilized for the training set, and the rest are the test set. The GS-SVM algorithm is compared with existing cutting-edge algorithms with the same data set, illustrated in Table 3, and the experimental results with different data sets are shown in Fig. 7.

Like many other studies, the prediction accuracy of the algorithm mainly depends on the MAPE value [37] in this work. Combined with Table 3, and Figs. 7 and 8, the MAPE value of the optimized GS-SVM algorithm is far lower than that of SVR, ARIMA, LSTM, and CNN. More specifically, the experimental results with the data set on 1 day, 5 days, 10

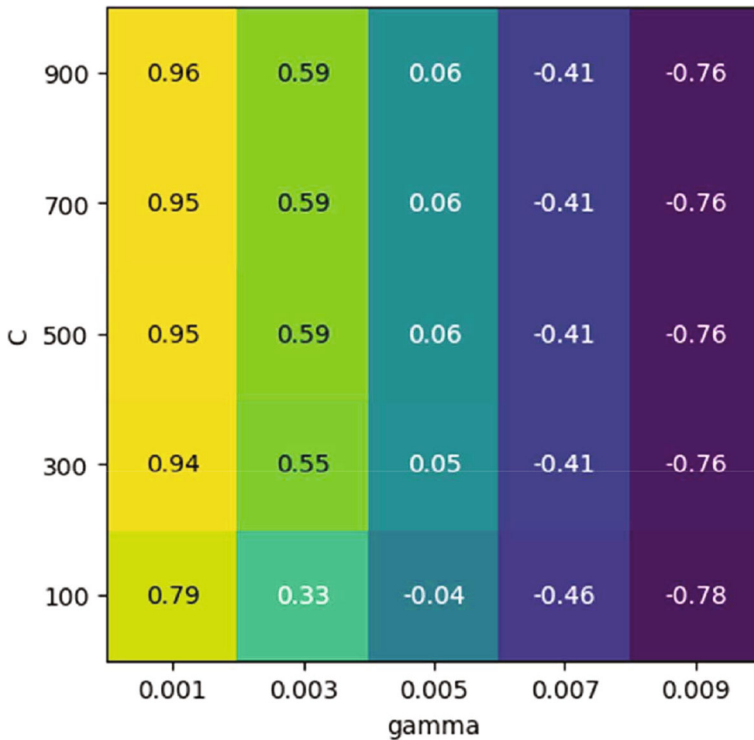


Fig. 6 Parameter tuning

days, 15 days, 20 days, 25 days, and 30 days are illustrated in Table 3. With the data set on 30 days, the MAPE value of GS-SVM is 99%, 78.8%, 85.4%, and 96% lower than that of ARIMA, SVR, LSTM, and CNN, respectively. The MAPE value obtained by GS-SVM is 99%, 92.8%, 88.9%, 78.4%, 78.6%, 81.3%, and 78.8% lower when seven groups of data sets are processed, respectively. The aforementioned results show that the SVM algorithm optimized by the GS approach has better accuracy in predicting passenger hotspots.

Table 2 Performance comparisons of GS-SVM ( $C = 900, \gamma = 0.001$ ) on different data sets

Data sets	GS-SVM ( $C = 900, \gamma = 0.001$ )						
	MAPE	MAE	RMSE	ME	Training set score	Test set score	Best cross-validation score
1 day	0.002	0.068	0.075	0.103	1.000	1.000	0.960
5 days	0.005	0.063	0.071	0.101	1.000	1.000	0.960
10 days	0.005	0.071	0.089	0.602	1.000	1.000	1.000
15 days	0.008	0.296	4.656	94.156	1.000	0.979	1.000
20 days	0.006	0.062	0.070	0.100	1.000	1.000	0.990
25 days	0.006	0.274	5.513	143.745	1.000	0.974	0.990
30 days	0.007	0.576	7.914	161.698	1.000	0.959	0.990



