



An effective co-evolutionary algorithm based on artificial bee colony and differential evolution for time series predicting optimization

Yun Yang¹ · Zongtao Duan¹

Received: 20 November 2019 / Accepted: 26 April 2020 / Published online: 18 May 2020
© The Author(s) 2020

Abstract

Non-linear model optimization for predicting time series is a challenge problem. In Intelligent Transportation Systems (ITS) application, the indispensable short-term traffic flow prediction with big data makes the problem worst. To improve the prediction accuracy and ensure real-time performance in the big data environment, we propose a novel co-evolutionary artificial bee colony (ABC) improved by differential evolution (DE) optimization algorithm combined with a traffic flow predicting model trained by extreme learning machine (ELM) neural network. The proposed model can inherit the better generalization performance and the less training time consumption of the standard ELM, and can achieve a more balanced search strategy with the optimized weights and biases to overcome the random initialization deficiency of the typical ELM, and successfully obtain higher prediction accuracy compared with state-of-the-art methods. To verify the efficiency of the proposed model, we apply it to Lozi and Tent chaotic time series simulations and measured traffic flow time series experiments. Simulation and experimental results demonstrate that the proposed model has superior performance and competitive computational efficiency.

Keywords Artificial bee colony · Differential evolution · Extreme learning machine · Traffic flow prediction

Introduction

An accurate and rapid short-term traffic flow prediction is important to the performance of many advanced applications of the intelligent transportation system (ITS). The forecast results can help road users make better travel routes, reduce traffic congestion and improve the capacity of the existing road networks. The non-linearity and complexity of short-term traffic flow prediction have attracted researchers to propose many solutions to solve this challenging problem. The traffic flow data are a typical time sequence signal in which values are collected from traffic sensors based on a uniform time interval and a fixed sampling rate. Meanwhile, urban traffic flow data different with traditional time series data can reflect the periodical similarity like morning and night peak in big cities and the zonal relevancy like schools or hospitals or edge zones. The unique temporal and spatial

characteristics of traffic flow data make it very difficult to accurately predict in real time. Therefore, the forecast results often have an obvious deviation compared with real measured traffic flow data. There are two major reasons. First, earlier short-term traffic flow prediction methods mainly depend on the limited historical detecting data because of lack of traffic flow data collecting infrastructure. Second, it is hard to build one predictive model that has better scalability and fit all traffic situations. Luckily, we have entered the era of big data transportation with the development of infrastructure and new data transmission technology.

Nowadays there are various sensors to collect real-time traffic flow data including radars, cameras, mobile Global Position System (GPS), social media, etc. However, the big data ITS also induce some new problems for short-term traffic flow prediction. Therefore, considering the remarkable big traffic flow data feature, how to efficiently use the tremendous traffic flow data to improve the accuracy and timely prediction result has grown into a hot spot [1,2]. Recently, a deep-learning-based traffic flow prediction method has been successfully used to extract deep features for prediction, which have achieved superior performance [1,3]. However, the computational cost during both training and prediction processes is expensive for all deep learning models [4]. Apart

✉ Yun Yang
yangyun@chd.edu.cn

Zongtao Duan
ztduan@chd.edu.cn

¹ School of Information and Engineering, Chang'an University, Xi'an, China

from deep learning models, the popular autoregressive integrated moving average (ARIMA), Neural network (NN), and Support Vector machine (SVM) are successfully used for short-term traffic flow prediction. The NN models are widely utilized in engineering applications of ITS because of the good performance and the compact architecture.

A novel NN called extreme learning machine (ELM) can obtain good generalization at fast speed, which is suitable to predict short-term traffic flow in real time. However, the random weights and biases of ELM could lead to the instability of ELM output even based on same training dataset. To improve the stability of ELM and keep the prediction accuracy, the input weights and hidden bias in ELM can be optimized by some search algorithms [19]. However, single optimization mechanism has different emphasis on local searching or global searching. Therefore, combining different optimization algorithms to build hybrid strategy for ELM predicting traffic flow is an effective and interesting research thought, which is the main motivation of our work.

In this paper, our contributions are as follows. First, we propose a novel hybrid NN model which combines a novel co-evolutionary artificial bee colony (ABC) optimization algorithm improved by differential evolution (DE) with a traffic flow predictor trained by extreme learning machine (ELM) algorithm. Second, we use the Lozi and Tent chaotic mapping to verify the improved ABC–ELM model to outperform than several traditional models and standard ELM. Finally, we collect real traffic flow sensors data to train and test the improved ABC–ELM model compared with the standard ELM model to certify the competitive performance and the computational cost in the real big data ITS environment.

The organization of this paper is as follows. The next section reviews the popular short-term traffic flow prediction solutions followed by which the detailed implementation of the proposed hybrid prediction model is explained. The results of simulation and experiment are shown in the subsequent section. The concluding remarks and future works are given in the last section.

Related works

In addition to the works mentioned in “Introduction” section, we further review works closely related to research and theoretical analysis in this paper. Over the latest few years, many data analysis models have been developed to solve the short-term traffic forecast problem.

