



Time Series Prediction of ESN Based on Chebyshev Mapping and Strongly Connected Topology

Minzhi Xie¹ · Qianxue Wang¹ · Simin Yu¹

Accepted: 27 November 2023 / Published online: 13 February 2024
© The Author(s) 2024

Abstract

This paper introduces a novel approach called Chebyshev mapping and strongly connected topology for optimization of echo state network (ESN). To enhance the predictive performance of ESNs for time series data, Chebyshev mapping is employed to optimize the irregular input weight matrix. And the reservoir of the ESN is also replaced using an adjacency matrix derived from a digital chaotic system, resulting in a reservoir with strong connectivity properties. Numerical experiments are conducted on various time series datasets, including the Mackey–Glass time series, Lorenz time series and solar sunspot numbers, validating the effectiveness of the proposed optimization methods. Compared with the traditional ESNs, the optimization method proposed in this paper has higher predictive performance, and effectively reduce the reservoir’s size and model complexity.

Keywords Echo state network · Time series predicting · Chebyshev mapping · Strongly connected topology

1 Introduction

Recursive neural network (RNN) has been widely used in chaotic time series prediction [1]. However, it’s structure is relatively complex, and there are problems such as slow convergence, high training complexity, susceptibility to local optima, and vanishing gradients [2]. Although some researchers have proposed corresponding improvement schemes to solve some problems, they have not solved the essential problem of high training complexity. The fundamental problem of high training complexity was not solved until the reservoir computing (RC) [3]. The essential idea of RC is that the reservoir itself remains unchanged with

All the authors have contributed equally to this work.

✉ Qianxue Wang
wangqianxue@gdut.edu.cn

Minzhi Xie
13517968892@163.com

Simin Yu
siminyu@163.com

¹ School of Automation, Guangdong University of Technology, Guangzhou 510006, Guangdong, China

random and fixed connections between the neurons, and only the output layer's neural networks can be trained. Based on the idea of RC, Jaeger et al. proposed ESN in 2001 [4, 5]. Different from traditional RNNs, the ESN internally contains a sparse, recursively connected reservoir, which stores historical information, thereby acting as an echo [6]. The reservoir replaces the fully connected hidden layer of traditional RNNs. In the ESN, the input weight matrix and the internal connection weight matrix of the reservoir remain unchanged after random initialization [7]. The mapping from the reservoir to the output layer is linear, and only the output weight is obtained through linear regression training, thus overcoming the disadvantages of slow convergence, susceptibility to local optima, and vanishing gradients in traditional RNNs. The ESN model is simple and its training speed is fast. It has been successfully applied in time series prediction [8–14], time series classification [15–18], speech recognition [19], non-linear control [20, 21] and other fields.

However, in traditional ESN, the input weight matrix is randomly generated, and the internal neurons of the reservoir are randomly connected. The input weight matrix defines how external input influences the internal state of the reservoir. This matrix determines how external input propagates within the network, thereby affecting the state evolution and output of the reservoir. The reservoir is the core component of the ESN, comprising a large number of neurons and generating nonlinear dynamics through internal connections. The complexity, size, and nonlinear nature of the reservoir profound impact the expressive and generalization capabilities of ESNs. Larger and more complex reservoirs can better capture features and patterns in the data but are also prone to overfitting. The randomness of the input weight matrix and reservoir leads to issues in ESNs, including performance instability, unclear dynamic behavior, difficulty in ensuring the generation of ESN models tailored to specific tasks, and susceptibility to overfitting. In recent years, domestic and foreign experts and scholars have made relevant improvements to the problems existing in the ESN, mainly including the following three aspects: topology structure design, optimization of input layer preprocessing, optimization of reservoir parameters selection, and selection of neural activation functions and related training and learning algorithm improvements.

Regarding the design of ESN structures, reference [22] introduces a Fractional Order ESN for time series prediction. This network utilizes fractional-order differential equations to describe the dynamic characteristics of the reservoir state, enabling a more accurate representation of the dynamic features of a specific class of time series. To enhance the predictive performance, the paper also presents a fractional-order output weight learning method and a fractional-order parameter optimization method for training output weights and optimizing reservoir parameters. Reference [23] introduces an improved ESN with an enhanced topology. This improvement involves studying and optimizing echo state properties, designing smooth activation functions, and constructing a reservoir with rich features. The goal is to achieve more precise and efficient time series prediction. Through a series of experiments and comparisons with other models, this approach demonstrates significant performance improvements in various tasks. In Ref. [24], the authors presented an approach to optimize the storage reservoir topology of an ESN by using the corresponding adjacency matrix of the digital chaotic systems. They constructed a chaotic ESN and showed that its prediction performance is higher than that of traditional ESN. By adjusting the topology structure of ESN, its predictive performance can be effectively improved while reducing overfitting. References [25–29] have introduced and enhanced deep echo state networks (DESNs) composed of multiple stacked reservoirs. These layers progressively extract abstract features from the data. With the addition of intermediate layers, DESNs are capable of addressing more complex time series problems. Experimental results demonstrate the significant impact of DESNs on improving ESN's predictive performance.

In addressing the optimization of reservoir parameters, Ref. [30] proposes the use of particle swarm optimization (PSO) to optimize the selection of reservoir parameters in ESN. This pre-training approach effectively reduces the impact of random initialization variables on the prediction performance of the network. The experimental results demonstrate that the optimized ESN using PSO algorithm has higher stability and better generalization ability. Reference [31] introduces a novel model that incorporates logistic mapping and bias dropout algorithms with the objective of optimizing, the irregular input weight matrix and generate a more optimal and simplified reservoir structure. Experimental results demonstrate that the proposed model exhibits significant improvements in reducing testing time, reservoir size and model complexity, while also enhancing the performance compared to traditional ESN. In addition, Ref. [32] presents a design method for optimizing the global parameters, topology and weights of the reservoir using a simultaneous search approach. The internal neuron state of the reservoir is not solely determined by fixed weights, but is also influenced by input signals. By optimizing the reservoir parameters, the capability of ESN to handle input sequences can be enhanced, thereby improving the accuracy of prediction. Employing different reservoir parameters for different tasks increases the flexibility of ESN applications.

However, adjusting the reservoir parameters can be time-consuming and requires careful selection to prevent decreased prediction accuracy, reduced robustness of the network model, and overfitting. Reference [33] proposes a time series prediction model that combines ESN with adaptive elastic network algorithms for neuron selection and training learning algorithms. The adaptive elastic algorithm is used to solve the linear regression problem of ESN. Experiments have shown that choosing appropriate neuron activation functions and training learning algorithms can enhance the nonlinear representation ability of ESN for multivariate chaotic time series, surpassing the prediction ability of other ESN models. However, using complex activation functions and training learning algorithms may lead to overfitting of the network and increase computational costs. Therefore, careful consideration is needed when selecting activation functions and training algorithms based on specific tasks and datasets. Additionally, cross-validation and other methods can be employed to evaluate the model's performance and avoid overfitting. References [34–36] have employed techniques such as filtering, reservoir module methods, and probabilistic regularization to optimize the structure of the reservoir and the training process of ESN, resulting in a significant improvement in ESN performance.

In order to improve the prediction performance and stability of ESN, this paper proposes an novel ESN model based on Chebyshev mapping and strongly connected topology. On the one hand, Chebyshev mapping is utilized to generate an input weight matrix with convergence characteristics and rich chaotic characteristics. On the other hand, strong connected structures are used to optimize the topology of the reservoir, making it have strong connected characteristics. This article is arranged as follows: Sect. 2 briefly introduces the ESN, including the structure and training prediction algorithm of the ESN. Section 3 describes in detail the steps of optimizing the ESN, including the algorithm for generating the input weight matrix and the creation of strongly connected topology. Section 4 demonstrates the excellent prediction performance and stability of the optimized ESN through simulation of chaotic time series and real-world time series prediction, and compares it with other prediction models. Section 5 concludes the paper.

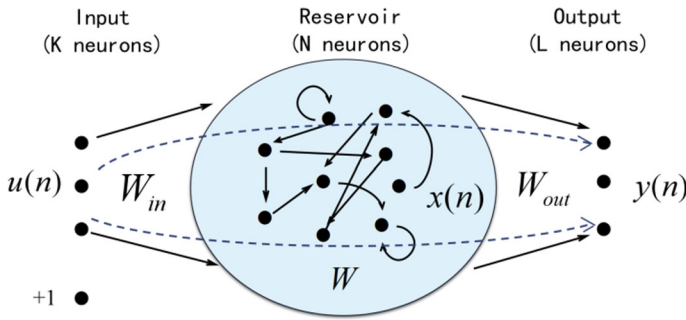


Fig. 1 Echo state network

2 Echo State Network

Jaeger [4] introduced the ESN, a distinctive type of recurrent neural network that incorporates a dynamic reservoir of neurons comprising hundreds or thousands of units. The reservoir is a vast, randomly generated, sparsely connected recursive structure, resembling biological neural networks more closely than traditional recurrent networks. ESN has found applications in various domains, including time series prediction, time series classification, nonlinear control and other related areas.

