



2019

## Structure Optimization for Echo State Network Based on Contribution

Dingyuan Li

*College of Communication Engineering, Jilin University, Changchun 130022, China,*

Fu Liu

*College of Communication Engineering, Jilin University, Changchun 130022, China,*

Junfei Qiao

*Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China,*

Rong Li

*Department of Information Technology, Beijing Vocational College of Agriculture, Beijing 100124, China.*

Follow this and additional works at: <https://tsinghuauniversitypress.researchcommons.org/tsinghua-science-and-technology>



Part of the [Computer Sciences Commons](#), and the [Electrical and Computer Engineering Commons](#)

### Recommended Citation

Dingyuan Li, Fu Liu, Junfei Qiao et al. Structure Optimization for Echo State Network Based on Contribution. *Tsinghua Science and Technology* 2019, 24(1): 97-105.

This Research Article is brought to you for free and open access by Tsinghua University Press: Journals Publishing. It has been accepted for inclusion in *Tsinghua Science and Technology* by an authorized editor of Tsinghua University Press: Journals Publishing.

# Structure Optimization for Echo State Network Based on Contribution

Dingyuan Li, Fu Liu\*, Junfei Qiao, and Rong Li

**Abstract:** Echo State Network (ESN) is a recurrent neural network with a large, randomly generated recurrent part called the dynamic reservoir. Only the output weights are modified during training. However, proper balancing of the trade-off between the structure and performance for ESN remains a difficult task. In this paper, a structure optimized method for ESN based on contribution is proposed to simplify its network structure and improve its performance. First, we evaluate the contribution of reservoir neurons. Second, we present a pruning mechanism to remove the unimportant connection weights of reservoir neurons with low contribution. Finally, the new output weights are learned with the pseudo inverse method. The novel optimized ESN, named C-ESN, is tested on a Lorenz chaotic time-series prediction and an actual municipal sewage treatment system. The simulation results show that the C-ESN can have better prediction and generalization performance than ESN.

**Key words:** neural network; structural design; time-series prediction

## 1 Introduction

The Echo State Network (ESN), first proposed by Jaeger<sup>[1,2]</sup>, is a novel kind of Recurrent Neural Network (RNN). Unlike the traditional RNN, the ESN usually comprises a large number of internal neurons (typically, RNNs contain 5–30 internal neurons) and only the output weights are trained using linear regression method, overcoming problems such as the local optima and vanishing gradients. Nowadays, owing to its simplicity and effectiveness, the ESN has been successfully applied in different fields, such as language modeling<sup>[3]</sup>, radio signal strength prediction<sup>[4]</sup>, lake water level forecasting<sup>[5]</sup>, microscopic cellular image

segmentation<sup>[6]</sup>, and motor control<sup>[7]</sup>.

The application of the ESN necessitates balancing the trade-off between network structure and network performance. If the network structure is too small, the network may present limited information processing capacity. The ESN cannot achieve satisfactory accuracy for complex problems. On the other hand, if the network structure is too large, the network may exhibit poor generalization performance during the testing process. Thence, selecting the optimal network structure for operators in practice often presents difficulty.

In recent years, many scholars have focused their efforts on new methods to optimize the structure of the ESN. Wang and Yan<sup>[8]</sup> proposed a sensitive iterative pruning algorithm method to optimize the Simple Cycle Reservoir Network (SCRN). The algorithm was used to prune out the least sensitive internal units one by one according to the sensitive analysis. The results demonstrated that the proposed method can optimize the structure and improve the generalization performance of the SCRN. However, SCRN is a kind of minimal-complexity ESN with deterministically constructed connectivity and weight structure<sup>[9]</sup>. Unlike SCRN, the original ESN includes a dynamic reservoir

- 
- Dingyuan Li and Fu Liu are with College of Communication Engineering, Jilin University, Changchun 130022, China, E-mail: dyl12@mails.jlu.edu.cn; liufu@jlu.edu.cn.
  - Junfei Qiao is with Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China, E-mail: junfei@bjut.edu.cn.
  - Rong Li is with Department of Information Technology, Beijing Vocational College of Agriculture, Beijing 100124, China. E-mail: 445422612@qq.com.

\* To whom correspondence should be addressed.

Manuscript received: 2018-01-03; accepted: 2018-01-10

with random topologies and weights. However, several reservoir properties of ESN are poorly understood. Consequently, difficulty arises from determining which reservoir neurons are actually useful for the output.