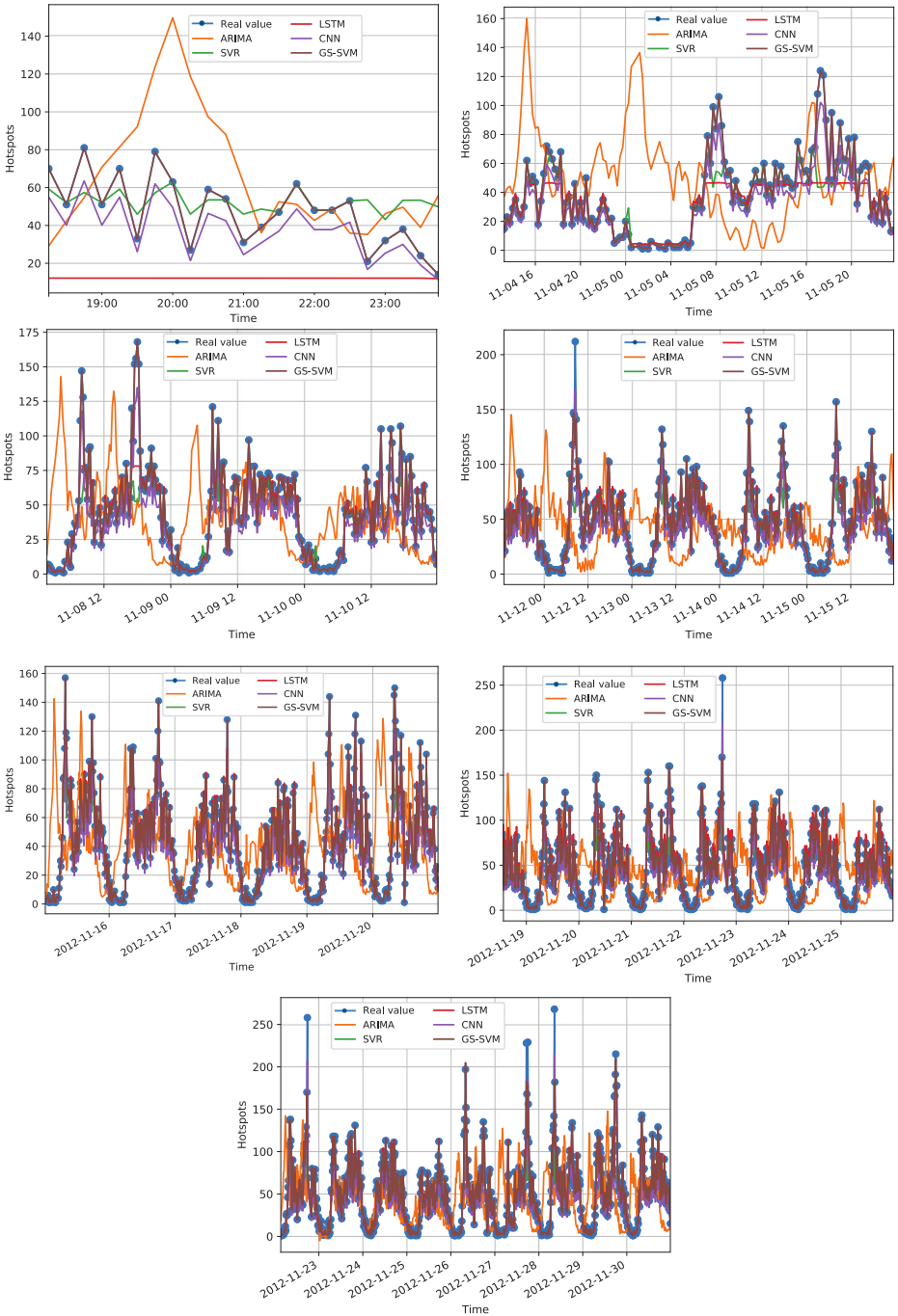
**Table 3** MOEs of ARIMA, SVR, LSTM, CNN, and GS-SVM