These approaches can be divided into two categories, namely parametric methods and non-parametric methods. Among the parametric methods, the ARIMA model and many variants of ARIMA were widely used to predict short-term freeway traffic flow [5]. Parametric methods can obtain a good performance when traffic flow shows regular variations,

but they cannot deal with the obvious chaotic characteristics and nonlinear nature of traffic flow data. Therefore, many researchers have focused on nonparametric methods, such as K-nearest neighbor (KNN) model [6], back propagation (BP) neural network model [7], radial basis function (RBF) neural network model [8], and support vector regression (SVR) [9]. In these classical models, RBF neural networks have not only more powerful approximation but also better auto-adaptability. It becomes the researched focus of many experts in traffic flow forecasting field [2].

Recently, a new RBF network called ELM has been verified in the benchmark regression and classification data sets very well [10,11]. For non-linear chaotic time series prediction problem, the ELM using sigmoid activation function can obtain high accuracy [12]. Recently, researchers proposed a hybrid framework optimized ELM by self-adaptive differential harmony search for financial time series prediction [13]. Extensive comparison experiments based on BP and RBF with ELM improved by the global search optimization approach could achieve the superior performance and not easily get trapped in local minima. However, the traditional optimization method in the proposed ELM model could not handle uncertainties and outliers in the complex time series data. More recently, time series prediction has become more accurate because of the deep learning development. In [14], ELM for wind speed time series forecasting optimized by stacked auto-encoders (SAE) is compared with current deep learning models like a deep belief network (DBN) and a restricted Boltzmann machine (RBM). The evaluation results indicated that the proposed ELM model could obtain average accuracy of 93.73% on single datasets and 94.04% on combined datasets. However, ELM with SAE is time consuming. To overcome these drawbacks and improve the stability of ELM network, how to optimize the parameters of ELM for time series prediction is still a challenging problem.

During the last decades, the biological population-based optimization provides robust and simple solution for multi-modal and multi-objective optimization problems. The classical biological optimization technique includes early genetic algorithms (GA) based on Darwinian evolution theory of the living beings. GA, as the most popular optimization method, has the obvious advantage of the powerful local search ability. Similar to GA including mutation, crossover and selection operation, the advanced differential evolution (DE) overcomes the drawback of GA easily falling into local optima and performs well in many fields. Compared with GA and DE, the particle swarm optimization (PSO) inspired by bird behavior has obvious advantages of easy implementation and less tuning parameters. For short-term traffic flow forecasting, the latest work in [15] proposed PSO–ELM to improve the accuracy and obtain a competitive performance by comparing with several state-of-the-art methods. However, the drawbacks of PSO algorithm are it is easily trapped in local

optima and premature convergence occurs. The latest artificial bee colony (ABC) algorithm is a novel optimization which is good at global search and can be directly combined with a neural network [16]. However, the local research ability of ABC is poorer than other biological optimization algorithms. To improve the GA, the DE, the PSO and the ABC performance, some various combination attempts have been down in hybrid GA, DE, PSO and ABC model. In [17], authors present the performance comparison of DE, PSO and ABC based on benchmark test functions. The experimental result shows that the DE algorithm can obtain a better solution than ABC and DE, and the computation time is minimum for ABC and maximum for PSO based on same iterations. To achieve the short-term traffic flow prediction goal, we propose a novel ABC–DE algorithm to optimize the ELM neural network parameters.

The standard ABC algorithm starts from random solutions to search for better solutions through iteration according to the fitness result, and it has slow convergence speed and is easily prone to premature convergence [18]. Considering the real-time and accurate traffic flow prediction requirement in the big data ITS, we improve the local searching space with new optimal searching strategy [19–21] of ABC based on differential evolution (DE) [22] algorithm to optimize the input weights and hidden biases of ELM [22]. To verify the proposed model in the big data ITS, Lozi and Tent chaotic functions [2,24,25] and real measured traffic flow data can be applied to simulate and implement experiments. Compared with ARIMA, KNN, BP, RBF, SVR and standard ELM, the proposed model can obtain a better generalization and higher accuracy and competitive time consumption in the big data ITS.

Proposed network architecture for traffic flow prediction

In this section, we propose a novel neural network architecture for short-term traffic flow prediction in a big data environment based on the ELM and improved ABC, including the theoretical analysis, model confirmation and optimization, respectively.

ELM neural network

ELM proposed by Guangbin Huang [10,16] was originally inspired by biological learning and aimed to overcome these limitation drawbacks faced by conventional machine learning theories and techniques. From the neural network architecture point of view, the output function with L hidden nodes for a standard ELM [16] can be defined as follows:

$$y_i = \sum_{j=1}^L \beta_j f(\omega_{i,j}x_i + b_j), \tag{1}$$

where the activation function $f(\bullet)$ is a nonlinear piecewise continuous function like sigmoid and radial basis. Where ω is the random weights of the input layer which connects the hidden node with i th input vector, and b is the random biases of the hidden layer. For the given n training samples (x_i, t_i) , where the input vector $x_i = [x_{i1}, x_{i2}, \dots, x_{im}] \in R^m$, where the target label $t_i = [t_{i1}, t_{i2}, \dots, t_{ip}] \in R^p$, $i = \{1, 2, \dots, n\}$, the output value of ELM network can be defined as $y_i = [y_{i1}, y_{i2}, \dots, y_{ip}] \in R^p$.

