ISSN: 2664-9640

An Improved Traffic Prediction Model for Communication Base stations

Shaobo Ding*, Yueru Wang, Yiman Wu, Long Zhang

Harbin University of Science and Technology, Harbin, Heilongjiang, 150080, China

Abstract

In this paper, we use the improved grey wolf algorithm to optimize support vector machine regression to improve the traffic prediction accuracy of communication base stations. Firstly， we use the advantages of Cat mapping and backward learning theory, particle swarm algorithms, nonlinear control parameter balancing algorithms, and Lévy flight theory to synthesize and optimize the Gray Wolf algorithm. Then the penalty factor and kernel function parameters of support vector machine regression are optimized with the improved gray wolf algorithm to build a traffic prediction model based on communication base stations. Finally, simulation comparisons are performed, and the results show that the improved gray wolf algorithm optimized support vector machine regression prediction model established in this paper has the highest prediction accuracy compared to the standard gray wolf algorithm optimized support vector machine regression prediction model, simulated annealing algorithm optimized BP neural network prediction model, and particle swarm algorithm optimized BP neural network prediction model.

Keywords

GWO; PSO; Lévy flight theory; Cat mapping; backward learning.

1. Introduction

To address the current problem of large traffic prediction errors of communication base stations, this paper designs the Improved Gray Wolf Algorithm (LIGWO) optimized Support Vector Regression (SVR) communication base station traffic prediction model, hereafter referred to as LIGWO_SVR. Firstly, the initialization method using a combination of Cat chaos mapping and backward learning replaces the standard gray wolf algorithm using random numbers to initialize the population, which lays the foundation for population diversity in the global search process of the algorithm; secondly, the individual position update idea of the particle swarm algorithm is used to improve the gray wolf position update formula and reduce the risk of the algorithm falling into local optimum; thirdly, the nonlinear control parameters are used to replace the standard gray wolf algorithm's linear control parameters, so that the convergence factor decreases slowly in the early stage and rapidly in the late stage, thus improving the global search ability in the early stage of the algorithm and the local exploitation ability in the late stage of the algorithm; fourthly, when α wolf position is updated, the Levy flight theory is used to conduct global search for α wolves to prevent the wolf population from losing diversity and reduce the risk of premature convergence in the late stage of the algorithm; fifthly, LIGWO is used to optimize penalty factor and kernel function parameters of SVR. Simulation experiments were conducted using traffic data from two communication base stations in Harbin, and the results proved that the LIGWO_SVR traffic prediction model has higher prediction accuracy in base station traffic prediction compared with the standard gray wolf algorithm optimized support vector machine regression prediction model (GWO_SVR), simulated annealing algorithm optimized BP neural network prediction model (SA_BP), and particle swarm algorithm optimized BP neural network prediction model (PSO_BP).

2. Standard Gray Wolf Algorithm(GWO)

In the standard gray wolf algorithm, the population is divided into 4 orders α, β, δ, and ω from high to low. In the D-dimensional search space, assume that n gray wolves form a population $X=(X_1,X_2,\cdots,X_n)$. Define the position of the X_i^d gray wolf as $X_i=X_i^1,X_i^2,\cdots,X_i^D$, where $d<D$, X_i^d denotes the position of the ith gray wolf in the dth dimension. In the process of searching for the prey, the gray wolf gradually approaches and surrounds the prey, and the mathematical model shown in Equation (1) is satisfied for the dth dimensional position of the ith gray wolf.

$$X_i^d(t+1)=X_p^d(t)-A_i^d\,|C_i^d X_p^d(t)-X_i^d\,| \tag{1}$$

Where t denotes the number of current iterations, $X_p=(X_p^1,X_p^2,\cdots,X_p^D)$ indicates the location of the prey, $A_i^d\,|C_i^d X_p^d(t)-X_i^d(t)|$ is the encircling step. Among them A_i^d is convergence factor, C_i^d is oscillation factor. They are defined as shown in equations (2) and (3).

$$A_i^d=2a\times rand_1-a \tag{2}$$

$$C_i^d=2\times rand_2 \tag{3}$$

Where $rand_1$ and $rand_2$ are random variables between [0 1]. a is a linear control parameter whose value decreases linearly from 2 to 0 as the number of iterations increases, with the expression shown in equation (4)

$$a=2-\frac{t}{t_{\max}} \tag{4}$$

Where, $t_{\max}$ is the maximum number of population iterations.

