Neural Computing and Applications (2021) 33:4997–5010
https://doi.org/10.1007/s00521-020-05286-8

ORIGINAL ARTICLE

Ensemble echo network with deep architecture for time-series modeling

Ruihan Hu[1,6] · Zhi-Ri Tang[2,6] · Xiaoying Song[3] · Jun Luo[4] · Edmond Q. Wu[5] · Sheng Chang[6]

Received: 17 October 2019 / Accepted: 7 August 2020 / Published online: 24 August 2020

Abstract

Echo state network belongs to a kind of recurrent neural networks that have been extensively employed to model time-series datasets. The function of reservoir in echo state network is expected to extract the feature context from time-series datasets. However, generalization of echo state networks is limited in real-world application because the architectures of the network are fixed and the hyper-parameters are hard to be automatically determined. In the present study, the ensemble Bayesian deep echo network (EBDEN) model with deep and flexible architecture is proposed. Such networks with deep architecture progressively extract more dynamic echo states through multiple reservoirs than those with the shallow reservoir. To enhance the flexibility of the configuration for the network, this study investigates the Bayesian optimization procedure of hyper-parameters and ensures the suitable hyper-parameters to activate the network. In addition, when dealing with more complex time-series datasets, ensemble mechanism of EBDEN can measure the redundancy for the channels of the time series without sacrificing the algorithm's performance. In this paper, the deep, optimization and ensemble architectures of EBDEN are verified by experiments benchmarked on multivariate time-series repositories and realistic tasks such as chaotic series representation and Dansgaard–Oeschger estimation tasks. According to the results, EBDEN achieves high level of the goodness-of-fit and classification performance in comparison with state-of-the-art models.

Keywords Echo state network · Deep reservoir · Bayesian optimization · Ensemble mechanism · Multivariate time series

1 Introduction

On the basis of the units in human brain, which are interconnected by synapses to generate decision making and coordination, the researchers developed artificial neural network models in machine learning to solve real-world tasks. At present, a lot of deep-learning methods [1–4] have been successfully applied into time-series tasks (TSKs). Convolution neural networks with their variants such as temporal convolutional neural network (TCNN) [5], time-series encoder (TSE) [6], multi-channel deep convolutional neural network (MCDCNN) [7] and time-CNN [8] were proposed to model the time domain with one-dimensional convolution templates. As a natural extension of convolutional networks, recurrent neural network (RNN) models, such as long short-term memory (LSTM) [9], have been designed for linking and memorizing the past and current information using recurrent connection. These classic ANN models such as LSTMs and CNNs [5–9] employed multiple layers computation framework to adapt the large-scale inputs, while the

✉ Sheng Chang
changsheng@whu.edu.cn

Ruihan Hu
rh.hu@giim.ac.cn

[1] Guangdong Key Laboratory of Modern Control Technology, Guangdong Institute of Intelligent Manufacturing, Guangzhou, China

[2] Department of Computer Science, City University of Hong Kong, Hong Kong, China

[3] The Engineering Research Center of Metallurgical Automation and Measurement Technology, Wuhan University of Science and Technology, Wuhan, China

[4] China Electronic Product Reliability and Environmental Testing Research Institute, Guangzhou, China

[5] Department of Automation, Shanghai Jiao Tong University, Shanghai, China

[6] School of Physics and Technology, Wuhan University, Wuhan, China

backpropagation mechanism between these layers consumed too much time for convergence [10, 11]. Echo state networks (ESNs) [12, 13] used the gradient-free method to model the time-series datasets. Echo states in ESNs achieved the feed-forward transition in the reservoirs and did not need to propagate the gradients through time step. The echo state networks [12, 13] can effectively suppress the local minima and gradient vanishing in backpropagation procedure [14], in which general ANNs cannot avoid the above problems.

Besides these advantages, the architecture of the reservoir for ESNs is kept fixed and only governed by hyper-parameters such as the number of echo units, leaky rate and spectral radius. Therefore, the learning performance of the ESNs critically depends on the configuration of hyper-parameters. However, the ESNs [12, 13] cannot effectively represent the TSKs in a lot of actual applications because the shallow reservoir architecture of network is hard to portray the complicated feature of TSKs. According to the previous work [15–19], the spatial scale of the stacked reservoirs organization of the ESNs has shown the powerful hierarchical temporal feature representation with respect to the shallow ESNs. Due to the improvement of the fitting capacity with multiple reservoirs stacked, the ESNs have been extensively applied into the complex tasks such as solar irradiance prediction [15–18] application.

When meeting with the multivariate time-series (MTS) datasets, the single ESN may achieve accurate performance on univariate channel and poorly on another. Inspired by the advantages of ensemble learning [20], the mechanisms can provide the trade-off diagram for the accuracy and diversity of the base learners. It can be considered that combining multiple ESNs to form an ensemble ESN can yield better results beyond the multivariate series than single ESN. The ensemble selection mechanisms for the general ensemble models [21, 22] tend to select the diverse base learners from a number of trained learners to enhance the learning performance. For example, each echo state network [21, 22] is applied to model the 3D motion and steady-state visual evoked potentials (SSVEPs) datasets to build the ensemble echo state network. In these works, the tuning of hyper-parameters for ensemble ESN is always solved by manual or grid search modulation based on trial-and-error method. Previous works [21, 22] demonstrate that such solution suffers from the problems of search space complexity, growing exponentially with the number of tuned hyper-parameters.

To introduce the multiple architecture and ensemble selection mechanisms to the echo state networks, in this paper, ensemble Bayesian deep echo network (EBDEN) is proposed to model the time-series datasets. Admittedly, it is the first attempt to fuse the Bayesian into the optimization of hyper-parameters tuning for deep echo state network. The above merits are attributed to the following two contributions of the proposed method:

- To extend the shallow layer to the deep architecture, the multiple-scale reservoirs with bidirectional connection are fused across the time steps to build the deep architecture of EBDEN. In order to ensure the performance of the multiple-scale reservoirs, the hyper-parameters in multiple-scale reservoirs can improve the learning performance which are explored by employing Bayesian optimization (BO) in EBDEN.
- To solve the ensemble selection for the echo state networks, due to the redundancy for modeling multiple time domain channels of multivariate time series (MTS), EBDEN can determine the optimal ensemble weights and avoid overfitting problem.

