Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Wavelet-denoising multiple echo state networks for multivariate time series prediction

Meiling Xu, Min Han*, Hongfei Lin

Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Chuangxin Building, Dalian 116023, China

ARTICLE INFO

Article history:
Received 12 April 2017
Revised 15 June 2018
Accepted 8 July 2018
Available online 10 July 2018

Keywords:
Echo state networks
Wavelet denoising algorithm
Multivariate time series
Prediction

ABSTRACT

Motivated by the idea of ’decomposition and ensemble’, this paper proposes a novel method based on the wavelet-denoising algorithm and multiple echo state networks to improve the prediction accuracy of noisy multivariate time series. The noisy time series is first denoised by a wavelet soft thresholding algorithm and decomposed into a set of well-behaved constitutive series. Each constitutive series is then predicted by a separate echo state network with proper parameters that match the specified dynamics. Finally, the overall prediction is achieved by a linear combination of the constitutive series. For each constitutive series, we use the correlation integral method to select the phase-reconstruction parameters and to construct the appropriate input. Two sets of multivariate time series are investigated using the proposed model and some other related work. The simulation results demonstrate the effectiveness of the proposed method.

1. Introduction

Multivariate time series prediction is an open problem in which past observations of the same variables are collected and analyzed to model the underlying relationships between nonlinear systems [4,7,33,35]. It requires dynamic computational models to store and access the time history. The most widely used dynamic model is recurrent neural networks (RNNs) [4] that couple delay lines in a nonlinear architecture to achieve time embedding. However, one of the key problems associated with RNNs is the difficulty in adapting the weights. A variety of algorithms have been used to train RNNs such as the Levenberg–Marquardt method and quasi-Newton method; however, these algorithms suffer from high computational complexity, slow training speed, local minimum, and potential instability [23,39].

In the past few decades, randomized algorithms for training neural networks have become popular [28,38,45,51]. One such algorithm is the reservoir computing algorithm used in echo state networks (ESNs), which has been proposed for modeling nonlinear dynamic systems [21,26,30]. The main concept of this algorithm is to train the output weights alone by using a simple linear regression method while the other parameters remain fixed once they are properly initialized [27]. The nonlinear reservoir derives from the input signals and generates a high-dimensional dynamic echo response. Then, the echo state is used as a non-orthogonal basis to construct the desired outputs. Recently, ESNs are being widely used in the field of multivariate time-series prediction [6,19,50].

Despite the above mentioned advantages, the setting of ESN parameters, such as input-variable selection and reservoir-parameter initialization, is still an open issue [8]. Several problems exist, including the unnecessarily large sets of input that

* Corresponding author.
E-mail address: minhan@dlut.edu.cn (M. Han).

https://doi.org/10.1016/j.ins.2018.07.015
0020-0255/© 2018 Published by Elsevier Inc.

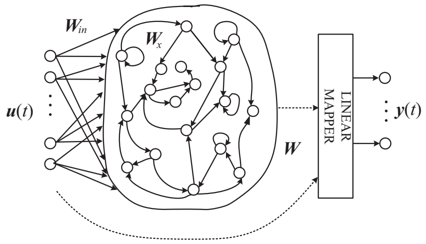


Fig. 1. Structure of a standard ESN.

require long computational time for model training. The input scaling parameter rescales the amplitude of the input and balances the effects of the input and the previous echo state. The reservoir size depends on the task complexity and the size of the training dataset. In addition, the largest singular value of the reservoir-to-reservoir weight matrix plays a key role in the stability of the ESN [2,32].

Before ESNs can be applied in real-world applications, several unresolved issues must be addressed. A major obstacle is the difficulty in denoising the noise and outliers [22]. Some researchers have shown that prediction accuracy can be improved by eliminating noise using methods such as principal component analysis [48], singular spectrum analysis (SSA) [24], wavelet analysis [17]. Gholipour et al. [16] proposed a hybrid model to make long-term prediction of chaotic time series, where they extracted the principal components (PCs) of the time series by SSA, and then built a linear or nonlinear model for each PC to achieve accurate prediction. Zhou et al. [53] proposed a wavelet–autoregressive-moving-average (ARMA) model for the prediction of the monthly discharge time series. The main advantage of wavelet analysis over traditional methods like SSA is its ability to transform the original complex series into different frequencies and times. Each component can be studied with a resolution that matches its scale, which is especially suitable for complex time-series prediction [14,22].

Ensemble learning methods, which achieve better forecasting performance by strategically combining multiple learning algorithms, have been widely used in time-series prediction. Among the various ensemble methods, fast algorithms are more likely to be used with ensembles, such as ARMA, ESNs, and extreme learning machines [5]. Motivated by the idea of 'decomposition and ensemble', the original time series is decomposed into several easy prediction tasks and fine results, and the final prediction values are obtained by adding up the predicted values of each constitutive series [12,34,40,44,46]. In [16] and [53], linear models were proposed; however, they failed to capture the complex nonlinear dynamics sufficiently. Therefore, we combine the wavelet-denoising algorithm with multiple ESNs. The contribution of our work is the use of the wavelet-denoising method to decompose a noisy multivariate time series into a set of noiseless constitutive series. The universal threshold method is employed to eliminate the additive noise [47]. Then, the correlation integral method (C–C method) [15] is used to select an appropriate size for the input time window, and five-fold cross validation is applied to select the best parameters of the ESN for each constitutive series. The future behavior of the constitutive series is predicted using multiple ESNs. Through linear combinations of these constitutive series, the overall prediction is generated.

The rest of this paper is organized as follows. Section 2 reviews the theory of ESNs and the wavelet-denoising algorithm. Section 3 introduces and explains the details of the proposed model for predicting the multivariate time series. Section 4 presents our simulation results on Lorenz series and runoff series. Runoff series prediction is very important in modern water resource management and has received increasing attention from researchers in recent years. Finally, in Section 5, conclusions are drawn and the future work is described.

2. Preliminaries

Multivariate time series are inherently nonstationary signals with observed noise; hence, their behaviors can be analyzed by decomposing on both time and frequency, which are obtained by the wavelet-denoising algorithm. Meanwhile, ESNs as a new kind of RNNs with a simple and fast training process have demonstrated the promising ability to predict complex dynamic behavior. Therefore, the combination of the two methods will lead to more accurate prediction. In this section, we will describe the basic theory of ESNs and the wavelet-denoising algorithm.

2.1. Echo state networks

An ESN consists of an input layer of L units, a hidden layer of M recurrent-connected units (i.e. the reservoir), and an output layer of L typically linear and non-recurrent-connected units. The basic idea of an ESN is to use the larger-than-normal reservoir as a supplier of interesting dynamics [26]. Fig. 1 shows the structure of a standard ESN.

We denote $\boldsymbol{s}(t) = [s_1(t), s_2(t), \ldots, s_L(t)]^T \in \mathbf{R}^{L\times 1}$ as the collected time series, $\boldsymbol{u}(t) = [\boldsymbol{s}(t), \boldsymbol{s}(t-\tau), \ldots, \boldsymbol{s}(t-(m-1)\tau)]$ as the network input, and $\boldsymbol{y}(t) = \boldsymbol{s}(t+p)$ as the network output at time step t, where p is the prediction horizon, m is the embedding dimension, and τ is the delay time [41]. The reservoir state $\boldsymbol{x}(t) \in \mathbf{R}^{M\times 1}$ is generated from the input $\boldsymbol{u}(t)$ and the previous state $\boldsymbol{x}(t-1)$ as follows:

$$\boldsymbol{x}(t) = \tanh(\boldsymbol{W}_{in}\boldsymbol{u}(t) + \boldsymbol{W}_x\boldsymbol{x}(t-1)) \tag{1}$$

where $\boldsymbol{W}_{in} \in \mathbf{R}^{M\times mL}$ is the input-to-reservoir weight matrix and $\boldsymbol{W}_x \in \mathbf{R}^{M\times M}$ is the reservoir-to-reservoir weight matrix. $\tanh(\cdot)$ is the hyperbolic tangent activation function of neurons and is applied to each element. The reservoir state $\boldsymbol{x}(t) \in \mathbf{R}^{M\times 1}$ contains information on the input history in a way that reflects the recent history well and vanishes gradually over time [26]. The initial state of the reservoir is typically set as a zero vector. The linear output layer is defined as

$$\boldsymbol{y}(t) = \boldsymbol{W}^T[\boldsymbol{u}(t) : \boldsymbol{x}(t)] \tag{2}$$

where $[\cdot : \cdot]$ stands for a vertical vector concatenation and $\boldsymbol{W} \in \mathbf{R}^{(L+M)\times L}$ is the reservoir-to-output weight matrix.

All weight matrices to the reservoir ($\boldsymbol{W}_{in}$ and $\boldsymbol{W}_x$) are generated randomly, and are not adaptable, whereas $\boldsymbol{W}$ can be adapted via supervised learning [49]. Not every choice of $\boldsymbol{W}_{in}$ and $\boldsymbol{W}_x$ is valid. The elements of $\boldsymbol{W}_{in}$ are chosen randomly from the range $[-\gamma, \gamma]$, where γ is called the input scaling parameter [20]. Generally, it is a constant less than one. The values of γ are quite different for different applications. A smaller γ corresponds to a situation where the reservoir state $\boldsymbol{x}(t)$ depends more on $\boldsymbol{x}(t-1)$ than on $\boldsymbol{u}(t)$, whereas a larger γ corresponds to a situation where the reservoir state $\boldsymbol{x}(t)$ depends more on $\boldsymbol{u}(t)$ than on $\boldsymbol{x}(t-1)$.

Similarly, the elements of $\boldsymbol{W}_x$ are set as random values from the range $[-1, 1]$, but are largely set to zero based on a fixed percentage of connectivity (less than 10%). The idea behind this is that, a sparsely-connected reservoir ensures rich dynamics and outweighs the fully connected ones in terms of computational efficiency. In a sense, the state of the network is an echo of its input history, i.e. the reservoir dynamics depend only on its past input, and are independent of its initial state values [13]. To ensure this echo state property, the largest singular value of the reservoir-to-reservoir weight matrix $\boldsymbol{W}_x$ ($\sigma(\boldsymbol{W}_x)$) is set to be less than one. This is also a sufficient condition for the global stability of the network [32]. For random initial states $\boldsymbol{x}(0)$ and $\boldsymbol{x}'(0)$,

$$\begin{aligned}
&\|\boldsymbol{x}(t) - \boldsymbol{x}'(t)\|_2 \\
&= \|\tanh(\boldsymbol{W}_{in}\boldsymbol{u}(t) + \boldsymbol{W}_x\boldsymbol{x}(t-1)) - \tanh(\boldsymbol{W}_{in}\boldsymbol{u}(t) + \boldsymbol{W}_x\boldsymbol{x}'(t-1))\|_2 \\
&\le \max(|\tanh'|)\|\boldsymbol{W}_x\boldsymbol{x}(t-1) - \boldsymbol{W}_x\boldsymbol{x}'(t-1)\|_2 \\
&\le \|\boldsymbol{W}_x\|_2\|\boldsymbol{x}(t-1) - \boldsymbol{x}'(t-1)\|_2 \\
&\le \cdots \\
&\le \|\boldsymbol{W}_x\|_2^t\|\boldsymbol{x}(0) - \boldsymbol{x}'(0)\|_2 \to 0,\ \ if\ t \to \infty
\end{aligned} \tag{3}$$

when t goes to infinity, the distance between states $\boldsymbol{x}(t)$ and $\boldsymbol{x}'(t)$ approaches zero for any choice of initial state. Thus, the contractility of the Euclidean norm is ensured, and the echo state property holds. After a sufficiently long period of input sequence, the reservoir state depends only on the input sequence.

We dismiss the first *Init* states, which may be affected by the initial conditions, and collect the subsequent states. Let us denote $\boldsymbol{X}$ as the matrix whose rows are $[\boldsymbol{u}(Init+1): \boldsymbol{x}(Init+1)]^T, \ldots, [\boldsymbol{u}(train): \boldsymbol{x}(train)]^T$. Likewise, we arrange the corresponding target vectors $\boldsymbol{y}(Init+1)^T, \ldots, \boldsymbol{y}(train)^T$ row-wise into matrix $\boldsymbol{Y}$. Assuming that the rows of $\boldsymbol{X}$ are independent and identically distributed, the ordinary least-squares method is the simplest method to estimate the output weights by minimizing the sum of squares between the target values and predicted values.

$$\min \|\boldsymbol{XW} - \boldsymbol{Y}\|_2^2 \tag{4}$$

We can obtain an accurate closed-form solution of the following form [6] without significant computational cost

$$\boldsymbol{W} = \boldsymbol{X}^{\dagger}\boldsymbol{Y} \tag{5}$$

where † denotes the pseudo inverse of $\boldsymbol{X}$.