2.1 Network Mode

The structure of the ESN is shown in Fig. 1, which is a three-layer recurrent neural network composed of an input layer, a reservoir and an output layer. The reservoir is comprised of 100s–1000s of sparsely connected recursive neurons, with their connection weights being randomly generated and fixed thereafter.

Let the number of neurons in the input layer, reservoir, and output layer of the ESN be denoted by K , N and L , respectively.

$$\begin{cases} u(n) = [u_1(n), u_2(n), \dots, u_K(n)]^T \\ x(n) = [x_1(n), x_2(n), \dots, x_N(n)]^T \\ y(n) = [y_1(n), y_2(n), \dots, y_L(n)]^T \end{cases} \quad (1)$$

where $u(n) \in \mathbb{R}^{K \times 1}$ refers to a vector consisting of K input neurons, $y(n) \in \mathbb{R}^{L \times 1}$ refers to a vector consisting of L output neurons, and $x(n) \in \mathbb{R}^{N \times 1}$ denotes a vector of N neurons within the reservoir that are activated.

According to the ESN proposed in Ref. [37], which uses a type of RNN called the leaky integrated discrete-time continuous-value unit, the update equation is as follows:

$$x(n) = f(W_{in}[1; u(n)] + Wx(n - 1)) \quad (2)$$

where $W_{in} \in \mathbb{R}^{N \times (1+K)}$ represents the input weight matrix with elements ranging between $[-1, 1]$, $W \in \mathbb{R}^{N \times N}$ denotes the sparse internal connection matrix within the reservoir. $[1; u(n)]$ represents the connection between vector 1 and vector $u(n)$. The input weight matrix W_{in} and the internal connection matrix W within the reservoir are both randomly generated and remain fixed during the training phase of the ESN. $x(n)$ is the activation vector of neurons in the reservoir. f represents the internal neuron activation function, which is usually a *tanh* function or *sigmoid* function.

To ensure that the internal connection matrix W within the reservoir is sufficiently sparse, we express W mathematically in Eq. 3.

$$W = R \frac{W_R \odot W_S}{|\lambda_{\max}|} \tag{3}$$

where R is the spectral radius of W , and its value interval is $(0, 1)$. $W_R \in \mathbb{R}^{N \times N}$ is a random matrix in which the values of matrix elements are uniformly distributed over $(-0.5, 0.5)$. $W_S \in \mathbb{R}^{N \times N}$ is a randomly generated matrix in which the values of elements are 0 or 1. The symbol \odot indicates that the elements of the same row and column of the two matrices W_R and W_S are multiplied, and λ_{\max} is the maximum eigenvalue of matrix $W_R \odot W_S$.

The linear output equation is:

$$y(n) = W_{out}[1; u(n); x(n)] \tag{4}$$

where $W_{out} \in \mathbb{R}^{N \times (1+L+K)}$ is the output weight matrix, $[1; u(n); x(n)]$ represents the connection between vector 1, vector $u(n)$ and vector $x(n)$.

2.2 Training Process

In the ESN, only the output weight matrix W_{out} needs to be trained. Before training, it is necessary to initialize the reservoir. Generally, the internal state vector of the reservoir is set to $x(0) = 0$, and a certain time point T_0 ($T_0 < T$) is selected as the initialization time point of the reservoir. All internal state vectors before T_0 are discarded. T is the data amount of the training dataset, and T_0 is the initialized data amount.

Since the output weight matrix W_{out} of the ESN is generally linear, Eq. 4 can be written as:

$$Y = W_{out}H \tag{5}$$

where $Y \in \mathbb{R}^{L \times T}$ represents all output vectors $y(n)$, in order to simplify the symbol, H is used instead of $[1; U; X]$, so $H \in \mathbb{R}^{(1+k+N) \times T}$.

To find the optimal output weight matrix W_{out} to minimize the square error between the actual output $y(n)$ and the target output $y_{target}(n)$, this process is equivalent to solving a typical optimization problem.

$$\min \|W^{out}H - Y_{target}\|_2^2 \tag{6}$$

where $Y_{target} \in \mathbb{R}^{L \times T}$, there are many training algorithms that can solve Eq. 6. Here, the most conventional ridge regression is selected

$$W_{out} = Y_{target}H^T(HH^T + \beta I)^{-1} \tag{7}$$

where β is the regularization coefficient and I is the identity matrix [38].

2.3 Key Parameter

The core of an ESN is its reservoir, whose performance is mainly related to two important parameters: reservoir size N , and reservoir spectral radius R .

(1) Reservoir size

The reservoir size N is the number of neurons in the reservoir, and increasing N generally improves the accuracy of the model. However, when N is too large, it may lead to increased

computational complexity and even overfitting. On the other hand, when N is too small, the model may suffer from low accuracy and underfitting.

(2) Reservoir spectral radius

The spectral radius R of the reservoir is one of the central global parameters of the ESN, and the spectral radius is the absolute value of the maximum eigenvalue of the sparse connection matrix W within the reservoir. Generally, the value interval of spectral radius R is $(0, 1)$.

According to Ref. [39], if the spectral radius is too large, the reservoir of an ESN may develop multiple fixed points, periodic, or even chaotic spontaneous attractor patterns (when the reservoir is sufficiently nonlinear), which violates the characteristic of the ESN. On the other hand, if the spectral radius is too small, the output weight values become very small, leading to low prediction accuracy of the network and making it challenging to match the target system.

3 Echo State Network Based on Chebyshev Mapping and Strongly Connected Topology

In this study, we propose an optimized ESN that uses Chebyshev mapping and strongly connected topology. Firstly, the input connection weight matrix W_{in} is randomly generated using Chebyshev mapping to construct a Ch-ESN model. Then, a strongly connected topological structure is incorporated into the reservoir of the ESN to enhance its connectivity and improve prediction performance, resulting in the construction of an SC-ESN. Finally, we combine these two techniques to create the Ch-SC-ESN model.

3.1 Chebyshev Mapping

The Chebyshev chaotic system [40] belongs to a one-dimensional chaotic system, and the corresponding Chebyshev mapping equation is

$$x(n+1) = \cos(k \cdot \arccos x(n)) \quad (8)$$

where k is the control parameter of the Chebyshev mapping. When k is > 2 , the system has a positive Lyapunov exponent and is in a chaotic state.

The first phase is the stable phase, where $k \in [0, 1]$ and x approaches a fixed value of 1 ($x \rightarrow 1$).

The second phase is the critical chaos phase, where $k \in (1, 2]$ and x gradually loses its stability. At this phase, the Lyapunov exponent of the system is negative, but it does not exhibit chaotic behavior.

The third phase is the chaotic phase, where $k \in (2, 4)$ and the Lyapunov exponent of the system is positive. In this phase, x exhibits chaotic dynamic behavior.

3.2 Optimizing the Input Weight Matrix with a Chebyshev Mapping

The fundamental idea of ESN is to transform the original input signal into a high-dimensional feature space through the input weight matrix W_{in} . In traditional ESN, the input weight matrix W_{in} is randomly generated with element values ranging from -1 to 1 . In contrast, the input

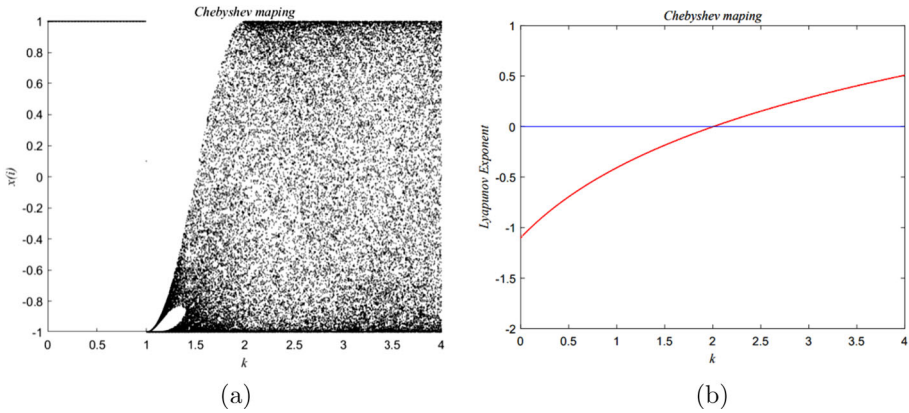


Fig. 2 Chebyshev mapping. **a** Bifurcation graph of Chebyshev mapping, **b** Lyapunov exponential of Chebyshev mapping

weight matrix W_{in}^{Ch} of Ch-ESN is generated through Chebyshev mapping iteration.

$$W_{in}^{Ch} = \begin{bmatrix} w_{in}(1, 1) & w_{in}(1, 2) & \cdots & w_{in}(1, 1 + K) \\ w_{in}(2, 1) & w_{in}(2, 2) & \cdots & w_{in}(2, 1 + K) \\ \dots & \dots & \dots & \dots \\ w_{in}(N, 1) & w_{in}(N, 2) & \cdots & w_{in}(N, 1 + K) \end{bmatrix} \tag{9}$$

Each element in input weight matrix W_{in}^{Ch} of Ch-ESN is generated using the following method.