The reminder of this study is organized as follows: In Sect. 2, the methodology of EBDEN is described. The benchmarks and characteristics of EBDEN are discussed in Sect. 3. The experimental results by leveraging EBDEN to solve various TSKs, such as multivariate time series (MTS), chaotic time-series representation and Dansgaard–Oeschger estimation, are shown in Sect. 4. The conclusion and future work are presented in Sect. 5.

2 Architecture of EBDEN

Particularly, the boldface letter and the italic letter denote the matrix and the vector in this paper, respectively. In Fig. 1, the EBDEN containing three modules, namely input, reservoir and readout layers, is shown. In terms of general multivariate time-series (MTS) sequences $\mathbf{x}$, they incorporate M samples ($\mathbf{x} = (x_1, x_2, \ldots, x_M)$) and K channels. According to Fig. 1, the input sequences $\mathbf{x}$ are progressively fed into the multiple-scale bidirectional reservoirs to extract the echo state sequences. Compared with the shallow architecture of the reservoir, the deep one provides richer and more differentiated echo state sequences to discriminate the past and current information. Apart from the global hyper-parameters of EBDEN like the number of scales L, each reservoir is activated by local hyper-parameters, such as unit numbers in the reservoir N, the leaky rate λ, the spectral radius ρ, the scaling coefficient ω and the sparsity connection degree η. However, due to the fixed form of the hyper-parameters without any optimization, the performance of the multiple-scale echo state network fails to achieve the competitive fitting ability [15–19]. To achieve this target, the suitable hyper-parameters of EBDEN are optimized by the BO method.

2.1 Deep multiple-scale reservoir of EBDEN

There are several curious hyper-parameters in EBDEN, such as internal weights $\mathbf{W_{in}}$, reservoir weights $\mathbf{W_r}$, the spectral radius ρ, the scaling coefficient ω and the sparsity connection

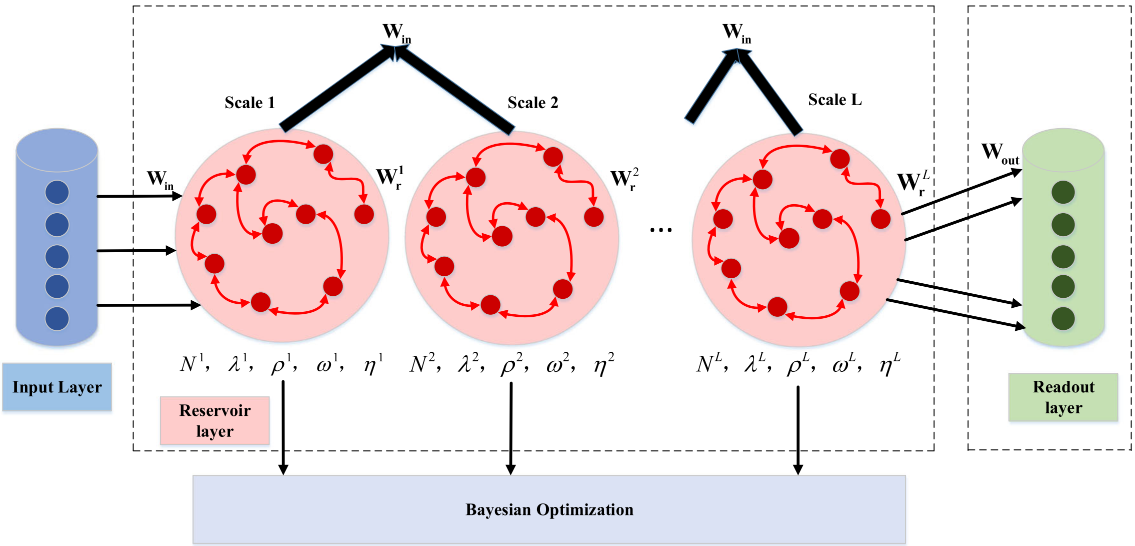


Fig. 1 The illustration of basic architecture of EBDEN. The EBDEN is based on the echo state network, and time-series inputs are incorporated into the T scale of the reservoirs to compute the feature. Each reservoir contains several hyper-parameters, such as the number of units in each reservoir N, leaky rate λ, scaling coefficient ω, spectral radius ρ, connection degree η and scale number L

degree η. With the initialization of $\mathbf{W_{in}}$ following the uniform distribution in $[-1, 1]$, the internal weights $\mathbf{W_{in}}$ range from $[-\omega, \omega]$ after being scaled by the scaling coefficient ω. Besides, ρ denotes the spectral radius of the reservoir weights $\mathbf{W_r}$ that can be calculated as follows:

$$\mathbf{W}_r = \rho \frac{\mathbf{W}}{\mathrm{Eigen}_{\max}(\mathbf{W})} \tag{1}$$

where $\mathbf{W}$ follows the uniform distribution in $[-0.5, 0.5]$ in this paper. According to the echo memory mechanism [19], ranges of ρ and the sparsity connection degree β in reservoirs are both $[0, 1]$. Time-series inputs $\mathbf{x}$ are fed into N units of L scales of the reservoir through the internal synapse $\mathbf{W_{in}^0}$. All units in reservoirs are kept the same size of the reservoir weight matrix $\mathbf{W_r}$, which is $N * N$. In this paper, the echo state $\mathbf{S(t)}$ is extracted by the reservoir. When L equals 1, the echo state $\mathbf{S(t)}$ is activated by external input $\mathbf{x}$, the size of which is $M * N$ for each time step. The $\mathbf{S(t)}$ can be written as follows:

$$\mathbf{S}(t) = (1 - \lambda)\mathbf{S}(t-1) + \lambda \mathbf{H}(t) \tag{2}$$

$$\mathbf{H}(t) = \tanh(\mathbf{W_r}\mathbf{S}(t-1) + \mathbf{W_{in}}\mathbf{x}(t)) \tag{3}$$

where λ is the leaky rate of the echo units that controls the speed of state dynamics. Besides, larger λ denotes faster dynamics for EBDEN. $\mathbf{H(t)}$ represents the intermediate variable at time step t, which incorporates the feedforward $\mathbf{x}(t)$ and the echo state from last time step $\mathbf{S}(t-1)$. Meanwhile, it is bounded by the tanh function, which can influence eigenvalues of the incoming weight matrix and ensure the echo state property [23] of EBDEN. Considering the deep architecture of EBDEN, when L is greater than 1, the L_{th} scale of variables $\mathbf{S}^L(\mathbf{t})$ and $\mathbf{H}^L(\mathbf{t})$ can be rewritten as follows:

$$\mathbf{S}^L(t) = (1 - \lambda)\mathbf{S}^L(t-1) + \lambda \mathbf{H}^L(t) \tag{4}$$

$$\mathbf{H}^L(t) = \tanh(\mathbf{W_{in}}^{L-1}\mathbf{S}^{L-1}(t) + \mathbf{W_r}^L\mathbf{S}^L(t-1)) \tag{5}$$