2.2. Wavelet-denoising algorithm

A wavelet transform provides decomposition in terms of time and frequency, or scale and position [9]. A variety of base functions with different properties can be used. The continuous wavelet transform of a signal $\boldsymbol{f}(t)$ can be expressed as

$$W(\boldsymbol{f})(a, b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{+\infty} \boldsymbol{f}(t)\psi\left(\frac{t-b}{a}\right)dt \tag{6}$$

where a is the scaling parameter, b is the translation parameters, and $\psi(t)$ is the mother wavelet [1]. Each base function $\psi(\frac{t-b}{a})$ is a scaled and translated version of the mother wavelet function ψ. These base functions meet the condition:

$$\int \psi\left(\frac{t-b}{a}\right)dt = 0 \tag{7}$$

Continuous wavelet transform necessitates a large amount of computation, while discrete wavelet transform requires less computation and is easier to implement than continuous wavelet transform. Discrete wavelet transform involves choosing scales and positions, called dyadic scale and positions, based on powers of two. This is achieved by modifying the wavelet representation as

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k) \tag{8}$$

where j and k are integers that control the wavelet dilation and translation, respectively. Using the wavelet discretization, the time-scale space can be sampled at discrete levels.

The wavelet transform has the appealing property of compression, i.e. a large proportion of the coefficients of the transform can be set to zero without appreciable loss of information [52], even for signals that contain occasional abruptness. The number of coefficients of these filters and the value of each coefficient depend on the mother wavelet function [17]. There is a vast number of choices for the mother wavelet [31]. Our primary research shows that longer wavelets with higher regularity tend to give slightly better results in terms of noise reduction. However, if the wavelet filter is too long, the details of the time series may be over smoothed. At the same time, the computational complexity will be nearly proportional to the length of the wavelets. Therefore, in this paper, we use the Daubechies four-coefficient wavelet as the mother wavelet, which achieves a balance between noise reduction and computational efficiency [18].

The essence of the denoising algorithm using wavelet analysis is the reduction of the noise in the wavelet-transform domain. Methods based on thresholding are widely used and have the ability to shrink the wavelet coefficients for reducing the noise observed in signals. Assuming that we have a length-N noisy observation $\boldsymbol{S} = \{\boldsymbol{s}(1), \ldots, \boldsymbol{s}(t), \ldots, \boldsymbol{s}(N)\}$,

$$\boldsymbol{s}(t) = \boldsymbol{f}(t) + \boldsymbol{\varepsilon}(t),\ t = 1, 2, \ldots, N \tag{9}$$

where $\{\varepsilon(t)\}_{1 \le t \le N}$ is a centred Gaussian white noise with unknown variance σ^2 and $\{\boldsymbol{f}(t)\}_{1 \le t \le N}$ is an unknown signal to be recovered from the observations. For a given orthogonal wavelet basis denoted by $((\phi_{K,\ k}(t))_{k \in \mathbf{Z}}, (\psi_{j,\ k}(t))_{1 \le j \le K, k \in \mathbf{Z}})$, where K is the decomposition level, $\phi_{K,\ k}(t)$ is the associated scaling function, and $\psi_{j,\ k}(t)$ is generated from $\psi(t)$ by dilations and translations as (8), the time series $\boldsymbol{s}(t)$ can be decomposed as

$$\boldsymbol{s}(t) = \sum_k \boldsymbol{cA}_{K,\ k}\phi_{K,\ k}(t) + \sum_{j=1}^{K}\sum_k \boldsymbol{cD}_{j,\ k}\psi_{j,\ k}(t) \tag{10}$$

where the coefficients $\boldsymbol{cA}_{K,k}$ are called the approximating coefficients and the coefficients $\boldsymbol{cD}_{j,\ k}$ are called the detail coefficients.

There are two types of thresholding methods for denoising the series, namely, soft thresholding and hard thresholding [43]. The wavelet coefficient will be set to zero if its amplitude is smaller than a predefined threshold; otherwise, it will be kept unchanged (hard thresholding) or its absolute value is decreased with respect to the threshold (soft thresholding). Because the hard-thresholding method discontinues at the threshold, additional oscillation may be imposed on the original series. The continuous soft-thresholding method can, however, generate a smooth series. Therefore, we use the soft-thresholding method to reduce the noise and outliers. Denoting c_j as the original wavelet coefficients ($\boldsymbol{cA}_{K,k}$ and $\boldsymbol{cD}_{j,\ k}$) and T_j as the threshold for the jth constitutive series, the denoised coefficients c'_j ($\boldsymbol{cA'}_{K,k}$ and $\boldsymbol{cD'}_{j,\ k}$) can be calculated as

$$c'_j = \begin{cases} c_j - T_j, & c_j \ge T_j \\ 0, & |c_j| < T_j \\ c_j + T_j, & c_j \le -T_j \end{cases} \tag{11}$$

If the absolute value of c_j is less than T_j, c_j corresponds to noise. The key to this method is to select an appropriate threshold T_j. Since the noise variances are not known in advance, T_j will be estimated from the data. If the selected value is too small, the recovered time series will remain noisy; if the value is too large, important time series details may be smoothed out. There are a number of approaches to choose the threshold. We find the universal threshold rule to be very simple and efficient [47], which is given by

$$T_j = \sqrt{2\log(N)}\sigma_{n_j} \tag{12}$$

where N is the sample size and σ_{n_j} is the noise standard deviation of the jth constitutive series. Subsequently, a denoised version of the original signal can be reconstructed from the approximation coefficient $\boldsymbol{cA'}_{K,k}$ and detail coefficients $\boldsymbol{cD'}_{j,\ k}$, using the inverse wavelet transform.

3. Proposed methodology

Inspired by the idea of 'decomposition and ensemble', which works by recursively breaking down a problem into two or more related sub-problems until these become simple enough to be solved directly, we decompose the time series into several sub-series in different scales or space/time positions by wavelet decomposition. An ESN is applied to each sub-series including approximation component and detail components. The first part in (10) represents the approximation of $\boldsymbol{s}(t)$, while the second part represents the details of $\boldsymbol{s}(t)$. The predictions of the sub-series are then combined to give an overall prediction of the original series.

By wavelet decomposition, the coherent signals in the wavelet domain have been compressed into just a few large magnitude coefficients, whereas incoherent noise is represented by lots of small magnitude coefficients. We handle the wavelet coefficients $\boldsymbol{cA}_{K,k}$ and $\boldsymbol{cD}_{j,k}$ by the above described soft-thresholding method, yielding $\boldsymbol{cA'}_{K,k}$ and $\boldsymbol{cD'}_{j,k}$. Then, the discrete time series $\boldsymbol{s}(t),\ t = 1, \ldots, N$ can be reconstructed with the following form:

$$\boldsymbol{s}(t) = \boldsymbol{A}_K(t) + \sum_{j=1}^{K} \boldsymbol{D}_j(t),\ t = 1, \ldots, N \tag{13}$$

where

$$\boldsymbol{A}_K(t) = \sum_k \boldsymbol{cA'}_{K,\,k} \phi_{K,\,k}(t) \tag{14}$$

$$\boldsymbol{D}_j(t) = \sum_k \boldsymbol{cD'}_{j,\,k} \psi_{j,\,k}(t) \tag{15}$$

The first item $\boldsymbol{A}_K(t), t = 1, \ldots, N$ presents the trend of the series, and is characterized by slow dynamics. The second item $\boldsymbol{D}_j(t), t = 1, \ldots, N, j = 1, \ldots, K$ presents the local details of the series, and is characterized by fast dynamics. Typically, the scale K is calculated as

$$K = \log_2 N - 5 \tag{16}$$

Let us express the signal $\boldsymbol{s}(t)$ by (13) to $K{=}k$ scale and $K{=}k-1$ scale separately, that is, $\boldsymbol{s}(t) = \boldsymbol{A}_k(t) + \sum_{j=1}^{k} \boldsymbol{D}_j(t) = \boldsymbol{A}_k(t) + \boldsymbol{D}_k(t) + \sum_{j=1}^{k-1} \boldsymbol{D}_j(t)$ for $K{=}k$, and $\boldsymbol{s}(t) = \boldsymbol{A}_{k-1}(t) + \sum_{j=1}^{k-1} \boldsymbol{D}_j(t)$ for $K{=}k-1$. Comparing the two expressions, we get

$$\boldsymbol{A}_{k-1}(t) = \boldsymbol{A}_k(t) + \boldsymbol{D}_k(t) \tag{17}$$

To obtain the predicted value $\hat{\boldsymbol{y}}(t)\boldsymbol{=}\ \hat{\boldsymbol{s}}(t+p)$, where p is the prediction horizon, we first predict $\hat{\boldsymbol{A}}_K(t+p)$ and $\hat{\boldsymbol{D}}_j(t+p)$, $j = 1, 2, \cdots, K$. To do this, let us write

$$\hat{\boldsymbol{A}}_K(t+p) = g_0[\boldsymbol{A}_K(t), \boldsymbol{A}_K(t-\tau_0), \ldots, \boldsymbol{A}_K(t-(m_0-1)\tau_0)] \tag{18}$$

Similarly,

$$\hat{\boldsymbol{D}}_j(t+p) = g_j\big[\boldsymbol{D}_j(t), \boldsymbol{D}_j(t-\tau_j), \ldots, \boldsymbol{D}_j(t-(m_j-1)\tau_j)\big],\ j = 1, 2, \ldots, K \tag{19}$$

where $g_0, g_1, \ldots, g_K$ denote nonlinear predictors related to the dynamic behavior of the constitutive series. In this paper, they are chosen as ESNs with different parameters.

In ESNs, the size of the input time window is important. If the time window is too large, some of the input variables may have little or no relevance to the output variables. For a finite dataset, some random correlations may exist between the irrelevant inputs and the output, making it hard for an ESN to set the coefficients of irrelevant inputs to zero; the irrelevant inputs will affect the model's performance adversely. In order to solve this problem, autocorrelation functions or mutual information are used to estimate the embedding parameters. However, autocorrelation functions are weak in treating the nonlinearity and may yield incorrect values, while the process of calculating mutual information is complicated.

Fortunately, the C–C method has been proposed to calculate the time window [15]. It is easier to implement and has lower computational cost compared with the mutual-information method. We thus apply the C–C method [15] to choose a short-term history for the high-frequency components and a long-term history for the low-frequency components. Through this, we can effectively exploit both the 'detailed' and 'trend' information of the time series. The reservoir parameters are modified further to produce optimal performance. The size of the reservoir determines the memory capacity of the network, the largest singular value of $\boldsymbol{W}_x$ determines the dynamic memory properties and ensures the stability of the network, and the sparseness of $\boldsymbol{W}_x$ controls the diversity of the reservoir states.

The overall forecast is then constructed from a linear combination of the predictions in each resolution. Therefore,

$$\hat{\boldsymbol{y}}(t) = \beta_0 \hat{\boldsymbol{A}}_K(t+p) + \sum_{j=1}^{K} \beta_j \hat{\boldsymbol{D}}_j(t+p) \tag{20}$$

where $\beta_j,\ j = 0, 1, \cdots, K$ are the weighted coefficients of each constitutive series.

Based on the above analysis, the structure of the proposed wavelet-denoising multiple ESN model (WDMESN) is illustrated in Fig. 2. It works as follows.

In the first stage, a time series is decomposed by wavelet transform as (10). The trend coefficients $\boldsymbol{cA}_K$ contain the slowest dynamics and are practically noise-free. The detail coefficients $\boldsymbol{cD}_j,\ j = 1, 2, \ldots, K$ contain the dynamics at a certain intermediate scale. The smaller the value of j is, the faster the dynamics will be. Because the high-frequency series may be corrupted by noise [42], we use the soft-thresholding method to denoise the wavelet coefficients. Then, the 'clean' wavelet coefficients $\boldsymbol{cA'}_K$ and $\boldsymbol{cD'}_j$, $j = 1, 2, \ldots, K$ are used to reconstruct the constitutive series.