Data sets	MOEs	Algorithms				
		ARIMA	SVR	LSTM	CNN	GS-SVM
1 day	MAPE	0.319	0.382	0.659	0.213	0.002
	RMSE	20.646	15.746	39.902	11.429	0.075
	MAE	15.326	11.141	35.598	10.655	0.068
	ME	52.180	35.834	68.978	18.307	0.103
5 days	MAPE	0.854	0.069	0.188	0.162	0.005
	RMSE	15.442	15.460	17.798	8.094	0.071
	MAE	11.869	4.836	9.645	6.749	0.062
	ME	53.711	80.591	77.828	20.674	0.108
10 days	MAPE	0.775	0.045	0.123	0.219	0.005
	RMSE	17.244	16.799	12.688	12.357	0.089
	MAE	12.607	4.049	4.597	10.125	0.071
	ME	80.500	115.510	89.740	37.740	0.602
15 days	MAPE	0.791	0.037	0.044	0.176	0.008
	RMSE	17.123	15.852	9.492	11.367	4.656
	MAE	12.460	3.651	2.476	9.347	0.296
	ME	81.162	155.903	115.230	42.108	94.156
20 days	MAPE	0.818	0.028	0.042	0.182	0.006
	RMSE	17.449	10.882	4.754	10.634	0.070
	MAE	12.663	2.405	2.129	8.688	0.010
	ME	79.582	86.173	46.833	30.859	0.100
25 days	MAPE	0.821	0.032	0.038	0.172	0.006
	RMSE	18.052	15.045	7.017	11.129	5.513
	MAE	12.966	3.302	1.720	8.925	0.274
	ME	132.095	195.890	139.023	49.616	143.745
30 days	MAPE	0.730	0.033	0.048	0.173	0.007
	RMSE	18.621	20.069	11.046	12.523	7.914
	MAE	13.109	4.103	3.607	9.746	0.576
	ME	141.719	202.111	139.212	53.535	161.698

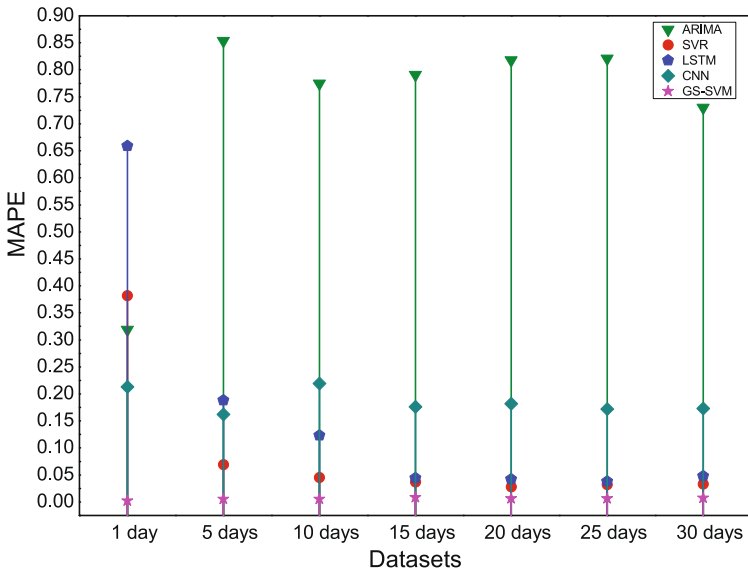
To further evaluate the universality of the parallel GS-SVM algorithm, we select 00 grid, 55 grid, 99 grid, and other surrounding grids to produce the mean value of MAPE under the data set on 30 days, and plot the results in Tables 4, 5, and 6, respectively.