The $\mathbf{W_{in}}^{L-1}$ in (5) denotes the internal weight matrix between the $L - 1_{\mathrm{th}}$ and L_{th} scale reservoir. In the conventional mechanism of the echo state network, the echo state $\mathbf{S}$ for whole timescale ($\mathbf{S} = [\mathbf{S}(1), \mathbf{S}(2), \ldots, \mathbf{S}(T)]$) is weighted by readout weights $\mathbf{W_{out}}$, which is listed as follows:

$$\mathbf{y} = \mathbf{W_{out}}\mathbf{S} \tag{6}$$

Expected to track any forms of target dynamics $\hat{\mathbf{y}}$ like complex time-series patterns, the output $\mathbf{y}$ can be optimized by learning $\mathbf{W_{out}}$ in Eq. (6) and computing the mean square error with ridge regressor (RC) [24, 25] as shown below:

$$E(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 \tag{7}$$

When dealing with the classification problems, the support vector machines (SVMs) are always used in previous works [26]. Inspired by bidirectional LSTM [27], we adopt echo units with bidirectional connection in the EBDEN for the purpose of extracting more abundant context features from input $\mathbf{x}$. In reservoir space of the EBDEN, except for the forward computation along the time step for the echo state $\mathbf{S}(t)$, reverse computation along the reverse time step for the echo state is computed as well. Different from the representation of unidirectional computation, the echo state $\mathbf{S}(t)$ (described in (4)) and the intermediate variable $\mathbf{H}(t)$ (described in (5)) are represented as $\overrightarrow{\mathbf{S}}^L(t)$, $\overrightarrow{\mathbf{H}}^L(t)$ and $\overleftarrow{\mathbf{S}}^L(t')$, $\overleftarrow{\mathbf{H}}^L(t')$, respectively:

$$\overrightarrow{\mathbf{H}}^{L}(t) = \tanh(\mathbf{W}_{\mathbf{in}}^{L-1}\overrightarrow{\mathbf{S}}^{L-1}(t) + \mathbf{W}_{\mathrm{r}}^{L}\overrightarrow{\mathbf{S}}^{L}(t-1)) \quad (8)$$

$$\overleftarrow{\mathbf{H}}^{L}(t') = \tanh(\mathbf{W}_{\mathbf{in}}^{L-1}\overleftarrow{\mathbf{S}}^{L-1}(t') + \mathbf{W}_{\mathrm{r}}^{L}\overleftarrow{\mathbf{S}}^{L}(t'-1)) \quad (9)$$

where $\overrightarrow{\mathbf{S}}^{L}(t)$ and $\overrightarrow{\mathbf{H}}^{L}(t)$ denote the L_{th} scale of forward states; $\overleftarrow{\mathbf{S}}^{L}(t')$ and $\overleftarrow{\mathbf{H}}^{L}(t')$ represent the reverse states. Accordingly, the start and the end of the simulation t' for $\overleftarrow{\mathbf{H}}^{L}(t')$ in (9) are the end and the start of the simulation t for intermediate states $\overrightarrow{\mathbf{H}}^{L}(t)$, respectively. Substitute (8), (9) into (4), and the following results of forward and reverse echo states are obtained:

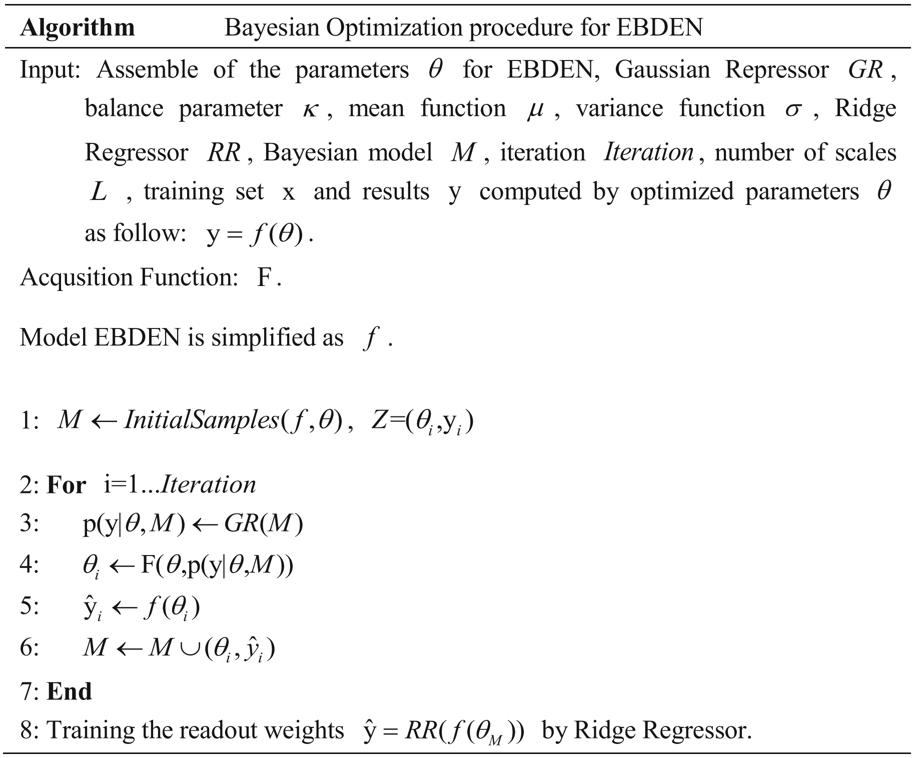
$$\overrightarrow{\mathbf{S}}^{L}(t) = (1-\lambda)\overrightarrow{\mathbf{S}}^{L}(t-1) + \lambda\overrightarrow{\mathbf{H}}^{L}(t) \quad (10)$$

$$\overleftarrow{\mathbf{S}}^{L}(t') = (1-\lambda)\overleftarrow{\mathbf{S}}^{L}(t'-1) + \lambda\overleftarrow{\mathbf{H}}^{L}(t') \quad (11)$$