During the hunting process, when the gray wolf population searches for the prey position, α-wolves, β-wolves, and δ-wolves are closest to the prey position, so the gray wolf population can calculate the position of the gray wolf moving toward the prey based on the positions X_α、X_β, X_δ of α wolf, β wolf, and δ wolf with the following equations.

The direction of movement of the remaining gray wolves is first calculated according to equations (5), (6) and (7).

$$X_{i,\alpha}^d(t+1)=X_\alpha^d(t)-A_{i,1}^d\,|C_{i,1}^d X_\alpha^d(t)-X_i^d(t)| \tag{5}$$

$$X_{i,\beta}^d(t+1)=X_\beta^d(t)-A_{i,2}^d\,|C_{i,2}^d X_\beta^d(t)-X_i^d(t)| \tag{6}$$

$$X_{i,\delta}^d(t+1)=X_\delta^d(t)-A_{i,3}^d\,|C_{i,3}^d X_\delta^d(t)-X_i^d(t)| \tag{7}$$

Then the position of the gray wolf is updated by equation (8).

$$X_i^d(t+1)=\frac{1}{3}\sum_{j=\alpha,\beta,\delta}X_{i,j}^d(t+1) \tag{8}$$

3. Improved gray wolf algorithm

3.1. A Population Initialization Strategy Combining Cat Chaos Mapping and Backward Learning

To ensure the diversity of the gray wolf population, we combine the chaotic mapping method and the backward learning method to propose the strategy of chaotic backward learning to initialize the population. Compared with most scholars at home and abroad who use a sequence of logistic-based mappings combined with intelligent optimization algorithms, this paper uses

Cat chaotic mappings with more uniform mappings. The Cat mapping is a two-dimensional reversible chaotic mapping with the kinetic equations shown in equation (9) [1].

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod 1 \tag{9}$$

The distribution of the chaotic sequences generated by the Cat chaotic mapping between [0, 1] is uniform, and the distribution of 2500 iterations is shown in Fig. 1 and Fig. 2.

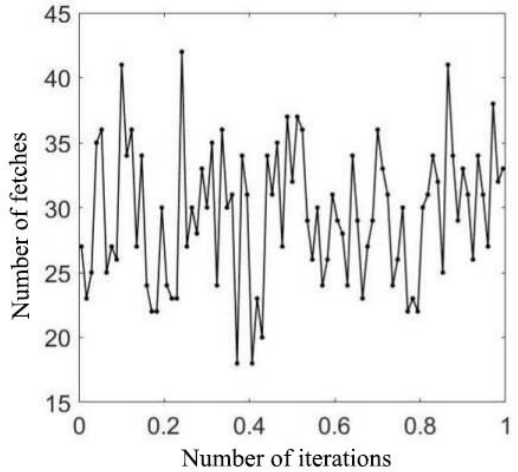


Figure 1: Number of times Cat chaotic variables take values

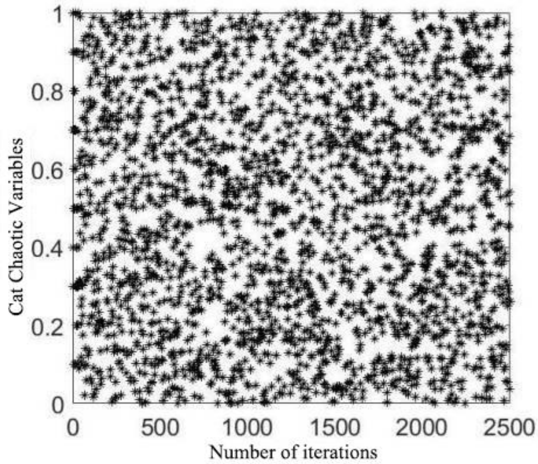


Figure 2: Cat chaotic variable taking values

The specific steps for initializing the population based on the chaotic backward learning strategy are as follows.

