Echo State Network Optimization Using Binary Grey Wolf Algorithm

Junxiu Liu[1], Tiening Sun[1], Yuling Luo[1*], Su Yang[2], Yi Cao[3], Jia Zhai[4]
[1] School of Electronic Engineering, Guangxi Normal University, Guilin, China
[2] School of Computing, Engineering and Intelligent Systems,
Ulster University, Londonderry, UK, BT48 7JL
[3] Management Science and Business Economics Group, Business School,
University of Edinburgh, 29 Buccleuch Place, Edinburgh, UK, EH8 9JS
[4] Salford Business School, University of Salford, Salford, UK, M5 4WT
*yuling0616@gxnu.edu.cn

Abstract: The echo state network (ESN) is a powerful recurrent neural network for time series modelling. ESN inherits the simplified structure and relatively straightforward training process of conventional neural networks, and shows strong computational capabilities to solve nonlinear problems. It is able to map low-dimensional input signals to high-dimensional space for information extraction, but it is found that not every dimension of the reservoir output directly contributes to the model generalization. This work aims to improve the generalization capabilities of the ESN model by reducing the redundant reservoir output features. A novel hybrid model, namely binary grey wolf echo state network (BGWO-ESN), is proposed which optimises the ESN output connection by the feature selection scheme. Specially, the feature selection scheme of BGWO is developed to improve the ESN output connection structure. The proposed method is evaluated using synthetic and financial data sets. Experimental results demonstrate that the proposed BGWO-ESN model is more effective than other benchmarks, and obtains the lowest generalization error.

Keywords: echo state network; binary grey wolf optimization; time series; network structure optimization

1. Introduction

Recurrent neural networks are often used to solve time series related prediction tasks [1]. However, there are a few problems hamper its practical applications, such as complicated training process, large amount of calculation, and slow convergence [2]. In order to reduce the computational complexity of the training process, the echo state network (ESN) and the liquid state machine (LSM) are proposed [3, 4], and the study in [5] has proved that ESN and LSM are conceptually similar, they are hence collectively summarized as “reservoir computing”. ESN is a special recursive structure that is known for its structural simplicity and high forecasting precision. Two main features distinguish ESN from other recursive networks: (a) the hidden layer of the ESN is a large-scale sparsely connected reservoir; and (b) only the connection weights between the reservoir and the system output need to be trained, where often only a linear regression problem need to be solved to finish the training process [3]. The ESN reservoir is designed to map low dimensional input to high dimensional space with a feedback connection between the reservoir and the output layer, which is ideal for dealing with dynamic system modelling problems [3]. The low computational cost and powerful performance of ESN make it widely used in a range of applications, e.g. classification task [6], stock prediction [7], electric load forecasting [8], nonlinear model predictive control [9], tourist arrival forecasting [10] etc. Although the ESN model shows strong learning ability, its rich dynamic characteristics and random weight structure have not been fully studied. Hence, how to build an optimized reservoir for a specific task is a challenge which needs to be solved [11]. Hyperparameter optimization is one of the common optimization strategies. Some intelligent algorithms such as fruit fly algorithm and particle swarm optimization are often used to optimize the important parameters of the ESN to get a better performance [12, 13]. The topology of the network is also often optimized to further enhance the competitiveness of ESN [14]. In addition, the optimal output connection of the ESN should be explored. The ESN reservoir is sparsely connected. Paradoxically, the output layer is fully connected. Research works have shown that setting the ESN output layer to a sparsely connected state is beneficial for the network performance [15], which is also in line with the sparse connections of brain neurons [16]. However, the method to optimize the connections is still a challenge which needs to be addressed.

The process of optimizing the connections between the reservoir and the output layer neurons can be considered as a feature selection problem [16]. In practical, the representation of data usually has many redundant features, some these replaceable features can be eliminated [16]. The purpose of feature selection is to improve the network performance by reducing the dimensionality of the feature. Recently, some researches have done to optimize the output connection of ESN. For example, some classical feature selection algorithms (such as least angle regression, backward selection, etc.) are used to trim redundant connections between the reservoir and the output layer, thereby improving the generalization ability of the network [15]. In addition, the greedy feature selection algorithm is also used to reduce the high computation complexity output connection [16]. However, many of these feature selection methods have limitations in solving ESN output connection optimization problem, and the evolutionary computation algorithms can avoid these limitations. The evolutionary computation algorithm is inspired by the process of natural evaluation, and a series of approaches for the global optimization are proposed based on the study of biota behaviours [17, 18]. Genetic algorithm (GA) [19] is one of the popular evolutionary computation algorithms. Compared with traditional optimization methods (e.g. enumeration, heuristic, etc.), it has good convergence speed and can be directly used for discrete problems such as feature selection. It has advantages of robustness and strong scalability, however is also complicated in programming, parameter tunings, and slow search speed [20]. The particle swarm optimization (PSO) algorithm on the other hand, has a fairly fast approximation speed of the optimal solution, which can effectively optimize the parameters of the system [13]. It uses the information of current position, global extremum and individual extremum to iteratively update the position of the particle. The advantage of the PSO algorithm is that it can solve the optimization problem of continuous functions. However, it is easy to converge in advance (especially while dealing with complex multi-modal problems), and the local optimization ability is poor. In addition, it cannot be directly applied to discrete feature selection problems. Thus a discrete binary particle swarm optimization (BPSO) algorithm is proposed in the approach of [21], whose trajectory coordinates are only zero or one.

The grey wolf optimization (GWO) is another novel evolutionary computation algorithm that simulates the social level and hunting behaviour of the grey wolf [22]. It is based on the tracking, enveloping, chasing, and attacking behaviours of wolves for optimization purposes. The binary grey wolf optimization (BGWO) algorithm is proposed in the approach of [23], which is used to select the classification purpose of the best feature subset. In this work, BGWO is used to optimize the output connections of the ESN, and synthetic and financial data are used to evaluate the performance of the proposed model. Experimental results show that the proposed BGWO-ESN model achieves a better performance and successfully reduces the generalization error of the original ESN, which reflect the superiority of the BGWO algorithm. The rest of this paper is organized as follows: Section 2 provides a brief introduction to the background of the ESN. Then the ESN optimization method using BGWO is presented in Section 3. Section 4 provides the experimental results and Section 5 concludes the paper.

2. Echo state network

In this section, the ESN will be briefly described. Two conventional evolutionary algorithms (i.e., GA and BPSO) used for comparison will also be introduced.