For the structure optimization design, several methods are used to remove the redundant connection weights to increase the performance of ESN. An effective algorithm for pruning redundant connections during training, which was proposed by Scardapane et al.<sup>[10]</sup>, is the only one that removes connections inside the dynamic reservoir. However, pruning operations on the reservoir can potentially disrupt the echo state property. Meanwhile, more researchers have concentrated on pruning redundant connections from the reservoir to the readout layer to optimize the ESN structure. The regularization methods are the most frequently used techniques to optimize the network structure. For example, Dutoit et al.<sup>[11]</sup> proposed several regularization methods, such as backward selection, least angle regression, and random deletion, to prune readout connections. These algorithms can increase the generalization capability of the ESN. However, the optimal value of the regularization parameter is difficult to set based on the dynamical organization of reservoirs and the number of training patterns. Moreover, the evolutionary computation algorithms have been applied to solve the problem of structure optimization for the ESN<sup>[12,13]</sup>. These algorithms include particle swarm optimization and genetic algorithms, which are used to optimize the weight connection structures of the ESN. However, the evolutionary computation technique is a heuristic algorithm that features high computational complexity and computational cost. Hence, this technique results in another problem: how to prune the redundant connection weights in an efficient manner while improving the network performance. To date, the method to optimize the network structure and performance remains an open problem in the field of ESNs.

Motivated by the above problems, in this paper, we propose a structure optimization approach based on the contribution of ESN. First, analyzing the information contribution between the reservoir neurons and output neurons, we define an effective criterion to measure the contribution of reservoir neurons. Second, a pruning mechanism is proposed to prune out the unimportant output connection weights of reservoir neurons with low contribution. Finally, the pseudo inverse method is used to compute the important output weights. To verify

the effectiveness of this proposed method, the C-ESN is tested on the Lorenz chaotic time-series prediction and an actual municipal sewage treatment system in a Waste Water Treatment Plant (WWTP).

The rest of this paper is organized as follows. Section 2 provides an overview of the ESN. Section 3 presents the details of the proposed structure optimization algorithm for the ESN based on contribution. Section 4 discusses the experimental results and analysis to demonstrate the performance of the C-ESN. Finally, Section 5 describes the conclusion based on this work.

## 2 Echo State Network

ESN is a new class of RNN. The special features of the ESN include a dynamic reservoir, which is a hidden layer with a number of random recurrent connections, and a simple training procedure, in which only the output weights are adaptable<sup>[2]</sup>. Figure 1 illustrates the basic structure of an original ESN. The ESN comprises an input layer with  $K$  input neurons, a reservoir layer with  $N$  reservoir neurons, and an output layer with  $L$  output neurons.

The state of reservoir neurons can be calculated according to the following:

$$\mathbf{x}(n+1) = f(\mathbf{W}_{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}_{fb}\mathbf{y}(n)) \quad (1)$$

where  $\mathbf{u}(n+1)$  refers to the input vector at time  $n+1$ ,  $\mathbf{x}(n)$  is the reservoir neuron state at time  $n$ , and  $\mathbf{y}(n)$  is the output vector at time  $n$ .  $f$  is the activation function (typically a sigmoid function) of the reservoir neurons.  $\mathbf{W}_{in}$ ,  $\mathbf{W}$ , and  $\mathbf{W}_{fb}$  are the input weight matrix, the reservoir weight matrix, and the feedback weight matrix, with sizes of  $N \times K$ ,  $N \times N$ , and  $N \times L$ , respectively. Here, we consider only the original ESN<sup>[1]</sup> without feedback connections.

For the ESN to work, the network should possess the echo state property. For practical purposes, the spectral radius (the highest among the absolute eigenvalues in a

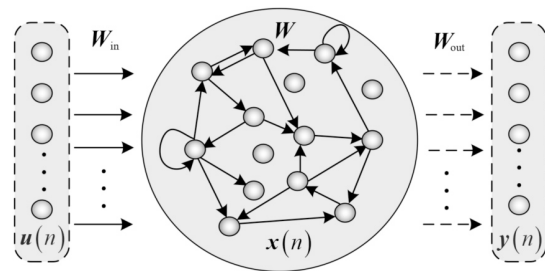


Fig. 1 The basic structure of the original ESN.

matrix) of  $\mathbf{W}$  is less than 1. To ensure the echo state property, the final  $\mathbf{W}$  will usually be scaled as follows:

$$\mathbf{W} = \alpha (\mathbf{W} / |\lambda_{\max}|) \quad (2)$$

where  $\alpha$  is the scaling parameter with a value between zero and one, and  $|\lambda_{\max}|$  is the spectral radius of  $\mathbf{W}$ . The necessary condition is satisfied.

The output signals  $\mathbf{y}(n)$  of the ESN can be expressed as follows:

$$\mathbf{y}(n) = f^{\text{out}}(\mathbf{W}_{\text{out}}\mathbf{x}(n)) \quad (3)$$

where  $f^{\text{out}}$  denotes the activation function of output neurons.  $\mathbf{W}_{\text{out}}$  is the output weight matrix with the size of  $L \times (N + K)$ .