In formula (1), the initial ω and b value can be randomly assigned by the ELM theory, and the output weights β can be calculated by the least squares solution. If the training of ELM network aims to reach not only the smallest training error, but also the smallest norm of output weights, which means the ELM network should be trained to approximate arbitrary samples with zero error [11], there exist β , ω and b that make $y_i = t_i$ hold true. Therefore, the compact vector version of ELM function could be expressed as follows:

$$T = H_{\omega,b,x}\beta, \quad T = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_L \end{bmatrix} \tag{2}$$

$$H_{\omega,b,x} = \begin{bmatrix} f(\omega_{1,1} \cdot x_1 + b_1) \cdots f(\omega_{1,L} \cdot x_1 + b_L) \\ \vdots \\ f(\omega_{n,1} \cdot x_n + b_1) \cdots f(\omega_{n,L} \cdot x_n + b_L) \end{bmatrix} \tag{3}$$

where $H_{\omega,b,x}$ is named as the hidden layer output matrix, the output weights β can easily be calculated by the least square solution as follows:

$$\beta = pinv(H_{\omega,b,x}) \cdot T, \tag{4}$$

where function $pinv(\cdot)$ means to compute the Moore–Penrose pseudo-inverse of the hidden layer output matrix $H_{\omega,b,x}$. It is easy for programming, so the output weights β can be successfully calculated by this function.

However, the convergence of the standard ELM is generally slow, because the training of a standard ELM needs a large number of hidden nodes to approach an appropriate result. To overcome this issue, the kernel-based ELM (KELM) is suggested by the authors [16]. To KELM network, the hidden layer output matrix can be presented as follows:

$$H_{\mu,\sigma,x} = \begin{bmatrix} k(\mu_1, \sigma_1, x_1) \cdots k(\mu_L, \sigma_L, x_1) \\ \vdots \\ k(\mu_1, \sigma_1, x_n) \cdots k(\mu_L, \sigma_L, x_n) \end{bmatrix}, \quad (5)$$

where the function $k(\bullet)$ is the ELM activation kernel function which can be but are not limited linear, polynomial, sigmoid, Fourier, hard limit or radial basis function (RBF). Generally, the RBF kernel $k(\mu_i, \sigma_i, x_j) = \exp(-\frac{\|x_j - \mu_i\|^2}{\sigma_i^2})$ can be chosen to achieve a competitive regression model and good generalization performance. Therefore, with the optimum condition of the Karush–Kuhn–Tucker, the output weights matrix β of KELM can be expressed as follows:

$$\beta = H_{\mu,\sigma,x}^T \left(\frac{I}{\varepsilon} + H_{\mu,\sigma,x}^T \cdot H_{\mu,\sigma,x} \right)^{-1} T, \quad (6)$$

where ε is defined as penalty coefficient, the μ, σ and ε parameters of KELM are similar to the input layer weights and the hidden layer biases of ELM and can be randomly initialized to train the KELM network. The random initialization of parameters can lead to the obvious unstable output of KELM or ELM network, which will reduce the advantage of the ELM or KELM generalization ability. To solve this problem, the self-adaptive optimization algorithm can be introduced to optimize the above parameters of ELM or KELM neural network.

ABC algorithm

An ABC optimization algorithm that imitates the foraging behaviors of honey bee colony is a new intelligent optimization algorithm. In ABC algorithm, there are three types of bee including employed, onlooker and scout bees searching for the best food source. The employed and onlooker bees perform the local search, and the scout bees control the global search [18]. The main steps of the ABC algorithm can be given as follows:

Step 1: Initialization stage

- Initialize the boundary of positions value including $[Min, Max]$.
- Randomly initialize each searching position X_i on the boundary, the initialization function can be defined as follows:

$$x_{ij} = Min + R_{ij} * (Max - Min), \quad (7)$$

where $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, and where $i = 1, 2, \dots, SN$, and where D means the number of opti-

mization parameters, and where R_{ij} is a random number in the range of $[-1, 1]$.

- Initialize the start searching position for the employed bees using the greedy policy to find the best solution in the initialization position matrix $X^{SN \times D}$.
- Step 2: Employed bee phase
- Employed bees generate new solutions using the following expression:

$$v_{ij} = x_{ij} + \phi_{ij} * (x_{ij} - x_{kj}). \quad (8)$$

In the above equation, v_{ij} is the candidate solution, where $i, k \in \{1, 2, \dots, SN\}$ and $j \in \{1, 2, \dots, D\}$ are randomly selected indexes, but $k \neq j$. And ϕ_{ij} is a random number between $[-1, 1]$.

- Select the best one between the new candidate solution v_{ij} and the last original solution x_{ij} using the greedy policy.
- Calculate the fitness values and the selection probability using the equation as follows:

$$fit_i = \begin{cases} \frac{1}{1+fit_i} & \text{if } fit_i \geq 0 \\ 1 + abs(fit_i) & \text{if } fit_i < 0 \end{cases}, \quad (9)$$

where fit_i is the fitness value of the i th solution X_i , and the f_i is the cost value of the i th solution X_i .