2.1 ESN

ESN is a type of recurrent neural networks (RNNs), which is composed of input layer, hidden layer and output layer. Unlike the conventional RNN, the connection weights from the input layer to the hidden layer, and the internal connection weights of the ESN reservoir are randomly initialized. In the process of training, only the connection weights of the hidden layer to the output layer need to be trained. As a linear regression problem, the training of ESN is very fast. The structure of the ESN is shown in the Figure 1. A reservoir is similar to the hidden layer of conventional neural network. The connection weights from the input layer to the reservoir is W_{in}, the connection weights of the neurons in the reservoir is W, and the connection weights from the reservoir to the output layer is W_{out}. In addition, there is a connection W_{back} from output layer to the reservoir. This connection (which is represented by the dotted arrow in the Figure 1) is not necessarily required. When W_{back} exists, the ESN can make multi-step predictions. Otherwise, it can only do single-step prediction.

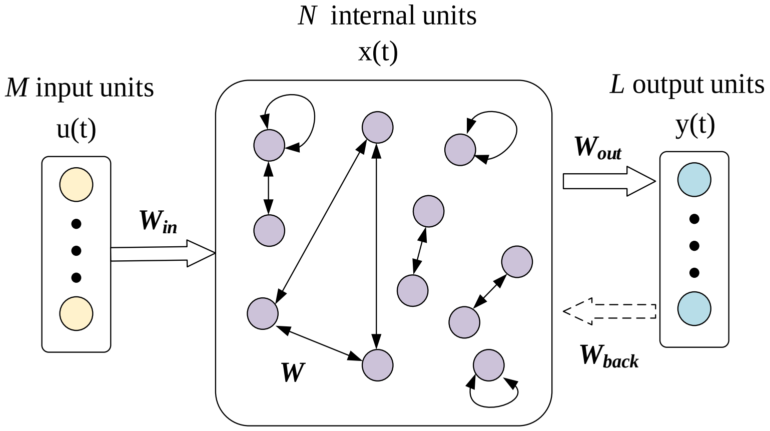

Figure 1. The structure of ESN.

As shown in Figure 1, the input at time *t* is $u(t)$, and there are *M* neuron nodes. The reservoir state is $x(t)$ and there are *N* neuron nodes. The output is $y(t)$, and there are *L* neuron nodes. The reservoir updates its state when each time u is entered. The state update of reservoir is described by

$$x(t+1) = f(Wx(t) + W_{in}u(t+1) + W_{back}y(t)), \qquad (1)$$

where the W_{in} and W are randomly initialized when the network is initially established, and they remain unchanged during training. The $u(t+1)$ is the input of the current time, and $x(t)$ is the reservoir state at the previous time. It can be initialized with zero. f is an activation function, which is usually set to *tanh*.

The output of ESN is described by

$$y(t+1) = f_{out}(W_{out}[x(t+1), u(t+1), y(t)] + W_{bias}^{out}), \qquad (2)$$

where f_{out} is the activation function of output layer, and W_{bias}^{out} is the bias term. After the reservoir state and the output mode of the ESN are determined, W_{out} can be determined according to the target output, so that the error between $y(t)$ and the target output is as small as possible. This is a simple linear regression problem and many methods can be used to calculate it [24]. In addition, ESN has a good advantage for the processing of one-dimensional time series. But for high-dimensional timing series, such as video frames, ESN is not very capable [3].

To make sure the ESN working properly, a few factors need to be considered carefully: (a) because the input from the previous moment will be echoing in the ESN reservoir, to avoid the explosion of the reservoir state, the eigenvalues of W must be less than or equal to one; (b) since W_{out} is the only variable in the network, W must be set large enough to represent as many different data rules as possible; and (c) the sparse degree of W also needs to be set appropriately, which is usually between 1% and 5%.

2.2 Genetic algorithm

Genetic algorithm (GA) is a classic evolutionary algorithm. The original GA has been improved by many researches, which greatly increase the convergence speed and accuracy of the algorithm. It uses the choice, crossover and mutation operations to search for the optimal solution in the problem space. Classic GA first encodes parameters to generate a certain number of individuals. Each individual in the generated initial population can be a one-dimensional or multi-dimensional vector, represented by a string of binary numbers, i.e. a chromosome. Each binary number on a chromosome is called a gene. According to the choice of natural survival of the fittest, the fitness function is designed as a criterion for judging the performance of each individual. Individuals with good performance are selected as parents, and they have certain probability to participate in future genetic operations to generate a new generation of population. After repeating selection, hybridization, and mutation, the optimal population is finally produced.

2.3 PSO and BPSO

Particle swarm optimization (PSO) algorithm has been considered one of the most popular evolutionary algorithms in recent years. Similar to other evolutionary algorithms, PSO also adopts the concepts of “group” and “evolution”. Through the cooperation and competition among individuals, the search for the optimal solution in complex space is realized. The PSO first generates an initial population by randomly initializing a group of particles in the feasible solution space. Each particle is a feasible solution to the optimization problem, and the objective function determines a fitness value for it. The particles will move in the solution space and their direction and distance will be determined by a speed. Usually the particles follow the current optimal particle and move through the generations to get the optimal solution. In each generation, the particles track the positions of two optimal solutions. One is the optimal solution found by the particle itself, and the other is the optimal solution found in the whole population. PSO is often applied to continuous function optimization, but it is not good at solving discrete problems. To address its disadvantages on discrete problems, the binary particle swarm optimization (BPSO) algorithm was proposed [25], which is based on the PSO algorithm. Compare to the PSO, the position and velocity values in the BPSO algorithm are discrete values.

BPSO has a strong global search capability, but it cannot converge to the global optimal value. The approach [21] optimizes BPSO and overcomes the inherent shortcomings, so that the algorithm can converge better. In addition to GA and improved BPSO, the recently proposed BGWO algorithm has been proven to be competitive in the problems of feature selection. The algorithm will be used for the ESN output connection optimization in this work and more details will be introduced in the next section.

3. BGWO-ESN

The optimization of the ESN output connection is a problem of feature selection, and the BGWO algorithm is used to solve this problem in this section. This section introduces the basic theory of GWO and BGWO. Then the BGWO-ESN model is proposed and discussed in detail.

3.1 Grey Wolf Optimization (GWO)

The GWO algorithm is an evolutionary computation technique that mimics the predation behaviour of grey wolves [23]. A wolf group has ~5-12 intelligent individuals. They don’t act alone when they prey, but they are divided into small groups and act together. In the group, according to the dominant hierarchy, the grey wolves are divided into four categories: alpha (α), beta (β), delta (δ), and omega (ω) [23]. Alphas lead the pack and are responsible for making decisions about predation, rest, and action. The primary responsibility of betas is to help alphas in decision making and other activities. Deltas must be subject to alphas and betas, but they can control omegas. Omegas needs to obey all other dominant wolves.