As mentioned above, the ESN is trained in such a way that it only needs to adjust the output weights. During the training, the states of reservoir neuron are collected into a state matrix  $\mathbf{M}$  as follows:

$$\mathbf{M} = \begin{bmatrix} \mathbf{x}^T(t) \\ \mathbf{x}^T(t+1) \\ \vdots \\ \mathbf{x}^T(t+k-1) \end{bmatrix} \quad (4)$$

The corresponding output vectors are then collected into a target matrix  $\mathbf{T}$  as follows:

$$\mathbf{T} = \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{y}(t+1) \\ \vdots \\ \mathbf{y}(t+k-1) \end{bmatrix} \quad (5)$$

where  $t$  corresponds to the length of initial transient process. To reduce the influence of initial reservoir states,  $k$  is considered the number of training samples. Typically,  $f^{\text{out}}$  is an identity function. We can calculate the output weight matrix  $\mathbf{W}_{\text{out}}$  using the pseudo inverse algorithm, as shown in Eq. (6).

$$\mathbf{W}_{\text{out}} = \mathbf{M}^+ \mathbf{T} \quad (6)$$

where  $\mathbf{M}^+$  is the generalized inverse matrix of the state matrix  $\mathbf{M}$ , and Eq. (6) will be modified as follows:

$$\mathbf{W}_{\text{out}} = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{T} \quad (7)$$

### 3 Structure Optimization for ESN Based on Contribution

In this section, to simplify the network structure and improve the network performance, we propose a structure optimization method for ESN based on contribution. The details of structure optimization design of ESN are described below.

#### 3.1 Contribution measure of reservoir neurons

The ESN transforms an incoming time-series signal into a high-dimensional state space. However, not every dimension may contribute to the solution<sup>[14]</sup>. In other words, the ESN includes several reservoir neurons with relatively low contributions on the readout, along with a high training complexity. Based on the information theory, the mutual information can calculate both the linear and nonlinear dependences between two random variables. The mutual information can be used to analyze the Information Dependence (ID) between the output variables of two neurons. In this paper, we use the mutual information to measure the degree of ID between reservoir neuron  $X$  and output neuron  $Y$ ; this condition can be expressed as follows:

$$\text{ID}(X; Y) = H(X) + H(Y) - H(X, Y) \quad (8)$$

where  $H(X)$  and  $H(Y)$  are the entropy of  $X$  and  $Y$ , which describe the amount of information of reservoir neuron  $X$  and output neuron  $Y$ , respectively.  $H(X, Y)$  is joint entropy of  $X$  and  $Y$ , and it describes the amount of information shared by the two neurons.

Based on Shannon entropy<sup>[15]</sup>, the entropy and the joint entropy are calculated as follows:

$$H(X) = - \sum_{x \in X} \rho(x) \log \rho(x) \quad (9)$$

$$H(Y) = - \sum_{y \in Y} \rho(y) \log \rho(y) \quad (10)$$

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} \rho(x, y) \log \rho(x, y) \quad (11)$$

where  $\rho(x)$  and  $\rho(y)$  are the margin probability densities of  $X$  and  $Y$ , respectively.  $\rho(x, y)$  is the joint probability density of  $X$  and  $Y$ .

Then, Eq. (8) will be modified as follows:

$$\text{ID}(X; Y) = \sum_{x \in X} \sum_{y \in Y} \rho(x, y) \log \frac{\rho(x, y)}{\rho_X(x) \rho_Y(y)} \quad (12)$$

The degree of ID between two neurons  $X$  and  $Y$  is nonnegative (i.e.,  $\text{ID}(X; Y) \geq 0$ ) and symmetric. In this paper, we adopt the histogram method to approximate  $\text{ID}(X; Y)$ <sup>[16]</sup>. The method divides the vector variable  $X$  into  $K_x$  equally sized  $\Delta x$  segments, the vector variable  $Y$  into  $K_y$  equally sized  $\Delta y$  segments, and the bivariate variable  $(X, Y)$  into  $(K_x \times K_y)$  equally sized  $(\Delta x \times \Delta y)$  cells. Then, Eqs. (9)–(11) will be respectively modified as follows:

$$H(X) = - \sum_{k_i=1}^{K_x} \frac{n_{k_i}}{N_h} \ln \frac{n_{k_i}}{N_h} \quad (13)$$

$$H(Y) = - \sum_{k_j=1}^{K_y} \frac{n_{k_j}}{N_h} \ln \frac{n_{k_j}}{N_h} \quad (14)$$

$$H(X, Y) = - \sum_{k_i=1}^{K_x} \sum_{k_j=1}^{K_y} \frac{n_{k_i k_j}}{N_h} \ln \frac{n_{k_i k_j}}{N_h} \quad (15)$$

where  $N_h$  is the total number of the samples.  $n_{k_i}$  and  $n_{k_j}$  are the number of samples in segment  $k_i$  and segment  $k_j$ , respectively.  $n_{k_i k_j}$  is the number of samples in cells  $(k_i, k_j)$ . Hence, the  $ID(X; Y)$  can be estimated as follows:

$$ID(X; Y) = - \sum_{k_i=1}^{K_x} \frac{n_{k_i}}{N_h} \ln \frac{n_{k_i}}{N_h} - \sum_{k_j=1}^{K_y} \frac{n_{k_j}}{N_h} \ln \frac{n_{k_j}}{N_h} + \sum_{k_i=1}^{K_x} \sum_{k_j=1}^{K_y} \frac{n_{k_i k_j}}{N_h} \ln \frac{n_{k_i k_j}}{N_h} \quad (16)$$

where  $K_x = K_y = \text{round} \left\{ \frac{\xi}{6} + \frac{2}{3\xi} + \frac{1}{3} \right\}$ ,  $\xi = \sqrt[3]{8 + 324N_h + 12\sqrt{36N_h + 729N_h^2}}$ , which is defined in Ref. [16].