- (1) The initial value of the first row element in input weight matrix W_{in}^{Ch} is set as follows:

$$w_{in}(1, j) = p \cdot \sin\left(\frac{j}{K + 1} \cdot \frac{\pi}{q}\right) \quad (j = 1, 2, \dots, K + 1) \tag{10}$$

where adjustable parameters p and q are employed here, j represents the column index of an element, and K represents the number of input layer neurons [41]. In our experiments, p was set to 0.3, and q was set to 5.9. By utilizing Eq. 10, the first-row elements of the input weight matrix for Ch-ESN can be obtained.

- (2) Other row elements of W_{in}^{Ch} are iteratively generated using Chebyshev mapping

$$w_{in}(i + 1, j) = \cos[k \cdot \arccos(w_{in}(i, j))] \tag{11}$$

As shown in Fig. 2, the input weight matrix W_{in}^{Ch} generated iteratively by Chebyshev mapping has three different phases. W_{in}^{Ch} is very sensitive to the initial condition of the control parameter k , which indicates that a slight change in the initial value of the control parameter k will lead to a great change in the distribution of elements in W_{in}^{Ch} . At the same time, the distribution of elements in W_{in}^{Ch} will change the “echo” attribute of ESN, which has a great impact on the prediction performance of Ch-ESN model. How to select the appropriate control parameter k and the influence of the control parameter k on the final model performance will be discussed in Sect. 4.6.

- (3) Finally, the initial input weight matrix is replaced by input weight matrix W_{in}^{Ch} , which is generated iteratively by Chebyshev mapping, and the Ch-ESN model is established.

$$x(n) = \tanh\left(W_{in}^{Ch}[1; u(n)] + Wx(n - 1)\right) \tag{12}$$

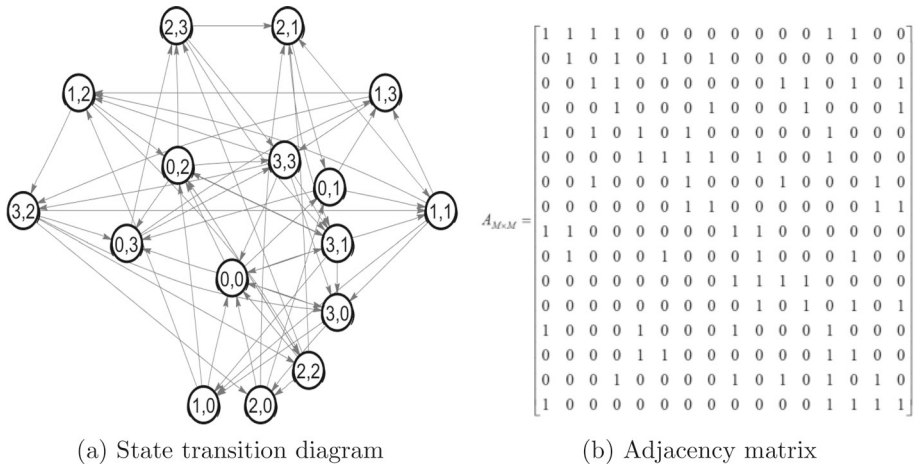


Fig. 3 Strongly connected state transition diagram and strongly connected adjacency matrix corresponding to Eq. 13 ($D = 2$)

3.3 Digital Chaotic System

Building upon the emerging theory of digital chaotic system, it is possible to directly establish a digital chaotic system under finite precision conditions and prove its chaotic characteristics. By making full use of the adjacency matrix associated with the digital chaotic system, we can construct a reservoir network structure suitable for ESNs, thereby transforming ESN into a chaotic ESN [24]. In this section, we will introduce the designed digital chaotic system.

For a two-dimensional digital chaotic system with finite precision, denoted as D .

$$\begin{cases} x_1(n) = x_1(n-1) \cdot \overline{s(n)} + \overline{x_1(n-1)} \cdot x_2(n-1) \cdot s(n) \\ x_2(n) = x_2(n-1) \cdot \overline{u(n)} + \left(\overline{x_1(n-1)} + x_1(n-1) \cdot \overline{x_2(n-1)} \right) \cdot u(n) \end{cases} \quad (13)$$

where $x_1(n), x_1(n-1) \in \{0, \dots, 2^D - 1\}$; $s(n)$ and $u(n)$ are two independent random sequences. \cdot means and operation, $+$ means or operation, \bar{x} means inverting x . The corresponding strongly connected topology is shown in Fig. 3.

$A_{M \times M}$ is a square matrix of order $M = 2^{2 \times D}$, where D denotes precision. As demonstrated in Ref. [24], state transition diagrams associated with $A_{M \times M}$ remain strongly connected at different precision levels. Therefore, by altering the precision level D of the chaotic system, one can change the size of $A_{M \times M}$.

3.4 Optimizing the Reservoir

In ESN, the reservoir is the core component of the network, comprising a large number of hidden layer neurons that generate nonlinear dynamics through internal connections. The introduction of strongly connected state transition diagrams ensures interconnections among neurons within the reservoir, ensuring the existence of one or more pathways from any neuron to others. A strongly connected reservoir structure holds significant importance in ESN for several reasons:

Dynamic Behavior and Memory Capacity: A strongly connected reservoir can facilitate complex dynamic behaviors, which are crucial for handling time-series data and capturing

long-term dependencies within the data. This connectivity allows the network to generate rich internal nonlinear dynamics, thereby endowing it with strong memory capacity and pattern extraction capabilities.

Stability and Performance: The strong connectivity condition can contribute to the stability and robustness of the network. When neurons within the reservoir are interconnected, and information can flow throughout the entire network, the network's dynamics tend to be more stable. This is important for maintaining the characteristics of an ESN and enhancing its predictive performance.

Decoupling and Information Flow: The strongly connected reservoir structure aids in decoupling the influence between different neurons, allowing information within the network to propagate among various neurons without being constrained by local connectivity patterns. This facilitates the network in handling complex input patterns more effectively.

Preventing Dead Neurons: Within a strongly connected reservoir, information can flow throughout the network, mitigating the issue of certain neurons remaining inactive due to insufficient input, thus preventing the occurrence of so-called "dead neuron" problems.

In summary, the strong connectivity condition contributes to the reservoir's generation of rich dynamic behaviors, enhances network stability and performance, and enables the network to better capture the structure and features of data, particularly when handling tasks involving time series analysis.

According to Ref. [24], the better the corresponding network performance, the richer the dynamic characteristics inside the reservoir of the ESN, and the internal dynamic characteristics of the reservoir are closely related to its internal topological structure. In the traditional ESN, the internal connection matrix W is generated by two random matrices W_R and W_S according to Eq. 3. Since both W_R and W_S are generated randomly, W usually does not satisfy the condition of strong connectivity, and it lacks the property of strong connectivity.

The strongly connected adjacency matrix $A_{M \times M}$ mentioned in Sect. 3.3 was used to replace the randomly generated W_S in Eq. 3, so that the internal connection matrix W of the reservoir satisfies the strong connectivity condition and has the strong connectivity characteristic, namely

$$W^{SC} = R \frac{W_R \odot A_{M \times M}}{|\lambda_{\max}|} \quad (14)$$

3.5 Ch-SC-ESN

The proposed Ch-SC-ESN model improves upon the traditional ESN in two ways. Firstly, the input weight matrix W_{in}^{Ch} is generated through Chebyshev mapping iterations, replacing the completely random generation of W_{in} in traditional ESN. Secondly, the strongly connected adjacency matrix $A_{M \times M}$ is used to replace the randomly generated W_S in the original ESN. This modification ensures that the internal connection matrix W of the reservoir satisfies the strong connectivity condition and has the strong connectivity characteristic.

The generation and training process of Ch-SC-ESN is as follows:

Step 1: Determine some basic parameters.

Step 2: Generate W_{in}^{Ch} : use Eq. 10 to generate the first row element of W_{in}^{Ch} , and then substitute the first row element into Eq. 11 to iteratively generate the other elements of W_{in}^{Ch} .

Step 3: Generate W^{SC} : Randomly generate a random matrix W_R with size $M \times M$ where $M = N$ and N represent the number of neurons in the reservoir, with elements ranging from $(-0.5, 0.5)$, use the two-dimensional digital chaotic system Eq. 13, let $D = \log_2 \frac{M}{2}$ generate strongly connected adjacency matrix $A_{M \times M}$. The randomly generated W_R and

strongly connected adjacency matrix $A_{M \times M}$ are substituted into Eq. 14 to generate W^{SC} with strongly connected characteristics, and let $W = W^{SC}$.