According to [23], the model of grey wolf predation consists of two steps. Firstly, the wolves surround the prey, which is described by

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) + \vec{A} \cdot \vec{D}, \tag{3}$$

where t is the iterations, $\vec{X}$ is the wolf position, $\overrightarrow{X_p}$ is the target position, and $\vec{A}$ is the coefficient constant. $\vec{D}$ is defined by

$$\vec{D} = \left|\vec{C} \cdot \overrightarrow{X_p}(t) - \vec{X}(t)\right|, \tag{4}$$

where $\vec{C}$ is coefficient vector. The $\vec{A}$ and $\vec{C}$ is defined by

$$\vec{A} = 2a \cdot \overrightarrow{r_1} - a, \tag{5}$$

and

$$\vec{C} = 2 \cdot \overrightarrow{r_2}, \tag{6}$$

where a decreases linearly from two to zero as the number of iterations increases. $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ are random vectors in [0, 1]. In the GWO algorithm, alphas are considered to be the best candidate solutions, betas are considered to be the second-best candidate solutions, and deltas are the third-best candidate solutions. Alphas, betas and deltas have the most of knowledge for the position of the food. When their best positions are obtained, other search individuals (including the omegas) are also forced to update their positions. Hence, to further attack the prey, wolves (including omegas) need to update their position by

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3}, \tag{7}$$

where $\overrightarrow{X_1}$, $\overrightarrow{X_2}$ and $\overrightarrow{X_3}$ are calculated by

$$\overrightarrow{X_1} = \left|\overrightarrow{X_\alpha} - \overrightarrow{A_1} \cdot \overrightarrow{D_\alpha}\right|, \tag{8}$$

$$\overrightarrow{X_2} = \left|\overrightarrow{X_\beta} - \overrightarrow{A_2} \cdot \overrightarrow{D_\beta}\right|, \tag{9}$$

$$\overrightarrow{X_3} = \left|\overrightarrow{X_\delta} - \overrightarrow{A_3} \cdot \overrightarrow{D_\delta}\right|, \tag{10}$$

where $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$, $\overrightarrow{X_\delta}$ are the optimal first three solutions for the each iteration, and $\overrightarrow{A_1}$, $\overrightarrow{A_2}$, $\overrightarrow{A_3}$ can be calculated by Eq. (5). The $\overrightarrow{D_\alpha}$, $\overrightarrow{D_\beta}$, $\overrightarrow{D_\delta}$ can be obtained by

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1} \cdot \overrightarrow{X_\alpha} - \vec{X}\right|, \tag{11}$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_2} \cdot \overrightarrow{X_\beta} - \vec{X}\right|, \tag{12}$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_3} \cdot \overrightarrow{X_\delta} - \vec{X}\right|, \tag{13}$$

where $\overrightarrow{C_1}$, $\overrightarrow{C_2}$ and $\overrightarrow{C_3}$ are calculated by Eq. (6). These two steps are repeated until the wolves capture the prey.

3.2 BGWO

In GWO, wolves can change their continuous positions to find prey. But in some specific tasks (such as feature selection), the problem is in binary space, and the solution is limited to zero and one values, which becomes a problem for the typical GWO. Hence, the BGWO has been proposed for feature selection tasks, where all solutions are represented in binary form. Two position update algorithms in the approach of [23] are used in this work, where the position update algorithm 1 (PUA1) and the position update algorithm 2 (PUA2) are expressed by

$$x_d^{t+1} = \begin{cases} x_1^d, & if\ rand < \frac{1}{3} \\ x_2^d, & \frac{1}{3} \leq rand < \frac{2}{3}, \\ x_3^d, & otherwise \end{cases} \tag{14}$$

and

$$x_d^{t+1} = \begin{cases} 1, & if\ sigmoid(\frac{x_1 + x_2 + x_3}{3}) \geq rand \\ 0, & otherwise \end{cases}, \tag{15}$$

respectively, where $rand$ is a random number of [0, 1] and obeys the uniform distribution. The x_d^{t+1} is the updated *d-dimensional* binary wolf position after the *t-th* iteration. The $sigmoid$ is defined by

$$sgimoid(x) = \frac{1}{1 + e^{-10(x-0.5)}}, \tag{16}$$

and the x_1, x_2, x_3 are binary vectors representing the effect of wolf move towards the alpha, beta, delta grey wolves. They are defined by

$$x_1^d = \begin{cases} 1, & if\ (x_\alpha^d + bstep_\alpha^d) \geq 1 \\ 0, & otherwise \end{cases}, \tag{17}$$

$$x_2^d = \begin{cases} 1, & if\ (x_\beta^d + bstep_\beta^d) \geq 1 \\ 0, & otherwise \end{cases}, \tag{18}$$

$$x_3^d = \begin{cases} 1, & if\ (x_\delta^d + bstep_\delta^d) \geq 1 \\ 0, & otherwise \end{cases}, \tag{19}$$

where x_α^d, x_β^d, x_δ^d are the positions of the alpha, beta and delta wolves respectively, and $bstep_\alpha^d$, $bstep_\beta^d$, $bstep_\delta^d$ are given by

$$bstep_\alpha^d = \begin{cases} 1, & if\ \ cstep_\alpha^d \geq rand \\ 0, & otherwise \end{cases}, \tag{20}$$

$$bstep_\beta^d = \begin{cases} 1, & if\ \ cstep_\beta^d \geq rand \\ 0, & otherwise \end{cases}, \tag{21}$$

$$bstep_\delta^d = \begin{cases} 1, & if\ \ cstep_\delta^d \geq rand \\ 0, & otherwise \end{cases}, \tag{22}$$

where $cstep_\alpha^d$, $cstep_\beta^d$, $cstep_\delta^d$ are described by

$$cstep_\alpha^d = \frac{1}{1 + e^{-10(A_1^d D_\alpha^d - 0.5)}}, \tag{23}$$

$$cstep_\beta^d = \frac{1}{1 + e^{-10(A_1^d D_\beta^d - 0.5)}}, \tag{24}$$

$$cstep_\delta^d = \frac{1}{1 + e^{-10(A_1^d D_\delta^d - 0.5)}}, \tag{25}$$

where the A_1^d, D_α^d, D_β^d and D_δ^d are calculated by Eq. (5), (11), (12), (13).