Table 4 explains the value of MAPE for the surrounding grid of 00 grid, 01 grid, 10 grid, and 11 grid with the data set on 30 days. Moreover, Table 5 shows the MAPE values of 44 grid, 45 grid, 46 grid, 54 grid, 56 grid, 64 grid, 65 grid, and 66 grid, which are around the 55 grid with the data set on 30 days. Finally, Table 6 illustrates the MAPE values of 88 grid, 89 grid, and 99 grid, around the 99 grid.

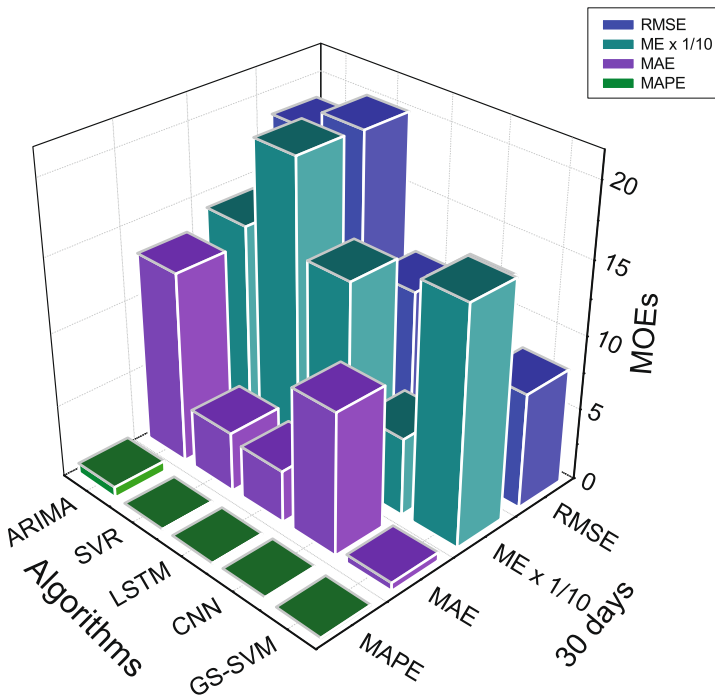
From Tables 4, 5, and 6, it is evident that the average values of MAPE generated from the parallel GS-SVM algorithm are 0.7%, 0.28%, and 0.27%, respectively. These experimental results show that it is reliable to predict passenger hotspots between different grids in different areas of passenger hotspots with the parallel GS-SVM algorithm on Spark.



**Fig. 7** Prediction results of GS-SVM on different data sets. (a) 1 day, (b) 5 days, (c) 10 days, (d) 15 days, (e) 20 days, (f) 25 days, and (g) 30 days



(a)



(b)

**Fig. 8** MOEs. (a) MAPE values of different algorithms on different data sets, and (b) MOEs of 30 days

**Table 4** MOEs values of different grids on the same data set

Data sets	MOEs			
	MAPE	MAE	RMSE	ME
00-30 days	0.007	0.576	7.914	161.698
01-30 days	0.006	0.071	0.168	4.38
10-30 days	0.007	0.349	5.331	109.373
11-30 days	0.007	0.108	1.034	28.317

**Table 5** MOEs values of different grids on the same data set

Data sets	MOEs			
	MAPE	MAE	RMSE	ME
44-30 days	0.003	0.904	10.823	235.615
45-30 days	0.003	0.071	0.102	2.044
46-30 days	0.003	0.072	0.078	0.153
54-30 days	0.003	0.544	9.203	266.466
55-30 days	0.006	0.065	0.073	0.100
56-30 days	0.003	0.069	0.130	2.283
64-30 days	0.001	0.302	3.618	77.418
65-30 days	0.002	0.091	0.410	7.795
66-30 days	0.001	0.090	0.362	10.761

**Table 6** MOEs values of different grids

Data sets	MOEs			
	MAPE	MAE	RMSE	ME
88-30 days	0.003	0.595	10.716	301.37
89-30 days	0.002	0.069	0.089	1.433
98-30 days	0.004	0.142	2.113	62.632
99-30 days	0.002	0.358	6.385	178.753