(1) Generate N initial solutions between [0, 1] using Cat chaotic mapping;

(2) Using the reverse learning theory to generate the corresponding reverse solution for each initial solution, the calculation formula is shown in Equation (10).

$$OP_i = K(X_{\min}^d + X_{\max}^d) - X_i \tag{10}$$

Where K is a random variable between [0, 1]. OP_i is the inverse solution corresponding to each initial solution X_i. $X_{\min}^d$, $X_{\max}^d$ denote the minimum and maximum values of the dth dimension vector in all initial solutions, respectively.

(3) Combining the initial and inverse solutions;

(4) The first N solutions with small fitness values are selected as the initial solutions to form a good population by ranking them from smallest to largest.

ISSN: 2664-9640

3.2. Introduction of particle swarm individual position update strategy

During the search for prey by gray wolves, it can be seen from equations (5), (6), (7) and (8) that the positions of gray wolves are updated mainly according to the positions X_α、X_β、X_δ of α-wolves, β-wolves and δ-wolves, and the exchange of information between individual gray wolves and their own experience is ignored in the position update [2]. In this paper, inspired by the position update strategy in the particle swarm algorithm, the position information of individual gray wolves is introduced in the gray wolf position update, and the gray wolf position update formula of Eq. (8) is adjusted to obtain Eq. (11) [3]

$$X_i^d(t+1) = \frac{1}{3}\sum_{j=\alpha,\beta,\delta} X_{i,j}^d + v_i^d(t) \tag{11}$$

Among them, the formula for updating the position of gray wolf individuals is shown in equation (12).

$$v_i^d(t+1) = \omega \cdot (v_i^d(t) + G_1 \cdot rand_3 \cdot (X_{i,\alpha}^d - X_i^d(t)) + G_2 \cdot rand_4 \cdot (X_{i,\beta}^d - X_i^d(t)) + G_3 \cdot rand_5 \cdot (X_{i,\delta}^d - X_i^d(t)) \tag{12}$$

$$G = 2 \cdot rand_6 \tag{13}$$

Where w is a random variable between [0, 1]. It can be proved from the literature that the algorithm has better search performance when w takes values between [0.6, 1]; when w is larger, the algorithm has better global search capability, and when w is smaller, the algorithm has better local search capability. $rand_3$, $rand_4$, $rand_5$ and $rand_6$ are random variables between [0, 1], G_1, G_2, G_3 are calculated from equation (13). $X_{i,\beta}^d$ and $X_{i,\delta}^d$ can be derived from equations (5), (6) and (7) respectively. $X_i^d(t)$ represents the current position of the gray wolf.

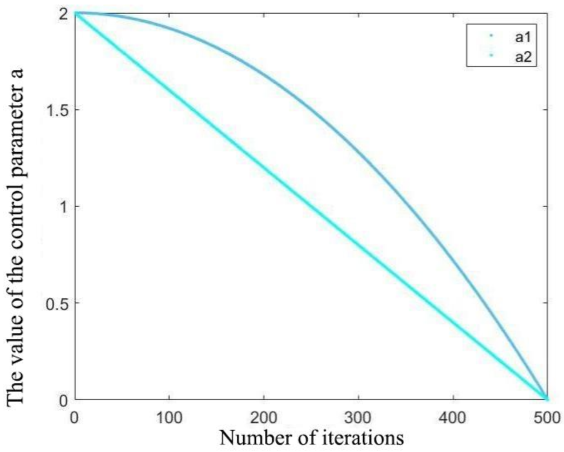
3.3. Nonlinear control parameter strategy