3.3 BGWO-ESN

In this paper, the BGWO algorithm is used to optimize the output connection structure of ESN, with the aim to remove the redundant connections between the reservoir and the output layer, i.e. besides the conventional connection weight computation, some of the connection weights in W_{out} will be selectively set to zero. The disconnection and connection of with respect to the output connectivity are hereby represented by 1 and 0, respectively. The optimization variable can then form a binary matrix (corresponding to the state of the ESN output connection), and its dimension is equal to the number of neurons in the reservoir.

The pseudocode and detailed optimization process are given by Algorithm 1, which includes the following steps:

(a) Initialize the parameters of the BGWO algorithm as shown by line #1 in the pseudocode, including the number of search agents, maximum number of iterations, and positions of search agents (which are represented by the matrices consisting of only 0 and 1).

(b) Initialize the ESN model (line #2). A properly sized reservoir is created using *tanh* as an internal activation function. The sparseness of the reservoir is maintained at 1% to 5%. The spectral radius needs to be less than or equal to 1 to ensure the stability of the model. Input weights and internal weights are randomly generated and remain unchanged during the subsequent training. Feedback connection weights can be set according to specific tasks.

(c) Calculate the objective function for each search agent (line #4-#8). In this paper, the ESN output connection (represented by the binary matrix) is used as the position of grey wolf, the normalized mean square error and root mean squared error between each predicted and actual values are used as the objective function, and the minimum value of the function is defined as the search target.

(d) Update alpha, beta and delta (line #10). They are calculated once per iteration.

(e) Update the position of search agents, including the omegas (line #11).

(f) Perform iterative optimization (line #3-#12). BGWO repeatedly updates and adjusts the wolf position until reaching the maximum iterations or the target accuracy, i.e. the iteration stops when the best connection weight vector is obtained.

(g) The optimal connection weight is applied to the ESN, and then the performance of the optimized ESN is evaluated (line #13).

Algorithm 1. BGWO-ESN algorithm.

```
Input: data ($\boldsymbol{X}$,$\boldsymbol{Y}$), reservoir size $M$, number of iterations for optimization $N_{iter}$,
number of grey wolves in the pack $n$;
Output: trained BGWO-ESN;
1.  Initialize a population of $n$ wolves positions at random∈[0,1] (step (a));
2.  Initialize the ESN, generate $\boldsymbol{W}_{\boldsymbol{in}}$, $\boldsymbol{W}$ and $\boldsymbol{W}_{\boldsymbol{back}}$ randomly (step (b));
3.  While stopping criteria not met do (step (f))
4.      for $wolf_i$ ∈ $pack$ do
5.          set ESN output connection to $wolf_i$ position;
6.          compute $\boldsymbol{W}_{\boldsymbol{out}}$;
7.          calculate objective function (step (c));
8.      endfor
9.      update $a$, $A$, $C$;
```

```
        update $\alpha$, $\beta$, $\delta$ (step (d));
        update $wolf_i$ position to a binary position according to Eq.(14), (15)(step (e));
    End
    Obtain optimal $\boldsymbol{W_{out}}$ (step (g));
```

4. Results

In this section, the two binary versions of BGWO algorithm (i.e., PUA1 and PUA2) are exploited in the optimization of the ESN output connection. Their performances are evaluated by two sets of experiment to demonstrate its generalization: the first one is a widely used benchmarking database for time series prediction, and the other is the financial datasets of the Standard & Poor's 500 index and high frequency foreign exchange data. For each experiment, a comparison of the accuracies with and without BGWO is performed. BGWO-ESN is compared to not only the original ESN but also other similar hybrid models (GA-ESN, BPSO-ESN). In fact, the network architecture of BGWO-ESN is critical in training process, especially for the size (N) and connectivity rate of the reservoir. These parameters are set based on the corresponding tasks. For BGWO, the number of search agents is 12, and the maximum number of iterations is 300. The problem dimension is the number of features in the data. In addition, the normalized mean square error (NMSE) and root mean squared error (RMSE) given in Eq. (26) and (27) are used for evaluation to intuitively analyse the performance of BGWO. All experiments were run on MATLAB R2016b platform. To avoid the effects of random initialization of some network parameters, each experiment is repeated 10 times and the average is chosen as the result.

$$NMSE = \sum_{i=1}^{M} \frac{(Q_i - P_i)^2}{M \cdot \sigma^2}, \tag{26}$$

$$RMSE = \sqrt{\sum_{i=1}^{M} \frac{(Q_i - P_i)^2}{M}}, \tag{27}$$

where Q_i, P_i are predicted and target value, respectively. M is the number of data samples. σ^2 is the variance of P_i.

4.1 Evaluation using synthetic data

In this section, two benchmarking sequences (i.e., nonlinear autoregressive moving average (NARMA) and mackey–glass (MG) time series) are used to test the proposed method.

1) NARMA prediction: NARMA is one of the popular benchmarks that has been widely used for ESN testing [13]. One appealing feature of the NARMA sequence is its unpredictable complexity, due to its high degrees of confusion and non-correlation of inputs. A detailed explanation of this sequence is described in [26], and its dynamic expression is given by

$$y(t+1) = c_1 y(t) + c_2 y(t) \sum_{i=0}^{k-1} y(t-i) + c_3 x\big(t-(k-1)\big)x(t) + c_4, \qquad (28)$$

where $y(t)$ and $x(t)$ are the output and input of the system at time t, respectively. The constant parameters c_i are set as 0.3, 0.05, 1.5, 0.1, respectively. Parameter k determines the complexity of NARMA. In general, k is either set to 10 or 30, which are the most common choices used in the literature [13]. In this test, k was 10. The input signal was assumed to be independent identically distributed random samples, and its value is in the range [0, 0.5].