## 5 Conclusions

In this paper, a parallel GS-SVM algorithm based on Spark was proposed to predict passenger hotspots with real-world GPS trajectories of taxicabs, which was beneficial for locating passengers timely and accurately. Specifically, the urban traffic road network was gridded on Spark, and passenger hotspots in the same grid were counted at an interval of 15 minutes. Then, the GS approach found the optimal combination of  $C$  (penalty factor) and  $\gamma$  (nuclear parameter) in the RBF function, and it was validated by cross-validation methodology. When the optimal combination of the RBF function is  $C = 900$  and  $\gamma = 0.001$ , both the correct rates of the training set and the test set are 1, and the cross-validation score reaches the peak, which indicates that finding the optimal parameter combination in the GS approach is the global optimal rather than the local optimal.

Finally, this combination was substituted into the RBF function. Compared with ARIMA, SVR, LSTM, and CNN, the MAPE value generated from the proposed GS-SVM algorithm was decreased by 99.0%, 92.8%, 88.9%, 78.4%, 78.6%, 81.3%, and 78.8% in seven groups of different data sets. The experimental results demonstrated that our algorithm had a high prediction accuracy and effectively process large-scale traffic data on the Spark platform. In addition, to validate the universality of GS-SVM, we predicted the surrounding grids by fixing a grid. The surrounding MAPE values of 00 grid, 55 grid, and 99 grid are 0.7%, 0.28%, and 0.27%, respectively, which verified the reliability of GS-SVM.

This work considers the number of passenger hotspots in each grid within 15 minutes but does not consider the impacts of weather factors, traffic conditions, and passenger mobility. For example, the travel rate of passengers on sunny days may be higher than on cloudy and rainy days. When the traffic conditions are good, the travel rate of residents is high, and passenger mobility may affect the accuracy of the number of passengers in the grid. On the other hand, this paper only employs the GPS trajectory data of taxis in Beijing to validate the universality of the algorithm, so it is unknown that the optimal combination of the parameters is still applicable under different data sets. In future work, we will incorporate the impacts of weather, traffic conditions, and passenger mobility into predicting passenger hotspots and utilize taxi GPS trajectory data in different cities to evaluate the algorithm.

**Acknowledgments** This work described in this paper was supported in part by the National Natural Science Foundation of China (Grant nos. 61762020, 62162012, 61773321, 62072061, and 62173278), the Science and Technology Talents Fund for Excellent Young of Guizhou (Grant no. QKHPTRC20195669), the Science and Technology Support Program of Guizhou (Grant no. QKHZC2021YB531), and the Scientific Research Platform Project of Guizhou Minzu University (Grant no. GZ-MUSYS[2021]04).

## Declarations

**Conflict of Interests** The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

1. Ali A, Zhu Y, Zakarya M (2021) A data aggregation based approach to exploit dynamic spatio-temporal correlations for citywide crowd flows prediction in fog computing. *Multimed Tools Appl*, pp 1–33
2. Bashir M, Ashraf J, Habib A, Muzammil M (2020) An intelligent linear time trajectory data compression framework for smart planning of sustainable metropolitan cities. *Transactions on Emerging Telecommunications Technologies* e3886

