

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Deep Learning Based Prediction of Transfer Probability of Shared Bikes Data

Wenwen Tu

Abstract

In the pile-free bicycle sharing scheme, the parking place and time of the bicycle are arbitrary. The distribution of the pile does not constrain the origin and destination of the journey. The travel demand of the user can be derived from the use of the shared bicycle. The goal of this article is to predict the probability of transition for a shared bicycle user destination based on a deep learning algorithm and a large amount of trajectory data. This study combines eXtreme Gradient Boosting (XGBoost) algorithm, stacked Restricted Boltzmann Machines (RBM), support vector regression (SVR), Differential Evolution (DE) algorithm, and Gray Wolf Optimization (GWO) algorithm. In an experimental case, the destinations of the cycling trips and the probability of traffic flow transfer for shared bikes between traffic zones were predicted by computing 2.46 million trajectory points recorded by shared bikes in Beijing. The hybrid algorithm can improve the accuracy of prediction, analyze the importance of various factors in the prediction of transfer probability, and explain the travel preferences of users in the pile free bicycle-sharing scheme.

Keywords: deep learning, restricted Boltzmann machines, support vector regression, eXtreme gradient boosting, shared bikes data

1. Introduction

Bicycle sharing is a new type of transportation with low energy consumption and emissions. It serves short-distance travel and helps solve the “last mile” problem [1]. With the rapid development of the mobile Internet, the pileless bicycle began to replace the pile station bicycle [2]. In the pile-free bicycle sharing scheme, the parking place and time of the bicycle are arbitrary. The distribution of the pile does not constrain the origin and destination of the journey. The travel demand of the user can be derived from the use of the shared bicycle. The distribution of destinations for shared bike users is a valuable study. However, the large number of shared bicycle tracks requires a lot of computation time. This paper sets up different traffic areas and studies the law of shared bicycle flow transfer between the traffic areas. On this basis, we predict the ratio of the traffic flow of shared bicycles between traffic areas. It can be considered as the probability that the shared bicycle user selects the traffic area A as the origin and the traffic area B as the destination.

The pile-free shared bicycle system puts a large number of bicycles in the city. The amount is much higher than the number of traditional piled public bicycles. Therefore, when dealing with massive volume of trajectory data volume as a data set, classical statistical methods and traditional neural network algorithms would have limited processing capabilities.

As a newly developed travel method, the algorithms for the destination prediction of trips based on shared bikes need to be researched in depth [3–5]. In Deep neural networks (DNN), the model with multi-hidden layers can be developed based on the artificial neural network. The hidden layers of DNN convert the input data into a more abstract compound representation [6–10].

The Restricted Boltzmann Machine (RBM) is an algorithm that can be used for dimensionality reduction, classification, regression, and feature learning problems. RBM reconstructs data in an unsupervised algorithm and adjusts the weight through the process of reverse transfer and forward transfer. The RBM gradually approaches the original input and learns the probability distribution on the input set [11–15].

In this paper, a stacked RBM-SVR algorithm is constructed by combining support vector regression (SVR) [16] and stacking RBM algorithm. RBM-SVR is used to predict continuous output values. The error penalty factor c_s and kernel function parameter γ_s are the basic parameters of the radial basis function of the SVR model. The value of c_s and γ_s will directly affect the fit and generalization ability of the SVR [17–19]. In order to improve the accuracy of prediction, this paper needs to introduce intelligent algorithms to optimize the selection of parameter values.

In machine learning algorithms, Mirjalili et al. [20] proposed Gray Wolf Optimizer (GWO) as a meta-heuristic algorithm for solving many multi-modal functions in 2014. In addition, Storn and Price [21] proposed a differential evolution (DE) algorithm. The DE algorithm is an optimization algorithm based on modern intelligent theory. DE intelligently optimizes the direction of search through groups generated by cooperation and competition among individuals. Based on the above theory, an algorithm called the differential evolution Gray Wolf Optimizer algorithm (DEGWO) is used to optimize c_s and γ_s . DEGWO generates initial populations, sub-populations, and variant populations for each iteration, and then uses the GWO's capabilities for global searching to optimize the c_s and γ_s parameters.

After processing by RBM algorithm, the input data are transformed into a sparse matrix containing a high amount of information. It can reduce the computation time of SVR prediction, but it may also increase the number of outliers in the SVR algorithm, increase the complexity of the SVR model and reduce the stability of the fitting process. To solve this problem, this paper proposes a hybrid algorithm that combines the eXtreme Gradient Boosting (XGBoost) algorithm with the stacked RBM-SVR algorithm. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable [22]. XGBoost uses the Exclusive Feature Bundling method to transform several sparse features into a dense matrix [23]. In the process, the approximate greedy algorithm is used to find the best combination of merged features when the number of bundles is the smallest. The XGB algorithm realizes the partition of nodes by the second order gradient and optimizes the greedy algorithm of splitting nodes by the local approximation algorithm [24].

The principal purpose of this paper is to build a hybrid model that combines the XGBoost model, the stacked RBM-SVR network, and the DEGWO optimization algorithm. This paper analyzes the trajectory data of shared bicycles, extracts the cell information, and predicts the probability of user destination selection in the traffic area, that is, predicts the transfer probability of shared bikes.

2. Background

Artificial intelligence (AI) is a domain of computer science that studies how to apply computings to simulate the fundamental theories, methods, and techniques of human knowledge. As the mainstream algorithm of artificial intelligence, deep learning is considered capable of solving many challenges in the field of computer vision, prediction, and optimization. It realizes the automatic positioning of targets and automated learning of target features, which improves the speed and accuracy of target detection. Artificial intelligence is mainly used to share bicycles in the following aspects.

First, the user's travel behavior and the law of spatial movement can be obtained through machine learning algorithms and statistical theory analysis. The user's travel preferences can be quantitatively analyzed. Researchers can discuss the impact of various influencing factors on shared bicycle usage, such as the mix of land use, the degree of convergence with public transport facilities, the sharing of bicycle infrastructure, rainfall and high temperatures [25, 26].

Second, through the deep learning algorithm of AI technology, the dynamic demand and parking demand of the shared bicycle users can be predicted. The focus of this paper is on this issue. This paper uses a deep learning algorithm to predict the probability of user destination selection. Xu et al. [27] prove that the long short-term memory neural networks (LSTM NNs) in deep learning algorithms are superior to traditional statistical metrology algorithms and advanced machine learning algorithms. LSTM NNs can better predict the riding demand dynamically. Besides, based on the distribution of road networks and travel needs, researchers can predict parking demand and develop better layout strategies for electronic fences [28].