Step 4: Substitute the generated W_{in}^{Ch} and W into the internal state update Eq. 12.

Step 5: From the linear output Eq. 4, the output weight matrix W_{out} is solved by ridge regression method.

At this point, the generation and training process of Ch–SC–ESN was completed.

3.6 Stability Analysis

Echo state property (ESP) refers to the property of an ESN's reservoir, wherein after a period of dynamic evolution, it gradually forgets its initial state and becomes relatively insensitive to changes in input data. In other words, even as the internal state of the network becomes highly complex over time, it still retains some information from previous input data. This allows the network to better capture long-term dependencies in time series data. The condition that the maximum singular value of the reservoir weight matrix is < 1 ensures that the ESN possesses the ESP property.

When employing a strongly connected topology to optimize the internal reservoir structure of a traditional ESN, the reservoir continues to exhibit the ESP. The presence of ESP in the reservoir can be validated by examining the reservoir's state. Specifically, the reservoir retains the ESP when the maximum singular value of the reservoir weight matrix W^{SC} , denoted as $\sigma(W^{SC}) = \|W^{SC}\| < 1$, remains < 1 .

Suppose $x(n+1) = \tanh(W_{in}^{Ch}[1; u(n)] + W^{SC}x(n))$ and $x'(n+1) = \tanh(W_{in}^{Ch}[1; u(n)] + W^{SC}x'(n))$ are two state vectors of the reservoir.

$$\begin{aligned} & \|x(n+1) - x'(n+1)\|_2 \\ &= \|\tanh(W_{in}^{Ch}[1; u(n)] + W^{SC}x(n)) - \tanh(W_{in}^{Ch}[1; u(n)] + W^{SC}x'(n))\|_2 \\ &\leq \|W_{in}^{Ch}[1; u(n)] + W^{SC}x(n) - (W_{in}^{Ch}[1; u(n)] + W^{SC}x'(n))\|_2 \\ &= \|W^{SC}x(n) - W^{SC}x'(n)\|_2 \\ &\leq \|W^{SC}\|_2 \|x(n) - x'(n)\|_2 \end{aligned} \quad (15)$$

when the maximum singular value of the weight matrix W^{SC} within the reservoir is < 1 , the states of the reservoir neurons tend to approach stability. Furthermore, the reservoir optimized with a strongly connected topology retains the ESP.

4 Experimental Results and Evaluation

In this section, the prediction performance of the proposed Ch–SC–ESN is verified by predicting chaotic time series. The proposed Ch–SC–ESN is evaluated by using three tasks of simulating chaotic time series, namely Mackey–Glass chaotic time series, discrete Lorenz chaotic time series prediction [14] and Rossler chaotic time series prediction. The use of two time series of real world problems, respectively, sunspots time series prediction [42] and solar power generation sequence prediction tasks (open source download link: <https://www.Nrel.gov/grid/solar-power-data.html>) to verify performance of the Ch–SC–ESN.

A comparative analysis of predictive performance has been conducted between Ch–SC–ESN and several other models, including traditional ESN, grouped ESN (GESN) [43], deep ESN (DESN) [44], and binary grey wolf algorithm-optimized ESN (BGW-ESN) [41]. Additionally, to assess the impact of Chebyshev mapping optimization on input weight matrices and strongly connected topology optimization on model performance, Ch–SC–ESN was

compared with two other models, namely Ch-ESN and SC-ESN. This comparison aimed to further validate the effectiveness of Ch-SC-ESN.

To ensure experimental fairness, in this study, we standardized the spectral radius (R) to 0.8 for all ESN models. Considering the positive correlation between reservoir size and the prediction accuracy of ESNs, we paid particular attention to this factor. To mitigate the impact of reservoir size on prediction performance, the reservoir size was set to 500 for the other ESN prediction models. Furthermore, to meet the condition of generating a strongly connected topology using digital chaotic systems, the reservoir size (N) employed in the Ch-SC-ESN model was set to 256, which is smaller compared to the reservoir size used in the other ESN prediction models. All experiments were conducted on an Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz, using MATLAB R2021 as the software version.

Normalized mean square error ($NMSE$), mean absolute error (MAE) and mean absolute percentage error ($MAPE$) are used to evaluate the prediction performance. In addition, $IM\%$ is used to represent the performance improvement percentage of different models compared with the traditional ESN. The following is the definition formula of several errors and $IM\%$:

$$NMSE = \sum_{n=1}^i \frac{(o(n) - y(n))^2}{i\sigma^2} \quad (16)$$

$$MAE = \sum_{n=1}^i \frac{|o(n) - y(n)|}{i} \quad (17)$$

$$MAPE = \frac{1}{i} \sum_{n=1}^i \left| \frac{o(n) - y(n)}{y(n)} \right| \times 100\% \quad (18)$$

$$IM\% = \frac{NMSE_{ESN} - NMSE_{Model}}{NMSE_{ESN}} \times 100\% \quad (19)$$

where $y_{\text{target}}(n)$ is the target output, $y(n)$ is the actual output, σ^2 represents the variance of $y_{\text{target}}(n)$, i represents the length of the output, $NMSE_{ESN}$ represents the normalized mean square error of the traditional ESN, and $NMSE_{\text{Model}}$ is the normalized mean square error of the model used for comparison.

4.1 Selection and Impact of Control Parameters

In this section, the selection of control parameter k is discussed using $NMSE$ as an indicator of validation error, and the minimum validation error obtained in the validation model is selected as the parameter for the final test.

In traditional ESN, the element values of the input weight matrix are typically limited to the range of -1 to 1 , which can result in feature loss during the transmission of input signals from the input layer to the reservoir. As outlined in Sects. 3.1 and 3.2, the control parameter k has a considerable impact on the bifurcation diagram of the input weight matrix in Ch-ESN. To determine the appropriate value of the control parameter k , the $NMSE$ is calculated during verification, as shown in Fig. 4. The value of k that yields the smallest verification error is selected as the final value of k . Figure 4 illustrates the verification and test $NMSE$ for various time series tasks with different values of k .

Depending on the different ranges of k between 0 and 4, the Chebyshev mapping can produce three different dynamic properties, which can be defined as three phases. Figure 5 illustrates the input weight values of the first 100 neurons for both the original ESN and

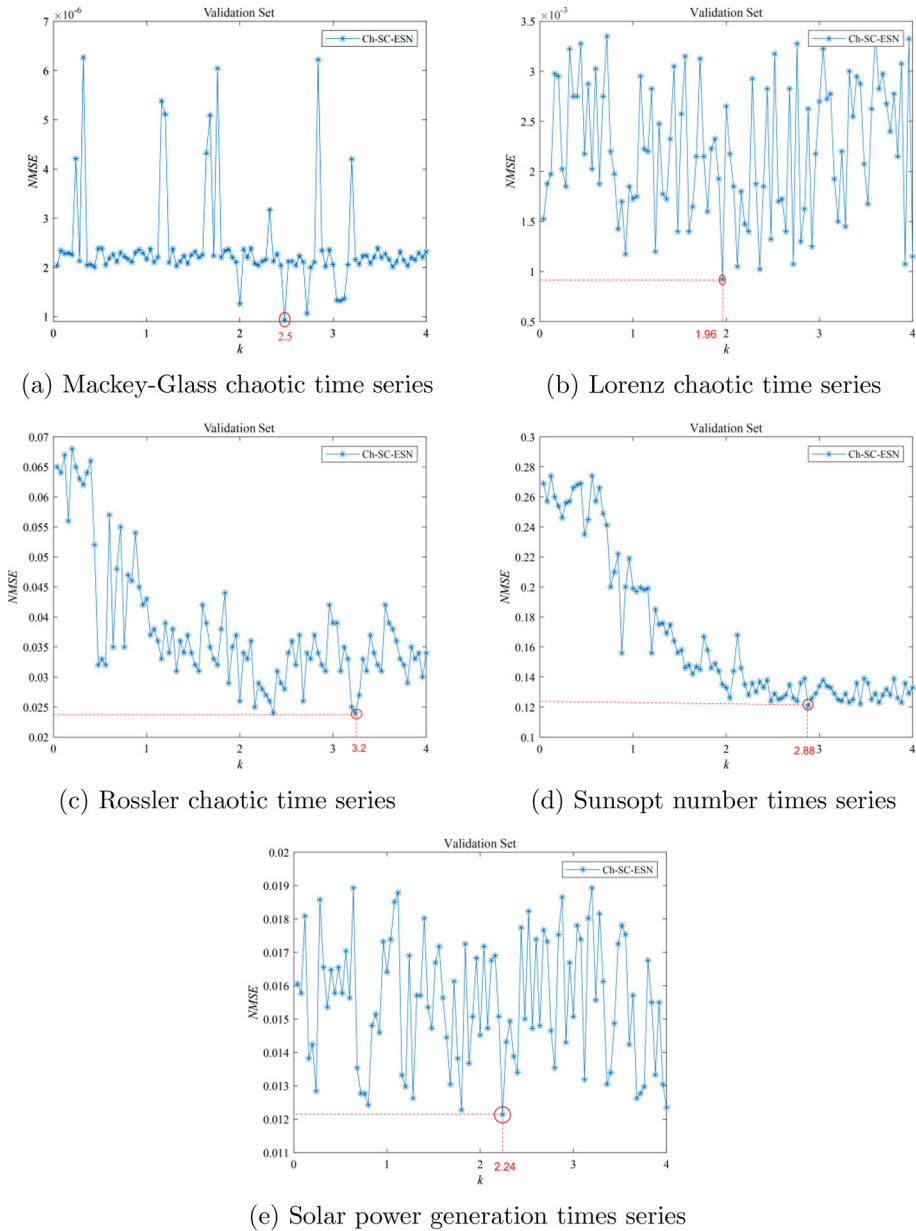


Fig. 4 Verification NMSE and test NMSE under different time series tasks and different control parameters k