3. Boeing G (2021) Spatial information and the legibility of urban form: Big data in urban morphology. *Int J Inf Manag* 56:102013
4. Chen L, Zheng L, Yang J, Xia D, Liu W (2020) Short-term traffic flow prediction: From the perspective of traffic flow decomposition. *Neurocomputing* 413:444–456
5. García FT, Villalba LJG, Orozco ALS, Kim T-H (2019) A comparison of learning methods over raw data: Forecasting cab services market share in new york city. *Multimed Tools Appl* 78:29783–29804
6. Gong Y, Jia L (2019) Research on SVM environment performance of parallel computing based on large data set of machine learning. *The Journal of Supercomputing* 75:5966–5983
7. Hao S, Lee D-H, Zhao D (2019) Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system. *Transportation Research Part C: Emerging Technologies* 107:287–300
8. Huan L, Lu Z (2020) Identification method of residents' medical travel behavior characteristics driven by mobile signaling data: A case study of kunshan. In: 2020 5Th international conference on information science, computer technology and transportation (ISCTT), IEEE, pp 198–207
9. Huang Z, Xu J, Zhan G, Zheng N, Xu M, Tu L (2019) Passenger searching from taxi traces using HITS-based inference model. In: 2019 20Th IEEE international conference on mobile data management (MDM), IEEE, pp 1440–149
10. Jamil MS, Akbar S (2017) Taxi passenger hotspot prediction using automatic ARIMA model. In: 2017 3Rd international conference on science in information technology (ICSITech), IEEE, pp 23–28
11. Ke J, Zheng H, Yang H, Chen XM (2017) Short-term forecasting of passenger demand under on-demand ride services: a spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies* 85:591–608
12. Kim T, Sharda S, Zhou X, Pendyala RM (2020) A stepwise interpretable machine learning framework using linear regression LR and long short-term memory LSTM: City-wide demand-side prediction of yellow taxi and for-hire vehicle FHV service. *Transportation Research Part C: Emerging Technologies* 120:1–15
13. Kuang L, Yan X, Tan X, Li S, Yang X (2019) Predicting taxi demand based on 3D convolutional neural network and multi-task learning. *Remote Sens* 11:1265
14. Li W, Luo Q, Cai Q (2019) Coordination of last train transfers using potential passenger demand from public transport modes. *IEEE Access* 7:126037–126050
15. Li X, Pan G, Wu Z, Qi G, Li S, Zhang D, Zhang W, Wang Z (2012) Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science* 6:111–121
16. Li W, Wang X, Zhang Y, Wu Q (2021) Traffic flow prediction over multi-sensor data correlation with graph convolution network. *Neurocomputing* 427:50–63
17. Li M, Yan M, He H, Peng J (2021) Data-driven predictive energy management and emission optimization for hybrid electric buses considering speed and passengers prediction. *Journal of Cleaner Production*, pp 127139
18. Li X, Zhang Y, Du M, Yang J (2020) The forecasting of passenger demand under hybrid ridesharing service modes: A combined model based on WT-FCBF-LSTM. *Sustainable Cities and Society* 62:1–39
19. Liu D, Cheng S-F, Yang Y (2015) Density peaks clustering approach for discovering demand hot spots in city-scale taxi fleet dataset. In: 2015 IEEE 18th international conference on intelligent transportation systems, IEEE, pp 1831–1836
20. Liu S, Pu J, Luo Q, Qu H, Ni LM, Krishnan R (2013) VAIT: A visual analytics system for metropolitan transportation. *IEEE Trans Intell Transp Syst* 14:1586–1596
21. Liu L, Wu C, Zhang H, Naji HAH, Chu W, Atombo C Research on taxi drivers' passenger hotspot selecting patterns based on GPS data: A case study in Wuhan. In: 2017 4Th international conference on transportation information and safety (ICTIS), IEEE, pp 432–441
22. Luo H, Cai J, Zhang K, Xie R, Zheng L (2020) A multi-task deep learning model for short-term taxi demand forecasting considering spatiotemporal dependences. *Journal of Traffic and Transportation Engineering (English Edition)*, pp 1–12
23. Markou I, Kaiser K, Pereira FC (2019) Predicting taxi demand hotspots using automated internet search queries. *Transportation Research Part C: Emerging Technologies* 102:73–86
24. Mouratidis K (2021) Urban planning and quality of life: A review of pathways linking the built environment to subjective well-being. *Cities* 115:1–12
25. Mridha S, Ghosh S, Singh R, Bhattacharya S, Ganguly N Mining Twitter and taxi data for predicting taxi pickup hotspots. In: Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining, pp 27–30
26. Mu B, Dai M (2019) Recommend taxi pick-up hotspots based on density-based clustering. In: 2019 IEEE 2Nd international conference on computer and communication engineering technology (CCET), IEEE, pp 176–181