At the final time step, the forward and reverse echo states are concatenated as the new type of $\mathbf{S}^{L}(t) = [\overrightarrow{\mathbf{S}}^{L}(t); \overleftarrow{\mathbf{S}}^{L}(t')]$. It can be seen that the L_{th} scale of the echo state $\mathbf{S}^{L}$ through T time steps in EBDEN has the size of $M * T * 2NL$.

2.2 Bayesian optimization of EBDEN

Unlike traditional reservoir computing frameworks, several hyper-parameters in EBDEN are unfixed during the learning procedure for the sake of efficaciously modeling time-series datasets. As mentioned before, these hyper-parameters are fitted by Gaussian regressor (GR), in which the Matern kernel is utilized and updated following BO. According to Fig. 1, several hyper-parameters, namely ρ, ω, η, λ, L and N, are sampled by BO, and the readout weight matrix $\mathbf{W_{out}}$ can be learnt to obtain the output $\mathbf{y}$ (according to Eq. 6). In EBDEN, such hyper-parameters and readout weights are optimized alternatively. When hyper-parameters are kept fixed, the readout weights of EBDEN are learnt by ridge regressor; when the latter is kept fixed, the suitable hyper-parameters capable of enhancing the performance are optimized by BO. For the optimization procedure of the hyper-parameters, the initial candidates for hyper-parameters are of equal size and predefined searching space. Furthermore, the average performance of the output $\mathbf{y}$ activated by these candidates is inferred by BO. In this paper, the lower confidence bound (LCB) mechanism, which is controlled by the predictive mean and variance functions, is used as the acquisition type of BO expected to query the suitable solutions to ensure the loss function with lower expectation under the candidates. Hence, suppose the acquisition function is denoted as F and assemble of the parameters as θ for EBDEN containing the spectral radius ρ, the scaling coefficient ω, the sparsity connection degree η, the leaky rate λ, the number of scales of the reservoir L and unit numbers in the reservoir N.



Algorithm Bayesian Optimization procedure for EBDEN

Input: Assemble of the parameters θ for EBDEN, Gaussian Repressor GR, balance parameter κ, mean function μ, variance function σ, Ridge Regressor RR, Bayesian model M, iteration $Iteration$, number of scales L, training set x and results y computed by optimized parameters θ as follow: $\mathrm{y} = f(\theta)$.

Acqusition Function: F.

Model EBDEN is simplified as f.

```
$M \leftarrow InitialSamples(f,\theta)$,  $Z=(\theta_i,\mathrm{y}_i)$
For  i=1...Iteration
    $\mathrm{p}(\mathrm{y}|\theta,M) \leftarrow GR(M)$
    $\theta_i \leftarrow \mathrm{F}(\theta,\mathrm{p}(\mathrm{y}|\theta,M))$
    $\hat{\mathrm{y}}_i \leftarrow f(\theta_i)$
    $M \leftarrow M \cup (\theta_i,\hat{y}_i)$
End
Training the readout weights  $\hat{\mathrm{y}} = RR(f(\theta_M))$  by Ridge Regressor.
```

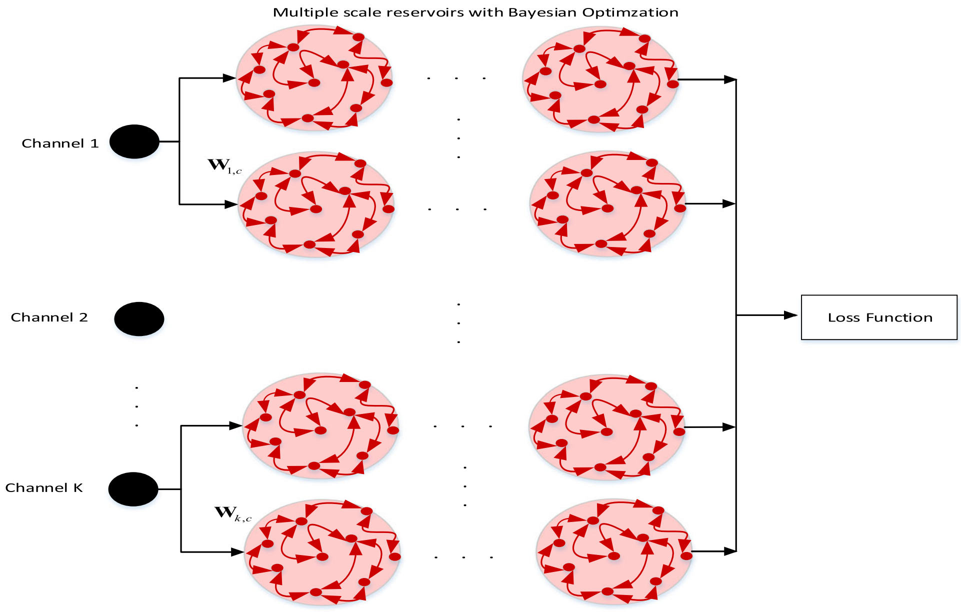


Fig. 2 The ensemble architecture of EBDEN

2.3 Ensemble architecture of EBDEN

When meeting with the complex time-series types, such as multivariate time-series (MTS), each time step stems from the multiple time domain channels. Hence, is it essential for each time domain channel to be modeled by EBDEN? In reality, the computation of such condition is very costly when time-series inputs $\mathbf{x}$ with multiple channels are incorporated into the deep echo network. Since the MTS datasets [28, 29] in the previous works are known to be sourced from the small subset of the total channels, it is eminent that not all time domain channels are deterministic for the computation of the network. In order to avert redundancy computation, ensemble echo network architecture is applied in EBDEN to determine which channels in MTS are with semblable temporal behavior.