In the second stage, to better utilize the detail information at high frequencies and the trend information at low frequencies, we apply the C–C method to choose short time windows $r_j = m_j \times \tau_j$ for the inputs to the ESNs at high frequencies,

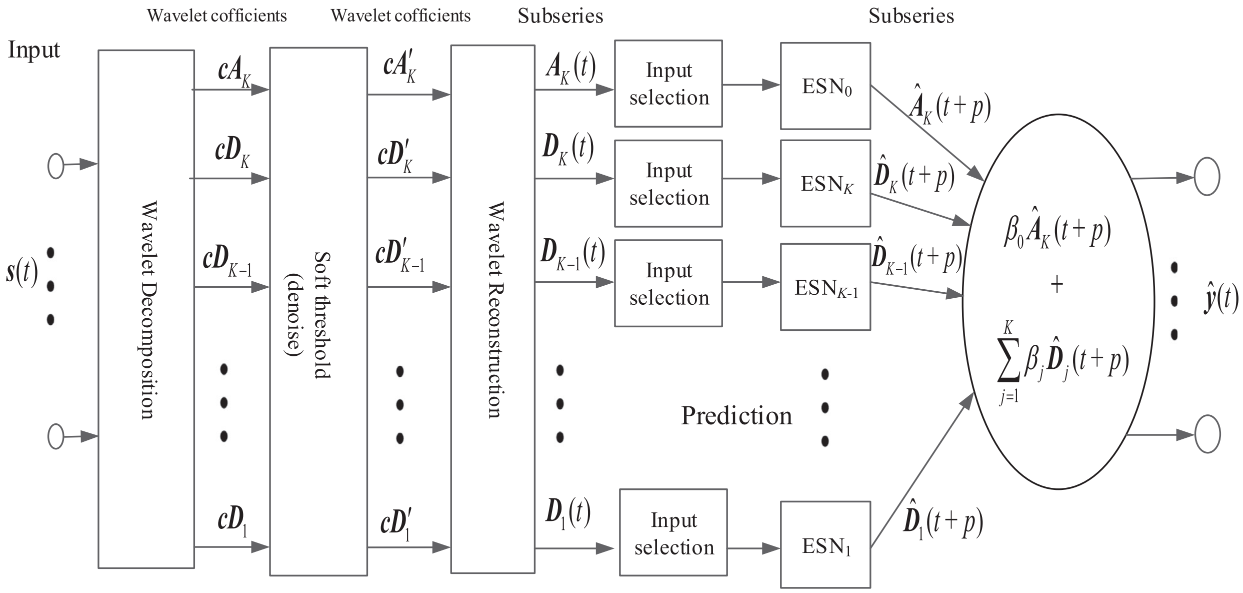


Fig. 2. Model structure.

and long time windows r_j for the inputs at low frequencies. A smaller r_j corresponds to faster dynamics, whereas a larger r_j corresponds to slower dynamics.

In the third stage, each constitutive series is modelled by an independent ESN. The values of the reservoir parameters should be set properly to match the target constitutive series dynamics. Under the constraints of reservoir parameters, as stated in Section 2, $\boldsymbol{W}_{in}$ and $\boldsymbol{W}_x$ in each ESN are assigned randomly and $\boldsymbol{W}_{out}$ is computed based on the least-squares estimation method. A five-fold cross-validation method is used to choose the optimal reservoir parameters. Each series is divided into five equal-sized subsets. In each fold, one subset is chosen as the testing set and the other four subsets are united as the training set. Each subset will be used once as the testing set, and the average of the five folds is recorded. The final parameters are obtained with respect to the lowest average [3].

In the fourth stage, the predicted results of each constitutive series are combined linearly to predict the overall output of the original time series. The lower the frequency is, the smoother the curve will be. The higher the frequency is, the noisier the curve will be, which makes prediction more difficult. Generally, the smooth wavelet constitutive series at low frequencies plays major roles.

The pseudocode of WDMESN is shown in Algorithm 1.

4. Experimental results

In this section, we provide a thorough experimental evaluation of the WDMESN model, considering two chaotic multivariate time series. One is the chaotic benchmark dataset—the Lorenz series, and the other is collected from real world—the annual runoff in the Yellow River and the annual number of sunspots. In order to demonstrate the effectiveness of the proposed model, we also conducted experiments using other similar works in literature, such as the echo state Gaussian process (ESGP) [6], regularized ESN (RESN) [37], support vector echo state machine (SVESM) [41], Elman network [29], and SSA [44] + MESN model.

Four measures were used to quantitatively evaluate the prediction performance from different aspects: root mean square error (RMSE), symmetric mean absolute percentage error (SMAPE), Pearson's correlation coefficient (R), and Nash–Sutcliffe efficiency coefficient (E).

The RMSE measures the differences between the predicted values and target values, and is defined by

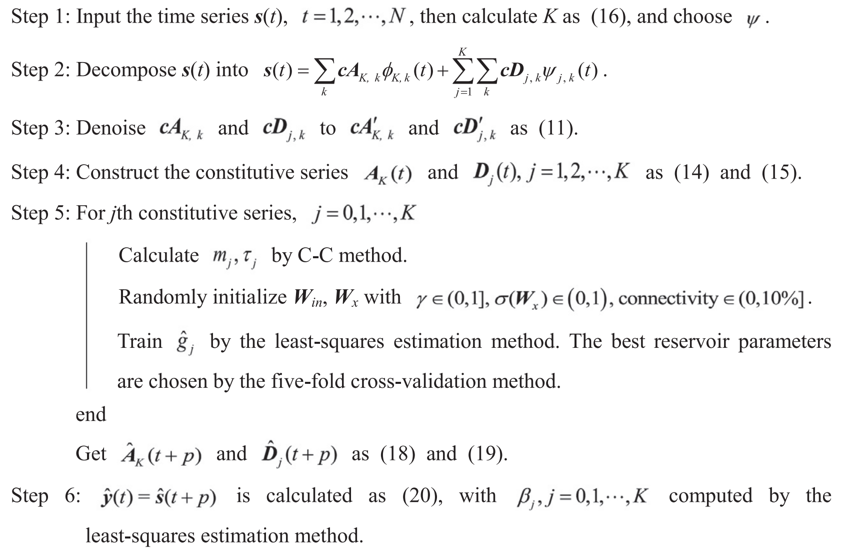
$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \hat{y}(t)\right)^2} \tag{21}$$

where $y(t)$ is the observed value at time step t, $\hat{y}(t)$ is the predicted value at time step t, and N denotes the length of the series considered for forecasting.

The SMAPE is an accuracy measure based on percentage errors, which is defined by

$$SMAPE = \frac{2}{N}\sum_{t=1}^{N}\left|\frac{y(t) - \hat{y}(t)}{y(t) + \hat{y}(t)}\right| \tag{22}$$

Algorithm 1 WDMESN algorithm.



```
Step 1: Input the time series $\boldsymbol{s}(t)$, $t=1,2,\cdots,N$, then calculate $K$ as (16), and choose $\psi$.
Step 2: Decompose $\boldsymbol{s}(t)$ into $\boldsymbol{s}(t)=\sum_k \boldsymbol{cA}_{K,k}\phi_{K,k}(t)+\sum_{j=1}^{K}\sum_k \boldsymbol{cD}_{j,k}\psi_{j,k}(t)$.
Step 3: Denoise $\boldsymbol{cA}_{K,k}$ and $\boldsymbol{cD}_{j,k}$ to $\boldsymbol{cA}'_{K,k}$ and $\boldsymbol{cD}'_{j,k}$ as (11).
Step 4: Construct the constitutive series $\boldsymbol{A}_K(t)$ and $\boldsymbol{D}_j(t)$, $j=1,2,\cdots,K$ as (14) and (15).
Step 5: For jth constitutive series, $j=0,1,\cdots,K$
        Calculate $m_j, \tau_j$ by C-C method.
        Randomly initialize $\boldsymbol{W}_{in}$, $\boldsymbol{W}_x$ with $\gamma\in(0,1]$, $\sigma(\boldsymbol{W}_x)\in(0,1)$, connectivity $\in(0,10\%]$.
        Train $\hat{g}_j$ by the least-squares estimation method. The best reservoir parameters
        are chosen by the five-fold cross-validation method.
    end
    Get $\hat{\boldsymbol{A}}_K(t+p)$ and $\hat{\boldsymbol{D}}_j(t+p)$ as (18) and (19).
Step 6: $\hat{\boldsymbol{y}}(t)=\hat{\boldsymbol{s}}(t+p)$ is calculated as (20), with $\beta_j, j=0,1,\cdots,K$ computed by the
    least-squares estimation method.
```

The Pearson's correlation coefficient (R) is a measure of the strength and direction of the linear relationship between the target values and predicted values, and is defined by

$$R=\frac{\sum_{t=1}^{N}(y(t)-\bar{y}(t))\big(\hat{y}(t)-\bar{\hat{y}}(t)\big)}{\sqrt{\sum_{t=1}^{N}(y(t)-\bar{y}(t))^2}\sqrt{\sum_{t=1}^{N}\big(\hat{y}(t)-\bar{\hat{y}}(t)\big)^2}} \tag{23}$$

where $\bar{y}(t)$ is the mean of $y(t)$, and $\bar{\hat{y}}(t)$ is the mean of $\hat{y}(t)$.

Finally, the Nash–Sutcliffe model efficiency coefficient (E) is another measure of the predictive power of models and is defined by

$$E=1-\frac{\sum_{t=1}^{N}\big(y(t)-\hat{y}(t)\big)^2}{\sum_{t=1}^{N}(y(t)-\bar{y}(t))^2} \tag{24}$$

RMSE and SMAPE with values of 0 stand for perfect prediction, while R and E with values of 1 stand for perfect prediction. In order to sustain the robustness of random assignments, we conducted 20 runs with different randomly-initialized reservoirs. The mean values and standard deviations of the 20 runs are given in the following experiments. In addition, we conducted Welch's t-test to determine whether the mean performances of two models were significantly different from each other. The T statistic can be calculated as follows:

$$T=\frac{\bar{\varepsilon}_1-\bar{\varepsilon}_2}{\sqrt{\left(s_{\varepsilon_1}^2+s_{\varepsilon_2}^2\right)/20}} \tag{25}$$

where $\bar{\varepsilon}_i, i=1,2$ denotes the prediction measure, e.g. RMSE, SMAPE, R or E, $s_{\varepsilon_i}, i=1,2$ denotes the standard deviation of the 20 runs for each measure. The test statistic has an approximate t distribution with v degrees of freedom given by Welch–Satterthwaite equation [10]:

$$v=\frac{\left(s_{\varepsilon_1}^2+s_{\varepsilon_2}^2\right)^2}{\left(s_{\varepsilon_1}^4+s_{\varepsilon_2}^4\right)/(20-1)} \tag{26}$$

In the following experiments, this test was performed with 0.05 significance level in two-tailed tests under the null hypothesis.

Besides Welch's t-test, another statistical test was applied to compare the performance of the two models based on the point-to-point forecast errors. The forecast error of the ith model at time step t is defined as

$$e_i(t)=\hat{y}_i(t)-y(t),\ i=1,2 \tag{27}$$

where $\hat{y}_i(t)$ denotes the predicted value of the ith model at time step t, and $y(t)$ denotes the observed value at time step t.

We assume that the loss function of model i is the square of $e_i(t)$; thus, the loss differential $d(t)$ between the two models is defined by

$$d(t)=e_1^2(t)-e_2^2(t) \tag{28}$$

The two models will have equal accuracy if and only if the loss differential has zero expectation for all t, i.e. the null hypothesis is

$$H_0 : E(d(t)) = 0, \ \ \forall t \tag{29}$$

and the alternative hypothesis is

$$H_1 : E(d(t)) \neq 0, \ \exists t \tag{30}$$

Under H_0, we get the following formula derived from (29)

$$\frac{\bar{d}}{\sqrt{2\pi f_d(0)/N}} \to N(0, 1) \tag{31}$$

where $f_d(0)$ measures the sample autocovariance of $d(t)$, and $\bar{d}$ is the mean value of $d(t)$, $t = 1, 2, \cdots, N$

$$\bar{d} = \sum_{t=1}^{N} d(t) \tag{32}$$

The Diebold–Mariano statistic [25] is denoted as

$$DM = \frac{\bar{d}}{\sqrt{2\pi \hat{f}_d(0)/N}} \tag{33}$$

where $\hat{f}_d(0)$ is a consistent estimate of $f_d(0)$ [11], defined by

$$2\pi \hat{f}_d(0) = \frac{1}{N} \sum_{t=1}^{N} \left[\left(d(t) - \bar{d} \right)^2 \right] \tag{34}$$

The Diebold–Mariano test was performed with α significance level in two-tailed tests under the null hypothesis.

4.1. Lorenz series

The Lorenz system is one of the most classical benchmarks for time-series prediction, and is described as follows

$$\frac{dx}{dt} = a(-x + y)\ , \quad \frac{dy}{dt} = bx - y - xz, \quad \frac{dz}{dt} = xy - cz \tag{35}$$

where $a = 10$, $b = 28$, and $c = 8/3$. We used the Runge–Kutta method to generate the Lorenz time series from the initial condition (12, 2, 9). The initial transient was washed out by employing a network warm-up time of 1000 steps. A zero-mean Gaussian noise with signal-to-noise ratio of 1 dB was added to the Lorenz series for the experiment. We generated 2500 samples to perform the simulation. First, the series was normalized. The procedure involves finding the maximum and minimum elements and normalizing the series $[x(t), y(t), z(t)]$ to the range $[-1, 1]$. According to (16), the decomposition scale for Lorenz series with length of 2500 is 6. We therefore decomposed the noisy Lorenz series up to a scale of 6.

In order to illustrate the wavelet denoising visually, we plotted the constitutive series of the Lorenz-x series in Fig. 3. The decomposed coefficients for the noisy Lorenz-x series are shown in Fig. 4(a). As can be seen, there is much noise in the coefficients of the $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, and $\boldsymbol{D}_3$ levels. Because the y and z series have similar results, we have not plotted them. The decomposed coefficients after wavelet denoising are shown in Fig. 4(b), and the constitutive series of different levels are shown in Fig. 5. As can be seen, almost all values of the $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, and $\boldsymbol{D}_3$ constitutive series are zero. Fig. 6 presents the comparison of the noisy Lorenz-x series and the denoised Lorenz-x series, and Fig. 7 presents the comparison of the denoised Lorenz-x series and the original noise-free Lorenz-x series. It is noteworthy that the denoised Lorenz-x series fits the noise-free Lorenz-x series very well, which validates the effectiveness of the wavelet-denoising algorithm. As almost all values of the $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, and $\boldsymbol{D}_3$ levels are noise, they will not be used in the next step.