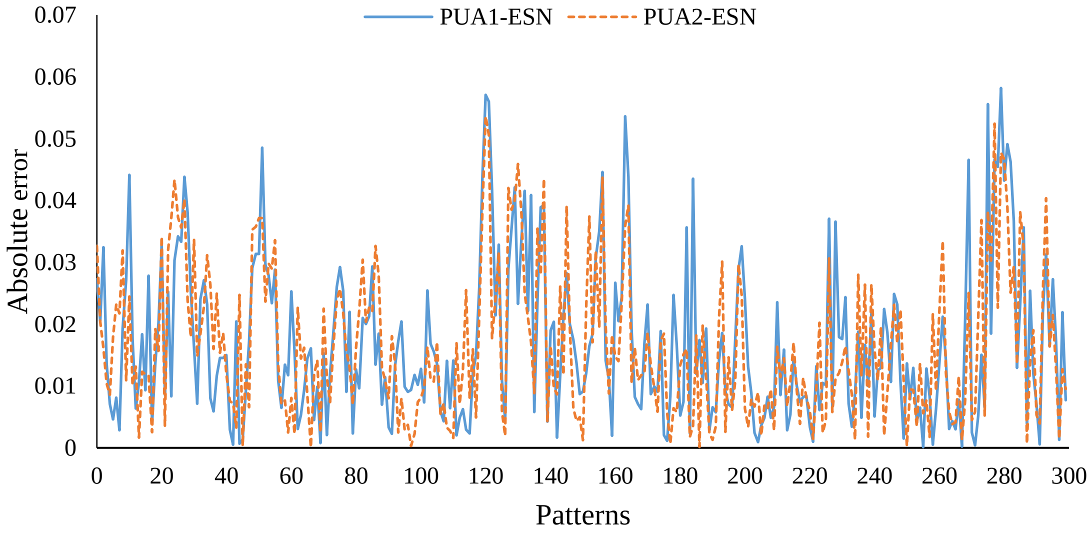

Figure 2. Evolution of the absolute error signal between the testing patterns and the desired ones (NARMA, order k=10).

In the experiment, 3,600 items were generated, where the first 1,200 items were used for training and 1,200 items were used for validation. The remaining 1,200 were used for testing. The size of the reservoir (N) were set to 200, 300 and 500, respectively. The connectivity rate was set to 5%. In addition, the output activation function of BGWO-ESN model was *linear*, and the W_{back} was zero matrix. Figure 2 tracks the absolute error between the test results of BGWO-ESN (including PUA1-ESN and PUA2-ESN) and the expected values, respectively. In order to further evaluate the performance of BGWO-ESN, five other conventional methods are chosen for performance comparisons, including pruning ESN output connection randomly (RAND-ESN), optimizing ESN output connection by least angle regression (LAR-ESN), optimizing ESN output connection by GA (GA-ESN), optimizing ESN output connection by BPSO (BPSO-ESN) and the original ESN. Table I, Table II and Table III give the comparison results.

Table I. NMSEs of ESN, LAR-ESN, RAND-ESN, GA-ESN, BPSO-ESN and BGWO-ESN under the NARMA predication task when *N*=200.

Method		NMSE	Accuracy improvement	Running time
ESN		0.0491	-	-
LAR-ESN [27]		0.0439	11%	-
RAND-ESN		0.0811	-39%	-
GA-ESN		0.0475	3%	12252s
BPSO-ESN [27]		0.0428	13%	-
BGWO-ESN	PUA1	0.0459	7%	1253s
	PUA2	**0.0419**	**15%**	**1109s**

Table II. NMSEs of ESN, LAR-ESN, RAND-ESN, GA-ESN, BPSO-ESN and BGWO-ESN under the NARMA predication task when *N*=300.

Method		NMSE	Accuracy improvement	Running time
ESN		0.0448	-	-
LAR-ESN [27]		0.0374	17%	-
RAND-ESN		0.0738	-39%	-
GA-ESN		0.0426	11%	15181s
BPSO-ESN [27]		0.0369	18%	-
BGWO-ESN	PUA1	0.0370	17%	1302s
	PUA2	**0.0315**	**27%**	**1221s**

Table III. NMSEs of ESN, LAR-ESN, RAND-ESN, GA-ESN, BPSO-ESN and BGWO-ESN under the NARMA predication task when *N*=500.

Method		NMSE	Accuracy improvement	Running time
ESN		0.0411	-	-
LAR-ESN [27]		0.0370	10%	-
RAND-ESN		0.0554	-26%	-
GA-ESN		0.0363	12%	17056s
BPSO-ESN [27]		0.0346	16%	-
BGWO-ESN	PUA1	0.0348	15%	**2625s**
	PUA2	**0.0293**	**29%**	2739s

Table I, Table II and Table III show that the ESN is greatly improved performance due to the use of BGWO algorithm to optimize the ESN output connection, where PUA2-ESN achieved the best performance in different reservoir size tasks. It also shows the superiority of the BGWO algorithm compare to other optimization algorithms. RAND-ESN got the worst result, and its average NMSE was 0.0701, which was 35% lower than ESN, i.e. blindly cutting the output connection could not improve model performance. The GA-ESN model also failed to achieve satisfactory results, and it was not even as good as the traditional LAR-ESN model. This maybe because there are many decision variables and large dimensions in the task, so GA algorithm cannot solve such problems well. In addition, GA algorithm often needs to adopt a large population and a large number of iterations, which leads to a long running time. The BGWO algorithms only need 5-12 individuals, and running times are very short, which is a great advantage over other evolutionary algorithms.

2) MG time series prediction: The MG is a typical chaotic system, which can be described by

$$\frac{dx(t)}{dt} = \frac{0.2 \cdot x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1 \cdot x(t), \tag{29}$$

where τ is an important parameter of the MG system and has been discussed in detail in [28]. τ is adjustable, which is often set to 17 and 30 [13]. In this work, τ was set to 17 for best system performance. 500 samples were used as training data sets and 500 samples were used for testing. The size of the reservoir was set to 200. Feedback connection was required for testing tasks at different reservoir sizes (N=50, 100 and 200). Figure 3 tracks the absolute error between the output values of BGWO-ESN (including PUA1-ESN and PUA2-ESN) and the target values. Table IV, Table V, Table VI and Table VII present a series of statistical comparisons to further explore the proposed method and its effectiveness in prediction tasks.