Step 3: Onlooker bee phase

- Select a new solution in all the employed bee solutions using the selection probability p_i as follows:

$$p_i = \frac{fit_i}{\sum_{k=1}^{SN} fit_k}. \quad (10)$$

In the standard ABC algorithm, it is difficult to balance the global search and the local search which will result in the slow convergence speed or the earlier local optimal solution. Therefore, accelerating convergence speed and avoiding local optima are two main research directions for the improved ABC algorithm.

DE strategies

Many DE mutation strategies have been widely described like DE/rand/1, DE/best/1, DE/current-to-best/1, and DE/rand/2. These strategies can be defined as follows:

- DE/rand/1: $V_i = X_{r1} + F(X_{r2} - X_{r3})$,
- DE/best/1: $V_i = X_{best} + F(X_{r1} - X_{r2})$,
- DE/current-to-best/1: $V_i = X_i + F(X_{best} - X_i) + F(X_{r1} - X_{r2})$,
- DE/rand/2: $V_i = X_{r1} + F(X_{r2} - X_{r3}) + F(X_{r4} - X_{r5})$,

where V_i is the perturbed vector, F is the control parameter, X_{best} is the global best solution found so far, and X_{r1} , X_{r2} , X_{r3} , X_{r4} , and X_{r5} are randomly selected from the target solution X_i . All mutation strategies have different searching ability. The DE/rand/1 strategy is good at a population diversity but is difficult to obtain an optimal solution. The DE/best/1 strategy can implement a fastest convergence but usually obtain a local optimum. The DE/rand/2 strategy involving two difference vectors may result in better perturbation than other strategies but easily lead to a premature convergence. Considering both a good quality solution and inferior solution, the DE/current-to-best/1 strategy is effective to solve multi-objective optimization problem and can achieve the convergence more quickly. According to the experimental results based on 26 standard benchmark numerical optimization functions for four conventional DE strategies in [19], we directly select the DE/current-to-best/1 strategy to enhance the local searching ability of the original ABC algorithm in the mutation operator phase.

Improved ABC algorithm by novel DE strategy

Comparing with popular evolutionary algorithms such as Genetic algorithm (GA), Particle swarm optimization (PSO), Differential Evolution (DE) and Ant colony optimization (ACO), the classical ABC can outperform to solve some multidimensional optimization problems. However, it also has some obvious drawbacks [20], which the search strategy of classical ABC can only update one element in a vector at each time, which will be good at global exploration and poor at local exploitation. Specifically, the update strategy of original ABC is different from the other representative population-based algorithms, which will result in ABC hardly taking the advantage of the best solution. However, the best solution information and the convergence performance are so important in solving some real engineering problems. Inspired by the searching strategy of the DE algorithm, we propose an improved ABC algorithm, which modifies the local searching mechanism of ABC in the onlooker bee phase, which is beneficial to exploitation by introducing the best-so-far solution. The novel co-evolutionary algorithm can combine ABC and DE strategies during the search process, so as to improve the search efficiency. For clarity, the main steps of the improved ABC by novel DE strategy function are described in Algorithm 1.

In the classical ABC algorithm, the onlooker behavior should be calculated by a same equation with employed bees. However, in the improved ABC algorithm, a new equation is proposed by the basic DE/current-to-best/2 mutation strategy which can easily achieve good convergence speed and obtain a better local optimum solution. Therefore, the improved ABC algorithm can also maintain the global search ability of

Algorithm 1 Improved ABC algorithm

Input: SN(population size), MCN(maximum iterations), LM(maximum searching limit), $Trial_i$ (memory trial numbers), Min(lower boundary), Max(upper boundary)

Output: BT(best solution)

Initialisation :

1: Initialize $Trial_i$ is zero

2: Generate X_i initial solutions by Eq.(7)

Repeat

Employed bees phase:

3: Generate a new candidate solution V_i by Eq.(8)

4: Select a greedy best one between X_i and V_i

5: Compute the fitness Fit_i of each solution by Eq.(9)

6: Calculate the probability P_i of each solution by Eq.(10)

Onlooker bees phase:

7: Using probability P_i to select a solution X_i

8: Search the best solutions x_{ij}^{best} among the random neighbours $x_{r1,j}$ and $x_{r2,j}$ of $x_{i,j}$ including itself

9: Using DE/current-to-best/2 mutation equation to generate a new candidate solution $v_{i,j}$ by

$$v_{ij} = x_{ij} + \phi_{ij} * (x_{ij}^{best} - x_{ij} + x_{r1,j} - x_{r2,j}) \quad (11)$$

10: Select a greedy best one between X_i and V_i

Scout bees phase:

11: **if** ($trail_i > LM$) **then**

12: replace X_i using new solution by Eq.(7)

13: **else**

14: $trail_i = trail_i + 1$

15: **end if**

16: Memorize the BT(best solution) detected so far

Repeat Until(cycle=MCN)

17: **return** BT

ABC and absorb the local optima of DE to perform higher accuracy.