After decomposition, we applied four individual ESNs with different reservoir parameters to the levels $\boldsymbol{A}_6$, $\boldsymbol{D}_6$, $\boldsymbol{D}_{5,}$ and $\boldsymbol{D}_4$, as shown in Table 1. The first 70% of the samples in each set were used for training and the remaining 30% were used for testing. We show the Lorenz-x series one-step-ahead predictions for each of the four levels ($\boldsymbol{A}_6$, $\boldsymbol{D}_6$, $\boldsymbol{D}_5$, and $\boldsymbol{D}_4$), and the overall prediction on both the training set and testing set in Fig. 8. It is noteworthy that all constitutive series of different resolutions are predicted accurately. The higher the scale is, the smoother the curve will be. Furthermore, we plotted the histogram of the overall prediction errors produced by WDMESN for Lorenz-x series on the testing dataset in Fig. 9. The Anderson–Darling test was used to test whether the prediction errors came from a normal distribution [36]. The p-value for the hypothesis test is 0.09, which is greater than the significance level 0.05; thus, we can confirm that the prediction errors follow a normal distribution with mean value of zero and standard deviation of 0.032. The dynamics of the Lorenz-x series has been sufficiently captured.

Table 1 gives the resulting RMSEs for each trained ESN over the training and testing datasets. As can be seen, a lower error is obtained for $\boldsymbol{D}_4$ compared with $\boldsymbol{D}_{5;}$ and for $\boldsymbol{D}_{6,}$ compared with $\boldsymbol{D}_5$. Based on the analysis in (13) and (17), $\boldsymbol{D}_i$ is obtained by applying a high-pass filter to $\boldsymbol{A}_{i\text{-}1}$, so that it contains the upper half of the frequencies in $\boldsymbol{A}_{i\text{-}1}$ and the remaining

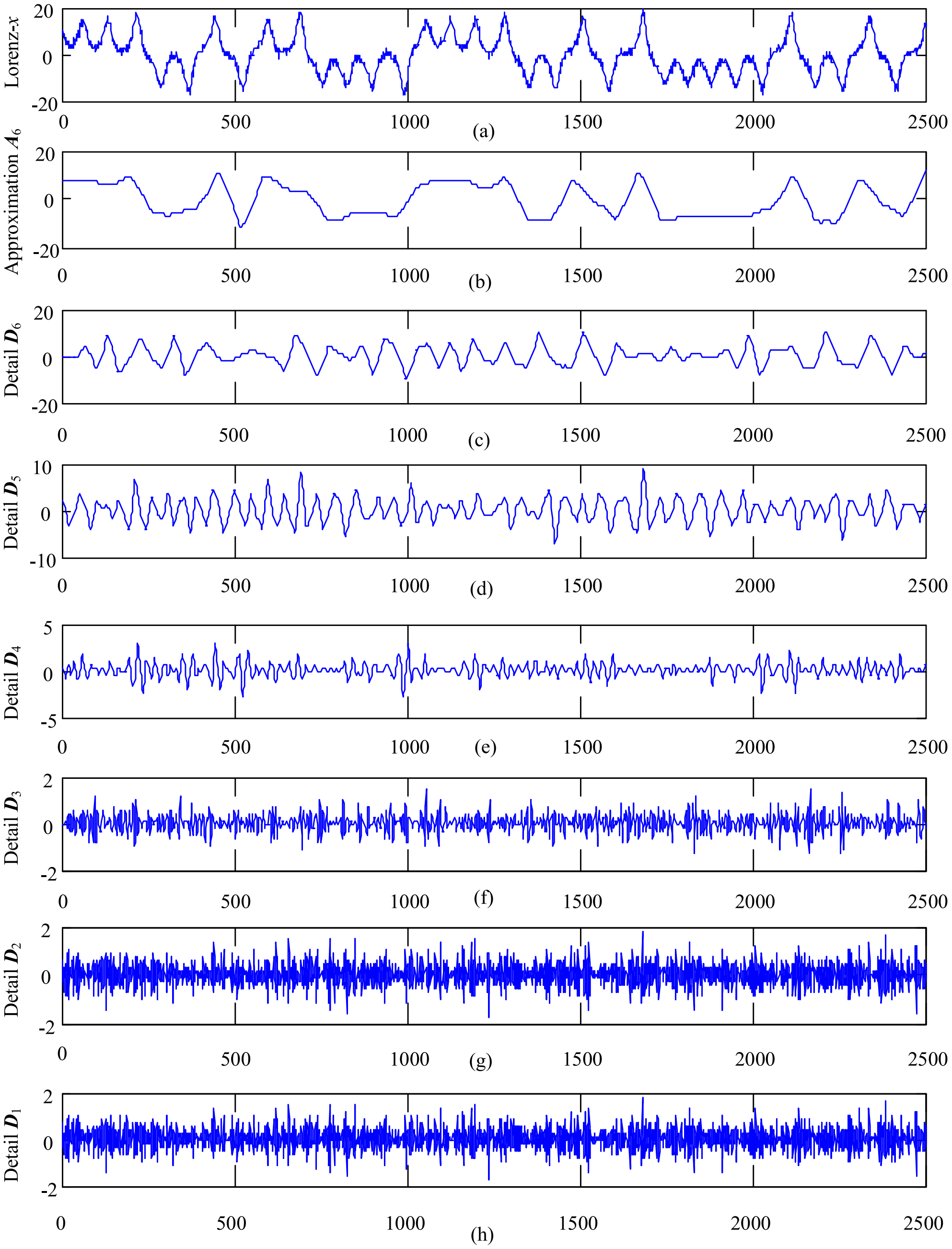


Fig. 3. Lorenz-*x* series with white Gaussian noise and its approximation and details up to a scale of 6.

Table 1
Best parameters of ESNs for different scales and their resulting RMSEs over the training dataset and testing dataset of Lorenz series.

Decomposed component	Embedding dimension	Delay time	Input scaling	Reservoir size	Reservoir connectivity	$\sigma(W_x)$	RMSE	
							Training	Test
$\boldsymbol{A}_6$	3	23	0.01	200	0.05	0.98	0.0066	0.0069
$\boldsymbol{D}_6$	2	26	0.01	200	0.05	0.9	0.0154	0.0173
$\boldsymbol{D}_5$	2	13	0.01	250	0.05	0.8	0.0249	0.0245
$\boldsymbol{D}_4$	2	8	0.01	250	0.05	0.7	0.0161	0.0177
Overall	–	–	–	–	–	–	0.0292	0.0318

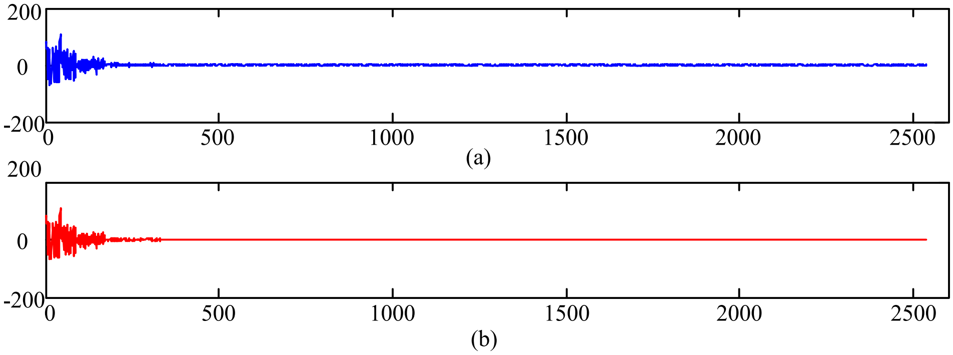


Fig. 4. Wavelet coefficients (a) before wavelet denoising (b) after wavelet denoising.

Table 2
Prediction performance of the Lorenz-*x* series of the evaluated methods.

Method	RMSE	SMAPE	*R*	*E*
WD + ESGP [6]	0.0552(0.0297)	0.0276(0.0236)	1.0000(0.0000)	0.9999(0.0001)
WD + RESN [37]	0.0746(0.0035)	0.0421(0.0320)	0.9999(0.0000)	0.9999(0.0000)
WD + SVESM [41]	0.1112(0.0066)	0.0369(0.0298)	0.9998(0.0000)	0.9997(0.0000)
WD + Elman [29]	0.2684(0.0872)	0.0977(0.0881)	0.9993(0.0005)	0.9985(0.0011)
SSA [44] + MESN	0.0564(0.0027)	0.0218(0.0141)	1.0000(0.0000)	0.9999(0.0000)
WDMESN	0.0318(0.0026)	0.0096(0.0051)	1.0000(0.0000)	1.0000(0.0000)

lower half of the frequencies appear in $\boldsymbol{A}_i$ as $\boldsymbol{A}_{i-1} = \boldsymbol{A}_i + \boldsymbol{D}_i$. In turn, $\boldsymbol{D}_i$ contains only the upper half of the frequencies in $\boldsymbol{A}_{i\text{-}1}$ that have no overlap with the frequencies in $\boldsymbol{D}_{i\text{-}1}$, and are cumulatively lower than the frequencies in $\boldsymbol{D}_{i-1}$. Therefore, different details ($\boldsymbol{D}_i$) extract different dynamics of separate and non-overlapping frequency bands from the original signal.

Since different values of parameters are expected for the best design of ESNs for predicting different constitutive series, the values of the parameters in trained ESNs for $\boldsymbol{A}_6$, $\boldsymbol{D}_6$, $\boldsymbol{D}_5$, and $\boldsymbol{D}_4$ series are not same, which is validated by Table 1. From the table, we can infer the following points, which give some guidelines for parameter setting.

First, we find that the input scaling parameter is 0.01 for all constitutive series. This implies that the current state of the Lorenz series depends much on the previous state.

Second, it is evident that the reservoir size of $\boldsymbol{D}_4$ and $\boldsymbol{D}_5$ is 250, whereas the reservoir size of $\boldsymbol{A}_6$ and $\boldsymbol{D}_6$ is 200, since the characteristics of $\boldsymbol{D}_4$ and $\boldsymbol{D}_5$ are more complex than those of $\boldsymbol{A}_6$ and $\boldsymbol{D}_6$. The reservoir size is the most important parameter that affects the prediction performance. A large size may lead to overfitting while a small size may lead to underfitting — it depends on the number of samples and the characteristics of the series.

Third, we note that the input time window (embedding dimension × delay time) of the low-frequency constitutive series is larger than that of the high-frequency constitutive series, as the dynamics of the low-frequency constitutive series varies slower than that of the high-frequency constitutive series. More historical information is needed for the low-frequency constitutive series. Then, we find that the largest singular values of $\boldsymbol{W}_x$ ($\sigma(\boldsymbol{W}_x)$) for $\boldsymbol{A}_6$, $\boldsymbol{D}_6$, $\boldsymbol{D}_5$, and $\boldsymbol{D}_4$ decrease in order, which means that the memory of the network gradually weakens.

Finally, in the experiment, we find that a reservoir connectivity of less than 10% is sufficient to ensure rich dynamics. The values 9% or 5% do not make much difference on prediction performance. Herein, we set the reservoir connectivity as 5%.

In Table 2, we present the prediction results of a few other methods applied on the testing dataset. The standard deviations of 20 runs are shown in brackets. In each run, the elements of $\boldsymbol{W}_{in}$ and $\boldsymbol{W}_x$ were randomly selected under the constraints of the reservoir parameters, which were chosen by cross-validation. The noisy Lorenz series was first pre-processed using the same wavelet-denoising algorithm used in WDMESN. Then, the different evaluation methods (ESGP, RESN, SVESM, Elman network) were applied to compare the prediction performance. In order to sustain the effectiveness of wavelet denoising, we also conducted an experiment combining SSA with MESN. The Welch's *t*-test results of the proposed model against other models for Lorenz-$x(t)$ series are given in Table 3, in which '+' means the proposed model significantly outperforms the other evaluated model, and '=' means there is no significant difference between two models.

As can be seen from Tables 2 and 3, WDMESN obviously outperforms WD + SVESM, WD + RESN, and WD + Elman; and slightly outperforms WD + ESGP and SSA + MESN. The Pearson's correlation coefficient of WDMESN, WD + ESGP and SSA + MESN are the same. The Elman network has the lowest prediction accuracy because there are too many unknown parameters to be trained, even when a small number of neurons are used in the hidden layer. This leads to underfitting. We further note that the standard deviation of WDMESN is small. Although the initialized values of the input-to-reservoir weights $\boldsymbol{W}_{in}$ and the reservoir-to-reservoir weights $\boldsymbol{W}_x$ are assigned randomly in the 20 runs, high prediction accuracy is always maintained.