Ch-SC-ESN with different values of k in three phases. It confirms that input weight matrix values within the control parameter range of k from 2 to 4 lead to chaotic behavior and exhibit superior randomness compared to other values.

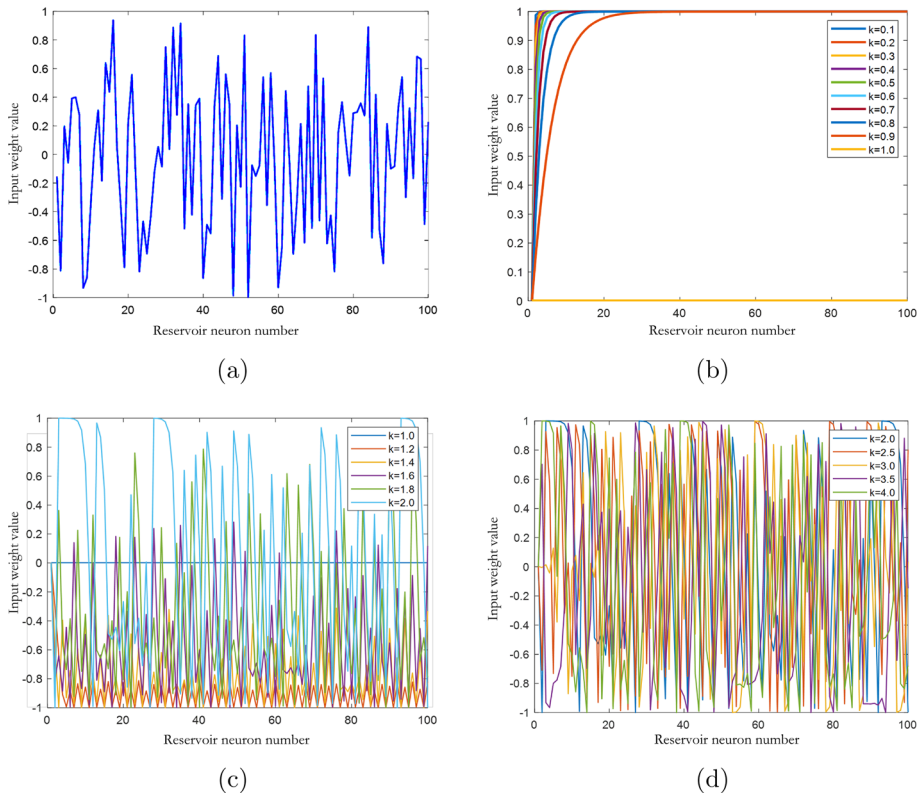


Fig. 5 When controlling different values of parameter k , input the weight value. **a** Random weight value; **b** the control parameter k is within $(0, 1)$; **c** the control parameter k is within $(1, 2)$; **d** the value of the control parameter k is within $(2, 4)$

4.2 Experiment on Mackey–Glass Chaotic Time Series Task

Mackey–Glass chaotic time series [45] prediction is a typical problem to verify the ability of ESN to process information. Mackey–Glass chaotic time series is generated by Eq. 20:

$$\frac{dx(t)}{dt} = \mu x(t) + \frac{\xi x(t - \tau)}{1 + x^{10}(t - \tau)} \tag{20}$$

The condition for Mackey–Glass time series to exhibit chaotic characteristics is $\tau > 16.8$. In order to make Mackey–Glass time series exhibit chaotic dynamic characteristics, each parameter in the experiment is set as $\tau = 17$, $\mu = -0.1$, $\xi = 0.2$ and the initial value $x(0) = 1.2$. The function `dde23` in Matlab tool was used to solve Eq. 20 to generate Mackey–Glass chaotic time series. Mackey–Glass chaotic time series was divided into initialization sequence, training sequence and test sequence, the lengths of which were $initLen = 100$, $trainLen = 2000$ and $testLen = 1000$ respectively.

As shown in Fig. 6b, the Ch–SC–ESN model excels in predicting the Mackey–Glass chaotic time series, with significantly lower prediction errors compared to the ESN model. This indicates a notable advantage of Ch–SC–ESN in addressing chaotic time series prediction problems. In Table 1, we conducted a comprehensive comparison of Ch–SC–ESN with

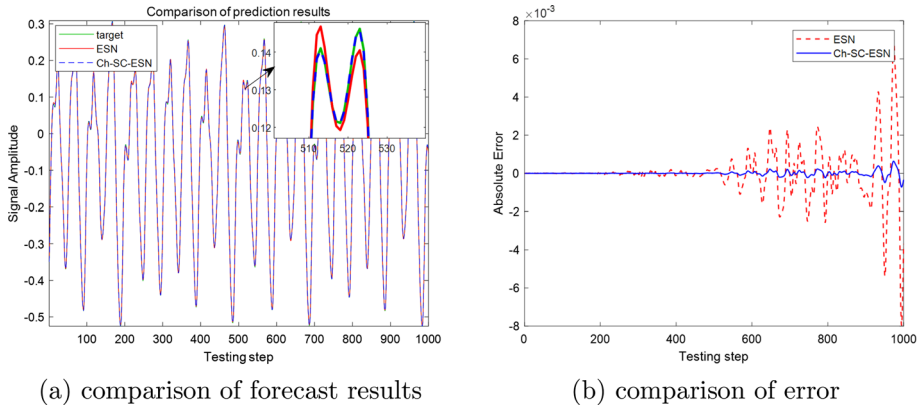


Fig. 6 Mackey–Glass chaotic time series prediction. **a** comparison of forecast results, **b** comparison of error

Table 1 Prediction performance ratios of Ch–SC–ESN and other prediction models for Mackey–Glass chaotic time series

	ESN	GESN	DESN	BGW–ESN	Ch–ESN	SC–ESN	Ch–SC–ESN
<i>NMSE</i>	$8.36e-04$	$9.75e-06$	$7.17e-06$	$5.79e-05$	$8.25e-05$	$3.04e-05$	$9.32e-07$
<i>MAE</i>	0.0085	0.0021	0.0017	0.0046	0.0019	0.0068	$1.89e-04$
<i>MAPE (%)</i>	46.15	5.07	4.76	19.47	4.977	35.64	2.19
<i>IM%</i>	–	98.83%	99.14%	93.07%	99.01%	63.64%	99.89%
Running time (s)	0.1997	0.2547	0.3659	0.1986	0.0827	0.0638	0.1195

other prediction models regarding their performance in predicting the Mackey–Glass chaotic time series. Through the comparison of these metrics, we gain a clearer understanding of how different prediction models perform in predicting the Mackey–Glass chaotic time series. The results in Table 1 clearly demonstrate that Ch–SC–ESN outperforms other models in all performance metrics. Specifically, its *NMSE*, *MAE* and *MAPE* values are significantly lower, and the performance improvement metric *IM%* reaches 99.89%, indicating that Ch–SC–ESN delivers more accurate prediction results.