Optimizing ELM neural network by improved ABC algorithm

The training of ELM is extremely fast and suitable for big data. For classical ELM, the output weights are calculated by analytical solution instead of the traditional gradient descent algorithm, and this change can obviously improve the speed and generalization ability of the ELM network [10]. The training goal of ELM is to find the optimal of the input weight matrix and hidden bias vector to make the network obtain the minimum error between the observed values and predicted values. Therefore, the input weights and biases and their amount dramatically affect the performance and accuracy of ELM. In the ELM model, the input weights and hidden biases are randomly assigned and unchanged during the training process. Therefore, the random parameters may be a set of non-optimal or unnecessary values, which make ELM network require a large number of hidden nodes to approach appropriate result and response slowly into testing data. Therefore, selecting the optimal input weights and

hidden biases can provide more compact ELM network architecture and better generalization performance.

Applying the ABC algorithm to train ELM is relatively straightforward. The main steps of the ABC–ELM model are described as follows:

Step 1, the parameters of ABC algorithm are initialized according to the initialization stage of Algorithm 1.

Step 2, the encoded input weights and hidden biases of ELM as the colony population of ABC algorithm is randomly created as follows:

$$\begin{aligned}
 X_i = & (w_{11}^i, w_{12}^i, \dots, w_{1L}^i, \\
 & w_{21}^i, w_{22}^i, \dots, w_{2L}^i, \\
 & \dots, \\
 & w_{n1}^i, w_{n1}^i, \dots, w_{nL}^i, \\
 & b_1^i, b_2^i, \dots, b_L^i),
 \end{aligned}
 \tag{12}$$

where the input weight w and the hidden biases b are randomly assigned in the range $[-1, 1]$, and where L is the number of ELM hidden nodes, $i = (1, 2, \dots, SN)$.

Step 3, the employed bee phase, the onlooker bee phase and the scout bee phase are executed in sequence according to Algorithm 1. Then the best solution of the ABC algorithm can be obtained after the iterations are met.

Step 4, the ELM network can be trained and tested with the best input weight and hidden bias solution obtained in Step 3.

Metrics for forecasting error

Three criteria are commonly used to evaluate the performance of the traffic flow forecast model. They are the root mean square error (RMSE), the mean absolute error (MAE) and the mean relative error (MRE) and are defined as follows, respectively:

$$RMSE = \left(\frac{1}{n} \sum_{s=1}^n |\hat{y}_s - y_s|^2 \right)^{1/2}
 \tag{13}$$

$$MAE = \left(\frac{1}{n} \sum_{s=1}^n |\hat{y}_s - y_s| \right)
 \tag{14}$$

$$MRE = \frac{1}{n} \sum_{s=1}^n \left| \frac{\hat{y}_s - y_s}{y_s} \right|.
 \tag{15}$$

Experiment results

To verify the validity of the optimized ELM by the improved ABC algorithm, Lozi and tent chaotic time series and measured traffic flow time series are applied into the experiments.

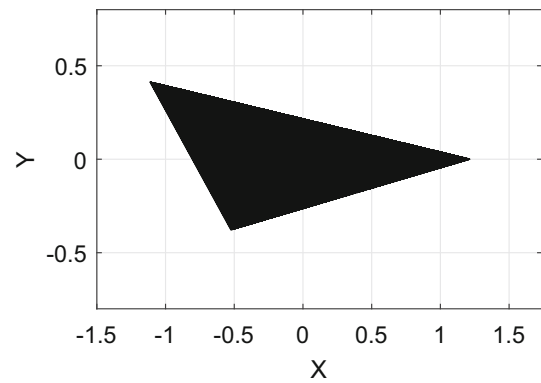


Fig. 1 Testing Lozi chaotic sequence generating function mapping

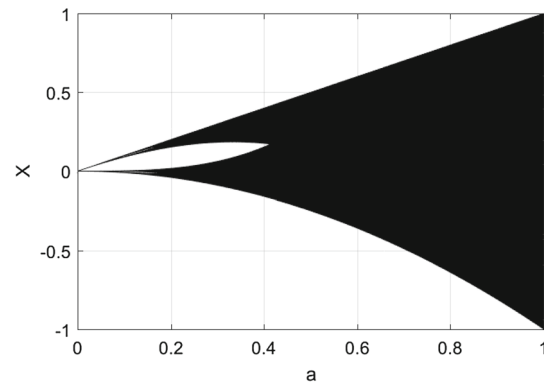


Fig. 2 Testing tent chaotic sequence generating function mapping

All the experiments are running on the computer equipped with a system with Intel (R) Xeon (R) CPU E3-1230 V2 (3.30 GHz), 16 GB DDR4.

For the improved ABC algorithm, the colony size should be set as 40 including the number of employed bees and onlooker bees, and the maximum control limit should be set as $100 * 2$, and the maximum iterations can be 6 times, and the number of optimized parameters for simulations is $101 * 15,000$, and the number of the optimized parameters for measurement is $101 * 8000$, which include input weights and hidden biases. Moreover, the lower bound and upper bound for parameters can be set as $[-1, 1]$, and the search radius of the onlooker bees can be set as $r_1 = 1, r_2 = 3$.