Based on the above analysis, in this paper, we can evaluate the contribution of the  $i$ -th reservoir neuron to the network output, which is related to the degree of ID between the state of reservoir neuron  $\mathbf{x}_i$  and the network output vector  $\mathbf{y}$  and calculated by the following:

$$m_i = \frac{ID(\mathbf{x}_i; \mathbf{y})}{\min(H(\mathbf{x}_i), H(\mathbf{y}))} \quad (17)$$

where  $m_i$  denotes the normalized ID of the  $i$ -th reservoir neuron to network output,  $0 \leq m_i \leq 1$ .

Furthermore, we propose an effective criterion for measuring the contribution of reservoir neurons inside the dynamic reservoir. We therefore define the contribution of the  $i$ -th reservoir neuron ( $c_i$ ), which is given by the following equation, in the ESN.

$$c_i = m_i / \sum_{i=1}^N m_i \quad (18)$$

where  $N$  is the number of reservoir neurons, with  $N > 0$ , that is, 50 to 1000 reservoir neurons.

### 3.2 Structure optimization algorithm for ESN based on contribution

In this subsection, we present an effective mechanism for pruning redundant connection weights based on contribution. The effectiveness of specific reservoir neurons is much weaker than that of the other neurons in the ESN. Thus, we prune the output connection weights of reservoir neurons with a low contribution value that is considerably unimportant. Specifically, the pruning method for the output connection weights of the  $i$ -th reservoir neuron is as follows:

$$w'_i = \begin{cases} 0, & \text{if } c_i \leq \rho; \\ w_i, & \text{else } c_i > \rho \end{cases} \quad (19)$$

where  $w_i$  and  $w'_i$  are the output connection weights of the  $i$ -th reservoir neuron before and after the network structure is adjusted, respectively.  $\rho$  is a prune parameter, which is a preset threshold value based on prior knowledge. The prune parameter  $\rho$  is used to delete unimportant output connection weights. If  $c_i > \rho$ , which indicates the  $i$ -th reservoir neuron features higher contribution value than the prune parameter, then the corresponding connection  $w_i$  is unchanged. If  $c_i \leq \rho$ , which shows that the  $i$ -th reservoir neuron features a lower contribution value than the prune parameter, then the corresponding output connection weight,  $w_i$ , will be removed. The number of output connection weights will be deleted depending on the prune parameter value. A higher value of the prune parameter will result in the removal of more output connection weights. Moreover, the network structure is simplified.

In summary, the main steps of the structure optimization algorithm for ESN based on contribution are as follows.

**Step 1.** Create an initial ESN to capture the dynamics of the underlying unknown process. Given the values of  $K$ ,  $N$ ,  $L$ , and  $\rho$ , an initial ESN ( $\mathbf{W}_{in}$ ,  $\mathbf{W}$ ,  $\mathbf{W}_{fb}$ ) is randomly created.

**Step 2.** Drive the reservoir by the training data, and update the reservoir states as Eq. (1). Store the reservoir states into a state matrix,  $\mathbf{M}$ , as follows:

$$\mathbf{M} = \begin{bmatrix} \mathbf{x}_1(1) & \mathbf{x}_1(2) & \cdots & \mathbf{x}_1(N_s) \\ \mathbf{x}_2(1) & \mathbf{x}_2(2) & \cdots & \mathbf{x}_2(N_s) \\ \cdots & \cdots & \ddots & \cdots \\ \mathbf{x}_N(1) & \mathbf{x}_N(2) & \cdots & \mathbf{x}_N(N_s) \end{bmatrix} \quad (20)$$

where  $N_s$  is the number of training data. Collect each desired output into a target matrix  $\mathbf{T}$ .

**Step 3.** Calculate the output weight matrix  $\mathbf{W}_{out}$  by Eq. (7), and calculate the training error and validation error  $E^v$  by the following equation:

$$E^v = \frac{1}{N_s} \sum_{n=1}^{N_s} (\mathbf{d}(n) - \mathbf{y}(n))^2 \quad (21)$$

where  $\mathbf{d}(n)$  is the target output and  $\mathbf{y}(n)$  is the prediction output of the trained network.

**Step 4.** Calculate the contribution value of each reservoir neuron according to Eq. (18).

**Step 5.** Prune the unimportant output connection weights of reservoir neurons using Eq. (19), and obtain the new state matrix  $\mathbf{M}_c$  of the optimized ESN as

follows:

$$\mathbf{M}_c = [x_1, x_2, \dots, x_{N_c}] \quad (22)$$

where  $N_c$  is the number of the selected reservoir neurons with higher contribution value.