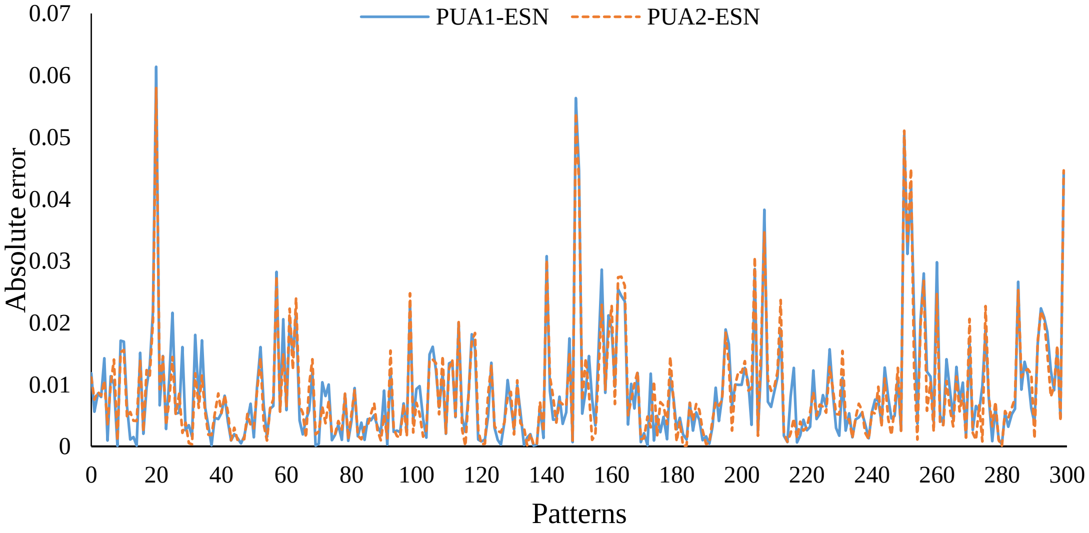

Figure 3. Evolution of the absolute error signal between the testing patterns and the desired ones (MG sequence, order τ=17).

Table IV. NMSEs of ESN, LAR-ESN, RAND-ESN, GA-ESN, BPSO-ESN and BGWO-ESN under the MG time series predication task when *N*=50.

Method		NMSE	Accuracy improvement	Running time
ESN		5.08e-03	-	-
RAND-ESN		7.24e-03	-30%	-
GA-ESN		4.90e-03	4%	1502s
BGWO-ESN	PUA1	4.52e-03	11%	342s
	PUA2	**4.18e-03**	**18%**	**338s**

Table V. NMSEs of ESN, LAR-ESN, RAND-ESN, GA-ESN, BPSO-ESN and BGWO-ESN under the MG time series predication task when *N*=100.

Method		NMSE	Accuracy improvement	Running time
ESN		3.33e-03	-	-
RAND-ESN		4.81e-03	-31%	-
GA-ESN		3.32e-03	0.3%	1702s
BGWO-ESN	PUA1	3.09e-03	7%	388s
	PUA2	**2.73e-03**	**18%**	**372s**

Table VI. NMSEs of ESN, LAR-ESN, RAND-ESN, GA-ESN, BPSO-ESN and BGWO-ESN under the MG time series predication task when *N*=200.

Method		NMSE	Accuracy improvement	Running time
ESN		4.56e-03	-	-
RAND-ESN		5.53e-03	-18%	-
GA-ESN		3.01e-03	34%	1937s
BGWO-ESN	PUA1	2.37e-03	48%	463s
	PUA2	**2.09e-03**	**54%**	**452s**

Table VII. The comparison of BGWO-ESN and other approaches under the Mackey and Glass time series task where delay τ=17.

Method	RMSE
ESN	1.72e-02
HMDDE-BBFNN [29]	1.70e-02

GA-BBFNN [30]		1.30e-02
Classical RBF [31]		1.14e-02
BGWO-ESN	**PUA1**	**1.16e-02**
	PUA2	**1.03e-02**

As can be seen from Table IV, Table V and Table VI, PUA2-ESN achieved the smallest NMSE value, i.e. it facilitated the best generalization performance for this MG predication task. The average prediction accuracies of PUA2-ESN were increased by 30%, 21% and 11%, respectively, compared to the standard ESN, GA-ESN and PUA1-ESN approaches. The results shown in Table VII also indicate the proposed PUA2-ESN achieved a good performance for the MG predication task.

The experimental results in this subsection show that the proposed BGWO-ESN demonstrated the reduced generalization errors for the predication tasks under NARMA and MG tasks, compared to other evolutionary methods. BPSO-ESN and GA-ESN achieved better test results than ESN, but their running times were generally higher than BGWO-ESN. The prediction accuracy of RAND-ESN was the lowest, suggesting that randomly pruning ESN output connections was detrimental. Additionally, the experimental results also demonstrate the effectiveness of BGWO as feature selection algorithm. In the next section, the real-world stock index data and foreign exchange data will be used to evaluate the BGWO-ESN model.

4.2 Tested using real-world financial datasets

The Standard & Poor's 500 index and high frequency foreign exchange data were used to further explore the performance of the BGWO-ESN in the real-life scenario.

1) S&P 500 index dataset: The S&P 500 index is a world-wide stock index. Data of this dataset is considered quite challenging for prediction due to its strong nonlinearity. This experiment will use the S&P 500 index as a research object to perform simple single-step prediction. The BGWO-ESN has four input variables (open, high, low, and close) and one output variable (opening price of the next day). In this task, the first 15,000 data samples are used for training, and the rest 2,000 rows are used for testing. Based on our pilot study, the input layer of the model has been set with four neurons and the output layer has one neuron. The size of the reservoir is 200 and the connectivity rate is 5%. In addition, the activation functions of the reservoir and the output layer are set to *tanh* and *identity*, respectively.

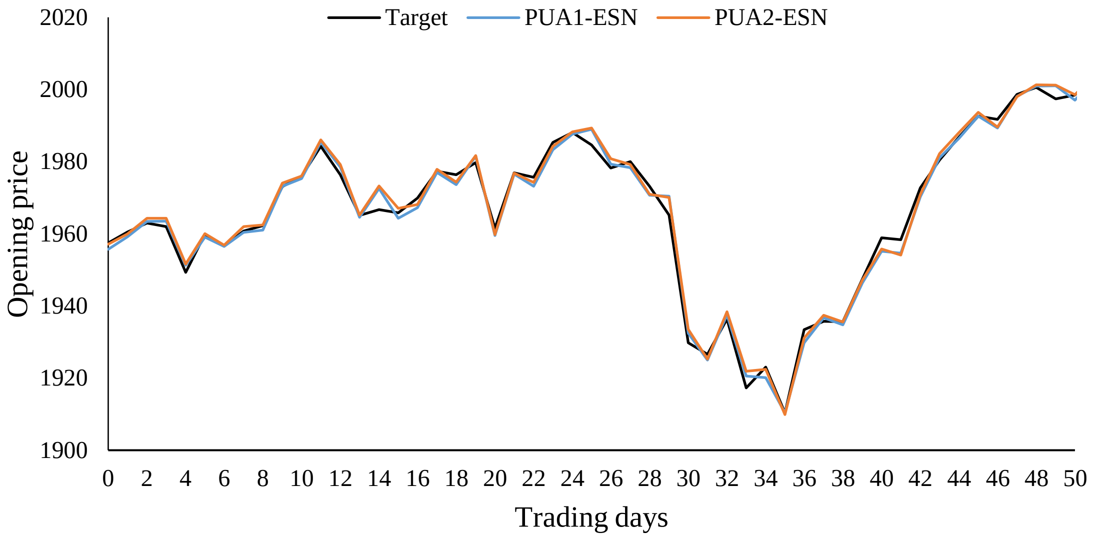

Figure 4. Results of opening prices, S&P 500 index using PUA1-ESN and PUA2-ESN methods.