The gray wolf algorithm is mainly composed of two parts: α, β, and δ wolves locating the prey and the remaining gray wolf individuals moving toward the prey according to the positions of α, β, and δ wolves. Numerous literatures have shown that convergence factor A_i^d plays an important role in balancing the global and local search capabilities of the GWO algorithm. This paper draws on the nonlinear variation strategy proposed in the literature [4], and to distinguish it from the linear control parameter a, the nonlinear control parameter is denoted as a^*:

$$a^*(t) = a_i - (a_i - a_f \times (\frac{t}{t_{\max}})^2)) \tag{14}$$

Where, a_i and a_f are the initial and final values of the control parameters, respectively; t is the current number of iterations; $t_{\max}$ is the maximum number of iterations.

ISSN: 2664-9640



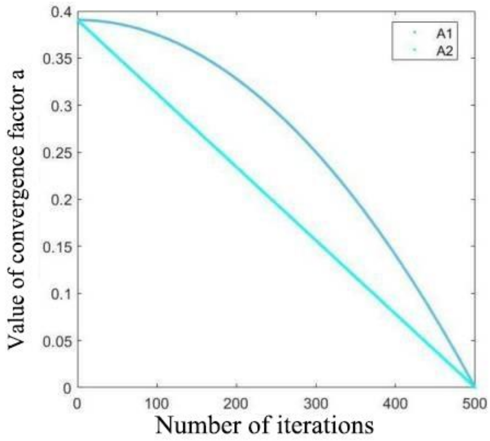


Figure 3 Figure 4

Figure 3: Variation curve of control parameter a

Figure 4: Dynamic change curve of convergence factor analysis

From Figure 4, we can see that the convergence factor of the nonlinear parameters A_i^d decreases slowly in the first period, which can increase the global search ability; in the later period the decreasing speed is fast, which can improve the local development ability.

3.4. Update α-wolves location with Lévy flight

In the gray wolf optimization algorithm, the position of the α-wolves represents the optimal problem. In the process of searching for prey, all individual gray wolves approach the optimal solution α-wolves, which can lead to the loss of diversity in the population and early convergence. [5] To address this drawback, this paper utilizes a strategy of position updating for α-wolves using Lévy flight. By introducing the Lévy flight, the formula for the new generation α-wolves position is shown in (15).

$$X_{i,\alpha}^d(t+1) = X_{i,\alpha}^d(t) - b \oplus Levy(\delta) \tag{15}$$

Where, $X_{i,\alpha}^d(t)$ denotes the position of α-wolves individuals at generation t. ⊕ represents point-to-point multiplication; b represents the random number of individual gray wolf locations, calculated from equation (16). $Levy(\delta)$ represents the random search path and is calculated by equation (17)

$$b = random(size(\alpha\_posion)) \tag{16}$$

$$Levy(\delta) \sim 0.01 \frac{u}{|v|^{\frac{1}{\delta}}} (X_{i,\alpha}^d(t) - X_{i,\alpha best}^d) \tag{17}$$

In equation (17), δ is generally taken as 1 < δ < 3. In the paper, δ = 1.5; $X_{i,\alpha best}^d$ denotes the location of the historical optimal α-wolves and v follows equation(18)、(19)

$$u \sim N(0, \sigma_u^2) \tag{18}$$

$$v \sim N(0, \sigma_v^2) \tag{19}$$

σ_u and σ_v are calculated by equations (20) and (21), respectively

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\frac{\pi\beta}{2})}{\Gamma|(\frac{1+\beta}{2})|\beta 2^{\frac{(\beta-1)}{2}}} \right\}^{\frac{1}{\beta}} \tag{20}$$

$$\sigma_v = 1 \tag{21}$$

3.5. Steps of the improved gray wolf algorithm

In summary, the steps of the improved gray wolf optimization algorithm are as follows.