Finally, according to the deep reinforcement learning algorithm in AI technology, a shared bicycle scheduling model can be constructed. Deep reinforcement learning combines the perception of deep learning with the decision-making ability of reinforcement learning. It can be directly controlled according to the original input data. It is an artificial intelligence method that is closer to human thinking. Based on this algorithm, the dynamic scheduling model can efficiently optimize goals such as improving user satisfaction and reducing system cost [29–31].

3. A stacked RBM_SVR deep learning algorithm

RBM_SVR is a deep learning model that connects three stacked RBM models and one SVR model. First, in RBM_SVR, the bottommost RBM is trained with the original input data, and the top RBM takes the feature extracted by the bottom RBM as input and continues training. RBM_SVR repeats this process until the topmost RBM model is trained. Secondly, RBM_SVR fine-tunes the network through the traditional global learning algorithm (BP algorithm), so that the model can converge to the local best. Finally, RBM_SVR can efficiently train a deep network and output the predicted probability value according to the SVR model.

Each RBM model has a visible layer v and a hidden layer h . The neurons inside the RBM layer are unconnected, but the neurons between the layers are fully connected. The value of the RBM node variable is 0 or 1. The number of layers of visible layers and hidden layers of the RBM_SVR model are n and m , respectively. The energy equation of RBM_SVR is given by Eq. (1)

$$E(v, h|\theta) = - \sum_{i=1}^n a_i v_i - \sum_{j=1}^m b_j h_j - \sum_{i=1}^n \sum_{j=1}^m v_i w_{ij} h_j \quad (1)$$

where $\forall i, j \in \{0, 1\}$, $\theta = \{w_{ij}, a_i, b_j\}$ involves the parameters of RBM; w_{ij} is a connection weight between the visible layer i and the hidden layer j ; a_i represents the bias of the visible layer, and b_j denotes the bias of the hidden layer. $P(v, h|\theta) =$

$\frac{e^{-E(v, h|\theta)}}{\sum_{v, h} e^{-E(v, h|\theta)}}$ is the joint probability distribution of this set of states (v, h) [32]. $P(v|\theta) = \frac{\sum_h e^{-E(v, h|\theta)}}{\sum_{v, h} e^{-E(v, h|\theta)}}$ is the marginal distribution of $P(v, h|\theta)$. Since each visible layer and each hidden layer are independent, the activation probability of the hidden layer j and the i th visible layer are shown in Eqs. (2) and (3), respectively [33].

$$P(v_i = 1|h, \theta) = \sigma\left(a_i + \sum_j w_{ij} h_j\right) \quad (2)$$

$$P(h_j = 1|v, \theta) = \sigma\left(b_j + \sum_i v_i w_{ij}\right) \quad (3)$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$. In RBM_SVR, the number of neurons per layer of RBM is 300. Based on the abstracted vector output from the stacked RBM model, the SVR model predicts the probability of the traffic transfer among traffic zones y_d , as shown in Eq. (4).

$$y_d = \text{RBM.SVR}(x) \quad (4)$$

where x is the input dataset, RBM.SVR represents the RBM_SVR model.

4. An improved RBM-SVR algorithm

4.1 Principles of GWO algorithm

Assume that in a D-dimensional search space, the population size $X = [X_1, X_2, \dots, X_N]$ is composed of N individuals. $X_{i_h} = (X_{i_h}^1, X_{i_h}^2, \dots, X_{i_h}^D)$ is the location of the gray wolf i_h -the solution to the optimization problem. The top three wolves of the optimal solution of the objective function are wolf α , wolf β , and wolf δ , respectively. They are also the main wolves that guide the rest of the wolves to explore the optimal solution. The rest of the solution corresponds to the wolf as wolf ω . The parameters and explanations of the GWO algorithm are shown in **Table 1**.

The update process of $X(t)$ is given by Eq. (5). The first three obtained optimal values are saved to enforce other searching individuals (including ω) to constantly update their positions according to the position of the optimal value, and the calculation method is expressed as Eqs. (6)–(7).

$$X(t+1) = X_p(t) - A|CX_p(t) - X(t)| \quad (5)$$

$$X_\mu(t+1) = X_\pi(t) - A_\mu D_\pi \quad (6)$$

$$X_p(t+1) = \frac{1}{3} \sum_{\mu=1}^3 X_\mu \quad (7)$$

Parameter	Interpretation
t	The number of current iterations
C	The swing factor $C = 2r_1$
$X_p(t)$	The position of the prey after the t th iteration
$X(t)$	The position of the gray wolf during the t th iteration
r_1	A random number within $[0, 1]$
A	The convergence factor, $A = 2a \cdot r_2 - a$
r_2	A random number uniformly distributed within $[0,1]$
a	a linearly decreases from 2 to 0 with the increase of the number of iterations
D_π	The distances between the individual gray wolves, $D_\pi = C_\mu X_\pi(t) - X(t) $

Table 1.
Parameters and explanations of the GWO algorithm.

where $\pi = \alpha, \beta, \delta; \mu = 1, 2, 3$. The distances between the other individual gray wolves and α, β , and δ , as well as the distances $D_\pi = |C_\mu X_\pi(t) - X(t)|$ between them and the updated position of the gray wolf are be determined by and (6). Then, the position of the prey can be determined by Eq. (7).

4.2 Principles of the DE algorithm

Assume that in the D-dimensional search space, in the population size NP, $Z(g)$ is the g th generation of the population, $Z(g) = \{Z_1(g), Z_2(g), \dots, Z_{np}(g)\}$. $Z_k(g)$ is the k th individual in the g th generation of the population, $Z_k(g) = [Z_{k,1}(g), Z_{k,2}(g), \dots, Z_{k,D}(g)]$, $k = 1, 2, \dots, NP$, $g = 1, 2, \dots, g_{\max}$, and g_{\max} is the number of the last iteration.

4.2.1 Initialization of the population

Initially, the algorithm randomly generates the 0th generation of the population over the entire search space, and the value of the individual $z_{k,q}(0)$ in each dimension q is generated according to Eq. (8).

$$z_{k,q}(0) = z_{k,q}^L + rand(0, 1) \left(z_{k,q}^U - z_{k,q}^L \right) \tag{8}$$

where $q = 1, 2, \dots, D$, $rand(0, 1)$ is a random number, which is uniformly distributed within $[0, 1]$, $z_{k,q}^L$ is the lower threshold of the individual population, $z_{k,q}^U$ is the upper threshold of the individual population.