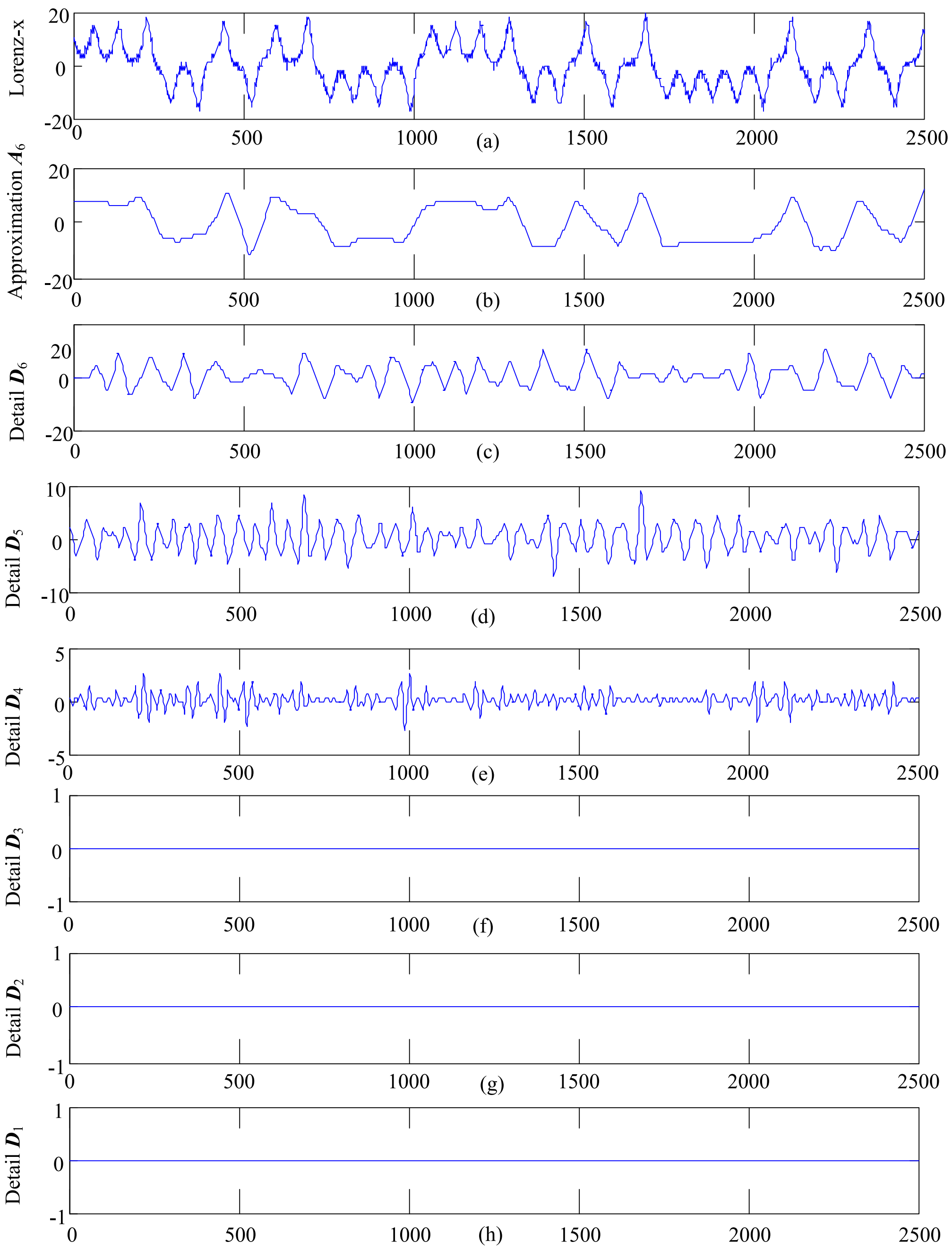


Fig. 5. Denoised Lorenz-*x* series with its approximation and details up to scale 6.

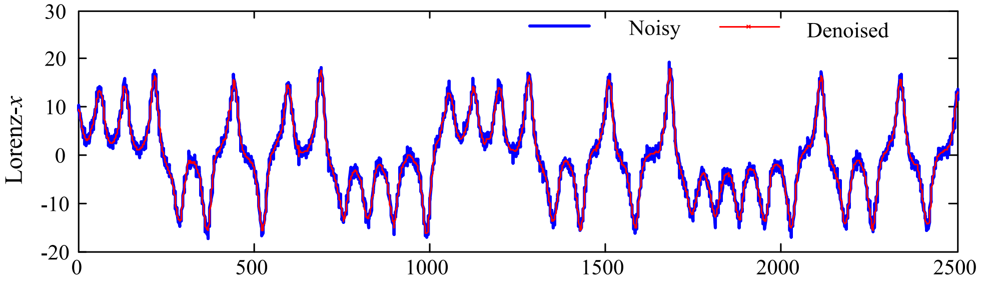


Fig. 6. Comparison of noisy Lorenz-*x* series and the denoised Loren-*x* series.

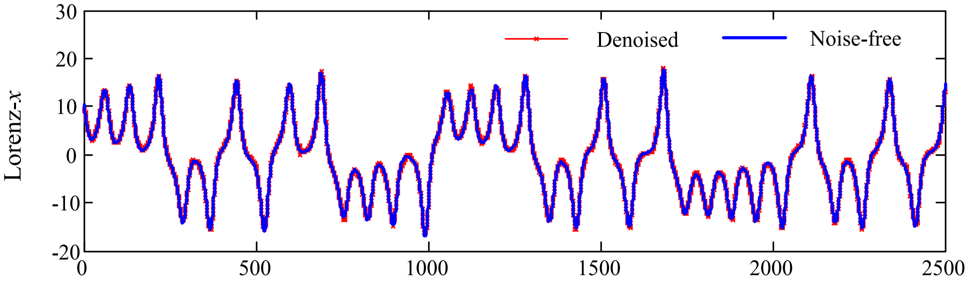


Fig. 7. Comparison of the denoised Lorenz-x series and the original noise-free Lorenz-x series.

Table 3
Results of Welch's t-test for Lorenz-x series.

Method	RMSE	SMAPE	R	E
WDMESN vs. WD + ESGP [6]	reject H_0 +	reject H_0 +	not reject H_0 =	reject H_0 +
WDMESN vs. WD + RESN [37]	reject H_0 +	reject H_0 +	reject H_0 +	reject H_0 +
WDMESN vs. WD + SVESM [41]	reject H_0 +	reject H_0 +	reject H_0 +	reject H_0 +
WDMESN vs. WD + Elman [29]	reject H_0 +	reject H_0 +	reject H_0 +	reject H_0 +
WDMESN vs. SSA [44] + MESN	reject H_0 +	reject H_0 +	not reject H_0 =	reject H_0 +

Table 4
Results of Diebold–Mariano test for Lorenz-x series prediction.

Method	DM	$\alpha = 0.05$	$\alpha = 0.10$
WDMESN vs. WD + ESGP [6]	3.9127	reject H_0 +	reject H_0 +
WDMESN vs. WD + RESN [37]	8.6105	reject H_0 +	reject H_0 +
WDMESN vs. WD + SVESM [41]	8.9561	reject H_0 +	reject H_0 +
WDMESN vs. WD + Elman [29]	16.3231	reject H_0 +	reject H_0 +
WDMESN vs. SSA [44] + MESN	4.8463	reject H_0 +	reject H_0 +

Table 5
Prediction performance of the Lorenz-y series of the evaluated methods.

Method	RMSE	SMAPE	R	E
WD + ESGP [6]	0.1426(0.0637)	0.0705(0.1194)	0.9998(0.0002)	0.9997(0.0004)
WD + RESN [37]	0.1053(0.0017)	0.0285(0.0021)	0.9999(0.0000)	0.9998(0.0000)
WD + SVESM [41]	0.1673(0.0108)	0.0853(0.0463)	0.9998(0.0000)	0.9996(0.0001)
WD + Elman [29]	0.4848(0.1316)	0.3730(0.4613)	0.9983(0.0009)	0.9964(0.0019)
SSA [44] + MESN	0.0629(0.0021)	0.0118(0.0008)	1.0000(0.0000)	0.9999(0.0000)
WDMESN	0.0597(0.0031)	0.0355(0.0541)	1.0000(0.0000)	1.0000(0.0000)

The Diebold–Mariano test was performed at the 0.05 and 0.10 significance levels in two-tailed tests under the null hypothesis of equal forecast accuracy for WDMESN and other methods. The test results shown in Table 4 indicate that the performance of the proposed method is superior to other methods for approximating the Lorenz characteristics.

In order to investigate the impact of reservoir parameters, taking γ for example, we conducted the experiment with γ increasing from 0.0001 to 1, while the other parameters were the same as in WDMESN. Fig. 10 shows the relationship between the RMSEs and γ in the logarithmic coordinates. As can be seen, WDMESN with small γ can achieve satisfactory prediction performance for the Lorenz time series. The smallest RMSE of 0.0318 ($\lg RMSE = -1.4976$) occurs when γ is $0.01(\lg \gamma = -2)$. This implies that the previous reservoir state contributes significantly to the current prediction. This is consistent with our previous conclusion that γ plays an important role in the reservoir's ability to characterize chaotic time series.

To sustain the effectiveness of the proposed model, we also provide the prediction results of the Lorenz-y series in Table 5, Welch's t-test results in Table 6, and Diebold–Mariano test results in Table 7. As can be seen, SSA+MESN achieved the lowest testing SMAPE of 0.0118 among all the six methods, however, WDMESN achieved the lowest testing RMSE of 0.0597. For RMSE, the differences between WDMESN and the other models are significant, but for SMAPE, the differences between WDMESN and some other models are not significant. Although the average SMAPE of WDMESN is bigger than that of SSA + MESN and WD + RESN, the overall performance of all the 20 runs are equal. Fig. 11 shows the prediction curves for each sub-series and the overall series on both the training dataset and testing dataset obtained by WDMESN. The predicted curves almost overlap with the observed curves. Fig. 12 gives the histogram of the overall prediction errors on the testing dataset. After the Anderson-Darling test, we find that the overall errors follow a normal distribution with zero mean value and 0.5667 standard deviation. Thus, the WDMESN model is suitable for Lorenz-y series prediction.

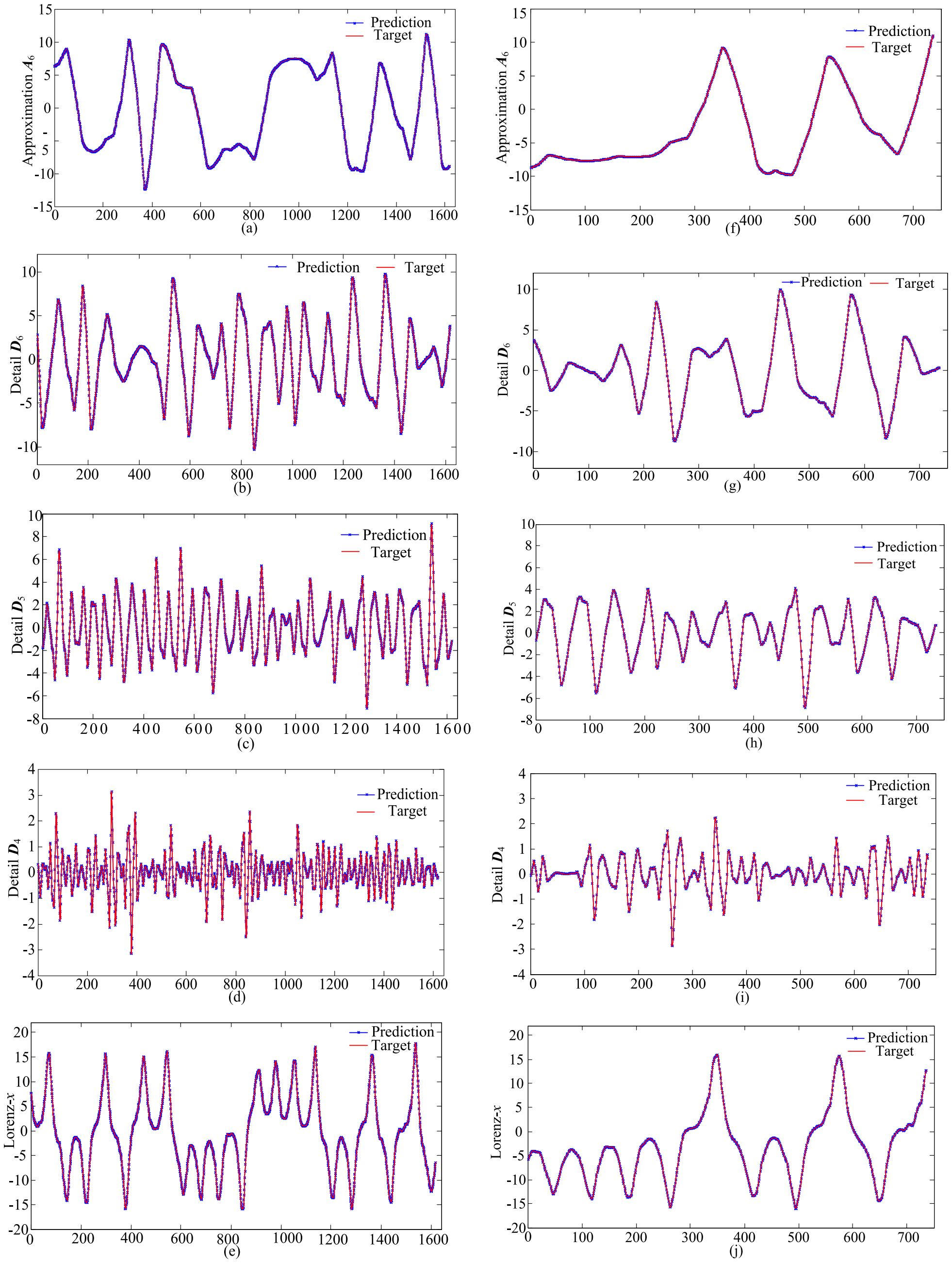


Fig. 8. From top to bottom: one-step-ahead predictions for $\boldsymbol{A}_6$, $\boldsymbol{D}_6$, $\boldsymbol{D}_5$, and $\boldsymbol{D}_4$ constitutive series and the overall Lorenz-*x* series on training dataset (a)–(e) and test dataset (f)–(j). In each figure, the solid line represents the target values and the solid line with stars represents the prediction values.