**Step 6.** Update the output weight matrix  $\hat{\mathbf{W}}_{\text{out}}$  of the optimized ESN by pseudo inverse algorithm.

**Step 7.** Calculate and record the validation error  $E_c^v$  using the validation data set.

**Step 8.** If the stopping criterion (the  $E_c^v$  reaches an acceptable range) is satisfied, then turn to Step 9; otherwise, go to Step 4 and continue to adjusting the network structure.

**Step 9.** Test the optimized ESN, and calculate the testing error.

According to the recorded validation error, the model with the minimal validation error will be tested on the testing data.

## 4 Experimental Results and Analysis

To demonstrate the effectiveness of the C-ESN model, it is tested on two scenarios, including the Lorenz chaotic time-series prediction and an actual municipal sewage treatment system in a WTP. In both experiments, each data set is divided into three subsets: a training set, a validation set, and a testing set. The lengths of the training, validation, and testing sets are  $L_{\text{train}}$ ,  $L_{\text{validation}}$ , and  $L_{\text{test}}$ , respectively. The length of initial transient process is  $L_{\text{washout}}$ , which is used to wash out the influence of initial states. All the experiments are carried out in an identical software and hardware environment.

### 4.1 Lorenz chaotic time-series prediction

The Lorenz time-series prediction is a benchmark problem for ESN. The Lorenz dynamic system is described by the following equation:

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x); \\ \frac{dy}{dt} = -xz + r(x - y); \\ \frac{dz}{dt} = xy - bz \end{cases} \quad (23)$$

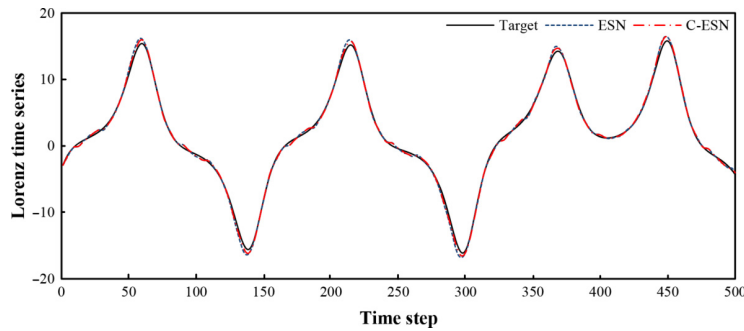
where  $\sigma = 10$ ,  $r = 2$ , and  $b = 8/3$ , the system shows a chaotic state and generated chaotic time series. The initial values are set as  $x(0) = 12$ ,  $y(0) = 2$ , and  $z(0) = 9$ . The data set is generated using four-order Runge-Kutta method, and split into three parts with  $L_{\text{train}} = 1500$ ,  $L_{\text{validation}} = 500$ ,  $L_{\text{test}} = 600$ , and  $L_{\text{washout}} = 100$ . To investigate the anti-noise capability of C-ESN, the training sequences are inserted with zero-mean Gaussian noise such that the square deviation is 1. The networks (C-ESN and ESN) possess the same reservoir parameters of reservoir. The selected reservoir size is 600, and the spectral radius and sparse connectivity of  $\mathbf{W}$  are set at 0.95 and 0.002, respectively.

The Root Mean Square Error (RMSE) is used to evaluate the performance of the C-ESN, which is calculated as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_1^{N_s} (\mathbf{y}(t) - \hat{\mathbf{y}}(t))^2}{(N_s - 1)}} \quad (24)$$

where  $\mathbf{y}(t)$  is the target output,  $\hat{\mathbf{y}}(t)$  is the prediction output of the trained network, and  $N_s$  is the number of samples.

Figure 2 shows the predictive output of the C-ESN and ESN for jerky Lorenz time series. Figure 2 reflects that the noise seriously affects the prediction accuracy of the networks (C-ESN and ESN). However, the C-ESN performs slightly better than ESN for predicting jerky Lorenz time series. Figure 3 shows the prediction error curves of the C-ESN and ESN for jerky Lorenz chaotic time series, where C-ESN exhibits a less prediction error compared with the ESN. As shown



**Fig. 2** Prediction results based on C-ESN and ESN for jerky Lorenz time series.

in Figs. 2 and 3, the C-ESN presents better prediction performance than the ESN after some unimportant connection weights are pruned.

Figure 4 shows the error draw curve drawn after each pruning operation in the Lorenz time-series prediction problem. The initial number of reservoir neurons is 200 in the C-ESN. As presented in Fig. 4, the validation RMSE shows no marked drop in the beginning of pruning operations, when the amount of selected reservoir neurons is small, the low contribution is also small. When more reservoir neurons with low contribution are pruned, the validation RMSE decreases. We can prune 153 reservoirs connections with the proposed structure optimization algorithm, whereas the validation RMSE is minimized. However, as the number of pruning increases, the reservoir neurons with high contribution are pruned, and the validation RMSE increases rapidly. Therefore, the proposed structure optimization algorithm based on contribution not only simplifies the network structure but also improves the prediction performance of ESN.