4.2.2 Mutation

Mutant individual is generated via Eq. (9).

$$\tau_{k,q}(g) = z_{p_1} + F(z_{p_2} - z_{p_3}) \tag{9}$$

where $z_{p_1}, z_{p_2}, z_{p_3}$ are three different parameter vectors randomly selected from the current population, and $z_{p_1} \neq z_{p_2} \neq z_{p_3} \neq i$; F is an amplifying factor within $[0,1]$.

4.2.3 Crossover

The crossover process in the DE algorithm is expressed as Eq. (10).

$$\mu_k(g+1) = \begin{cases} \tau_{k,q}(g), & \text{rand}(0,1) \leq CR \text{ or } q = \text{rand}(0,1) \\ \tau_{k,q}(g), & \text{rand}(0,1) \geq CR \text{ or } q \neq \text{rand}(0,1) \end{cases} \quad (10)$$

where CR is the crossover probability within $[0, 1]$, and $\text{rand}(0, 1)$ is a random number, which is uniformly distributed within $[0, 1]$ and used to guarantee that at least one-dimensional component comes from the target vector Z_k .

4.2.4 Selection

Selection operation compares the vector $\mu_k(g+1)$ and the vector $z_k(g)$ by an evaluation function, which is given by Eq. (11).

$$z_{k,q}(g+1) = \begin{cases} \mu_k(g+1), & f[\mu_k(g+1)] < f[z_k(g)] \\ z_k(g), & f[\mu_k(g+1)] \geq f[z_k(g)] \end{cases} \quad (11)$$

Therefore, this mechanism allows the populations of the offspring to evolve based on the current population. This optimization mechanism can improve the average optimization ability of the population and converge the optimal solution.

Algorithm 1. RBM_SVR_DEGWO

Input: $D_{train} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_{m'}, y_{m'})\}$, $D_{test} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_{n'})\}$, $S_{degwo.dbn}$;

Output: $[r_{test}, Z_{parent.\alpha}]$

Initialize a , A , C , Z_{parent} and objective function V_{parent}

for each individual wolf k **do**

$V_{parent} \leftarrow \text{RBM_SVR}(D_{train}, D_{test}, Z_{parent})$

end for

sort V_{parent}

compute top three gray wolf individuals $\{X_\alpha, X_\beta, X_\delta\}$

for each generation g **do**

update $a \leftarrow 2 - g \cdot (2/g_{max})$

for each individual wolf k **do**

$Z_{parent} \leftarrow X_p$

$V_{parent} \leftarrow \text{RBM_SVR}(D_{train}, D_{test}, Z_{parent})$

compute mutant individuals $\tau_{k,q} \leftarrow z_{p_1} + F \cdot (z_{p_2} - z_{p_3})$

compute children population $Z_{child} \leftarrow \mu_{k,q}$

$V_{child} \leftarrow \text{RBM_SVR}(D_{train}, D_{test}, Z_{child})$

end for

for each individual wolf k **do**

update Z_{parent} and V_{parent}

end for

end for

update the parameters in DBN $S \leftarrow Z_{parent.\alpha}$

$\text{RBM_SVR}(D_{train}, D_{test}, S)$

return $[r_{test}, Z_{parent.\alpha}]$

Table 2.

The procedure of RBM_SVR_DEGWO algorithm.

4.3 DEGWO algorithm

In the DEGWO algorithm, $S_{degwo.dbn} = (NP, g_{\max}, CR, D, ub, lb, F)$ where NP denotes population size, g_{\max} denotes the maximum number of iterations, ub and lb are the search range. r_{test} and r_{train} denote the error in test and learning procedure respectively. **Table 2** is the specific procedure employing the DE and the GWO algorithms to optimize parameters c_s and γ_s in the RBM-SVR deep learning model.

5. A hybrid model based on RBM SVR DEGWO and XGBoost

5.1 XGBoost principle

5.1.1 Model function form

This paper assumes that the data set as the input sample is $D = \{(X_i, z_i)\}$ and the XGBoost model is a linear model (logistic regression, linear regression). The linear model is an additive model. The number of learning trees is n [34]. The XGBoost model uses a pattern of linear superposition to calculate the predicted value, as shown in Eq. (12).

$$\hat{z}_i = \sum_n^N h_n(x_j) \quad (12)$$

Here, x_i is a feature vector and i is the number of data points. $h_n(x_i)$ is the regression tree function. $h_n \in H$, H is the set space of the regression trees.

$$H = \{h_n(z) = \alpha_{f(x)}\} \quad (13)$$

In Eq. (13), $f: R^m \rightarrow T$, $f(X)$ indicates that sample X is classified on a leaf node. T represents the number of leaf nodes of the tree. α is the score of the leaf node. $\alpha_{f(x)}$ represents the predicted value of the regression tree for the sample.

5.1.2 XGBoost learning objective function

The objective function based on the parameter space is shown in the following Eq. (14).

$$\kappa(\phi) = L(\phi) + \Omega(\phi) = \sum_{i=1}^I l(z_j, \hat{z}_i) + \sum_n^N \Omega(h_n) \quad (14)$$

where $\Omega(\phi)$ is a regularization term, indicating a penalty value for the complexity of the model. The regular term $\Omega(\phi)$ in the linear model includes: the regular term L_1 , $\Omega(\alpha) = \lambda \|\alpha\|_1$, and the regular L_2 , $\Omega(\alpha) = \lambda \|\alpha\|^2$. $L(\phi)$ is an error function that measures the fitting accuracy of the model. A can reduce model bias, such as square loss, exponential loss. Compared to GBDT, XGBoost adds a regular term to the objective function. XGBoost punishes the complexity of each regression tree and avoids overfitting during learning. XGBoost measures the complexity of the tree such as the number of internal nodes, the depth of the tree, the number of leaf nodes T , the leaf node score α , etc. XGBoost uses the regular term as shown in Eq. (15).

$$\Omega(h) = \gamma T + \frac{1}{2} \lambda \|\alpha\|^2 \quad (15)$$

5.1.3 Model optimization

In the model parameter optimization process, each iteration model is always added a new function on the optimal model obtained from the previous training. After the k th iteration, the prediction of the model is equal to the prediction function of the first $k - 1$ th model prediction function combined with the k th tree, as shown by Eq. (16).

$$\hat{z}_i^{(k)} = \sum_{n=1}^I h_n(x_j) = \hat{z}_i^{(k-1)} + h_i(x_j) \quad (16)$$

The objective function can be rewritten to Eq. (17).