Table 6
Results of Welch's *t*-test for Lorenz-*y* series.

Method	RMSE	SMAPE	*R*	*E*
WDMESN vs. WD + ESGP [6]	reject H_0 +	not reject H_0 =	reject H_0 +	reject H_0 +
WDMESN vs. WD + RESN [37]	reject H_0 +	not reject H_0 =	reject H_0 +	reject H_0 +
WDMESN vs. WD + SVESM [41]	reject H_0 +	reject H_0 +	reject H_0 +	reject H_0 +
WDMESN vs. WD + Elman [29]	reject H_0 +	reject H_0 +	reject H_0 +	reject H_0 +
WDMESN vs. SSA [44] + MESN	reject H_0 +	not reject H_0 =	not reject H_0 =	reject H_0 +

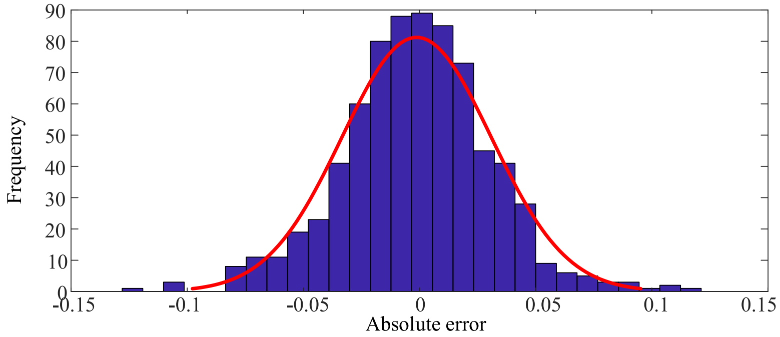


Fig. 9. Lorenz-x series: Histogram of the prediction errors of WDMESN.

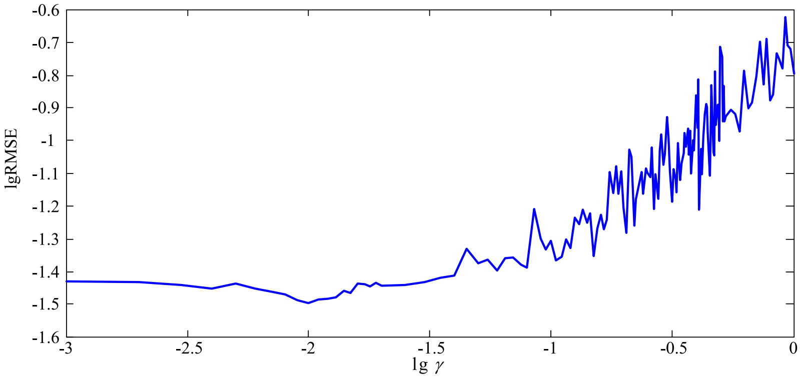


Fig. 10. RMSEs with γ increasing from 0.001 to 1, in the logarithmic coordinates.

Table 7
Results of Diebold–Mariano test for Lorenz-y series prediction.

Method	DM	$\alpha = 0.05$	$\alpha = 0.10$
WDMESN vs. WD + ESGP [6]	3.7299	reject H_0 +	reject H_0 +
WDMESN vs. WD + RESN [37]	4.6363	reject H_0 +	reject H_0 +
WDMESN vs. WD + SVESM [41]	12.4888	reject H_0 +	reject H_0 +
WDMESN vs. WD + Elman [29]	17.5850	reject H_0 +	reject H_0 +
WDMESN vs. SSA [44] + MESN	3.0216	reject H_0 +	reject H_0 +

4.2. Runoff in Yellow River series and sunspot series

In this experiment, we considered a real-world dataset: the annual runoff in the Yellow River and the number of sunspot series. The runoff series in the Yellow River observed at the Sanmenxia station and the number of sunspot series were considered over the years from 1749 to 2012, totalling 264 samples. The collected series were rescaled into the range [−1, 1] for use in training ESNs with the tanh activation function in the reservoirs. Because the observed runoff series contained much noise, a wavelet-decomposition method was first used to remove the noise. According to (16), the decomposition scale for the runoff-sunspot series with length of 264 is 3.

The runoff series with noise is depicted in Fig. 13(a). We decomposed the noisy runoff series into four resolution levels, which are shown in Fig. 13(b)–(e). The different levels represent the changing frequencies, amplitudes, and wavelengths. $\boldsymbol{A}_3$ has the maximum amplitude, lowest frequency, and longest wavelength. The following levels ($\boldsymbol{D}_3$, $\boldsymbol{D}_2$, and $\boldsymbol{D}_1$) show decreasing amplitudes and wavelengths and increasing frequencies. From the results, we can see that the $\boldsymbol{D}_1$ constitutive series are very noisy. Fig. 14 shows the decomposed constitutive series at different levels after wavelet denoising. Most of the noise has been eliminated.

Subsequently, we applied multiple ESNs with different parameters to the different levels, as shown in Table 8. The parameters of phase reconstruction, i.e. the delay times and embedding dimensions, were calculated using the C–C method. The first 70% of data were used for training and the remaining 30% were used for testing. As can be seen from Table 8, the input scaling parameter for the $\boldsymbol{A}_3$ series is 0.1, while that for the $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, and $\boldsymbol{D}_3$ series is 0.01. This means that, for the $\boldsymbol{A}_3$ series, both the previous reservoir state and current input have a significant impact on the current state, whereas for the $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, and $\boldsymbol{D}_3$ series, the previous reservoir state plays a key role in the current state.

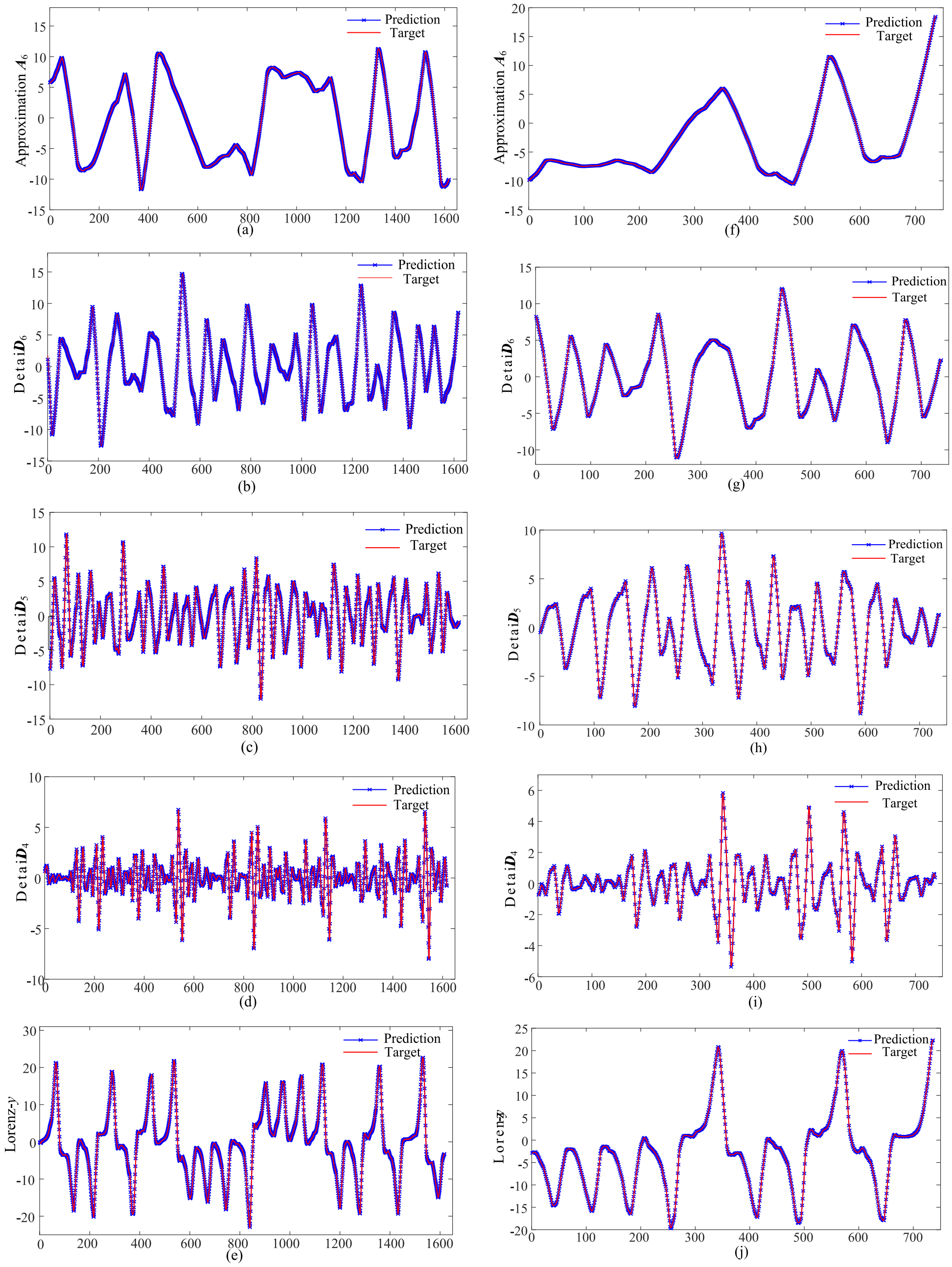


Fig. 11. From top to bottom: one-step-ahead predictions for $\boldsymbol{A}_6$, $\boldsymbol{D}_6$, $\boldsymbol{D}_5$, and $\boldsymbol{D}_4$ constitutive series and the overall Lorenz-y series on training dataset (a)–(e) and test dataset (f)–(j). In each figure, the solid line represents the target values and the solid line with stars represents the prediction values.

Table 8
Best parameters of ESNs for different scales and their resulting RMSEs over the training dataset and testing dataset of runoff series.

Decomposed component	Embedding dimension	Delay time	Input scaling	Reservoir size	Reservoir connectivity	$\sigma(W_x)$	RMSE	
							Training	Test
$\boldsymbol{A}_3$	2	12	0.1	40	0.05	0.98	3.0047	6.0491
$\boldsymbol{D}_3$	4	4	0.01	60	0.05	0.8	0.9977	4.4043
$\boldsymbol{D}_2$	3	6	0.01	50	0.05	0.6	4.9182	16.7994
$\boldsymbol{D}_1$	2	3	0.01	60	0.05	0.5	4.6145	7.4766
Overall	–	–	–	–	–	–	6.9143	24.4775

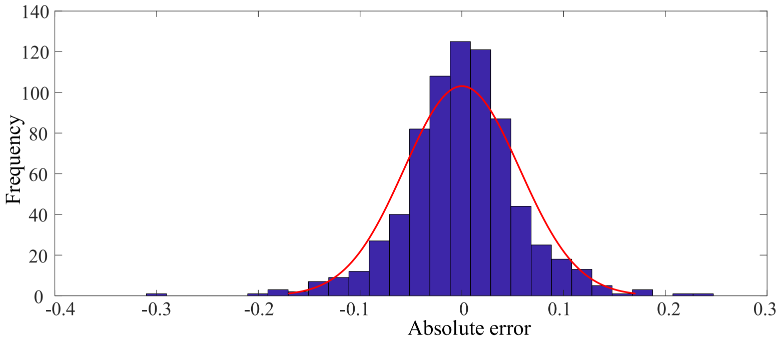


Fig. 12. Lorenz-*y* series: Histogram of the prediction errors of WDMESN.