27. Niu K, Cheng C, Chang J, Zhang H, Zhou T (2018) Real-time taxi-passenger prediction with l-CNN. *IEEE Trans Veh Technol* 68:4122–4129
28. Ou J, Sun J, Zhu Y, Jin H, Liu Y, Zhang F, Huang J, Wang X (2020) Stp-trellisnets: Spatial-temporal parallel trellisnets for metro station passenger flow prediction. In: Proceedings of the 29th ACM international conference on information and knowledge management, association for computing machinery, pp 1185–1194
29. Peng H, Wang H, Du B, Bhuiyan MZA, Ma H, Liu J, Wang L, Yang Z, Du L, Wang S (2020) Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Inf Sci* 521:277–290
30. Qin L, Li W, Li S (2019) Effective passenger flow forecasting using STL and ESN based on two improvement strategies. *Neurocomputing* 356:244–256
31. Qu B, Yang W, Cui G, Wang X (2019) Profitable taxi travel route recommendation based on big taxi trajectory data. *IEEE Trans Intell Transp Syst* 21:653–668
32. Saadallah A, Moreira-Matias L, Sousa R, Khiari J, Jenelius E, Gama J (2020) Bright-drift-aware demand predictions for taxi networks. *IEEE Trans Knowl Data Eng* 32:234–245
33. Sai J, Wang B, Wu B Bppgd: Budgeted parallel primal gradient descent kernel SVM on Spark. In: 2016 IEEE First international conference on data science in cyberspace (DSC), IEEE, pp 74–79
34. Shen J, Deng RH, Cheng Z, Nie L, Yan S (2015) On robust image spam filtering via comprehensive visual modeling. *Pattern Recogn* 48:3227–3238
35. Shen J, Wang HH (2020) Fusion effect of SVM in Spark architecture for speech data mining in cluster structure. *Int J Speech Technol* 23:481–488
36. Silva RA, Pires JM, Datia N, Santos MY, Martins B, Birra F (2019) Visual analytics for spatiotemporal events. *Multimed Tools Appl* 78:32805–32847
37. Smith BL, Williams BM, Oswald RK (2002) Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies* 10:303–321
38. Wang L, Qian X, Zhang Y, Shen J, Cao X (2020) Enhancing sketch-based image retrieval by cnn semantic re-ranking. *IEEE Trans Cybern* 50:3330–3342
39. Wang H, Xiao Y, Long Y (2017) Research of intrusion detection algorithm based on parallel SVM on Spark. In: 2017 7th IEEE international conference on electronics information and emergency communication (ICEIEC), IEEE, pp 153–156
40. Xia D, Lu X, Li H, Wang W, Li Y, Zhang Z (2018) A MapReduce-based parallel frequent pattern growth algorithm for spatiotemporal association analysis of mobile trajectory big data. *Complexity* 2018
41. Xia D, Zhang M, Yan X, Bai Y, Zheng Y, Li Y, Li H (2021) A distributed WND-LSTM model on MapReduce for short-term traffic flow prediction. *Neural Comput Applic* 33:2393–2410
42. Xu J, Rahmatizadeh R, Bölöni L, Turgut D (2017) Real-time prediction of taxi demand using recurrent neural networks. *IEEE Trans Intell Transp Syst* 19:2572–2581
43. Yan B, Yang Z, Ren Y, Tan X, Liu E (2017) Microblog sentiment classification using parallel SVM in Apache Spark. In: 2017 IEEE International congress on big data (BigData Congress), IEEE, pp 282–288
44. Yang X, Xue Q, Yang X, Yin H, Qu Y, Li X, Wu J (2021) A novel prediction model for the inbound passenger flow of urban rail transit. *Information Sciences*
45. Yu H, Chen X, Li Z, Zhang G, Liu P, Yang J, Yang Y (2019) Taxi-based mobility demand formulation and prediction using conditional generative adversarial network-driven learning approaches. *IEEE Trans Intell Transp Syst* 20:3888–3899
46. Zhang S, Tang J, Wang H, Wang Y, An S (2017) Revealing intra-urban travel patterns and service ranges from taxi trajectories. *J Transp Geogr* 61:72–86
47. Zhao W, Gao Y, Ji T, Wan X, Ye F, Bai G (2019) Deep temporal convolutional networks for short-term traffic flow forecasting. *IEEE Access* 7:114496–114507
48. Zhao T, Zhang B, He M, Wei Z, Zhou N, Yu J, Fan J (2018) Embedding visual hierarchy with deep networks for large-scale visual recognition. *IEEE Trans Image Process* 27:4740–4755
49. Zhou Y, Li J, Chen H, Wu Y, Wu J, Chen L (2020) A spatiotemporal attention mechanism-based model for multi-step citywide passenger demand prediction. *Inf Sci* 513:372–385