According to Fig. 2, the number of echo networks is assigned to ensembles, and each sub-ensemble is independently evaluated by the deep echo network. As mentioned in Sect. 2, there are K time domain channels and C ensembles, with hyper-parameters (ρ, ω, η, λ, L and N) optimized by BO briefly presented as θ. Furthermore, $f(\theta)$ denotes the loss function in line with Eq. (7) and the ensemble weight matrix is denoted as the optimal ensemble weight $\mathbf{W}_{\mathbf{K},\mathbf{C}}$, which measures the importance of each channel individually as follows:

$$\mathbf{W}_{k,c} = \frac{\mathrm{e}^{-f_k(\theta_c)}}{\sum_{c=1}^{C} \mathrm{e}^{-f_k(\theta_c)}} \tag{12}$$

where $f_k(\theta_c)$ denotes the real loss for the c_{th} sub-ensemble when dealing with the k_{th} time domain channel. The total loss of EBDEN is calculated with ensemble weights and the loss function $f(\theta)$:

$$\mathrm{Loss}_{\mathrm{total}} = \frac{1}{C}\sum_{k=1}^{K}\sum_{c=1}^{C}\mathbf{W}_{k,c} f_k(\theta_c) \tag{13}$$

According to (13), the learning stopping procedure of EBDEN for total loss is dominated by two factors: the ensemble weight $\mathbf{W}_{k,c}$ and the function loss $f_k(\theta_c)$ optimized by BO. The overfit learning phenomenon can be relieved by EBDEN when the loss is not merely determined by the loss of sub-ensembles.

3 Datasets and characteristics for EBDEN

In this section, descriptions of some benchmarks, such as Baydogan MTS task, SantaFe Competition A, the NARMA system and chaotic attractors, are discussed. Subsequently, the metric memory capacity is used to explore the advantage of deep reservoirs of EBDEN. The time-consuming performance between EBDEN using BO and grid search mechanisms is calculated to demonstrate the superiority of BO. Finally, the test error performances of EBDEN, EBDEN without the ensemble and EBDEN with global ensemble are compared.

3.1 Datasets

- The public database Baydogan archive [30] containing 13 multivariate time-series datasets is adopted to evaluate the performance of EBDEN and other comparison algorithms.

- Far-infrared-laser SantaFe time-series competition A [31] contains 9000 training and 1000 testing samples, respectively.
- The nonlinear autoregressive moving average (NARMA) [32] system of 10 orders activated by uniform distribution is used.
- The three chaotic attractors [33] are activated among total 2500 time steps. These trajectories for attractors are 1-channel, 2-channel (each channel denotes the *X*- and *Y*-axis) and 3-channel (each channel denotes the *X*-, *Y*- and *Z*-axis) time-series datasets, respectively. The dynamic formulation of the Lorenz attractor system is:

$$\begin{aligned} \dot{x} &= 10(y-x) \\ \dot{y} &= x(28-z)-y \\ \dot{z} &= xy-\frac{8}{3}z \end{aligned} \tag{14}$$

The dynamic formulation of the Rossler attractor system is:

$$\begin{aligned} \dot{x} &= -(y+z) \\ \dot{y} &= x+0.2y \\ \dot{z} &= 0.2+xz-8z \end{aligned} \tag{15}$$

The dynamic formulation of the Mackey–Glass attractor system is:

$$\dot{x} = \frac{0.2x(t-17)}{1+x(t-17)^n} - 0.1x(t) \tag{16}$$

where x, y and z denote the corresponding channels to activate the dynamic systems.

3.2 Contraction mapping analyzation for EBDEN

As described in Sect. 2, the bidirectional deep architecture of the reservoirs contained in EBDEN is composed of the multiple scales of the echo states. To the best of our knowledge, contractivity is used widely in previous works [34] in the aspect of measuring the echo space of reservoir computing. In this section, the exploration for the superiority of the bidirectional multiple-scale reservoir mechanism in the theoretical level is analyzed by contraction mapping.

There are two inputs postulated to have the same time series, with one decayed by Gaussian noise and the other undisturbed, both of which are learnt by EBDEN. Euclidean distance is perceived as the metric to measure the corresponding echo states. The total L scale reservoirs of two echo states, namely $\mathbf{S} = \left(\mathbf{S}^1, \ldots, \mathbf{S}^L\right)$ and $\hat{\mathbf{S}} = \left(\hat{\mathbf{S}}^1, \ldots, \hat{\mathbf{S}}^L\right)$, are calculated, while EBDEN deals with the undisturbed and the disturbed inputs, respectively. As described in Eqs. (4) and (5), the L_{th} scale of the echo state $\mathbf{S}^L$ is dominated by that of the last scale echo state $\mathbf{S}^{L-1}$. Suppose there is a constant $C^{(L)}$, and the L_{th} scale network satisfies the contraction condition in the light of Lipschitz continuity, which can be listed as follows:

$$\begin{aligned} &\left\| F^{(L)}\left(\mathbf{x}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L)}\right) - F^{(L)}\left(\mathbf{x}, \hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L)}\right) \right\| \\ &\quad \le C^{(L)} \left\| \left(\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L)}\right) - \left(\hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L)}\right) \right\| \end{aligned} \tag{17}$$

where $F^{(L)}$ denotes the L_{th} scale state transition function. According to Eqs. (4) and (5), the state transition function is measured by the adaptive process of the echo states, which can be simplified as $\mathrm{s}^{(1)}(t) = F^{(1)}(x(t), s^{(1)}(t-1))$ when the number of scales L is equal to 1. When the number of scales L is larger than 1, the adaptive process of echo states can be simplified as $\mathrm{s}^{(L)}(t) = F^{(L)}(x(t), s^{(1)}(t-1), \ldots, s^{(L)}(t-1))$. With the range of $C^{(L)}$ in $[0,1]$, the higher value of $C^{(L)}$ denotes less contractive dynamics. There is evidence that the range of $C^{(L)}$ is indeed in $[0,1]$ and the contractivity of EBDEN in L scale reservoirs satisfies the Lipschitz continuity lemma.