***Step*1:** Set the population size N, the maximum number of iterations $t_{\max}$, the nonlinear adjustment parameter k, the individual learning factors G1, G2, G3, the initial and initial values of the nonlinear control parameter a_i , the final value a_f , and the Lévy flight parameter delta;

***Step*2:** Initialize the population X_i using the chaotic backward learning strategy described in Section 3.1， $i = 1, 2, \cdots, N$;

***Step*3:** The fitness values of individual gray wolves in the population are calculated and sorted from smallest to largest, and the position of the individual gray wolf with the smallest fitness value is taken as the historical optimal solution X_α , the position of the individual gray wolf with the second smallest fitness value is taken as the second optimal solution X_β , and the position of the individual gray wolf with the third smallest fitness value is taken as the third optimal solution X_δ

***Step*4:** Update the location of α wolves by conducting a global search for α-wolves using Eqs. (15) ~ (21) of Section 3.4.

***Step*5:** Compute the nonlinear control parameter a*using equation (14) of Section 3.3 and update the convergence factor according to equation (2).

***Step*6:** Calculating the direction of gray wolf movement according to equations (5), (6) and (7) for individual gray wolves other than the α, β and δ wolves identified in Step3.

***Step*7:** Update the location of individual gray wolves using Equation (11) in Section 3.2.

***Step*8:** Judge whether the current number of iterations of the algorithm reaches the maximum number of iterations $t_{\max}$, if it is satisfied, output the fitness value of α wolf, otherwise return to execute Step3 to recalculate the fitness value and update $X_\alpha, X_\beta, X_\delta$.

3.6. Support vector regression

SVR converts a regression problem into a quadratic programming problem, suitable for handling prediction problems with small sample sizes [6]. SVR uses the nonlinear mapping function $\Phi(x_i)$ to map the input sample x_i into the high-dimensional space *D*, and creates the feature function in *D* as shown in equation (22).

$$f(x) = h\Phi(x) + b \tag{22}$$

Where *h* is a vector of weights and $h \in D$; *b* is the deviation value *and* $b \in R$.

For the regression fitting problem, an insensitive loss function ω and positive and negative relaxations λ, λ^* are introduced with the constraints shown in equation (23).

$$\begin{cases} \min\left\{\frac{1}{2}\|h\|^2 + C\sum_{i=1}^{S}(\lambda_i+\lambda_i^*)\right\} \\ s.t\begin{cases} h\Phi(x_i)-y_i+b \le \varepsilon+\lambda_i^* \\ y_i-h\Phi(x_i)-b \le \varepsilon+\lambda_i \\ \lambda_i \ge 0, \lambda_i^* \ge 0 \end{cases} \end{cases} \tag{23}$$

Where C is the penalty factor C ($C > 0$); S is the number of traffic samples from a single communication base station. In order to solve Eq. (23), the Lagrangian function is introduced to find the partial derivatives of each variable in Eq. (23), and the problem to be solved is transformed into Eq. (24) by the pairwise principle.

$$\begin{cases} \max\left\{-\frac{1}{2}\sum_{i,j}^{S}(\beta_i-\beta_i^*)(\beta_j-\beta_j^*)K(x_i,x_j)-\varepsilon\sum_{i=1}^{S}(\beta_i+\beta_i^*)+\sum_{i=1}^{S}y_i(\beta_i-\beta_i^*)\right\} \\ s.t.\sum_{i=1}^{S}(\beta_i-\beta_i^*)=0, \beta_i\beta_i^* \in [0,C] \end{cases} \tag{24}$$

Where $K(x_i,x_j)=\Phi(x_i)\Phi(x_j)$ is the kernel function; β_i、 β_i^* are Lagrangian daily numbers. At this point, the regression function can be obtained as shown in equation (25).

$$f(x)=\sum_{i=1}^{S}(\beta_i-\beta_i^*)K(x_i,x_j)+b \tag{25}$$

There are various kernel functions for SVR, and in this paper, the RBF (Radial Basis Function) kernel function is used, as shown in equation (26).

$$K(x_i,x_j)=\exp(-\gamma\|x_i-x_j\|^2) \tag{26}$$

Where γ is the kernel function parameter. In this paper, we refer to the literature [7] and use the mean squared error during model training as the fitness function, which is calculated as shown in equation (27).

$$f_{MSE}=\frac{1}{S_T}\sum_{i=1}^{S_T}(y_p-y_r)^2 \tag{27}$$

Where y_p is the predicted flow value and y_r is the true flow value.