Time series prediction simulation

Lozi time series is a kind of discrete chaotic system which has randomness and ergodicity features [23]. The Lozi equation is given as follows:

$$\begin{cases} X_{n+1} = 1 - a |X_n| + bY_n \\ Y_{n+1} = X_n \end{cases}
 \tag{16}$$

Fig. 3 Traffic flow prediction by improved ABC–ELM of Freeway SR29-S-402863

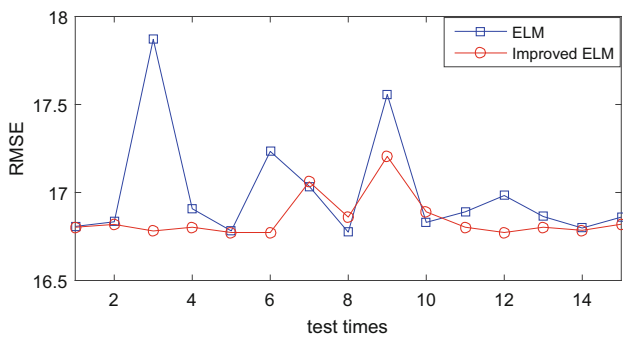
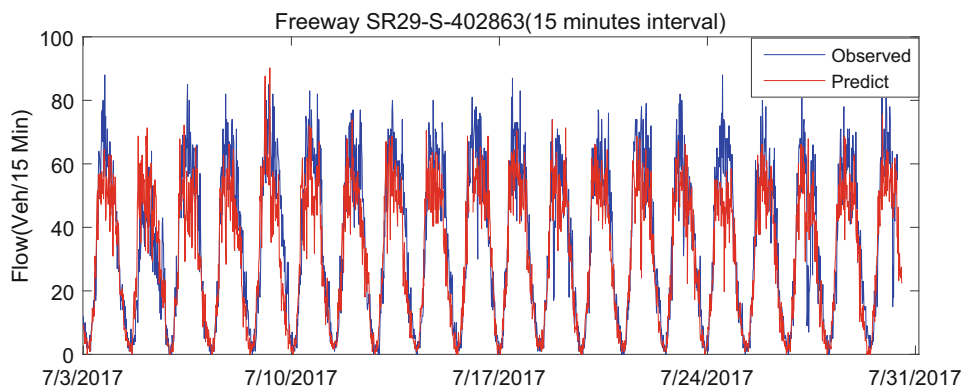


Fig. 4 Testing RMSE comparing ELM with improved ABC–ELM of Freeway SR29-S-402863

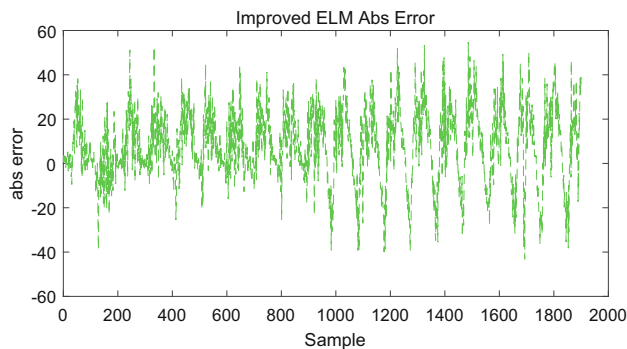


Fig. 5 Abs error of the improved ABC–ELM prediction for Freeway SR29-S-402863

The parameters used are $a = 1.7$ and $b = 0.5$ as suggested in [2] and the initial conditions are $X_0 = -0.1$ and $Y_0 = 0.1$. Figure 1 shows the Lozi chaotic generator mapping.

Tent time series is a simple chaotic system which has uniform probability and power spectral density feature [24]. Therefore, it suits to simulate the computational processing of the big data.

Figure 2 shows the tent chaotic function mapping. The tent mathematical model is defined as follows:

$$X_{n+1} = a - (a + 1) |X_n| \quad a \in [0, 1]. \tag{17}$$

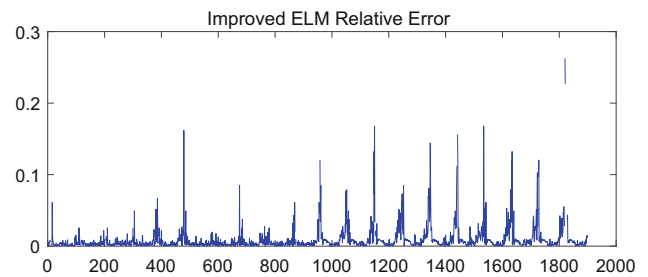


Fig. 6 Relative error of the improved ABC–ELM prediction for Freeway SR29-S-402863

In the experiment, the first 15000 samples of the time series (75%) are selected as the training dataset and the remaining 5000 samples (25%) are taken as the testing dataset.

The SR29-S-402863 dataset is collected from the Caltrans Performance Measurement System (PeMS) database [26]. In this paper, the traffic flow data collected on the weekdays of the first 7 months of the year 2017 are used for the experiments. The 8000 samples for the first 6 months are chosen as the training set, and the remaining 1 month’s 1900 data are selected as the testing dataset. Table 1 shows the prediction errors and run time of four kinds of models compared with the Lozi system, the tent system, and the SR29-S-402863 dataset. According to the comparison, the parametric ARIMA and KNN model can obtain highest accuracy, but their disadvantages are highest time cost which is not suitable for the modern ITS. The traditional neural network model such as the BP, RBF, and SVR present poor performance. Therefore, the stable accuracy and lowest time consumption of the ELM and improved ELM model should be more suitable for modern ITS big traffic flow real-time prediction problem.