4.3 Experiment on Lorenz Times Series Task

The Lorenz system is a classic benchmark function for time series prediction [44], and its formula is as follows:

$$\begin{cases} dx/dt = a_1(y - x) \\ dy/dt = -xz + a_2x - y \\ dz/dt = xy - a_3z \end{cases} \quad (21)$$

among them, a_1 , a_2 and a_3 are system parameters. To ensure that the Lorenz system has chaotic characteristics, the typical values of these system parameters are $a_1 = 10$, $a_2 = 28$, $a_3 = 8/3$. $x(t)$, $y(t)$ and $z(t)$ is the three-dimensional space vector of the Lorenz system. The fourth order Runge Kutta method is used to generate 10,000 sample datasets, and the x dimension sample $x(t)$ is used as a time series prediction. In the data sample set, the first 500

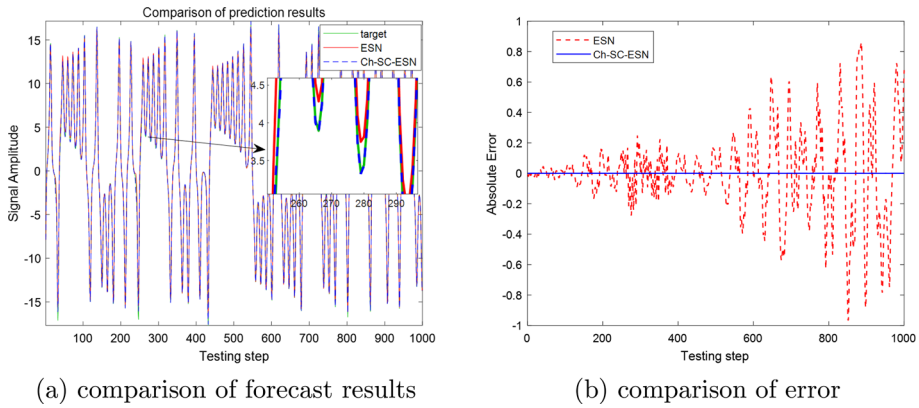


Fig. 7 Lorenz chaotic time series prediction. **a** Comparison of forecast results, **b** comparison of error

Table 2 Prediction performance ratios of Ch–SC–ESN and other prediction models for Lorenz chaotic time series

	ESN	GESN	DESN	BGW–ESN	Ch–ESN	SC–ESN	Ch–SC–ESN
<i>NMSE</i>	0.00197	0.00156	0.00086	0.00194	0.00167	0.00143	0.00096
<i>MAE</i>	0.00367	0.00278	0.00166	0.00438	0.00315	0.00298	0.00207
<i>MAPE (%)</i>	25.96	19.69	6.77	24.36	22.43	18.66	8.99
<i>IM%</i>	–	20.81%	56.35%	1.52%	15.23%	27.41%	51.27%
Running time (s)	0.1758	0.3554	0.4726	0.1635	0.0727	0.0536	0.1169

samples are used for initialization, the next 2500 are used for training, and the next 2000 are used for testing, namely $initLen = 500$, $trainLen = 2500$ and $testLen = 1000$.

Figure 7a, b represent the comparison of prediction results and errors between Ch–SC–ESN and ESN in forecasting the Lorenz chaotic time series. From these two figures, it is evident that Ch–SC–ESN significantly outperforms ESN in predicting the Lorenz chaotic time series. To further quantify the superior performance of Ch–SC–ESN compared to other prediction models in forecasting the Lorenz chaotic time series, we have listed various evaluation metrics in Table 2 for comparing Ch–SC–ESN with other models. Upon analyzing Table 2, it is apparent that the deep ESN (DESN) excels in predicting the Lorenz chaotic time series, with a performance improvement metric (*IM%*) reaching 56.36%. Ch–SC–ESN’s prediction performance is close to that of DESN, with an *IM%* of 51.27%, surpassing other prediction models and demonstrating outstanding performance.

4.4 Experiment on Rossler Times Series Task

The ordinary differential equation for Rossler time series is:

$$\begin{cases} dx/dt = -z - y \\ dy/dt = x + ay \\ dz/dt = b + z(x - c) \end{cases} \quad (22)$$

when $a = 0.15$, $b = 0.2$ and $c = 10$, the Rossler system has chaotic characteristics. The Runge Kutta method is used to generate discrete Rossler chaotic time series on the x

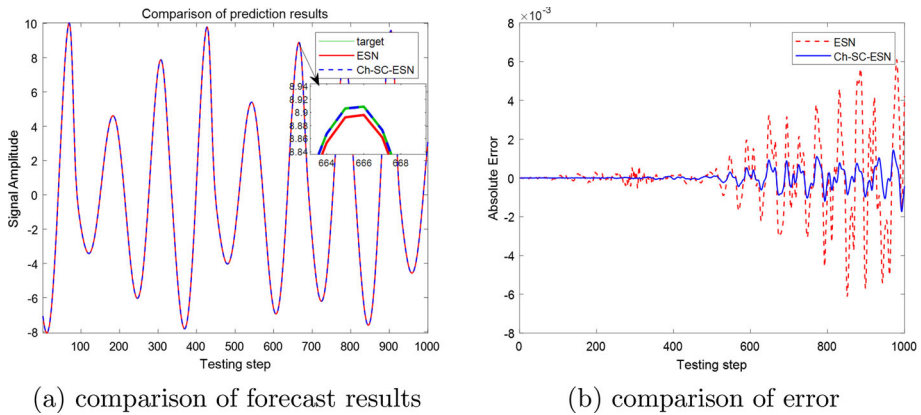


Fig. 8 Rossler chaotic time series prediction. **a** comparison of forecast results, **b** comparison of error

dimension. Generate 10,000 x dimensional sample data $x(t)$ for the prediction task of the time series. In the data sample set, the first 500 samples are used for initialization, the next 2500 are used for training, and the next 2000 are used for testing, namely $initLen = 500$, $trainLen = 2500$ and $testLen = 2000$.

By observing Fig. 8a, b, it is clearly evident that Ch–SC–ESN exhibits a significant advantage over ESN in predicting the Rossler chaotic time series, with significantly lower prediction errors. To provide a clearer comparison of the performance of Ch–SC–ESN against other models in forecasting the Rossler chaotic time series, we have presented various predictive performance evaluation metrics in Table 3. An analysis of the performance indicators in Table 3 reveals that Ch–SC–ESN has the lowest $NMSE$, MAE and $MAPE$ values, and it achieves a performance improvement metric ($IM\%$) of 59.23%. As a result, Ch–SC–ESN demonstrates superior performance in predicting the Rossler chaotic time series.

4.5 Experiment on Sunspot Number Times Series Task

Sunspots are a crucial characteristic of solar activity that have a significant impact on Earth. Accurately modeling sunspots is crucial for predicting solar activity. However, due to the complexity of solar activity and the lack of appropriate mathematical and statistical models, predicting sunspot time series is a challenging task. In this study, we used sunspot sequences to evaluate the predictive ability of Ch–SC–ESN for real-world time series problems. The dataset

Table 3 Prediction performance ratios of Ch–SC–ESN and other prediction models for Rossler chaotic time series

	ESN	GESN	DESN	BGW–ESN	Ch–ESN	SC–ESN	Ch–SC–ESN
$NMSE$	0.0596	0.0473	0.0247	0.0376	0.0297	0.0386	0.0243
MAE	0.0468	0.0320	0.0267	0.0355	0.0335	0.0427	0.0269
$MAPE$ (%)	6.96	5.68	3.52	4.79	3.98	5.13	3.49
$IM\%$	–	20.64%	58.56%	36.91%	50.16%	35.23%	59.23%
Running time (s)	0.2672	0.3421	0.5869	0.2245	0.1146	0.0929	0.1419

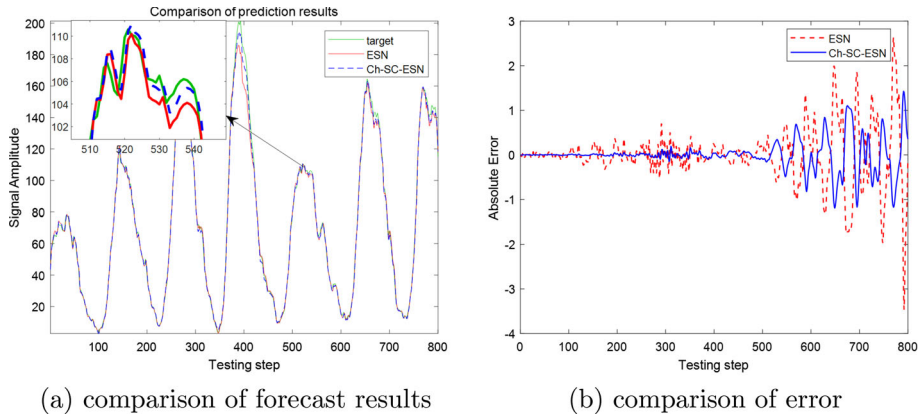


Fig. 9 sunspot number time series prediction. **a** Comparison of forecast results, **b** comparison of error

used in this study consisted of the average monthly sunspot count from June 1759 to April 2021, with a length of 3142. Specifically, the first 200 samples were used for initialization, the next 2000 samples were used for training, and the subsequent 800 samples were used for testing purposes, namely $initLen = 200$, $trainLen = 2000$ and $testLen = 800$.