Table 1 shows the performance comparison results of the C-ESN and ESN with different reservoir sizes ( $N = 200, 400, 600$ ). According to Table 1, the training error of the networks (C-ESN and ESN) gradually reduce with the increasing reservoir size. The training

error of C-ESN is larger than of the ESN. However, the C-ESN produces better validation and test results than the ESN under different reservoir sizes. Thus, the proposed structure optimization algorithm can adjust the network structure effectively and improve the network prediction performance. Consequently, the C-ESN achieves a better prediction and generalization performance than the ESN.

#### 4.2 Total Phosphorus (TP) prediction in a WWTP

The WWTP is complex and difficult to model for complex biochemical reactions. The effluent TP concentration is one of the most important effluent quality indexes for WWTP. To demonstrate the performance of the C-ESN, we apply the C-ESN in the soft measurement model for TP in the WWTP. The experimental data are collected from a sewage treatment plant in Beijing, with a total of 360 data samples. The influent TP, the dissolved oxygen in biological reactor, the influent and effluent oil, the effluent temperature, and effluent  $\text{NH}_4\text{-N}$  are selected as subsidiary variables for effluent TP prediction. The data set is split into three parts with  $L_{\text{train}} = 180$ ,  $L_{\text{validation}} = 90$ ,  $L_{\text{test}} = 90$ , and  $L_{\text{washout}} = 5$ . The networks (C-ESN and ESN) possess the same reservoir parameters. The selected reservoir size is 90, and the spectral radius and sparse

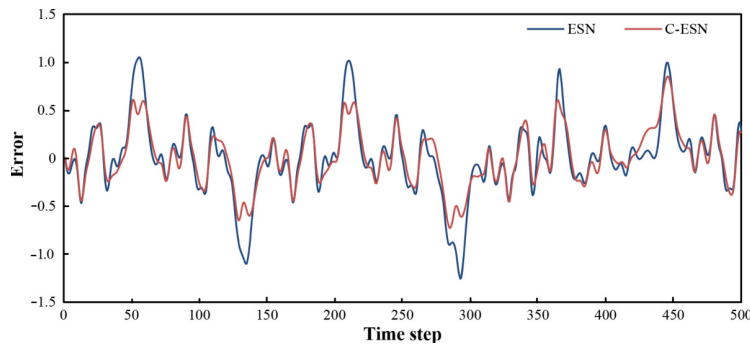


Fig. 3 Prediction errors based on C-ESN and ESN for jerky Lorenz time series.

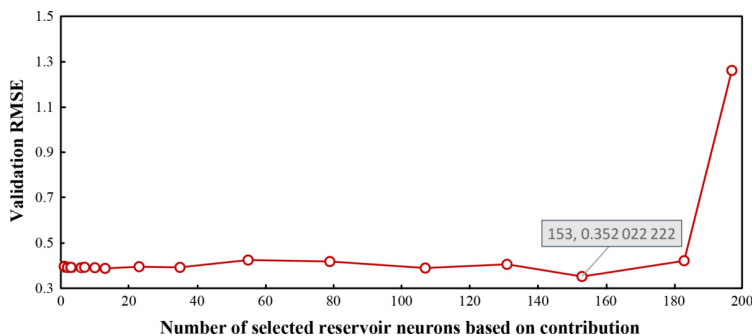


Fig. 4 Validation RMSE of C-ESN after each pruning operation when  $N=200$ .