$$\kappa(k) = \hat{z}_i^{(k)} = \sum_{i=1}^I l\left(z_i, \hat{z}_i^{(k-1)} + h_i(X_i)\right) + \Omega(h_i) \quad (17)$$

In formula (17), the model's goal is to learn the function of the k th tree. When the error function is replaced by a second-order Taylor expansion, the objective function can be rewritten as Eq. (18).

$$L^{(k)} = \sum_{i=1}^I \left[l\left(z_i, \hat{z}_i^{(k-1)} + v_i h_i(X_i) + \frac{1}{2} g_i h_k^2(X_i)\right) \right] + \Omega(h_k) \quad (18)$$

When $v_i = \partial_{\hat{z}_i^{(k-1)}} l\left(z_i, \hat{z}_i^{(k-1)}\right)$ and $g_i = \partial_{\hat{z}_i^{(k-1)}}^2 l\left(z_i, \hat{z}_i^{(k-1)}\right)$, the objective function is Eq. (19).

$$\tilde{L}^{(k)} = \sum_{i=1}^I \left[v_i h_i(X_i) + \frac{1}{2} g_i h_k^2(X_i) \right] + \Omega(h_k) \quad (19)$$

This objective function solves regression, classification, and sorting problems. Eqs. (20) and (21) are in the form of a tree structure of the regression tree function and the regular term. The objective function can be updated to Eq. (22).

$$h(X) = \alpha_{f(x)} \quad (20)$$

$$\Omega(h) = \gamma T + \frac{1}{2} \lambda \|\alpha\|^2 \quad (21)$$

$$\tilde{L}^{(k)} = \sum_{i=1}^I \left[v_i \alpha_{f(x_i)} + \frac{1}{2} g_i h_k^2 \alpha_{f(x)}^2 \right] + \gamma T + \lambda \frac{1}{2} \sum_{j=1}^T \alpha_j^2 \quad (22)$$

This article defines the sample set on each leaf node as $J_j = \{i | f(x_i) = j\}$. The objective function based on the form of leaf node accumulation is Eq. (23).

$$\begin{aligned} \tilde{L}^{(k)} &= \sum_{j=1}^T \left[\left(\sum_{i \in G_j} f_i \right) \alpha_j + \frac{1}{2} \left(\sum_{i \in G_j} g_i + \lambda \right) \alpha_j^2 \right] + \gamma T \\ &= \sum_{j=1}^T \left[\delta_j \alpha_j + \frac{1}{2} (\eta_j + \lambda) \alpha_j^2 \right] + \gamma T \end{aligned} \quad (23)$$

This paper assumes that the structure of the tree is a certain value (i.e., $f(x_i)$ is determined). To solve the problem of minimizing the objective function, we can make the derivative of the objective function zero. The optimal predicted score for each leaf node is Eq. (24). The formula for the minimum loss function is Eq. (25), which can be thought of as a function that scores the tree structure. The tree structure is gradually optimized as the score is reduced.

$$A_j^* = -\frac{\delta_j}{\eta_j + \lambda} \quad (24)$$

$$\tilde{L}^{(*)} = -\sum_{j=1}^T \frac{\delta_j^2}{\eta_j + \lambda} + \gamma T = -\frac{1}{2} \sum_{j=1}^T \left(\left(\sum_{i \in G_j} f_i \right)^2 / \sum_{i \in G_j} g_i + \lambda \right) + \gamma T \quad (25)$$

5.1.4 Structure score

Eq. (25) is a function for scoring a tree structure, called structure score. The smaller the score, the better the tree structure is. The algorithm searches for the optimal tree structure by using Eq. (25). \tilde{L}^* represents the contribution of the leaf node to the overall loss. The goal of the algorithm is to minimize the loss, so the larger part of $\frac{\delta_j^2}{\eta_j + \lambda}$ could be as good as possible. This article expands a leaf node and defines the gain as shown in Eq. (26).

$$gain = \frac{1}{2} \left[\frac{\delta_j^2}{\eta_L + \lambda} + \frac{\delta_R^2}{\eta_R + \lambda} - \frac{(\delta_L + \delta_R)_j^2}{\eta_L + \eta_R + \lambda} \right] - \gamma \quad (26)$$

In Eq. (26), $\frac{\delta_j^2}{\eta_L + \lambda}$ is the score of the left subtree, $\frac{\delta_R^2}{\eta_R + \lambda}$ is the score of the right subtree, $\frac{(\delta_L + \delta_R)_j^2}{\eta_L + \eta_R + \lambda}$ is the score without division, and γ is the cost of the complexity after introducing the new leaf node. The larger the value of $gain$, the more loss after splitting is reduced. Therefore, when segmenting a leaf node, we calculate the $gain$ corresponding to all candidate features and select the segment with the largest $gain$.

5.1.5 Best branch

The core part of the XGBoost algorithm is to obtain the optimal node based on the maximum gain obtained. XGBoost looks for the best branch using a greedy algorithm. The greedy algorithm traverses all possible segmentation points of all features, calculating the Gain value and selecting the maximum value to complete the segmentation. The greedy algorithm is an algorithm that controls the local optimum to achieve global optimization. The decision tree algorithm can also be considered as a method of greedy algorithm. XGBoost is an integrated model of the tree. If each leaf is optimal, the overall generated tree structure is optimal. This avoids enumerating all possible tree structures. XGBoost uses the objective function to measure the structure of the tree, and then let the tree grow from depth 0. Each time a branch calculation is implemented, XGBoost calculates the reduction in the objective function. When the reduction is below a certain value, the tree will stop growing.

5.2 Hybrid model based on RBM_SVR_DEGWO and XGBoost

After the boosting tree is created, the XGBoost algorithm extracts the importance score for each attribute. The XGBoost importance score measures the value of

Algorithm 2. Hybrid Algorithm based on RBM_SVR_DEGWO and XGBoost**Input:** $D = \{(X_i, z_i)\}, J_j = \{i | f(x_i) = j\}$

data sets normalization

Initialize $gain$ Initialize δ_L, η_L **for** each k **do** $\delta_L \leftarrow 0, \eta_L \leftarrow 0$ **for** j in sorted $(J, \text{by } X_{jk})$ **do** $\eta_L \leftarrow \eta_L + g_j, \delta_L \leftarrow \delta_L + f_j$ $\eta_R \leftarrow \eta - \eta_L, \delta_R \leftarrow \delta - \delta_L$ $score \leftarrow \max(score)$ **end for****end for**

Split with max score

 $y_d = \text{RBM.SVR.DEGWO}(X_i)$ $y_e = \text{EXGBOOST}(X_i)$ $y_{\text{hybrid}} = E[y_d, y_e]$ **Output:** y_{hybrid} **Table 3.***Hybrid algorithm based on RBM_SVR_DEGWO and XGBoost.*

features in improving decision tree construction. The more an attribute is used to build a decision tree, the more important it is [35]. In order to further improve the accuracy of prediction and analyze the importance of feature quantity, this paper uses XGBoost to extract the feature quantity importance score. By combining the proposed RBM_SVR_DEGWO model prediction value, this paper proposes a hybrid prediction model, as shown in **Table 3**.