**Dawen Xia** is currently a Professor at the College of Data Science and Information Engineering & Key Laboratory of Pattern Recognition and Intelligent Systems of Guizhou Province, Guizhou Minzu University, Guiyang, China. He received the Ph.D. degree from the College of Computer and Information Science & College of Software, Southwest University, Chongqing, China, in June 2016. He was a Visiting Scholar supported by China Scholarship Council with the Management Science and Information Systems, Rutgers, the State University of New Jersey, USA, from November 2019 to November 2020. His research interests include big data analytics, artificial intelligence, and data mining.



**Yongling Zheng** is currently working toward the M.S. degree at the College of Data Science and Information Engineering, Guizhou Minzu University, Guiyang, China. Her research interests include big data analytics, artificial intelligence, and data mining.





**Yu Bai** is currently working toward the M.S. degree at the College of Data Science and Information Engineering, Guizhou Minzu University, Guiyang, China. Her research interests include big data analytics, artificial intelligence, and data mining.



**Xiaobo Yan** received the M.S. degree from the College of Data Science and Information Engineering, Guizhou Minzu University, Guiyang, China. His research interests include big data analytics, artificial intelligence, and pattern recognition.



**Yang Hu** is currently an Assistant Lecturer at the Department of Automotive Engineering, Guizhou Traffic Technician and Transportation College, Guiyang, China. He received the M.Eng. degree from the College of Big Data and Information Engineering, Guizhou University, Guiyang, China, in June 2018. From September 2016 to February 2017, he was a visiting student with the Department of Computer Science and Information Engineering, National Taiwan University. His research interests include artificial intelligence, pattern recognition, and big data analytics.



**Yantao Li** is currently a tenure-track assistant professor at the College of Computer Science, Chongqing University, Chongqing, China. He received the Ph.D. degree from the College of Computer Science, Chongqing University, Chongqing, China, in December 2012. His research area includes wireless communication and networking, sensor networks and ubiquitous computing, and artificial intelligence. He was a recipient of the Outstanding Ph.D. Thesis Award in Chongqing in 2014, and the Outstanding Master's Thesis Award in Chongqing in 2011.



**Huaqing Li** is currently a Professor at the College of Electronic and Information Engineering & Key Laboratory of Nonlinear Circuits and Intelligent Information Processing of Chongqing, Southwest University, Chongqing, China. He received the B.S. degree in information and computing science from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2009, and the Ph.D. degree in computer science and technology from Chongqing University, Chongqing, China, in 2013. He was a Postdoctoral Researcher with the School of Electrical and Information Engineering, University of Sydney, Sydney, NSW, Australia, from September 2014 to September 2015, and the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from November 2015 to November 2016. His main research interests include artificial intelligence, nonlinear dynamics and control, multi-agent systems, and distributed optimization. He serves as a Regional Editor for *Neural Computing and Applications* and an Editorial Board Member for *IEEE ACCESS*.