Table VIII. RMSEs of different methods for S&P 500 index predication.

Method		RMSE	Accuracy improvement
ESN		61.8445	-
RAND-ESN		73.1039	-15%
GA-ESN		17.8065	44%
BPSO-ESN		4.8872	92%
BGWO-ESN	PUA1	4.5423	93%
	PUA2	**3.4957**	**94%**

Figure 4 shows the actual opening price and the prediction outputs of PUA1-ESN and PUA2-ESN. Both methods appear to fit the curve moderately well. To further verify the performance of the proposed PUA1-ESN and PUA2-ESN, the RMSE values were computed and compared with a few other evolutionary algorithms (e.g. GA and BPSO), which were also applied to ESN for network connectivity evaluation. The experimental results are shown in Table VIII.

Results in Table VIII shows that the PUA2-ESN model received the lowest generalization error amongst other models. In consistent with the conclusions of [15], it is proved that the optimization of the ESN output connection is advantageous. The proposed PUA2-ESN hybrid algorithm achieved the best performance with RMSE of 3.4957. Compared to standard ESN, GA-ESN, BPSO-ESN and PUA1-ESN, the accuracy of PUA2-ESN is increased by 94%, 80%, 28% and 23%, respectively.

2) *EURO/USD exchange dataset*: The EURO/USD database is featured by its high frequency (minute-sampling level) foreign data exchanging rate, which has

strong nonlinearity and complexity. In this experiment, the BGWO-ESN was applied to the dataset for single-step prediction. The input of BGWO-ESN has four financial variables, i.e., open, high, low, and close prices; the output is one variable, which indicates the opening price of the next minute. Therefore, the number of input nodes is set to four, the number of output node is set to one. In this task, the first 10,000 data samples are used for training, and the rest 5,000 rows are used for testing. Based on our empirical exploration, the number of nodes in the ESN reservoir is set to 200, and the connectivity rate of the reservoir is 5%. Figure 5 shows the target value and the output of the PUA1-ESN, PUA2-ESN. Meanwhile, Table IX shows the comparison results of BGWO-ESN and the other two methods (GA-ESN, BPSO-ESN).

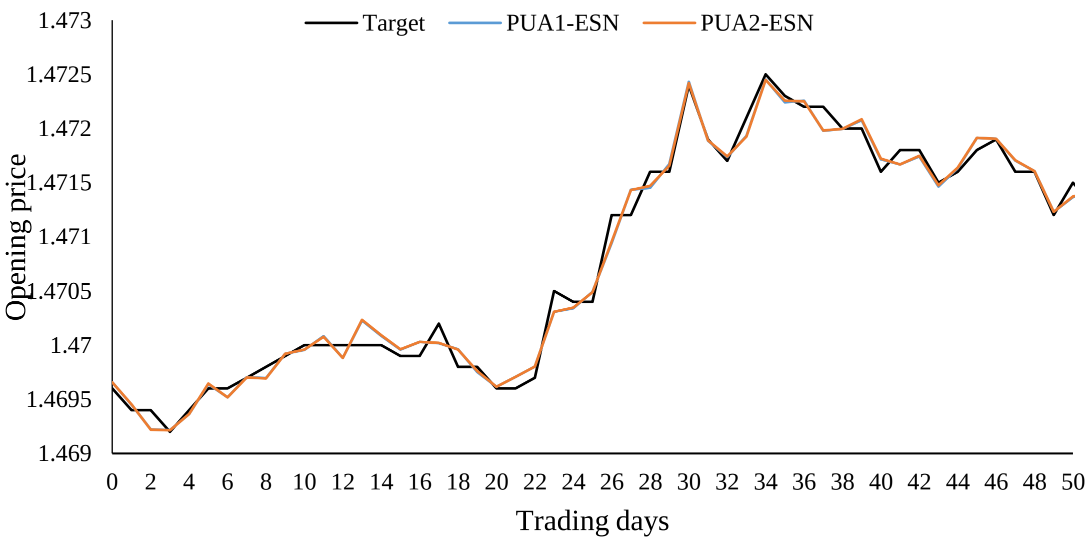

Figure 5. Results of opening prices, EURO/USD using PUA1-ESN and PUA2-ESN methods.

Table IX. RMSEs of four methods for EURO/USD predication.

Method		RMSE	Accuracy improvement
ESN		2.42e-04	-
RAND-ESN		4.92e-04	-51%
GA-ESN		2.02e-04	17%
BPSO-ESN		1.23e-04	49%
BGWO-ESN	PUA1	1.98e-04	18%
	PUA2	**9.85e-05**	**59%**

It can be seen that the proposed PUA2-ESN model has a good performance on the foreign exchange data set, which received an improvement of 59%, 51%, 20% and 50% compared to the ESN, GA-ESN, BPSO-ESN and PUA1-ESN, respectively. The results of RAND-ESN are still found been the worst, which means the optimization of

ESN output connection requires an appropriate algorithm, and PUA2-ESN is the solution we proposed in this study. These experimental results indicate the optimization of the ESN output connection can reduce the generalization error. At the same time, it is also verified that PUA2 [23] can improve the ESN performance by optimizing the connections between the reservoir and output layer.

5. Conclusion

ESN reservoir can map input signals to high-dimensional space, but not every dimension directly contributes to the output neurons, i.e. certain connections from the reservoir to the output layer are redundant. In this work, a hybrid methods combining ESN and BGWO was proposed to address this problem. Data was used to train a properly sized ESN reservoir. BGWO was then used to optimize the output connection. The proposed method was used for predications under the synthetic benchmarks and financial time series. Two other evolutionary algorithms (i.e., GA and BPSO) were compared to evaluate the performance of BGWO-ESN. The results show that compared to conventional ESN, the proposed methods of BGWO-ESN (including PUA1-ESN and PUA2-ESN) are able to reduce the generalization error. Moreover, the PUA2-ESN model achieves the best performance, and has a great advantage in running speed compared to other algorithms. Therefore, BGWO is an excellent optimization algorithm for feature selection, which has improved the predictive power of ESN. The proposed model has "echo" characteristic, which suits time series prediction tasks very well [3]. In the future, we will further optimize the model and apply the proposed model in other application domains, such as pattern recognition, robot control, event monitoring etc.