3.7. Improving the grey wolf arithmetic optimization support vector regression prediction model

When using SVR for prediction, the values of the penalty factor C and the kernel function parameter γ can have an important impact on the prediction accuracy [8]. Therefore, in order to improve the accuracy of SVR prediction, we use LIGWO to compute the penalty factor and kernel function parameters of SVR by finding the best, and establish the LIGWO_SVR prediction model. The steps of LIGWO_SVR prediction model for traffic prediction of communication base stations are as follows:

***Step*1:** The communication base station traffic history data is divided into training set and test set, and the normalized data matrix x^*, is obtained by normalizing the base station traffic data x using Equation (28), and the calculation formula is shown in equation (28).

$$x^*=\frac{x-\min(x)}{\max(x)-\min(x)} \tag{28}$$

***Step*2:** Set the range of values of dimension *Dim*, penalty factor C and kernel function parameter γ. Set the parameters of the improved gray wolf optimization algorithm using Step1 described in Section 3.5 of the text.

***Step*3:** Initialize the gray wolf population using *Step*2 as described in Section 3.5.

ISSN: 2664-9640

***Step*4:** The fitness of individual gray wolves was calculated according to equation (26), and the iterative gray wolf population was updated using *Step*3 ~*Step*8 described in Section 3.5 of the text, and the optimal values of penalty factors and kernel function parameters of SVR were finally found.

***Step*5:** The test set data are predicted using the SVR optimized by the LIGWO algorithm, and the output results are back-normalized to obtain the prediction results of the communication base station traffic.

4. Model Simulation Experiment

In order to avoid the chance of model prediction results, we use four prediction models to predict the traffic data of communication base station A and communication base station B in Harbin city respectively, and the traffic data are the hourly traffic of communication base stations from November 3 to 17, 2020. In order to exclude the influence of data set division on the prediction accuracy of the models and to ensure that each model conducts simulation experiments under the same conditions, we use the daily 24-hour traffic data from November 3 to November 16 as the training set and the 24-hour traffic data from November 17 as the test set.

4.1. Analysis of simulation experiment results

In order to verify the high prediction accuracy of the LIGWO_SVR prediction model proposed in this paper, the prediction results of the model are compared with the prediction results of the GWO_SVR model, the prediction results of the SA_BP model, the results of the PSO_BP model and the actual communication base station traffic data. The model parameters are set as follows: The maximum number of iterations for both the LIGWO_SVR model and the GWO_SVR model is $t_{\max}=100$, population size N=50， dimension Dim=2. The penalty factor C and the kernel function parameter γ both take values in the range [1×10^{-7},1×10^{6}]. The length of Markov chain of SA_BP model is L =10， initial temperature $T_{ini}=8$, final temperature $T_{fin}=3$, attenuation parameters *Dec*=0.85， *Metropolis Step Size Factor* M=0.2. PSO_BP model population size N=50， maximum number of iterations $t_{\max}=100$ ， individual learning factor $c1$=1.49, social learning factors $c2$=1.49, inertia factor ω=0.2. Number of nodes in the hidden layer of the BP neural network for SA_BP model and PSO_BP model *hid*=5. The predicted results of communication base stations A and B are shown in Fig. 5 and Fig. 6.