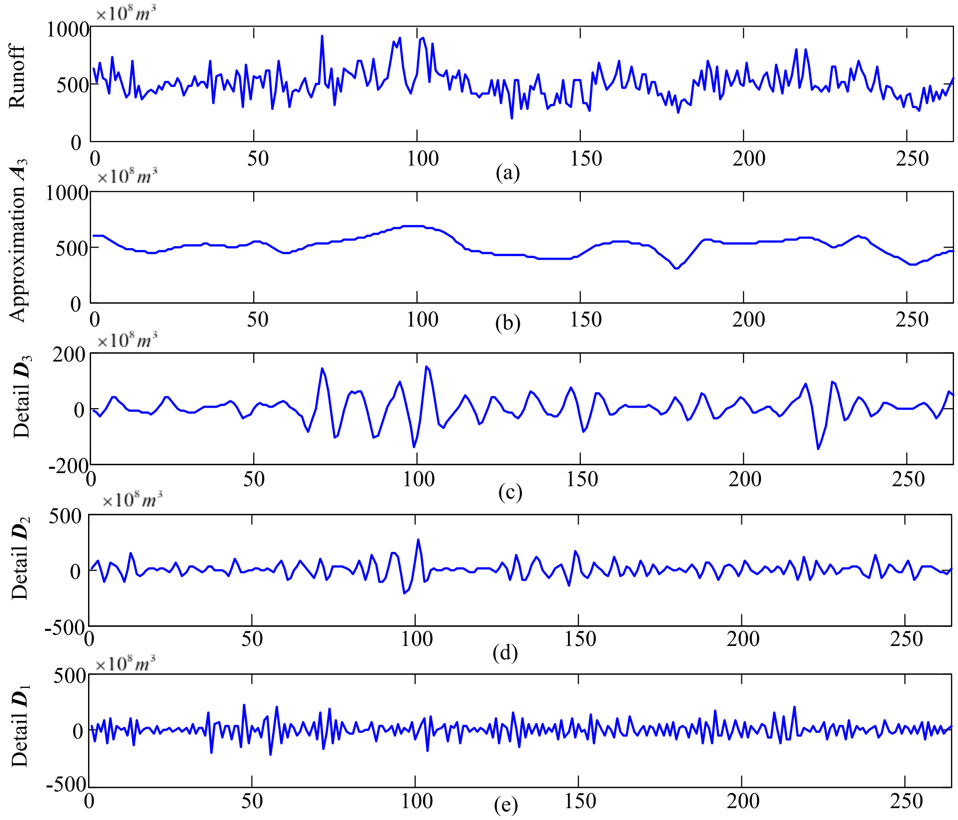


Fig. 13. Noisy annual runoff in Yellow River series and its approximation and details up to scale 3.

It is also noticeable that the reservoir size, which is a problem that depends on the size of the training set, is small for all constitutive series owing to the less than 200 training samples. We also note that both the largest singular value of $\boldsymbol{W}_x$ and the input time widow decrease as the frequency of the constitutive series increases. Higher-frequency constitutive series present faster dynamics, and thus the network needs less historical information.

We show the one-step-ahead prediction curves for each of the four constitutive series and the overall runoff series produced by WDMESN for both the training set and testing set in Fig. 15. The distribution of the overall prediction errors produced by WDMESN is presented in Fig. 16. After the Anderson–Darling test, we find that the errors follow a normal distribution with zero mean value and 16.5197 standard deviation, since the p value is 0.4629 larger than the significance level 0.05. This means that the WDMESN model is effective for runoff series prediction.

In Table 9, we present the experimental results of all the evaluated methods. The values in brackets are the standard deviations of 20 runs with randomly-initialized reservoir parameters. In SSA + MESN, the runoff-sunspots series are decomposed into five components by SSA, and each of the first four components is predicted by an ESN. Then, we obtain the final prediction by adding up the predicted values of the four components. Besides SSA + MESN, all other methods denoised the

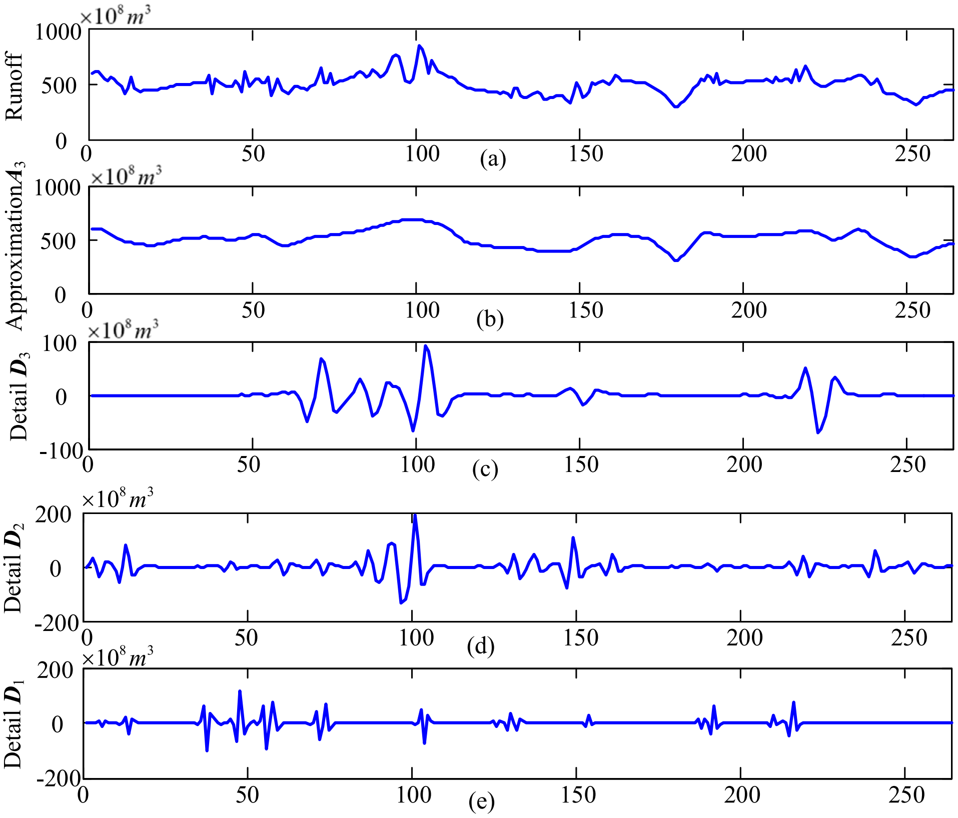


Fig. 14. Denoised annual runoff in Yellow River series and its approximation and details up to scale 3.

Table 9
Prediction performance for the runoff series of the evaluated methods.

Method	RMSE	SMAPE	R	E
WD + ESGP [6]	47.2627(2.2224)	0.0768(0.0051)	0.8801(0.0125)	0.7730(0.0234)
WD + RESN [37]	56.5105(2.7272)	0.0890(0.0049)	0.8267(0.0173)	0.6768(0.0333)
WD + SVESM [41]	63.5793(0.1284)	0.0955(0.0001)	0.6537(0.0013)	0.7590(0.0011)
WD + Elman [29]	62.0806(4.2364)	0.0961(0.0058)	0.7466(0.0257)	0.5514(0.0596)
SSA [44] + MESN	33.3800(2.9817)	0.0260(0.0023)	0.9492(0.0077)	0.8881(0.0172)
WDMESN	24.4775(2.7126)	0.0166(0.0019)	0.9634(0.0085)	0.9207(0.0181)

Table 10
Results of Welch's t-test for the runoff series.

Method	RMSE	SMAPE	R	E
WDMESN vs. WD + ESGP [6]	reject H_0 +	reject H_0 +	reject H_0 +	reject H_0 +
WDMESN vs. WD + RESN [37]	reject H_0 +	reject H_0 +	reject H_0 +	reject H_0 +
WDMESN vs. WD + SVESM [41]	reject H_0 +	reject H_0 +	reject H_0 +	reject H_0 +
WDMESN vs. WD + Elman [29]	reject H_0 +	reject H_0 +	reject H_0 +	reject H_0 +
WDMESN vs. SSA [44] + MESN	reject H_0 +	reject H_0 +	reject H_0 +	reject H_0 +

original series by using the wavelet-denoising algorithm. In Table 10, we give the Welch's t-test results of the WDMESN against other evaluated models for the runoff series, the differences in terms of all the prediction measures are significant. For real-world dataset, WDMESN are much more effective than other evaluated models.

From Table 9, we can get the similar conclusion. WDMESN outperforms WD + RESN with 56.69% and 81.35% reduction in RMSE and SMAPE, respectively, and 16.54% and 36.04% increase in R and E, respectively. This implies that the 'decomposition and ensemble' methods can capture the rich dynamics of complex chaotic time series as they can overcome the disadvantages of individual models by generating a synergetic effect in prediction. We also find that the average errors in WD + SVESM are larger than those in other models, but the standard deviations of WD + SVESM are very small, since WD + SVESM uses a static reservoir with only $\boldsymbol{W}_{in}$ being randomly initialized, which decreases the randomness of the model.

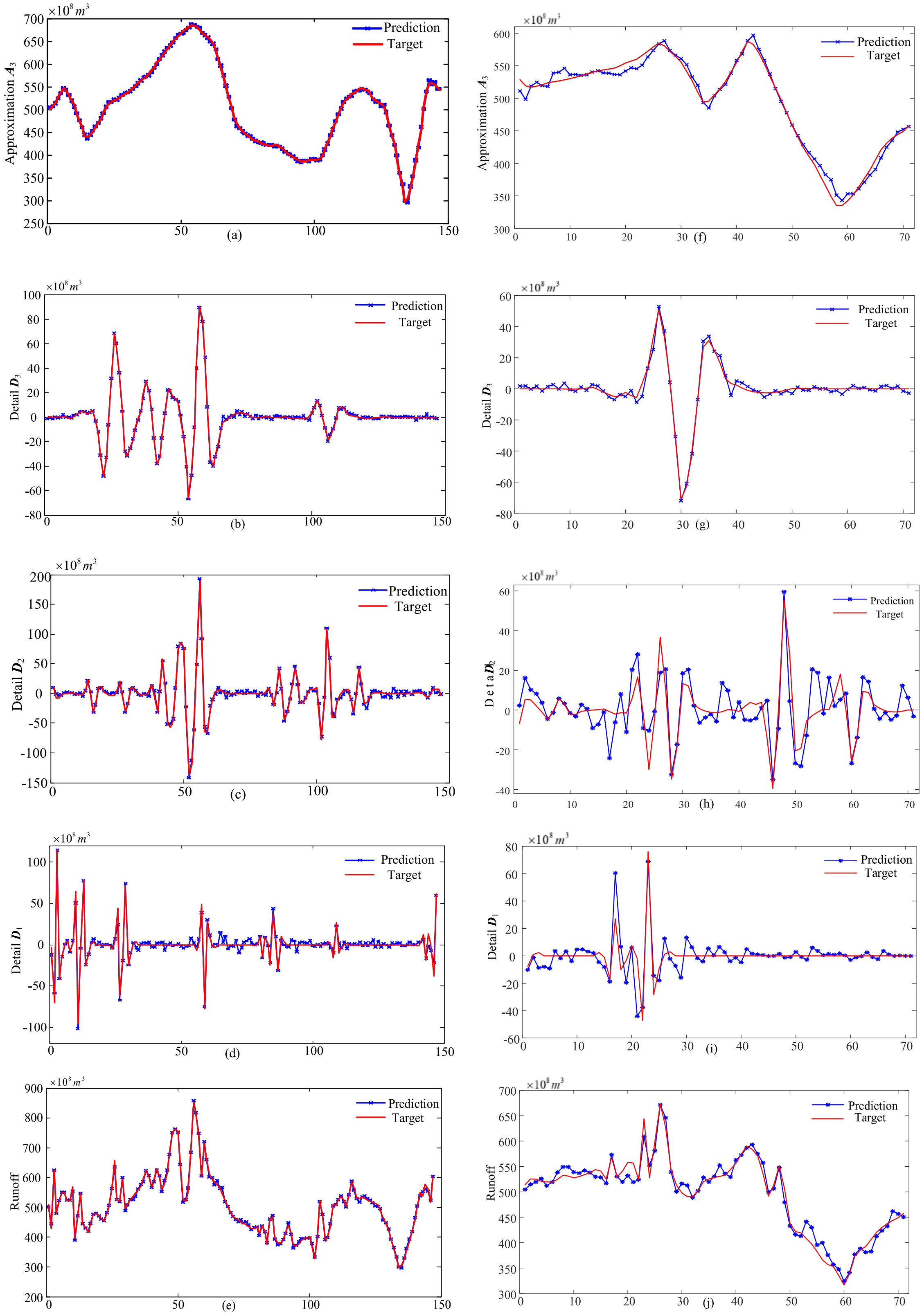


Fig. 15. From top to bottom: one-step-ahead predictions for $\boldsymbol{A}_3$, $\boldsymbol{D}_3$, $\boldsymbol{D}_2$, and $\boldsymbol{D}_1$ level series and the overall runoff series on training dataset (a)–(e) and test dataset (f)–(j). In each figure, the solid line presents the target values and the solid line with star presents the prediction values.