**Table 1** Performance comparison of ESN and C-ESN for jerky Lorenz time series when  $N=200, 400, \text{ and } 600$ .

Reservoir size ( $N$ )	No.	Train error (RMSE)		Validation error (RMSE)		Test error (RMSE)	
		ESN	C-ESN	ESN	C-ESN	ESN	C-ESN
200	1	0.571 41	0.676 19	0.370 25	0.332 42	0.350 67	0.324 14
	2	0.561 40	0.634 79	0.336 55	0.324 56	0.335 33	0.329 68
	3	0.563 52	0.570 68	0.355 44	0.346 06	0.357 14	0.323 29
	4	0.554 58	0.558 93	0.349 26	0.340 85	0.345 85	0.334 85
	5	0.517 80	0.581 96	0.334 65	0.293 20	0.290 53	0.273 46
	6	0.582 62	0.591 44	0.309 23	0.308 08	0.296 48	0.289 56
	7	0.587 25	0.620 32	0.386 72	0.374 42	0.344 68	0.336 77
	8	0.521 47	0.579 11	0.370 86	0.362 82	0.353 91	0.331 32
	9	0.582 62	0.591 44	0.309 23	0.308 08	0.296 48	0.289 56
	10	0.563 81	0.667 91	0.394 28	0.351 77	0.361 28	0.328 03
		<b>Mean</b>	<b>0.560 64</b>	<b>0.607 28</b>	<b>0.351 64</b>	<b>0.334 23</b>	<b>0.333 23</b>
400	1	0.418 16	0.460 08	0.360 58	0.298 67	0.321 10	0.280 79
	2	0.426 92	0.436 74	0.306 19	0.291 64	0.297 07	0.243 15
	3	0.381 11	0.385 16	0.324 63	0.316 39	0.339 69	0.329 92
	4	0.427 71	0.430 09	0.333 30	0.313 24	0.328 60	0.309 05
	5	0.465 60	0.509 35	0.329 50	0.297 02	0.271 18	0.247 46
	6	0.438 35	0.448 19	0.334 08	0.313 78	0.280 47	0.256 99
	7	0.450 86	0.462 08	0.337 38	0.320 12	0.311 16	0.264 74
	8	0.439 03	0.485 84	0.321 82	0.285 65	0.323 72	0.290 33
	9	0.404 84	0.472 07	0.365 65	0.341 89	0.347 01	0.309 38
	10	0.433 10	0.474 83	0.319 25	0.286 65	0.326 58	0.291 18
		<b>Mean</b>	<b>0.428 57</b>	<b>0.456 44</b>	<b>0.333 24</b>	<b>0.306 51</b>	<b>0.314 57</b>
600	1	0.390 51	0.408 06	0.315 14	0.263 66	0.307 79	0.281 82
	2	0.374 23	0.461 42	0.379 66	0.322 20	0.297 07	0.246 98
	3	0.328 60	0.401 90	0.345 02	0.283 80	0.300 11	0.281 54
	4	0.346 06	0.368 90	0.269 76	0.239 27	0.252 87	0.235 01
	5	0.345 82	0.370 26	0.300 71	0.262 52	0.272 77	0.262 19
	6	0.364 41	0.370 89	0.254 41	0.249 87	0.248 90	0.237 75
	7	0.331 27	0.356 45	0.237 46	0.221 51	0.255 58	0.234 70
	8	0.356 12	0.425 91	0.330 90	0.275 30	0.321 37	0.267 30
	9	0.338 44	0.353 89	0.312 48	0.294 11	0.294 02	0.263 92
	10	0.347 33	0.355 87	0.395 63	0.362 64	0.299 94	0.282 89
		<b>Mean</b>	<b>0.352 28</b>	<b>0.387 36</b>	<b>0.314 11</b>	<b>0.377 49</b>	<b>0.285 04</b>

connectivity of  $W$  are set at 0.98 and 0.01, respectively.

Figure 5 shows the predicted values of the C-ESN and the actual values of TP. The predicted results of the C-ESN can approximately fit the actual values. Figure 6 shows the error between the C-ESN predicted results and the real values. From Fig. 6, the C-ESN features a satisfactory performance on TP concentration estimation. The simulation results show that the proposed C-ESN is an effective method for predicting the TP concentrations in WWTPs.

## 5 Conclusion

In this paper, a structure optimization method for ESN is proposed based on the contribution for ESN. We

first propose an effective criterion for measuring the contribution of reservoir neurons. Then, we present the pruning mechanism based on contribution to remove the unimportant connections from reservoirs to the output layer of ESN. This approach can not only simplify the structure but also improve the prediction and generalization performance of the ESN. The performance of the optimized ESN model (C-ESN) has been demonstrated by several examples, including the Lorenz chaotic time-series prediction and the effluent TP prediction in a WWTP. Experimental results show that the C-ESN achieves better prediction and generalization performance than the ESN in nonlinear system modeling.



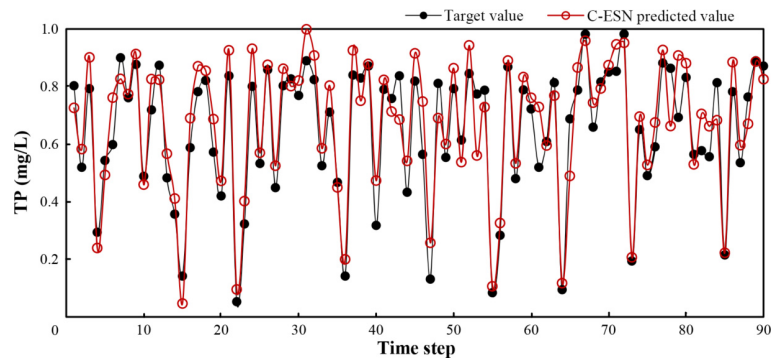


Fig. 5 Prediction results based on C-ESN for TP prediction.

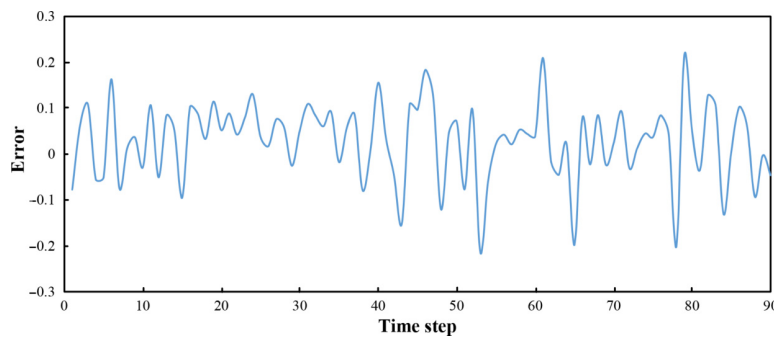


Fig. 6 Prediction errors curve based on C-ESN for TP prediction.