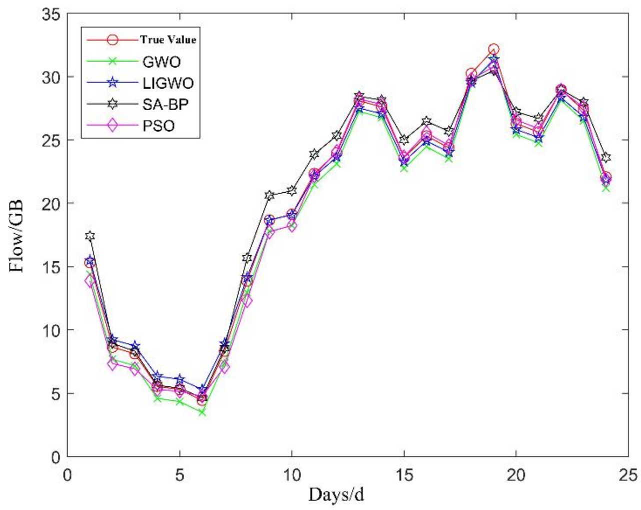


Figure 5: Traffic Forecast for Communication Base Station A

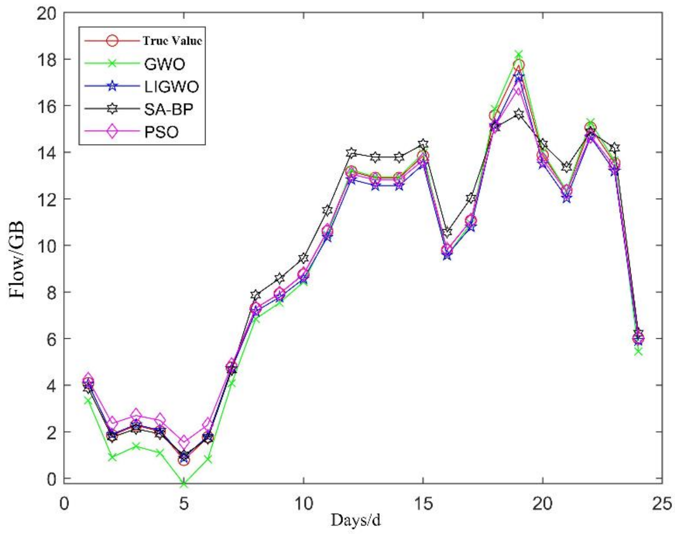


Figure 6: Traffic Forecast for Communication Base Station B

From Fig. 5 and Fig. 6, it can be seen that the LIGWO_SVR model fits the actual communication base station traffic data better than the other three prediction models. The comparison between GWO_SVR model and LIGWO_SVR model proves that LIGWO can effectively balance the global search ability and local exploitation ability, and reduce the risk that the search parameters fall into local optimal solutions.

4.2. Model prediction error analysis

The relative percentage error can effectively reflect the accuracy of the prediction method, and the relative percentage error σ is calculated as follows [9].

$$\sigma = \left| \frac{\hat{y} - y_i}{y_i} \right| \times 100\% \tag{29}$$

Where $\hat{y}_i$ is the predicted value of flow data and y_i is the true value of flow data, the relative percentage error comparison results are shown in Figure 7 and Figure 8.

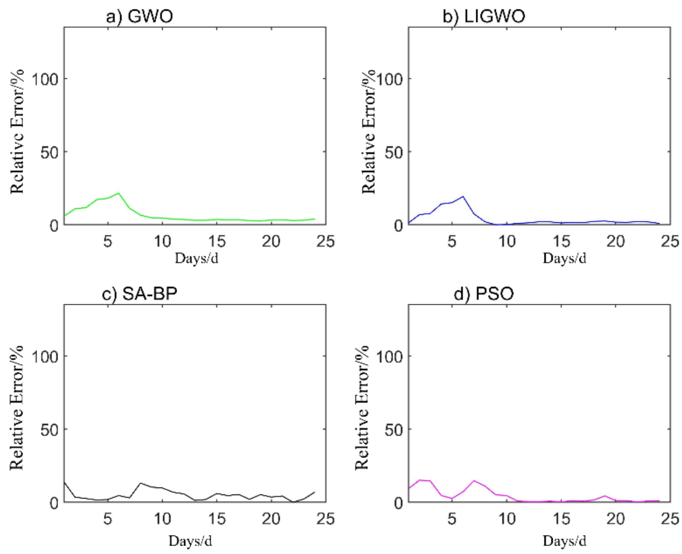


Figure 7: Relative percentage error of the prediction of traffic data for communication base station A by the four models