6. Experimental description and result analysis

This paper analyzes 2,468,059 trajectory data from Mobike's shared bikes. The data covers more than 300,000 users and 400,000 shared bikes. The data of each rental trip includes the start time, the end time, the Geohash code of the starting position, the Geohash code of the ending position, the bicycle ID and the user ID.

GeoHash is an algorithm for spatial indexing. In the GeoHash theory, the Earth is considered to be a two-dimensional plane that can be divided into multiple sub-regions. The latitude and longitude inside the sub-area will correspond to the same code. GeoHash-based spatial indexing can improve the efficiency of spatial data for latitude and longitude retrieval. In this paper, GeoHash encodes a square plane separated by a square of latitude and longitude of 0.001373. To improve the prediction accuracy, this paper combines nine adjacent areas into a square area with a length of 411.9873 meters. This paper divides Beijing into 10×10 traffic zones and numbers them from 1 to 100. Various indicators of the traffic area will be used as input data for the prediction model, as shown in **Table 4**.

The output of the model is the daily transfer probability of traffic flow among the traffic zones p_{IJ}^t , which is given by Eq. (27). In the cities of N interconnected traffic areas, p_{IJ}^t indicates the transfer probability of the traffic flow with the original point I and the destination J in day t .

$$p_{IJ}^t = \frac{d_{IJ}^t}{\sum_{J=1}^N d_{IJ}^t} \quad (27)$$

Number	Variable name
1	Zone number of origin traffic zone
2	Zone number of destination traffic zone
3	Longitude of the origin point
4	Latitude of the origin point
5	Longitude of the destination point
6	Latitude of the destination point
7	Distance between the center points of the traffic area
8	Absolute value of the difference in the numbers of traffic zone
9	Number of the day

Table 4.
Input variables and interpretation.

Test group	The first day of training data		The 2nd day of training data		Prediction data	
	Date	Data amount (trajectories)	Date	Data amount (trajectories)	date	Data amount (trajectories)
1	2017/5/10	262,569	2017/5/11	272,210	2017/5/12	265,173
2	2017/5/13	225,281	2017/5/14	236,594	2017/5/15	279,554
3	2017/5/14	236,594	2017/5/15	279,554	2017/5/16	288,719
4	2017/5/15	279,554	2017/5/16	288,719	2017/5/17	322,201
5	2017/5/16	288,719	2017/5/17	322,201	2017/5/18	315,758
6	2017/5/10	262,569	2017/5/15	279,554	2017/5/16	288,719

Table 5.
Experimental group training data and test data.

where $I = 1, 2, 3, \dots, N; J = 1, 2, 3, \dots, N; d_{IJ}^t$ refers to the traffic flow with the original point I and the destination J in day d . p_{IJ}^t represents the origin–destination (OD) probability distribution and reflects the distribution of demand in the city.

This paper builds a set of destinations that may correspond to the origin traffic zone of the test day. The calculated destination candidates can be used to predict the probability of the traffic flow among the traffic zones. In the experiment, we selected data of different adjacent days as 6 test groups (**Table 5**).

Based on data for the past 2 days as the training data, this paper predicts the subsequent third day of the transfer probabilities of bike-sharing traffic flow. **Figure 1** is the root mean square errors of a prediction result of transfer probabilities of bike-sharing traffic flow in Beijing based on the RBM_SVR_DEGWO algorithm.

Compared to the surrounding area, the central area of the city has higher shared bicycle usage and more bicycle trajectory data. Therefore, the Root Mean Square Error of the central region is smaller.

To illustrate the performance of the RBM_SVR_DEGWO algorithm, we calculated the predicted values of the SVR algorithm, the RBM_SVR algorithm, and the RBM_SVR_DEGWO algorithm based on the data from the experimental groups in **Table 5**. To ensure the fairness of the results, the data, network structure and parameter settings consistent. **Figure 2** shows the mean-square error bars of the predicted transfer probabilities of SVR, RBM_SVR, and RBM_SVR_DEGWO.

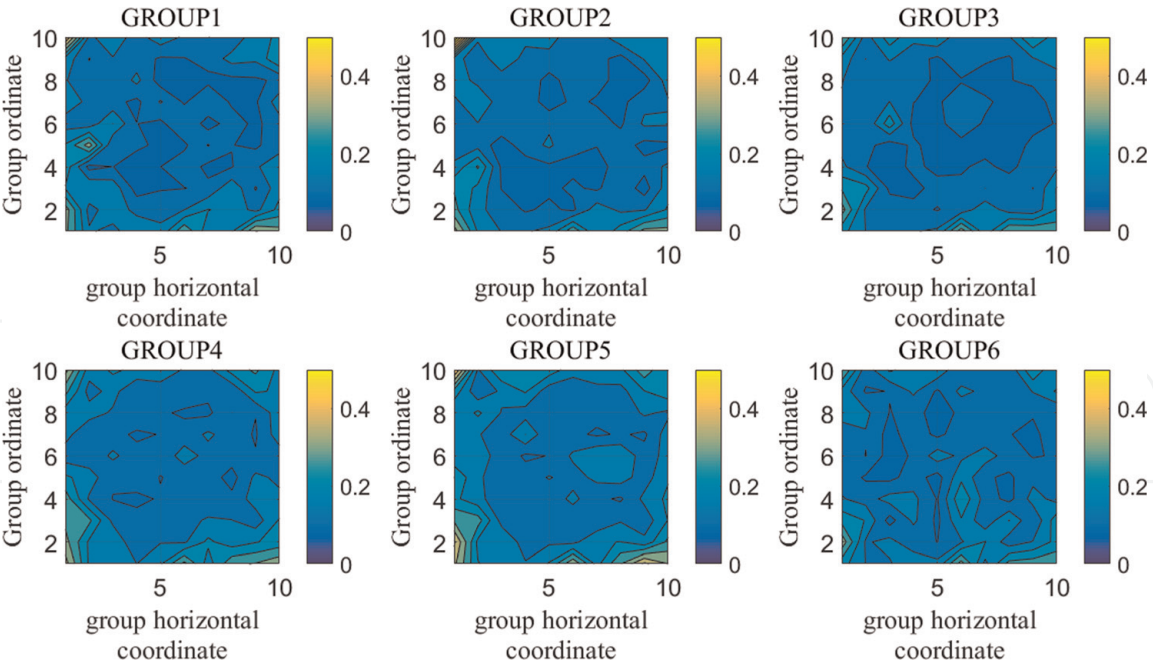


Figure 1.
The root-mean-square errors of the predicted transfer probabilities of bike sharing traffic flow.