Traffic flow prediction by improved ABC–ELM of freeway SR29-S-402863

The proposed ELM neural network model trained by the improved ABC algorithm is applied to the SR29-S-402863

Table 1 Predictive errors and run time of Lozi system, tent system, and SR29-S-402863

Prediction model	Dataset	RMSE	MAE	MRE	Runtimes(s)
ARIMA ($p = 1, d = 1, q = 1$)	Lozi	0.7668	0.6191	12.2854	248.1070
	Tent	0.7272	0.5890	3.5280	242.0442
	SR29-S-402863	14.7065	11.0178	NaN	182.5276
KNN ($k = 3$)	Lozi	0.6640	0.5450	18.9258	125.1184
	Tent	0.5802	0.4787	3.1493	124.8787
	SR29-S-402863	14.0554	10.5488	NaN	23.2309
BP ($h = 100, ep=800, lr=0.1$)	Lozi	1.0622	0.8540	10.1733	75.5460
	Tent	1.3805	1.0955	11.0894	59.1258
	SR29-S-402863	17.3484	13.1853	NaN	16.2755
RBF (dynamic $h = 150$)	Lozi	0.8077	0.6530	5.6868	31.3488
	Tent	0.6659	0.5461	5.8914	28.8251
	SR29-S-402863	43.1306	35.7384	NaN	15.2628
SVR (bestc=3.0314, bestg=0.0118)	Lozi	0.6168	0.5072	2.5302	84.8080
	Tent	0.5204	0.4395	3.0946	126.6523
	SR29-S-402863	17.9446	14.2106	NaN	29.4524
ELM (random w and $b, h = 100, act=sig$)	Lozi	0.6133	0.5079	2.9344	0.2483
	Tent	0.4925	0.4179	2.8410	0.2507
	SR29-S-402863	17.4315	13.7944	NaN	0.1744

Table 2 Predictive errors of optimization methods on SR29-S-402863

Method	RMSE	MAE
PSO-ELM($p = 40, Inertia=0.3593, Pbest=0.7238, Sbest=2.0289$)	17.1614	13.5223
DE-ELM($p = 40, CrossP=0.7122, DifferF=0.6301, r1=1, r2=3$)	17.0611	13.5181
ABC-ELM($p = 40, AccelF=1$)	16.9471	13.2742
Improved ABC-DE-ELM($p = 40, AccelF=1, r1=1, r2=3$)	16.5220	13.1509

dataset. The hidden nodes of ELM can be set as 100. Figure 3 presents the output of the proposed improved ABC-ELM model for the traffic flow prediction. The improved ABC-ELM comparing ELM results can be recorded in Fig. 4. It can be clearly seen the RMSE with the optimal results is more stable and better than that with ELM. Figure 5 shows the Abs error of the improved ABC-ELM predicting results, and Fig. 6 shows the relative error of the improved ABC-ELM predicting results.

Discussion of the results

The prediction error due to variance shows the variability of a model prediction between the training dataset and the testing dataset. In our experimental process, all models are iterated multiple times to decrease the variance error. The prediction error due to bias means the difference between the true value which the models are trying to predict and the expected value. After the successful iteration of all models, the testing results on Lozi, tent and SR29-S-402863 dataset are recoded in Table 1. Clearly, the RMSE and MAE of the improved

ABC-ELM model on Lozi, tent and SR29-S-402863 dataset are better than other models. For the MRE, the NaN results are derived the zero value in SR29-S-402863 testing dataset. We also compare the proposed improved algorithm with the ABC and DE to optimize ELM respectively. The predictive errors on SR29-S-402863 dataset is shown in Table 2. RMSE and MAE for the ABC-DE-ELM model indicate a good performance. Thus, the proposed improved ABC-ELM model can obtain higher accuracy than other comparing models. Of course, the time cost of the improved ABC-ELM model are higher than the standard ELM model and the standard ABC-ELM model because of the global searching function. However, the comparison results show that our proposed improved ABC-ELM model can obtain an optimal balance between high accuracy and low cost consumption, which are suitable for the traffic flow prediction in the big data era.

Conclusion

In this paper, we propose a novel and effective prediction model for traffic flow forecasting. This study adopts to

improve the ABC algorithm to optimize the input weights and hidden biases of the ELM neural network and simulates the prediction of two typical chaotic time series of big data comparing the accuracy with ARIMA model, KNN model, BP model, RBF model, and SVR model. With the application of this proposed model to predict real traffic flow measurement systems, the experimental results indicate that the proposed method has higher prediction accuracy and has more stability results. Therefore, we believe that the improved model can have good prospects in real time traffic flow prediction for the big data environment. The future work will focus on the optimal kernel parameters of KELM network to achieve better prediction results.