By observing Fig. 9a, b, it is clearly evident that Ch-SC-ESN exhibits a significant difference in predicting solar sunspot sequences compared to ESN, with markedly lower prediction errors. To provide a more intuitive comparison of the performance of Ch-SC-ESN against other models in predicting solar sunspot sequences, we have presented various predictive performance evaluation metrics in Table 4. An analysis of the predictive performance evaluation metrics in Table 4 reveals that Ch-SC-ESN has the lowest $NMSE$, MAE and $MAPE$ values, and it achieves a higher performance improvement metric ($IM\%$) compared to other predictive models. Therefore, Ch-SC-ESN outperforms other predictive models in forecasting solar sunspot sequences.

4.6 Experiment on Solar power Generation Times Series Task

The experimental dataset used for prediction is the solar power generation capacity in New York in 2021, with data recorded every 5 min. A total of 10,000 data samples were selected from the New York Power Plant, with the first 500 samples used for initialization, the next 6000 samples used as training data, and the final 1000 samples used for testing. By analyzing

Table 4 Prediction performance ratios of Ch-SC-ESN and other prediction models for sunspot number time series

	ESN	GESN	DESN	BGW-ESN	Ch-ESN	SC-ESN	Ch-SC-ESN
$NMSE$	0.01569	0.1325	0.1525	0.1403	0.1428	0.1367	0.1208
MAE	0.0945	0.0843	0.0938	0.0890	0.0915	0.0863	0.0765
$MAPE$ (%)	71.68	53.18	70.66	70.02	68.55	57.49	42.36
$IM\%$	–	15.55%	2.81%	10.58%	8.99%	12.87%	23.01%
Running time (s)	0.3625	0.4876	0.5866	0.3027	0.1926	0.1857	0.2216

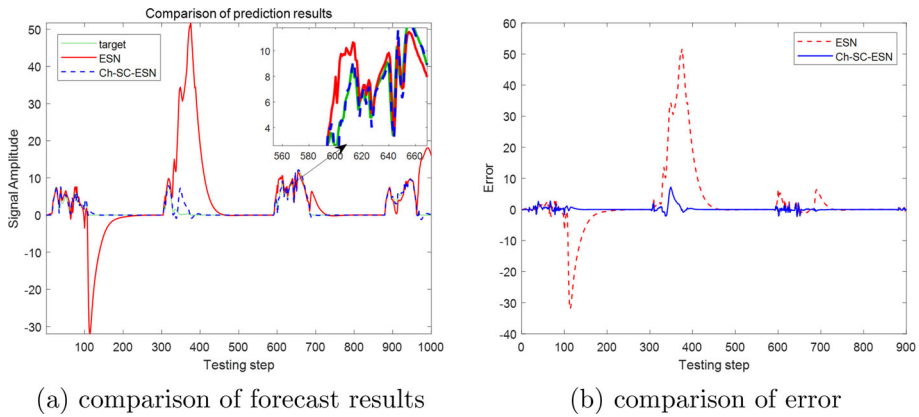


Fig. 10 Solar power generation time series prediction. **a** Comparison of forecast results, **b** comparison of error

Table 5 Prediction performance ratios of Ch–SC–ESN and other prediction models for solar power generation time series

	ESN	GESN	DESN	BGW–ESN	Ch–ESN	SC–ESN	Ch–SC–ESN
<i>NMSE</i>	0.0237	0.0119	0.0189	0.0161	0.0176	0.0202	0.0124
<i>MAE</i>	0.0386	0.0260	0.0390	0.0313	0.0356	0.0364	0.0289
<i>MAPE (%)</i>	25.96	11.60	13.23	15.88	15.33	21.67	12.68
<i>IM%</i>	–	49.37%	20.25%	32.07%	25.74%	14.77%	47.68%
Running time (s)	0.4657	0.5755	0.6979	0.3657	0.2897	0.2465	0.3122

Fig. 10a, b, it is evident that Ch–SC–ESN exhibits significant improvement in forecasting solar power generation time series compared to ESN, with significantly lower prediction errors. To provide a clearer illustration of the performance comparison between Ch–SC–ESN and other models in forecasting solar power generation time series, we have detailed various predictive performance evaluation metrics in Table 5. In-depth analysis of the predictive performance evaluation metrics in Table 5 reveals that grouped ESN (GESN) performs best in forecasting solar power generation time series, achieving a performance improvement metric (*IM%*) of 49.37%. Ch–SC–ESN’s predictive performance is close to that of GESN, with an *IM%* of 47.68%, and it continues to excel among the various models, significantly surpassing other predictive models.

4.7 Complexity Analysis

The complexity of ESN primarily manifests in the complexity of the reservoir network structure, predictive performance, and generalization capability. In the following sections, the complexity of Ch–SC–ESN will be analyzed from these two aspects.

(1) Reservoir Network Structure Complexity

Ch–SC–ESN significantly simplifies the structure of the reservoir network compared to traditional ESNs. Figure 11 illustrates the visualization of the reservoir for both ESN and Ch–SC–ESN, where each point represents a connection between the *x*th neuron and the *y*th neuron. In traditional ESNs, the internal connection matrix *W* within the reservoir is randomly

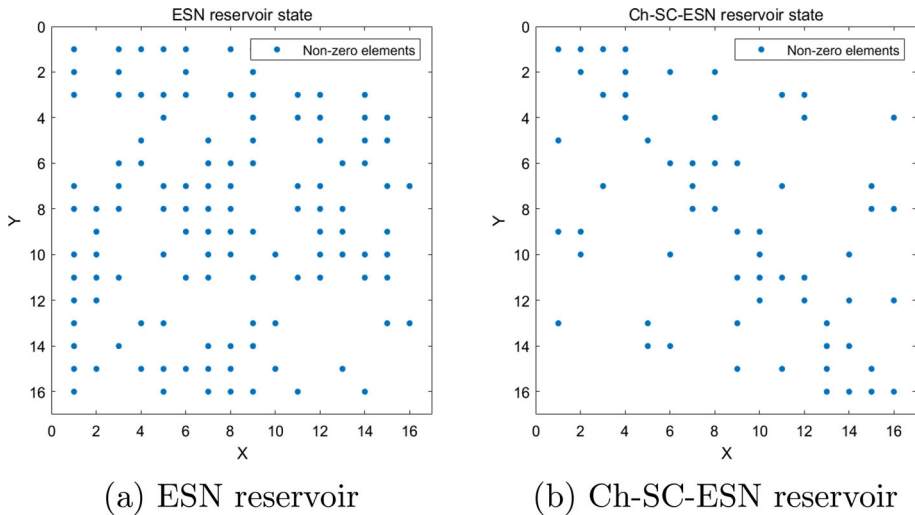


Fig. 11 The visualization of the reservoir. **a** ESN, **b** Ch-SC-ESN

generated, as shown in Fig. 11a. However, in Ch-SC-ESN, the internal connection matrix of the reservoir is obtained by multiplying a strongly connected adjacency matrix with a random matrix. After this optimization, the neuronal connections within the reservoir are depicted in Fig. 11b. Figure 11 clearly demonstrates that, compared to traditional ESNs, Ch-SC-ESN exhibits a sparser arrangement of neurons within the reservoir, resulting in a noticeable reduction in the complexity of the network structure.

(2) Predictive Performance and Generalization Capability

The predictive performance and generalization capability of ESNs are linked to the network's complexity in various time-series tasks. A more complex network may exhibit superior predictive performance in certain time-series tasks but may have limited generalization capability, leading to decreased predictive performance in other time-series tasks. Ch-SC-ESN utilizes Chebyshev mapping iteratively to generate input weight matrices and employs different control parameters k tailored to specific time series tasks. This adaptability allows Ch-SC-ESN to excel in various time series tasks, demonstrating high generalization capability. Simultaneously, through the analysis of predictive results across five distinct time-series tasks, Ch-SC-ESN demonstrates superior predictive performance in most time-series tasks compared to traditional ESNs. It closely competes with the best-performing models in predicting the Rossler time series and solar power generation time series tasks. Therefore, Ch-SC-ESN not only exhibits outstanding predictive performance but also possesses a high degree of generalization capability.

By analyzing the complexity of Ch-SC-ESN, it can be inferred that, relative to traditional ESNs, Ch-SC-ESN has a lower complexity in terms of its reservoir network structure. Furthermore, it demonstrates significant advantages in predictive performance and generalization capability.

4.8 Discussion

The experimental results presented in Sects. 4.1–4.5 demonstrate that the proposed Ch–SC–ESN achieves improved prediction performance while reducing model complexity and reservoir size. In comparison to other classical prediction models, Ch–SC–ESN performs better than most models in various time series prediction tasks, with the exception of the Rossler time series prediction and solar power generation time series prediction tasks, where its performance is comparable to the best-performing models.