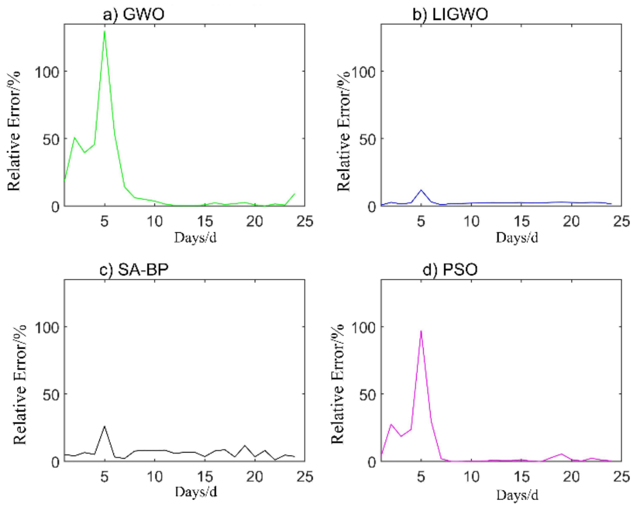


Figure 8: Relative percentage error of the prediction of traffic data for communication base station B by the four models

5. Conclusion

To further improve the accuracy of communication base station traffic prediction, the LIGWO_SVR prediction model is proposed in the paper by optimizing the penalty factor C and kernel function parameter γ of SVR using the LIGWO algorithm. Through simulation tests, the LIGWO_SVR model has higher prediction accuracy compared with other models in communication base station traffic prediction, providing a high-precision prediction model for communication base station traffic prediction.

References

[1] Luo Chongfan. Cat mapping-based multi-objective cat colony optimization algorithm and its application [D]. Lanzhou University,2017.
[2] Long W, Cai SH, Jiao JJ, Wu Tie-Bin. An improved gray wolf optimization algorithm [J]. Journal of Electronics,2019,47(01):169-175.
[3] Li Jing, Yang Fan. UAV agricultural spraying trajectory planning based on IGWO-A * algorithm [J]. Journal of Shenyang Agricultural University,2020,51(02):231-237.
[4] Teng ZJ, Lv JL, Guo LW, Xu YY. An improved algorithm for hybrid gray wolf optimization based on Tent mapping [J]. Journal of Harbin Institute of Technology,2018,50(11):40-49.
[5] Mo Yanhong, Nie Hui, Liu Zhenbei, Yang Huihua. Grey wolf optimization algorithm based on Lévy flight [J]. Microelectronics and Computing,2019,36(04):78-83.
[6] Zhang Yibiao, Jiao Jian, Wang Ke, Zheng Hanbo, Fang Ga Ke, Zhou H. An optimal support vector machine fault diagnosis model for power transformers based on imperial colonial competition algorithm [J]. Power Automation Equipment,2018,38(01):99-104.
[7] Liu Yugen, Chen Chao. Prediction model of corrosion rate of grounding network based on artificial bee colony algorithm optimized support vector machine [J]. Power Automation Equipment, 2019,39(05):182-186+200.
[8] Chen Wigen, Teng Li, Liu Jun, Peng Shangyi, Sun Caixin. A hot spot temperature prediction model for transformer windings based on genetic optimization support vector machine [J]. Journal of Electrical Engineering Technology,2014,29(01):44-51.
[9] Liu F, Li HJ, Liu YW, Zou QC. VMD and GWO-SVR based power load forecasting methods [J]. Modern Electronic Technology, 2020, 43(23):167-172.