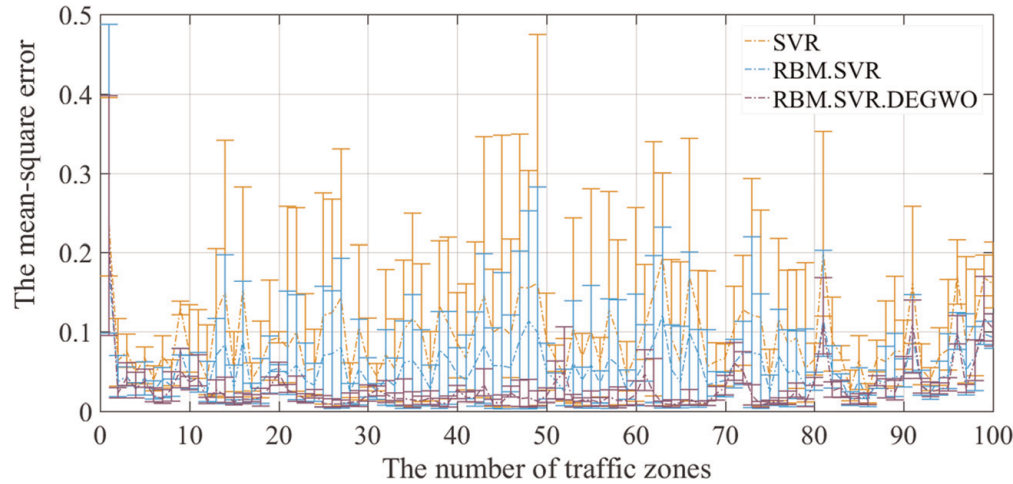


Figure 2.
The mean-square error bars of the predicted transfer probabilities of bike sharing traffic flow of six test groups for each type of comparison method.

The average values of the mean squared errors of the predicted values of the transfer probabilities of the algorithms SVR, RBM_SVR, and RBM_SVR_DEGWO are gradually reduced. The average mean square error of the SVR is 0.0916, the RBM_SVR is 0.0542, and the RBM_SVR_DEGWO is 0.0283. RBM improves the prediction accuracy of the model through the deep network structure. The DEGWO algorithm stabilizes the prediction value error to a lower value by optimizing the parameters of the RBM-SVR. Compared with SVR and RBM_SVR, RBM_SVR_DEGWO algorithm has better robustness.

According to the proposed hybrid algorithm of RBM_SVR_DEGWO and XGBoost, the value of transfer probabilities of bike-sharing traffic flow can be predicted. The data set for this experiment is from the grouped data of **Table 5**. The training data set, test data set, and feature variables are the same as those used in the previous experiments. **Table 6** is the parameters and explanation of the XGBoost model.

The root mean square error of the predicted values of the RBM_SVR_DEGWO algorithm, the XGBoost algorithm, and the hybrid algorithm is shown in **Figure 3**. In the six experimental groups, the mean, variance, kurtosis, maximum, minimum, and range of the predicted root mean square error of the RBM_SVR_DEGWO algorithm, the XGBoost algorithm, and the hybrid algorithm are shown in **Figure 3**.

The statistical characteristics of the proposed root mean square error of the algorithms are shown in **Figure 4**. The root-mean-square error of the predicted value of the mixed algorithm has a high kurtosis value. It indicates that the variance increases of root mean square error is caused by the extreme difference of low frequency greater than or less than the mean value. The plots of the minimum and variance indicate that RBM_SVR_DEGWO can achieve higher prediction accuracy than XGBoost. XGBoost is more stable than RBM_SVR_DEGWO in the prediction process. In the six experimental groups, compared with the RBM_SVR_DEGWO algorithm and the XGBoost algorithm, the mean, variance, maximum, minimum, and range of the root mean square error of the predicted value of the hybrid

Parameter	Value	Interpretation
early_stopping_rounds	200	If the loss function does not decrease after the model has been added to n trees continuously, the model will stop adding trees.
eval_metric	linear	Evaluation index
eta	0.3	The shrinking step size used in the update process. Eta is used to prevent overfitting.
min_child_weight	1	Min_child_weight refers to the sum of the weights of the smallest leaf nodes. If the sample weight of a leaf node is less than min_child_weight then the splitting process ends.
max_depth	6	Maximum depth of the tree
lambda	1	Penalty factor for L2 regular terms
alpha	0	Penalty factor for L1 regular terms
objective	linear	Loss function

Table 6.
Parameters and explanations of the XGBoost model.

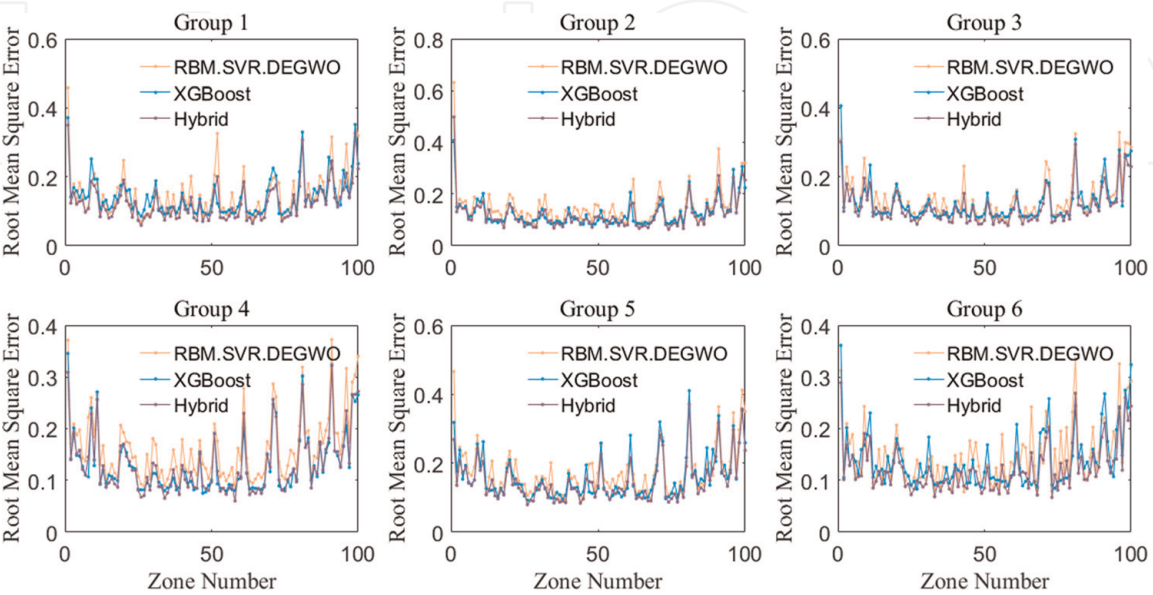


Figure 3.
The root mean square error of the predicted values of the RBM_SVR_DEGWO algorithm, the XGBoost algorithm, and the hybrid algorithm.