According to (4), (5) and (17), the difference between $F^{(L)}(x(t), s^{(1)}(t-1), \ldots, s^{(L)}(t-1))$ and $F^{(L)}(x(t), \hat{\mathrm{s}}^{(1)}(t-1), \ldots, \hat{s}^{(L)}(t-1))$ can be calculated as follows:

$$\begin{aligned} &\left\| F^{(L)}\left(\mathbf{x}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L)}\right) - F^{(L)}\left(\mathbf{x}, \hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L)}\right) \right\| \\ &\quad = \left\| (1-\lambda)\mathbf{S}^{(L)} + \lambda \tanh\left(\mathbf{W}_{\mathbf{r}}^{(L)}\mathbf{S}^{(L)} + \mathbf{W}_{\text{in}}^{(L)} F^{(L-1)}\left(\mathbf{x}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L-1)}\right)\right) \right. \\ &\qquad \left. -(1-\lambda)\hat{\mathbf{S}}^{(L)} - \lambda \tanh\left(\mathbf{W}_{\mathbf{r}}^{(L)}\hat{\mathbf{S}}^{(L)} + \mathbf{W}_{\text{in}}^{(\mathrm{L})} F^{(L-1)}\left(\mathbf{x}, \hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L-1)}\right)\right) \right\| \end{aligned} \tag{18}$$

Since the maximum for activation tanh is 1, (18) can be transformed to:

$$\begin{aligned} &\text{Eq. (18)} \\ &\quad \le \left\| (1-\lambda)\mathbf{S}^{(L)} + \lambda\left(\mathbf{W}_{\mathbf{r}}^{(L)}\mathbf{S}^{(L)} + \mathbf{W}_{\mathbf{in}}^{(L)} F^{(L-1)}\left(\mathbf{x}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L-1)}\right)\right) \right. \\ &\qquad \left. - (1-\lambda)\hat{\mathbf{S}}^{(L)} - \lambda\left(\mathbf{W}_{\mathbf{r}}^{(L)}\hat{\mathbf{S}}^{(L)} + \mathbf{W}_{\mathbf{in}}^{(L)} F^{(L-1)}\left(\mathbf{x}, \hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L-1)}\right)\right) \right\| \end{aligned} \tag{19}$$

When the range of λ is in $[0,1]$ and the condition of $\mathbf{S}^{(L)} - \hat{\mathbf{S}}^{(L)} \le ||\mathbf{S}^{(L)} - \hat{\mathbf{S}}^{(L)}||$ is met, $F^{(L-1)}(\mathbf{x}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L-1)}) - F^{(L-1)}(\mathbf{x}, \hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L-1)}) \le ||F^{(L-1)}(\mathbf{x}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L-1)}) - F^{(L-1)}(\mathbf{x}, \hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L-1)})||$. Equation (19) can be further derived as follows:

$$\begin{aligned} &\text{Eq. (19)} \\ &\quad \le (1-\lambda)\left\| \mathbf{S}^{(L)} - \hat{\mathbf{S}}^{(L)} \right\| + \lambda\left(\|\mathbf{W}_{\mathbf{r}}^{(L)}\| \left\| \mathbf{S}^{(L)} - \hat{\mathbf{S}}^{(L)} \right\| \right. \\ &\qquad \left. + \left\| \mathbf{W}_{\mathbf{in}}^{(L)} \right\| \left\| F^{(L-1)}\left(\mathbf{x}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L-1)}\right) - F^{(L-1)}\left(\mathbf{x}, \hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L-1)}\right) \right\| \right) \end{aligned} \tag{20}$$

Considering $||F^{(L-1)}(\mathbf{x}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L-1)}) - F^{(L-1)}(\mathbf{x}, \hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L-1)})|| \le \mathrm{C}^{(i-1)}||(\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L-1)}) - (\hat{\mathbf{S}}^{(1)},$

$\ldots, \hat{\mathbf{S}}^{(L-1)})||$ and $||\mathbf{S}^{(L)} - \hat{\mathbf{S}}^{(L)}|| \leq ||(\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L)}) - (\hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L)})||$, (20) is further enlarged as follows:

$$\begin{aligned}&\text{Eq. (20)}\\ &\quad\leq (1-\lambda)\left\|\left(\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L)}\right) - \left(\hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L)}\right)\right\|\\ &\quad+ \lambda\Big(\|\mathbf{W}_{\mathbf{r}}^{(L)}\| \left\|\left(\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L)}\right) - \left(\hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L)}\right)\right\|\\ &\quad+ \left\|\mathbf{W}_{\mathbf{in}}^{(L)}\right\| C^{(L-1)} \left\|\left(\mathbf{x}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(L-1)}\right) - \left(\mathbf{x}, \hat{\mathbf{S}}^{(1)}, \ldots, \hat{\mathbf{S}}^{(L-1)}\right)\right\|\Big)\end{aligned} \tag{21}$$

From the above, the recurrence formulation for L scale $C^{(L)}$ can be acquired as follows:

$$C^{(L)} = (1-\lambda) + \lambda\left(C^{(L-1)}\left\|\mathbf{W}_{\mathbf{in}}^{(L)}\right\| + \left\|\mathbf{W}_{\mathbf{r}}^{(L)}\right\|\right) \tag{22}$$

Hence, the L_{th} scale of the state transition function $F^{(L)}$ for EBDEN fulfills the requirement of contractivity when the following equation is satisfied:

$$0 < C^{(L)} = (1-\lambda) + \lambda(C^{(L-1)}||\mathbf{W}_{\mathbf{in}}^{(L)}|| + ||\mathbf{W}_{\mathbf{r}}^{(L)}|) < 1$$