Chebyshev mapping can generate three different dynamic characteristics based on the control parameter k within the range of 0–4, which can be defined as three phases and illustrated in Figs. 2 and 5. Results from Figs. 4 and 5 indicate that Ch–SC–ESN with chaotic input weights generally outperforms the traditional ESN in most cases. Additionally, as shown in Fig. 11 Ch–SC–ESN can effectively reduce the size of the reservoir and model complexity while improving prediction performance compared to traditional ESN.

5 Conclusion

In this paper, an input weight matrix $A_{M \times M}$ is generated using Chebyshev mapping for optimizing the one generated by traditional ESN, and a strong connected adjacency matrix W_{in} is also introduced for replacing in the internal connection matrix W . These two optimization steps ensure that the topological structure of the reservoir possesses strong connectivity characteristics. Building upon these enhancements, predictions were conducted on three simulated chaotic time series and two real-world time series. Experimental results demonstrate that the proposed Ch–SC–ESN model outperforms other prediction models in terms of predictive accuracy, exhibits strong generalization capabilities, and displays excellent adaptability across various time series prediction tasks. The Ch–SC–ESN model may not perform optimally in predicting certain datasets, such as the Lorenz chaotic time series and the solar power generation time series. Further optimization is required to tailor the model for these specific sequences.

Funding This work was supported by the National Natural Science Foundation of China under Grant 62271157 and the Natural Science Foundation of Guangdong Province Grant 22A1515010005.

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

1. Weerakody PB, Wong KW, Wang G, Ela W (2021) A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing* 441:161–178
2. Zhang H, Wang Z, Liu D (2014) A comprehensive review of stability analysis of continuous-time recurrent neural networks. *IEEE Trans Neural Netw Learn Syst* 25(7):1229–1262
3. Tanaka G, Yamane T, Héroux JB, Nakane R, Kanazawa N, Takeda S, Numata H, Nakano D, Hirose A (2019) Recent advances in physical reservoir computing: a review. *Neural Netw* 115:100–123

4. Jaeger H (2001) The “echo state” approach to analysing and training recurrent neural networks—with an erratum note, vol 148, p 13. GMD Technical Report, German National Research Center for Information Technology, Bonn, Germany
5. Jaeger H, Haas H (2004) Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* 304(5667):78–80
6. Massar M, Massar S (2013) Mean-field theory of echo state networks. *Phys Rev E* 87(4):042809
7. Chattopadhyay A, Hassanzadeh P, Subramanian D (2020) Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Process Geophys* 27(3):373–389
8. Li D, Han M, Wang J (2012) Chaotic time series prediction based on a novel robust echo state network. *IEEE Trans Neural Netw Learn Syst* 23(5):787–799
9. Hu H, Wang L, Tao R (2021) Wind speed forecasting based on variational mode decomposition and improved echo state network. *Renew Energy* 164:729–751
10. Han Z, Zhao J, Leung H, Ma KF, Wang W (2019) A review of deep learning models for time series prediction. *IEEE Sens J* 21(6):7833–7848
11. Rodan A, Tino P (2010) Minimum complexity echo state network. *IEEE Trans Neural Netw* 22(1):131–144
12. Qiao J, Li F, Han H, Li W (2016) Growing echo-state network with multiple subreservoirs. *IEEE Trans Neural Netw Learn Syst* 28(2):391–404
13. Xu M, Han M (2016) Adaptive elastic echo state network for multivariate time series prediction. *IEEE Trans Cybern* 46(10):2173–2183
14. Yang C, Qiao J, Han H, Wang L (2018) Design of polynomial echo state networks for time series prediction. *Neurocomputing* 290:148–160
15. Shi G, Zhao B, Li C, Wei Q, Liu D (2019) An echo state network based approach to room classification of office buildings. *Neurocomputing* 333:319–328
16. Verstraeten D, Schrauwen B, d’Haene M, Stroobandt D (2007) An experimental unification of reservoir computing methods. *Neural Netw* 20(3):391–403
17. Larger L, Baylón-Fuentes A, Martinenghi R, Udaltsov VS, Chembo YK, Jacquot M (2017) High-speed photonic reservoir computing using a time-delay-based architecture: million words per second classification. *Phys Rev X* 7(1):011015
18. Zhao K, Li L, Chen Z, Sun R, Yuan G, Li J (2022) A new multi-classifier ensemble algorithm based on DS evidence theory. *Neural Process Lett* 54(6):5005–5021
19. Peng Z, Wang J, Wang D (2017) Distributed containment maneuvering of multiple marine vessels via neurodynamics-based output feedback. *IEEE Trans Ind Electron* 64(5):3831–3839
20. Bo Y-C, Zhang X (2018) Online adaptive dynamic programming based on echo state networks for dissolved oxygen control. *Appl Soft Comput* 62:830–839
21. Peng Z, Wang J, Wang D (2017) Distributed containment maneuvering of multiple marine vessels via neurodynamics-based output feedback. *IEEE Trans Ind Electron* 64(5):3831–3839
22. Yao X, Wang Z (2020) Fractional order echo state network for time series prediction. *Neural Process Lett* 52:603–614
23. Li X, Bi F, Yang X, Bi X (2022) An echo state network with improved topology for time series prediction. *IEEE Sens J* 22(6):5869–5878
24. Wang Q, Yu S, Guyeux C, Wang W (2021) Constructing higher-dimensional digital chaotic systems via loop-state contraction algorithm. *IEEE Trans Circuits Syst I Regul Pap* 68(9):3794–3807
25. Gallicchio C, Micheli A, Pedrelli L (2017) Deep reservoir computing: a critical experimental analysis. *Neurocomputing* 268:87–99
26. Wang Z, Yao X, Huang Z, Liu L (2021) Deep echo state network with multiple adaptive reservoirs for time series prediction. *IEEE Trans Cogn Dev Syst* 13(3):693–704
27. Kim T, King BR (2020) Time series prediction using deep echo state networks. *Neural Comput Appl* 32:17769–17787
28. Li Z, Tanaka G (2022) Multi-reservoir echo state networks with sequence resampling for nonlinear time-series prediction. *Neurocomputing* 467:115–129
29. Zhang H, Hu B, Wang X, Xu J, Wang L, Sun Q, Wang Z (2021) Self-organizing deep belief modular echo state network for time series prediction. *Knowl Based Syst* 222:107007
30. Chouikhi N, Ammar B, Rokbani N, Alimi AM (2017) PSO-based analysis of echo state network parameters for time series forecasting. *Applied Soft Computing* 55:211–225
31. Wang H, Liu Y, Lu P, Luo Y, Wang D, Xu X (2022) Echo state network with logistic mapping and bias dropout for time series prediction. *Neurocomputing* 489:196–210
32. Ferreira AA, Ludermer TB, De Aquino RR (2013) An approach to reservoir computing design and training. *Expert Syst Appl* 40(10):4172–4182

33. Xu M, Han M (2016) Adaptive elastic echo state network for multivariate time series prediction. *IEEE Trans Cybern* 46(10):2173–2183
34. Chen X, Liu M, Li S (2023) Echo state network with probabilistic regularization for time series prediction. *IEEE/CAA J Autom Sin* 10(8):1743–1753
35. Viehweg J, Worthmann K, Mäder P (2023) Parameterizing echo state networks for multi-step time series prediction. *Neurocomputing* 522:214–228
36. Li Z, Liu Y, Tanaka G (2023) Multi-reservoir echo state networks with Hodrick–Prescott filter for nonlinear time-series prediction. *Appl Soft Comput* 135:110021
37. Lukoševičius M (2012) A practical guide to applying echo state networks, pp 659–686. Springer
38. Yildiz IB, Jaeger H, Kiebel SJ (2012) Re-visiting the echo state property. *Neural Netw* 35:1–9
39. Ren H-P, Yin H-P, Bai C, Yao J-L (2020) Performance improvement of chaotic baseband wireless communication using echo state network. *IEEE Trans Commun* 68(10):6525–6536
40. Liu Y, Luo Y, Song S, Cao L, Liu J, Harkin J (2017) Counteracting dynamical degradation of digital chaotic Chebyshev map via perturbation. *Int J Bifurc Chaos* 27(03):1750033
41. Liu J, Sun T, Luo Y, Yang S, Cao Y, Zhai J (2020) Echo state network optimization using binary grey wolf algorithm. *Neurocomputing* 385:310–318
42. Chen W, Xu H, Chen Z, Jiang M (2021) A novel method for time series prediction based on error decomposition and nonlinear combination of forecasters. *Neurocomputing* 426:85–103
43. Gallicchio C, Micheli A, Pedrelli L (2017) Deep reservoir computing: a critical experimental analysis. *Neurocomputing* 268:87–99
44. Long J, Zhang S, Li C (2019) Evolving deep echo state networks for intelligent fault diagnosis. *IEEE Trans Ind Inform* 16(7):4928–4937
45. Farmer JD, Sidorowich JJ (1987) Predicting chaotic time series. *Phys Rev Lett* 59(8):845

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