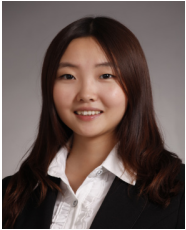
## Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61225016) and the Key Project of National Natural Science Foundation of China (No. 61533002).

## References

- [1] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, *Science*, vol. 304, no. 5667, pp. 78–80, 2004.
- [2] H. Jaeger, The “echo state” approach to analysing and training recurrent neural networks, GMD Report, German National Research Center for Information Technology, Sankt Augustin, Germany, 2001.
- [3] M. H. Tong, A. D. Bickett, E. M. Christiansen, and G. W. Cottrell, Learning grammatical structure with Echo State Networks, *Neural Networks*, vol. 20, no. 3, pp. 424–432, 2007.
- [4] K. Gideon, C. Nyirenda, and C. T. Nyah, Echo state network-based radio signal strength prediction for wireless communication in Northern Namibia, *IET Communications*, vol. 11, no. 12, pp. 1920–1926, 2017.
- [5] P. Coulibaly, Reservoir computing approach to Great Lakes water level forecasting, *Journal of Hydrology*, vol. 381, nos. 1&2, pp. 76–88, 2010.
- [6] B. Meftah, O. L  zoray, and A. Benyettou, Novel approach using echo state networks for microscopic cellular image segmentation, *Cognitive Computation*, vol. 8, no. 2, pp. 237–245, 2016.
- [7] M. Salmen and P. G. Ploger, Echo state networks used for motor control, in *Proc. IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2006, pp. 1953–1958.
- [8] H. S. Wang and X. F. Yan, Improved simple deterministically constructed Cycle Reservoir Network with Sensitive Iterative Pruning Algorithm, *Neurocomputing*, vol. 145, pp. 353–362, 2014.
- [9] A. Rodan and P. Tino, Minimum complexity echo state network, *IEEE Transactions on Neural Networks*, vol. 22, no. 1, pp. 131–144, 2011.
- [10] S. Scardapane, G. Nocco, D. Comminiello, M. Scarpiniti, and A. Uncini, An effective criterion for pruning reservoir’s connections in Echo State Networks, in *Proc. International Joint Conference on Neural Networks*, Beijing, China, 2015, pp. 1205–1212.
- [11] X. Dutoit, B. Schrauwen, J. V. Campenhout, D. Stoobandt, H. Van Brussel, and M. Nutton, Pruning and regularization in reservoir computing, *Neurocomputing*, vol. 72, nos. 7–9, pp. 1534–1546, 2009.
- [12] N. Chouikhi, B. Ammar, N. Rokbani, A. M. Alimi, and A. Abraham, A hybrid approach based on particle swarm optimization for echo state network initialization, in *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, Kowloon, China, 2016, pp. 2896–2901.
- [13] A. A. Ferreira and T. B. Ludermir, Comparing evolutionary methods for reservoir computing pre-training, in *Proc. International Joint Conference on Neural Networks*, San Jose, CA, USA, 2011, pp. 283–290.

- [14] H. U. Kobialka and U. Kayani, Echo state networks with sparse output connections, in *Proc. 20th Int. Conference on Artificial Neural Networks: Part I*, Thessaloniki, Greece, 2010, pp. 356–361.
- [15] C. E. Shannon, A mathematical theory of communication, *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.
- [16] A. H. Gharbi, P. Ravier, R. Harba, and T. Mohamadi, Low bias histogram-based estimation of mutual information for feature selection, *Pattern Recognition Letters*, vol. 33, no. 10, pp. 1302–1308, 2012.



**Dingyuan Li** received the BS degree from Jilin University in 2010. She is currently pursuing the PhD degree in Jilin University. Her current research interests include artificial intelligence, pattern recognition, neural network, and self-adaptive/learning systems.



**Junfei Qiao** received the BS and MS degrees from Liaoning Technical University in 1992 and 1995, respectively, and the PhD degree from Northeast University in 1998. He is currently a professor at Beijing University of Technology. His current research interests include neural networks, intelligent systems, self-adaptive/learning systems, and process control systems.



**Fu Liu** received the BS and MS degrees from Jilin University of Technology in 1991 and 1994, respectively, and the PhD degree from Jilin University in 2002. He is currently a professor at Jilin University. His research interests include artificial intelligence, machine vision, pattern recognition, and biometrics

and bioinformatics.



**Rong Li** received the BS degree from Yanshan University in 2006, and MS and PhD degrees from Beijing University of Technology in 2009 and 2017, respectively. She is currently an assistant professor at Beijing Vocational College of Agriculture. Her current research interests include neural networks and intelligent systems.