. From this recurrence formulation, one can see that when $C^{(L)}$ satisfies $0 \leq C^{(L)} \leq 1$, each reservoir layer and EBDEN can hold contractivity. According to the recurrence formulation for L scale Lipschitz constant $C^{(L)}$ in contraction analyzation in Eq. (22) and the transition function $\mathrm{s}^{(L)}(t) = F^{(L)}(x(t), s^{(1)}(t-1), \ldots, s^{(L)}(t-1))$ of the adaptive process of echo states, for the definition of echo state property, the distinction between two transition functions of EBDEN is close to a constant without any activation. For this reason, the external activation $\mathbf{0}$ is used in transition functions:

$$\begin{aligned}&\left\|F^{(L)}\left(\mathbf{0}, \mathbf{S}^{(L)}\right) - F^{(L)}\left(\mathbf{0}, \hat{\mathbf{S}}^{(L)}\right)\right\|\\ &\quad\leq (1-\lambda)\left\|\mathbf{S}^{(L)} - \hat{\mathbf{S}}^{(L)}\right\| + \lambda\Big(\|\mathbf{W}_{\mathbf{r}}^{(L)}\| \left\|\mathbf{S}^{(L)} - \hat{\mathbf{S}}^{(L)}\right\|\\ &\quad+ C^{(L-1)}\left\|\mathbf{W}_{\mathbf{in}}^{(L)}\right\| \left\|\left(\mathbf{0}, \mathbf{S}^{(L-1)}\right) - \left(\mathbf{0}, \hat{\mathbf{S}}^{(L-1)}\right)\right\|\Big)\end{aligned} \tag{23}$$

After simplification, Eq. (23) can be obtained as follows:

$$\text{Eq. (23)} \propto \lambda^2 C^{(L-2)}\left\|\mathbf{S}^{(L)} - \hat{\mathbf{S}}^{(L)}\right\| \propto \lambda^{L-1} C^{(1)}\left\|\mathbf{S}^{(L)} - \hat{\mathbf{S}}^{(L)}\right\| \tag{24}$$

Because the range of the leaky rate λ is $[0, 1]$, Lipschitz constant $C^{(L)}$ is thus the same. As the bound of $||F^{(L)}(\mathbf{0}, \mathbf{S}^{(L)}) - F^{(L)}(\mathbf{0}, \hat{\mathbf{S}}^{(L)})||$ is found to be a constant, the EBDEN satisfies the echo state property.

3.3 Quantitative analyzation for the BO and multiple-scale mechanisms

As described in Sect. 3.2, the theoretical level analyzation has proved that the architecture of the multiple-scale reservoir can be contractive. Considering that the echo state network with multiple-scale architecture will bring much more hyper-parameters than that with single-scale architecture, the Bayesian optimization (BO) in this paper is used to tune the hyper-parameters. The identical input time-series datasets are learnt by the EBDEN and EBDEN, which does not contain BO (EBDEN without BO). Except for the number of scales between EBDEN and EBDEN without BO, all configurations for these two models are kept consistent, with each result repeated 10 times.

The necessity of the BO for EBDEN is demonstrated in Fig. 3, and the testing error is used as the metric. It can be seen that the EBDEN and EBDEN without BO achieve the comparable performances when the number of scale equals 1. As for the performance of EBDEN without BO, it is not converged with the number of scales in accordance with the blue line in Fig. 3a. When there is an increase in the number of scales, more hyper-parameters are needed to be tuned, giving rise to the instability of the performance of the network if large amounts of the hyper-parameters are inadequately chosen. As comparison, the testing errors of EBDEN shrink with the number of scales, mainly owing to the BO mechanism of searching the appropriate hyper-parameters.

Furthermore, for the sake of quantitatively analyzing the necessity of the multiple-scale mechanism, we apply the memory capacity [35, 36] to measuring the effectiveness of ESNs. In this section, the performance of the short-term memory capacity between EBDEN and shallow EBDEN is compared by evaluating the models' compactness of recalling delayed time series. The calculation procedure of the memory capacity is:

$$C = \sum_{k=0}^{\infty} \mathrm{Corr}(\hat{\mathbf{y}}(t-\tau), \mathbf{y}(t))^2 \tag{25}$$

where C and Corr(.) denote the memory capacity and the correlation operator, respectively. From the above, it is observed that memory capacity is the computation of the squared correlation coefficient between the prediction $\mathbf{y}(t)$ and target $\hat{\mathbf{y}}$ with delay τ. Noticeably, the delay τ varies from 0 to the number of the input samples.

From Fig. 3b, c, the performance of EBDEN is seen to be superior to that of shallow EBDEN in the same configuration (such as leaky rates λ and spectral radius ρ).

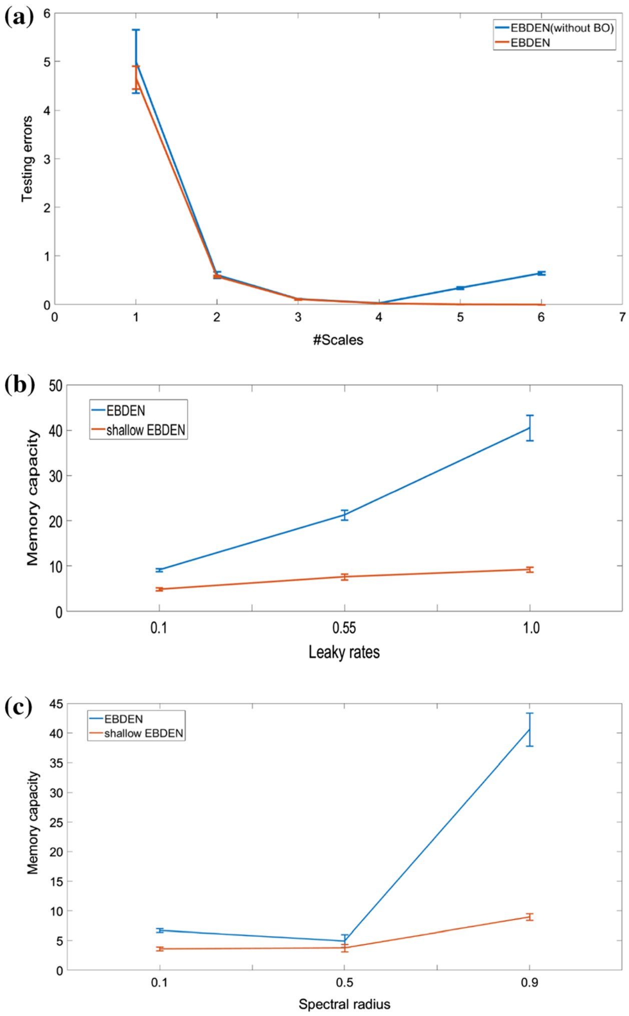


Fig. 3 The characteristics of EBDEN and shallow EBDEN. **a** The performances of EBDEN and EBDEN without BO with the number of scales. **b** The performances of EBDEN and shallow EBDEN with different leaky rates λ. **c** The performances of EBDEN and shallow EBDEN with dissimilar spectral radii ρ