Acknowledgements

This research is supported by the National Natural Science Foundation of China under Grants 61976063 and 61603104, the Guangxi Natural Science Foundation under Grant 2017GXNSFAA198180, the funding of Overseas 100 Talents Program of Guangxi Higher Education.

References

[1] M. Hüsken and P. Stagge, "Recurrent neural networks for time series classification," *Neurocomputing*, vol. 50, no. C, pp. 223–235, 2003.

[2] M. Lukoševičius and H. Jaeger, "Survey: Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, 2009.

[3] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural

networks," *Technol. GMD Tech. Rep. 148, Ger. Natl. Res. Cent. Information, Ger.*, 2001.

[4] W. Maass, T. Natschl, and H. Markram, "Real-time computing without stable states: a new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, 2002.

[5] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural Networks*, vol. 20, no. 3, pp. 391–403, 2007.

[6] S. E. Lacy, S. L. Smith, and M. A. Lones, "Using echo state networks for classification: a case study in Parkinson's disease diagnosis," *Artif. Intell. Med.*, vol. 86, pp. 53–59, 2018.

[7] X. Lin, Z. Yang, and Y. Song, "Short-term stock price prediction based on echo state networks," *Expert Syst. Appl.*, vol. 36, no. 3 PART 2, pp. 7313–7317, 2009.

[8] X. Yao, Z. Wang, and H. Zhang, "A novel photovoltaic power forecasting model based on echo state network," *Neurocomputing*, vol. 325, pp. 182–189, 2018.

[9] J. P. Jordanou, E. Camponogara, E. A. Antonelo, and M. A. Schmitz Aguiar, "Nonlinear model predictive control of an oil well with echo state networks," *IFAC-PapersOnLine*, vol. 51, no. 8, pp. 13–18, 2018.

[10] Y. Yao *et al.*, "A paired neural network model for tourist arrival forecasting," *Expert Syst. Appl.*, vol. 114, pp. 588–614, 2018.

[11] M. Xu, M. Han, and S. Member, "Adaptive elastic echo state network for multivariate time series prediction," *IEEE Trans. Cybern.*, vol. 46, pp. 2173–2183, 2017.

[12] J. Liu, T. Sun, Y. Luo, Q. Fu, and Y. Cao, "Financial data forecasting using optimized echo state network," in *International Conference on Neural Information Processing*, 2018, pp. 138–149.

[13] N. Chouikhi, B. Ammar, N. Rokbani, and A. M. Alimi, "PSO-based analysis of echo state network parameters for time series forecasting," *Appl. Soft Comput.*, vol. 55, pp. 211–225, 2017.

[14] A. Rodan, S. Member, and P. Tiˇ, "Minimum complexity echo state network," *IEEE Trans. Neural Networks*, vol. 22, no. 1, pp. 131–144, 2011.

[15] X. Dutoit, B. Schrauwen, J. Van Campenhout, D. Stroobandt, H. Van Brussel, and M. Nuttin, "Pruning and regularization in reservoir computing," *Neurocomputing*, vol. 72, no. 7, pp. 1534–1546, 2009.

[16] H. U. Kobialka and U. Kayani, "Echo state networks with sparse output

connections," in *International Conference on Artificial Neural Networks*, 2010, pp. 356 – 361.

[17] J. Liu *et al.*, "An Optimized Image Watermarking Method Based on HD and SVD in DWT Domain," *IEEE Access*, vol. 7, pp. 80849–80860, 2019.

[18] J. Liu, X. Huang, Y. Huang, Y. Luo, and S. Yang, "Multi-objective spiking neural network hardware mapping based on immune genetic algorithm," in *Artificial Neural Networks and Machine Learning – ICANN 2019: Theoretical Neural Computation*, 2019, pp. 745–757.

[19] W. Guo, M. Jiang, X. Li, and B. Ren, "Using a genetic algorithm to improve oil spill prediction," *Mar. Pollut. Bull.*, vol. 135, pp. 386–396, 2018.

[20] G. Shao, Y. Shangguan, J. Tao, J. Zheng, T. Liu, and Y. Wen, "An improved genetic algorithm for structural optimization of Au–Ag bimetallic nanoparticles," *Appl. Soft Comput. J.*, vol. 73, pp. 39–49, 2018.

[21] H. Nezamabadi-pour, M. Rostami-sharbabaki, and M. Maghfoori-Farsangi, "Binary particle swarm optimization: challenges and new solutions," *J. Comput. Soc. Iran Comput. Sci. Eng.*, vol. 6, no. 1-A, pp. 21–32, 2008.

[22] M. A. Al-Betar, M. A. Awadallah, H. Faris, I. Aljarah, and A. I. Hammouri, "Natural selection methods for grey wolf optimizer," *Expert Syst. Appl.*, vol. 113, pp. 481–498, 2018.

[23] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016.

[24] J. Liu, T. Sun, Y. Luo, S. Yang, Y. Cao, and J. Zhai, "An echo state network architecture based on quantum logic gate and its optimization," *Neurocomputing*, vol. 371, pp. 100–107, 2019.

[25] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *IEEE International Conference on Systems, Man, and Cybernetics*, 1997, pp. 4–8.

[26] A. F. Atiya and A. G. Parlos, "New results on recurrent network training: unifying the algorithms and accelerating convergence," *IEEE Trans. Neural Networks*, vol. 11, no. 3, pp. 697–709, 2000.

[27] H. Wang and X. Yan, "Optimizing the echo state network with a binary particle swarm optimization algorithm," *Knowledge-Based Syst.*, vol. 86, no. C, pp. 182–193, 2015.

[28] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[29] A. Miranian and M. Abdollahzade, "Developing a local least-squares support

vector machines-based neuro-fuzzy model for nonlinear and chaotic time series prediction," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 24, no. 2, pp. 207–218, 2013.

[30] R. Akbari and K. Ziarati, "A cooperative approach to bee swarm optimization," *J. Inf. Sci. Eng.*, vol. 27, no. 3, pp. 799–818, 2011.

[31] K. B. Cho and B. H. Wang, "Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction," *Fuzzy Sets Syst.*, vol. 83, no. 3, pp. 325–339, 1996.