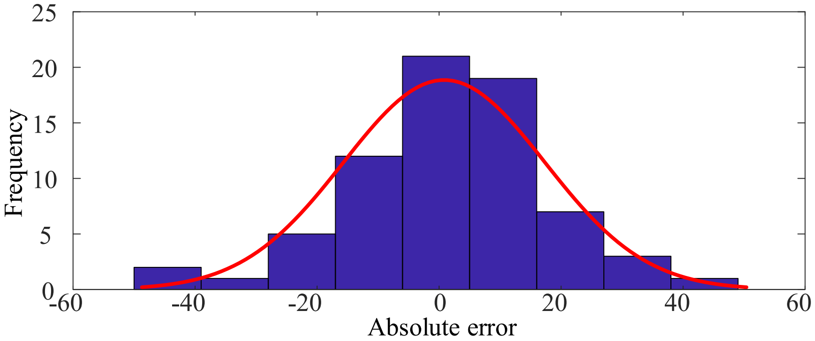


Fig. 16. Runoff series: Histogram of the prediction errors of WDMESN.

Table 11
Results of Diebold–Mariano test for runoff series prediction.

Method	DM	$\alpha = 0.05$	$\alpha = 0.10$
WDMESN vs. WD + ESGP [6]	5.3135	reject H_0 +	reject H_0 +
WDMESN vs. WD + RESN [37]	4.4492	reject H_0 +	reject H_0 +
WDMESN vs. WD + SVESM [41]	3.9904	reject H_0 +	reject H_0 +
WDMESN vs. WD + Elman [29]	4.2049	reject H_0 +	reject H_0 +
WDMESN vs. SSA [44] + MESN	1.8404	not reject H_0 =	reject H_0 +

Table 11 shows the Diebold–Mariano test results. When the significance level α is 0.10, the null hypothesis of no difference in accuracy between WDMESN and SSA + MESN is rejected, but when the significance level α decreases to 0.05, the null hypothesis is accepted. This implies that in terms of point-to-point errors, the WD + MESN is slightly superior to SSA + MESN but significantly superior to the other evaluated methods. In summary, the behavior of the proposed WDMESN method is appropriate for real-world noisy time-series prediction.

5. Conclusions

Multivariate time-series prediction is a difficult and complex problem involving the interaction of many variables. In this paper, we introduce a new prediction method that combines the shift-invariant wavelet transform with multiple ESNs. The proposed method has several advantages. First, the basic principle of wavelet denoising is simple, yet it can provide deep insight into the characteristics of the multivariate time series. Second, ESN is a simple, efficient, and effective algorithm for time series prediction. Third, the proposed model adopts the concept of 'decomposition and ensemble'. The decomposed components of time series are predicted individually by ESNs, and the overall prediction is reconstructed by adding up the outputs of multiple ESNs. We compared it with other methods by using simulations of two sets of multivariate chaotic time series. The simulation results demonstrate that WDMESN outperforms the other methods in terms of prediction accuracy.

Although the results of this paper are promising, additional research is necessary to further improve the generalization performance. The selection of reservoir parameters is based on cross-validation, which leads to unavoidably high computational complexity. A future work will be to use effective evolutionary algorithms to optimize the reservoir parameters. Besides, deep learning has attracted increasing attention in the field of machine learning, which attempts to extract important features from high-dimensional input by deep architecture in an unsupervised manner. It is also suggested as further studies to combine wavelet decomposition with deep neural networks in dealing with complex problems.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Nos. 61702077, 61773087) and Fundamental Research Funds for the Central Universities (No. DUT16RC(3)123).

References

[1] X. An, D. Jiang, C. Liu, M. Zhao, Wind farm power prediction based on wavelet decomposition and chaotic time series, Expert Syst. Appl. 38 (2011) 11280–11285.
[2] E.A. Antonelo, E. Camponogara, B. Foss, Echo state networks for data-driven downhole pressure estimation in gas-lift oil wells, Neural Netw. 85 (2017) 106–117.
[3] C. Bergmeir, J.M. Benítez, On the use of cross-validation for time series predictor evaluation, Inf. Sci. 191 (2012) 192–213.
[4] X. Cai, N. Zhang, G.K. Venayagamoorthy, D.C. Wunsch Ii, Time series prediction with recurrent neural networks trained by a hybrid PSO–EA algorithm, Neurocomputing 70 (2007) 2342–2353.
[5] J. Cao, Z. Lin, G.B. Huang, Composite function wavelet neural networks with extreme learning machine, Neurocomputing 73 (2010) 1405–1416.
[6] S.P. Chatzis, Y. Demiris, Echo state Gaussian process, IEEE Trans. Neural Netw. 22 (2011) 1435–1445.

[7] T.T. Chen, S.J. Lee, A weighted LS-SVM based learning system for time series forecasting, Inf. Sci. 299 (2015) 99–116.
[8] N. Chouikhi, B. Ammar, N. Rokbani, A.M. Alimi, PSO-based analysis of echo state network parameters for time series forecasting, Appl. Soft Comput. 55 (2017) 211–225.
[9] I. Daubechies, The wavelet transform, time-frequency localization and signal analysis, IEEE Trans. Inf. Theory 36 (1990) 961–1005.
[10] B. Derrick, D. Toher, P. White, Why Welch's test is Type I error robust, Quant. Methods Psychol. 12 (2016) 30–38.
[11] F.X. Diebold, Comparing predictive accuracy, twenty years later: a personal perspective on the use and abuse of Diebold–Mariano tests, J. Bus. Econ. Stat. 33 (2015) 1–9.
[12] S.W. Fei, Y. He, Wind speed prediction using the hybrid model of wavelet decomposition and artificial bee colony algorithm-based relevance vector machine, Int. J. Electr. Power Energy Syst. 73 (2015) 625–631.
[13] C. Gallicchio, A. Micheli, Architectural and Markovian factors of echo state networks, Neural Netw. 24 (2011) 440–456.
[14] S. Ganjefar, M. Tofighi, Single-hidden-layer fuzzy recurrent wavelet neural network: applications to function approximation and system identification, Inf. Sci. 294 (2015) 269–285.
[15] Z.K. Gao, N.D. Jin, Complex network from time series based on phase space reconstruction, Chaos Interdiscip. J. Nonlinear Sci. 19 (2009) 033137.
[16] A. Gholipour, C. Lucas, B.N. Araabi, M. Mirmomeni, M. Shafiee, Extracting the main patterns of natural time series for long-term neurofuzzy prediction, Neural Comput. Appl. 16 (2007) 383–393.
[17] A. Graps, An introduction to wavelets, IEEE Comput. Sci. Eng. 2 (1995) 50–61.
[18] H.T. Guo, J.E. Udegard, M. Lung, R.A. Gopinath, I.W. Selesnick, C.S. Burrui, Wavelet based speckle reduction with application to SAR based ATD/R, in: Proceddings of the IEEE International Conference on Image Processing, Austin, Texas, USA, 1994, pp. 75–79.
[19] M. Han, M. Xu, Laplacian echo state network for multivariate time series prediction, IEEE Trans. Neural Netw. Learn. Syst. 29 (2018) 238–244.
[20] M. Han, M. Xu, X. Liu, X. Wang, Online multivariate time series prediction using SCKF-γESN model, Neurocomputing 147 (2015) 315–323.
[21] S.I. Han, J.M. Lee, Fuzzy echo state neural networks and funnel dynamic surface control for prescribed performance of a nonlinear dynamic system, IEEE Trans. Ind. Electron. 61 (2014) 1099–1112.
[22] X. Han, X. Chang, An intelligent noise reduction method for chaotic signals based on genetic algorithms and lifting wavelet transforms, Inf. Sci. 218 (2013) 103–118.
[23] S.S. Haykin, Neural networks and learning machines, Pearson Education, third ed., United States, New Jersey, 2009.
[24] W. Huang, R. Wang, Y. Yuan, S. Gan, Y. Chen, Signal extraction using randomized-order multichannel singular spectrum analysis, Geophysics 82 (2017) 69–84.
[25] W.M. Hung, W.C. Hong, Application of SVR with improved ant colony optimization algorithms in exchange rate forecasting, Control Cybern. 38 (2009) 863–891.
[26] H. Jaeger, H. Haas, Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication, Science 304 (2004) 78–80.
[27] S. Lokse, F.M. Bianchi, R. Jenssen, Training echo state networks with regularization through dimensionality reduction, Cognit. Comput. 9 (2017) 364–378.
[28] M. Li, D. Wang, Insights into randomized algorithms for neural networks: Practical issues and common pitfalls, Inf. Sci. 382–383 (2017) 170–178.
[29] H. Liu, X.W. Mi, Y.F. Li, Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short term memory neural network and Elman neural network, Energy Convers. Manag. 156 (2018) 498–514.
[30] E. Maiorino, F.M. Bianchi, L. Livi, A. Rizzi, A. Sadeghian, Data-driven detrending of nonstationary fractal time series with echo state networks, Inf. Sci. 382–383 (2017) 359–373.
[31] W.K. Ngui, M.S. Leong, L.M. Hee, A. abdelrhman, Wavelet analysis: mother wavelet selection methods, Appl. Mech. Mater. 393 (2013) 953–958.
[32] M.C. Ozturk, D. Xu, J.C. Príncipe, Analysis and design of echo state networks, Neural Comput. 19 (2007) 111–138.
[33] S. Pravilovic, M. Bilancia, A. Appice, D. Malerba, Using multiple time series analysis for geosensor data forecasting, Inf. Sci. 380 (2017) 31–52.
[34] X. Qiu, Y. Ren, P.N. Suganthan, G.A.J. Amaratunga, Empirical mode decomposition based ensemble deep learning for load demand time series forecasting, Appl. Soft Comput. 54 (2017) 246–255.
[35] X. Qiu, L. Zhang, P.N. Suganthan, G.A.J. Amaratunga, Oblique random forest ensemble via least square estimation for time series forecasting, Inf. Sci. 420 (2017) 249–262.
[36] N.M. Razali, B.W. Yap, Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests, J. Stat. Model. Anal. 2 (2011) 21–33.
[37] R.F. Reinhart, J.J. Steil, Regularization and stability in reservoir networks with output feedback, Neurocomputing 90 (2012) 96–105.
[38] E.D.L Rosa, W. Yu, Randomized algorithms for nonlinear system identification with deep learning modification, Inf. Sci. 364–365 (2016) 197–212.
[39] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117.
[40] Y. Seo, S. Kim, O. Kisi, V.P. Singh, Daily water level forecasting using wavelet decomposition and artificial intelligence techniques, J. Hydrol. 520 (2015) 224–243.
[41] Z.W. Shi, M. Han, Support vector echo-state machine for chaotic time-series prediction, IEEE Trans. Neural Netw. 18 (2007) 359–372.
[42] S. Soltani, On the use of the wavelet decomposition for time series prediction, Neurocomputing 48 (2002) 267–277.
[43] M. Srivastava, C.L. Anderson, J.H. Freed, A new wavelet denoising method for selecting decomposition levels and noise thresholds, IEEE Access 4 (2016) 3862–3877.
[44] A.H. Vahabie, M.M.R. Yousefi, B.N. Araabi, C. Lucas, S. Barghinia, Combination of singular spectrum analysis and autoregressive model for short term load forecasting, in: Proceedings of the Power Tech Conference, Lausanne, Switzerland, 2007, pp. 1090–1093.
[45] D.H. Wang, Editorial: randomized algorithms for training neural networks, Inf. Sci. 364–365 (2016) 126–128.
[46] W.C. Wang, K.W. Chau, D.M. Xu, X.Y. Chen, Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition, Water Resour. Manag. 29 (2015) 2655–2675.
[47] H. Xie, L.E. Pierce, F.T. Ulaby, SAR speckle reduction using wavelet denoising and Markov random field modeling, IEEE Trans. Geosci. Remote Sens. 40 (2002) 2196–2212.
[49] L. Xu, J. Li, Y. Shu, J. Peng, SAR image denoising via clustering-based principal component analysis, IEEE Trans. Geosci. Remote Sens. 52 (2014) 6858–6869.
[49] M. Xu, M. Han, Adaptive elastic echo state network for multivariate time series prediction, IEEE Trans. Cybern. 46 (2016) 2173–2183.
[50] M.H. Yusoff, J. Chrol-Cannon, Y.C. Jin, Modeling neural plasticity in echo state networks for classification and regression, Inf. Sci. 364 (2016) 184–196.
[51] L. Zhang, P.N. Suganthan, A survey of randomized algorithms for training neural networks, Inf. Sci. 364–365 (2016) 146–155.
[52] Y.D. Zhang, Z.C. Dong, P. Phillips, S.H. Wang, G.L. Ji, J.Q. Yang, Exponential wavelet iterative shrinkage thresholding algorithm for compressed sensing magnetic resonance imaging, Inf. Sci. 322 (2015) 115–132.
[53] H.C. Zhou, Y. Peng, G.H. Liang, The research of monthly discharge predictor-corrector model based on wavelet decomposition, Water Resour. Manag. 22 (2008) 217–227.