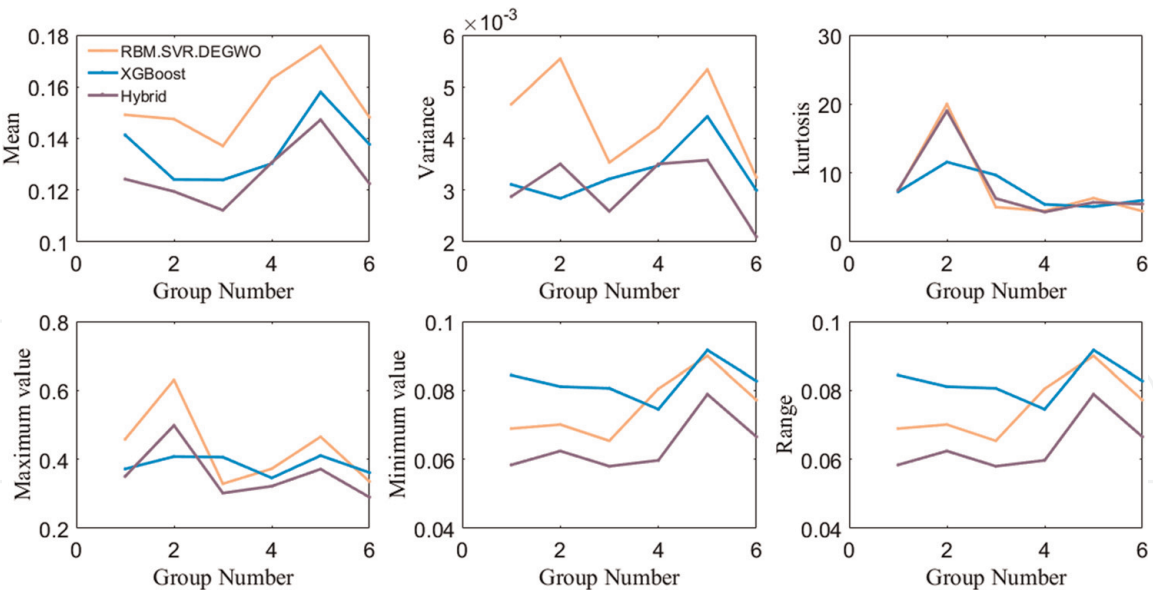


Figure 4.
Statistical characteristics of the root mean square error.

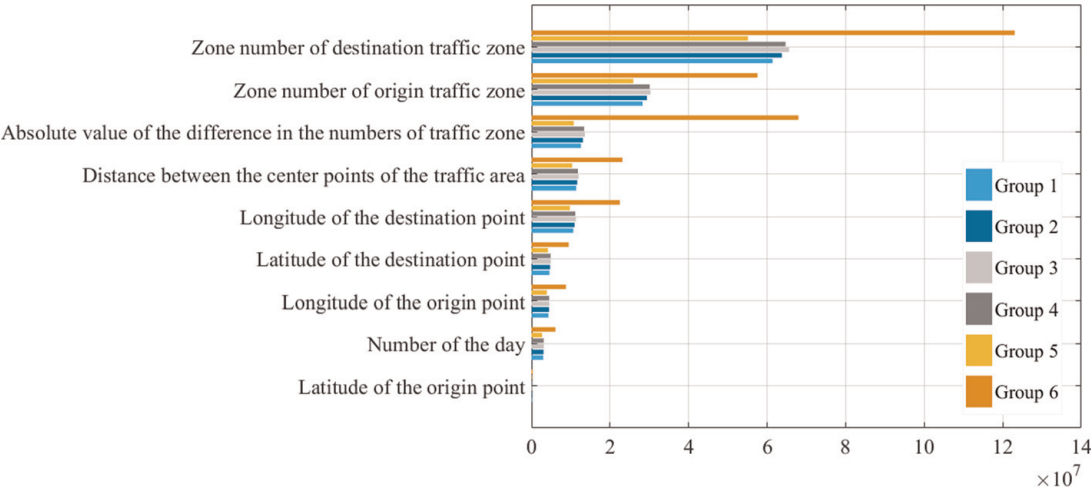


Figure 5.
Importance analysis of characteristic variables.

algorithm are lower. Therefore, by combining the prediction results of the RBM_SVR_DEGW0 algorithm and the XGBoost algorithm, the hybrid algorithm improves the prediction accuracy and obtains a lower root mean square error of the predicted value. XGBoost scores the importance of each feature based on the number of times the feature is used to segment the sample in all trees and the average gain in all trees. In the six experimental groups, the ranking of each input feature variable is as shown in **Figure 5**.

The main factors affecting the transfer probabilities of bike-sharing traffic flow are the destination traffic zone number, the origin traffic zone number, and the absolute value of the difference between the numbers of traffic zone. It shows that the shared bike rider's choice of destination is usually affected by the starting point, the end position and the distance of the journey. The shared bicycle service is suitable for short trips. Travel destinations for shared bike riders are usually nearby business and lifestyle centers, and bus stops. The dates of the data for the six groups of experiments are within weekdays. On a normal weekday, for the riders in the same community, the main travel destinations are somewhat similar and fixed. Therefore, information such as the cell number of the origin and destination becomes a key factor for predicting the probability of travel destination.

7. Conclusions

The principal objective of this study is to predict the traffic flow transfer probability of shared bicycle by proposing a hybrid deep learning algorithm and accurately reflect the transfer probability of the user's OD demand. First, this paper constructs a deep-structured RBM model and connects it to the SVR model for predicting continuous probability values. Furthermore, we utilize the DEGWO optimization algorithm, named, to optimize the parameters c_s and γ_s in the stacked RBM-SVR algorithm. XGBoost improves the prediction accuracy and analyzes the importance of the feature variables in the input data.