Acknowledgements The authors thank anonymous reviewers for their constructive comments and suggestions. This work was supported by Key research item for the industry of Shaanxi Province under Grant no. 2018GY-136.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Mackenzie J, Roddick JF, Zito R (2018) An evaluation of HTM and LSTM for short-term arterial traffic flow prediction. *IEEE Trans Intell Transp Syst* 99:1–11
- Chen D-W (2017) Research on traffic flow prediction in the big data environment based on the improved RBF neural network. *IEEE Trans Ind Inform* 13(4):2000
- Zhao Z, Chen W-H, Wu X-M, Chen PCY, Liu J-M (2017) LSTM network: a deep learning approach for short-term traffic forecast. *IET Intell Transp Syst* 11(2):68–75
- Wu X-M, Ding S-Y, Chen W-H, Wang J-H, Chen PC-Y (2018) Short-term urban traffic flow prediction using deep spatio-temporal residual networks. *IEEE Proc on Industrial Electronics and Applications (ICIEA)*, Wuhan, China, pp. 1073–1078
- Xu D-W, Wang Y-D, Jia L-M, Qin Y, Dong H-H (2017) Real-time road traffic state prediction based on ARIMA and Kalman filter. *Front Inf Technol Electron Eng* 18(2):287–302
- Fayed HA, Atiya AF (April 2009) A novel template reduction approach for the K-nearest neighbor method. *IEEE Trans on Neural Netw* pp 890–896
- Cao H-M, Feng H (2014) The urban arterial traffic flow forecasting based on BP neural network. *Computer, Communication and Control*, Harbin, China, Sept, *IEEE Proc. on Instrumentation and Measurement*, pp 393–397
- Lv Y, Duan Y-J, Kang W-W, Li Z-X, Wang F-Y (2014) Traffic volume forecasting based on radial basis function neural with the consideration of traffic flows at the adjacent intersections. *Transport Res C Emerg Technol* 47(2):276–281
- Kim EY (2017) MRF model based real-time traffic flow prediction with support vector regression. *Electron Lett* 53(4):243–245
- Huang GB, Zhou HM, Ding XJ, Zhang R (2012) Extreme learning machine for regression and multiclass' classification. *IEEE Trans Syst Man Cybern* 42(2):513–529
- Xing Y-M, Ban X-J, Liu R-Y (2018) A short-term traffic flow prediction method based on Kernel Extreme Learning Machine. Shanghai, China, May, *IEEE Proc. on Big Data and Smart Computing*, pp 533–536
- Singh R, Balasundaram S (2007) Application of extreme learning machine method for time series analysis, international journal of computer, electrical, automation, control and information. *Engineering* 1(11):3394–3400
- Dash R, Dash PK, Bisoi R (2014) A self adaptive differential harmony search based optimized extreme learning machine for financial time series prediction. *Swarm Evol Comput* 19(2014):25–42
- Yang H-F, Chen Y-PP (2019) Representation learning with extreme learning machines and empirical mode decomposition for wind speed forecasting methods. *Artif Intell* 277(103176):1–10
- Cai W-H, Yang J-J, Yu Y-D, Song Y-Y, Zhou T (2020) PSO-ELM: a hybrid learning model for short-term traffic flow forecasting. *IEEE Access* 8:6505–6514
- Huang G-B, Zhu Q-Y, Siew C-K (2009) A comparative study of artificial bee colony algorithm. *Appl Math Comput* 2(14):108–132
- Mohan M, Joseph MV (2019) A comparative study of different metaheuristic optimization algorithms using standard test functions. *Proceedings of International Conference on Applied Mechanics and Optimisation (ICAMeO-2019)* 2134(1):060001-6
- Bagis A, Senberber H (2016) Modeling of higher order systems using artificial bee colony algorithm. *Int J Opt Control Theor Appl* 6(2):129–139
- Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans Evol Comput* 13(2):398–417
- Gao H, Shi Y-J, Pun C-M, Kwong S (2018) An improved artificial bee colony algorithm with its application to metallographic image segmentation. *IEEE Trans Ind Inf* 99:1–13
- Tian D-X, Zhou J-S, Sheng Z-G (2017) An adaptive fusion strategy for distributed information estimation over cooperative multi-agent networks. *IEEE Trans Inf Theory* 63(5):3076–3091
- Konar M, Bagis A (2016) Performance comparison of particle swarm optimization, differential evolution and artificial bee colony algorithms for fuzzy modelling of nonlinear systems. *Elektronika Ir Elektrotehnika* 22(5):8–13
- Ma C (2017) An efficient optimization method for extreme learning machine using Artificial Bee Colony. *J Dig Inf Manag* 15(3):136–147
- Luo H-X., Zhang G-D, Shen Y-J, Hu J-L (2014) Mixed fruit fly optimization algorithm based on Lozi's chaotic mapping. *IEEE Proc. on P2P, Parallel, Grid, Cloud and Internet Computing*, Guangdong, China, pp. 179–183

25. Shan L, Qiang H, Li J, Wang Z-Q (2005) Chaotic optimization algorithm based on Tent map. *Control Decis* 20(2):179–182
26. Caltrans (2017) Performance measurement system (PeMS), [Online]. Available: <http://pems.dot.ca.gov>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.