Based on the comparison results, it demonstrates that the proposed hybrid algorithm outperformed the XGBoost model and RBM_SVR_DEGWO model. The XGBoost algorithm improves the stability of the prediction process and reduces the error of the RBM_SVR_DEGWO algorithm at extreme points. The deep-structured RBM algorithm simulates the probability distribution that best produces the training samples. In the case of massive training data, RBM improves the efficiency of algorithm calculation utilizing Gibbs sampling of small-batch data. In the DEGWO algorithm, the GWO algorithm guarantees the global search capability, and the DE algorithm avoids the fall into a local optimal through the mutant individual, cross-over, and selection operations.

Author details

Wenwen Tu
Southwest Jiaotong University, Chengdu, China

*Address all correspondence to: tuvivimic@gmail.com

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Vogel P, Greiser T, Mattfeld DC. understanding bike-sharing systems using data mining: Exploring activity patterns. *Procedia-Social and Behavioral Sciences*. 2011;**20**:514-523
- [2] Wu Y, Zhu D. Bicycle sharing based on PSS-EPR coupling model: Exemplified by bicycle sharing in China. *Procedia CIRP*. 2017;**64**:423-428
- [3] Chemla D, Meunier F, Calvo RW. Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*. 2013;**10**(2):120-146
- [4] Contardo C, Morency C, Rousseau L-M. Balancing a Dynamic Public Bike-Sharing System. Montreal: Cirrelt; 2012
- [5] Schuijbroek J, Hampshire RC, Van Hove W-J. Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research*. 2017;**257**(3):992-1004
- [6] Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy Layer-Wise Training of Deep Networks. *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press; 2007. pp. 153-160
- [7] Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. *Neural Computation*. 2006;**18**(7):1527-1554. DOI: 10.1162/neco.2006.18.7.1527
- [8] Hinton GE, Sejnowski TJ. Learning and Relearning in Boltzmann Machines, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge: MIT Press; 1986. pp. 282-317
- [9] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;**521**(7553):436. DOI: 10.1038/nature14539
- [10] LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, et al. Backpropagation applied to handwritten zip code recognition. *Neural Computation*. 1989;**1**(4):541-551. DOI: 10.1162/neco.1989.1.4.541
- [11] Hinton GE. A Practical Guide to Training Restricted Boltzmann Machines, *Neural Networks: Tricks of the Trade*. NY, USA: Springer; 2012. pp. 599-619
- [12] Larochelle H, Bengio Y. Classification using discriminative restricted Boltzmann machines. In: *Proceedings of the 25th International Conference on Machine learning*. NY, USA: ACM; 2008. pp. 536-543
- [13] Le Roux N, Bengio Y. Representational power of restricted Boltzmann machines and deep belief networks. *Neural Computation*. 2008; **20**(6):1631-1649
- [14] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning*. ICML-10. 2010. pp. 807-814
- [15] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering. In: *Proceedings of the 24th International Conference on Machine Learning*. NY, USA: ACM; 2007. pp. 791-798
- [16] Awad M, Khanna R. *Support Vector Regression*. Berkeley, CA: Apress; 2015
- [17] Drucker H, Burges CJC, Kaufman L, Smola AJ, Vapnik V. Support vector regression machines. *Advances in Neural Information Processing Systems*. 1997;**28**(7):779-784
- [18] Li L, Duan Y. A GA-based feature selection and parameters optimization for support vector regression. In: *International Conference on Natural Computation*. 2011. pp. 335-339

- [19] Wu CH, Wei CC, Su DC, Chang MH. Travel time prediction with support vector regression. In: Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems. Vol. 1432. 2004. pp. 1438-1442
- [20] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*. 2014;**69**:46-61. DOI: <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [21] Storn R, Price K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. 1997;**11**(4):341-359
- [22] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining. NY, USA: ACM; 2016. pp. 785-794
- [23] Chen T, He T, Benesty M, Khotilovich V, Tang Y. Xgboost: extreme gradient boosting. R package version 0.4–2. 2015. pp. 1-4
- [24] Tu W, Liu H. Transfer probability prediction for traffic flow with bike sharing data: A deep learning approach. In: Science and Information Conference. NY, USA: Springer; 2019. pp. 71-85
- [25] Li X, Zhang Y, Sun L, Liu Q. Free-floating bike sharing in Jiangsu: Users' behaviors and influencing factors. *Energies*. 2018;**11**(7):1664
- [26] Shen Y, Zhang X, Zhao J. Understanding the usage of dockless bike sharing in Singapore. *International Journal of Sustainable Transportation*. 2018;**12**(9):686-700
- [27] Xu C, Ji J, Liu P. The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets. *Transportation Research Part C: Emerging Technologies*. 2018;**95**:47-60
- [28] Zhang Y, Lin D, Mi Z. Electric fence planning for dockless bike-sharing services. *Journal of Cleaner Production*. 2019;**206**:383-393
- [29] Caggiani L, Camporeale R, Ottomanelli M, Szeto WY. A modeling framework for the dynamic management of free-floating bike-sharing systems. *Transportation Research Part C: Emerging Technologies*. 2018;**87**:159-182
- [30] Li M, Wang X, Zhang X, Yun L, Yuan Y. A multiperiodic optimization formulation for the operation planning of free-floating shared bike in China. *Mathematical Problems in Engineering*. 2018;**2018**:1-11
- [31] Liu Y, Szeto W, Ho SC. A static free-floating bike repositioning problem with multiple heterogeneous vehicles, multiple depots, and multiple visits. *Transportation Research Part C: Emerging Technologies*. 2018;**92**:208-242
- [32] Shim VA, Tan KC, Cheong CY, Chia JY. Enhancing the scalability of multi-objective optimization via restricted Boltzmann machine-based estimation of distribution algorithm. *Information Sciences*. 2013;**248**:191-213
- [33] Wen-juan X, Jian-feng L. Application of vision sensing technology in urban intelligent traffic control system. In: 2018 4th International Conference on Computer and Technology Applications (ICCTA). NY, USA: IEEE; 2018. pp. 74-77
- [34] Zheng H, Yuan J, Chen L. Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation. *Energies*. 2017;**10**(8):1168
- [35] Bai S, Li M, Kong R, Han S, Li H, Qin L. Data mining approach to construction productivity prediction for cutter suction dredgers. *Automation in Construction*. 2019;**105**:102833