3.4 Quantitative analyzation for BO versus grid search

In this section, the efficiency of using BO to optimize the EBDEN is measured by the comparison with that of using the grid search mechanism. It is worth noting that the range of the parameter configures is invariable when two optimization mechanisms are used. The space of the hyper-parameters for EBDEN is: the number of the units in each reservoir $N\{N \in (100, 1000)\}$, the scaling coefficient $\omega\{\omega \in (0, 1)\}$, the sparsity connection degree $\eta\{\eta \in (0.001, 1)\}$, the spectral radius $\rho\{\rho \in (0, 1.25)\}$, the leaky rate $\lambda\{\lambda \in (0, 1)\}$ and the number of scales $L\{L \in (1, 10)\}$. Moreover, there are 100 points initialized by BO in EBDEN, with the maximum number of iteration and the convergence criterion for BO pre-defined as 1000 and 0.001, respectively. With respect to the model EBDEN with grid search optimization, each hyper-parameter is taken 6 values as: the number of the units in each reservoir $N\{N \in (100, 325, 550, 775, 1000)\}$, the scaling coefficient $\omega\{\omega \in (0, 0.25, 0.5, 0.75, 1)\}$, the sparsity connection degree $\eta\{\eta \in (0.001, 0.0056, 0.031, 0.17, 1)\}$, the spectral radius $\rho\{\rho \in (0, 0.312, 0.625, 0.937, 1.250)\}$, the leaky rate $\lambda\{\lambda \in (0, 0.25, 0.5, 0.75, 1)\}$ and the number of scales $L\{L \in (1, 3, 5, 8, 10)\}$.

The experiments are performed on the three datasets that are described in Sect. 3.1. Besides, two metrics, such as time-consuming and test error, are used to measure the performance of EBDEN and EBDEN with grid search optimization. The experimental results are shown in Table 1.

As can be seen from Table 1, EBDEN can not only speed up the search of suited hyper-parameters in comparison with EBDEN with grid search optimization, but also achieve smaller test errors.

3.5 Quantitative analyzation for ensemble architecture

As mentioned in Sect. 2.3, several echo state networks are ensembled in EBDEN to restrict the redundant computation for each time domain channel. In this section, the edge of the ensemble architecture for EBDEN is explored by measuring whether it can contribute to the performance improvement of modeling the MTS dataset or not. The experiment is benchmarked on time-series dataset with 30 channels by comparing the performance between the EBDEN, EBDEN without ensemble and EBDEN with global ensemble. Note that the EBDEN without ensemble model means that all of the time-series' datasets are incorporated into the same echo state network architecture and the mean value performance is computed. EBDEN with global ensemble model signifies that every channel's

Table 1 The performance of EBDEN versus EBDEN with grid search

Dataset	EBDEN		EBDEN with grid search	
	Time-consuming (s)	Test error	Time-consuming (s)	Test error
Mackey–Glass	3792.605	0.0316	59,807.70194	0.0345
NARMA	4881.858	0.045	36,845.121	0.045
SantaFe Laser	305.535	0.005	1041.878	0.006

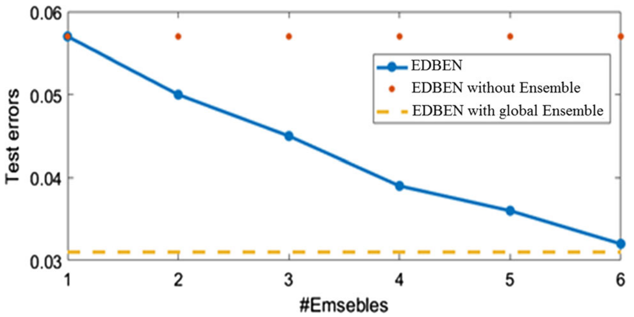


Fig. 4 The characteristic of EDBEN with the number of the ensembles

datasets are modeled by one echo state network (30 ensembles), with the performance shown in Fig. 4.

As is seen from Fig. 4, the number of ensembles for EBDEN ranges is within [1, 6]. As the number of the ensemble grows, the test error of EBDEN decreases into [0.057, 0.032]. When the ensemble number equals 6, the performance of EBDEN gets close to EBDEN with global ensemble.

Except for the comparison between the EBDEN with global ensemble and EBDEN in Fig. 4, in order to measure the necessary for the ensemble learning module in EBDEN, the prediction performance between the EBDEN and SESNE [22] which uses the simple least square regression is also compared. As described in Sect. 1, the optimal procedure is not contained in SESNE, but EBDEN uses the loss function with optimal weights in Eq. (13) to account the importance of MTS's channel individually. From Fig. 5, it can be clearly seen that the performance of EBDEN can be converged at 7 ensemble numbers, but the performance of SESNE cannot be converged with the number of ensembles. Figure 5 shows the optimal loss function of EBDEN can effectively improve the performance of the ensemble organization of the echo state network.

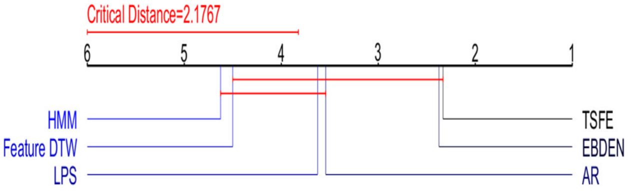


Fig. 6 The critical difference diagram for the comparison of EBDEN with five feature-based models

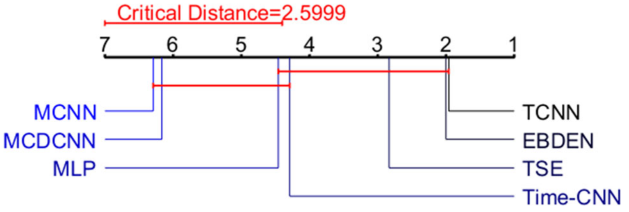


Fig. 7 The critical difference diagram for the comparison of EBDEN with six deep-learning-based models

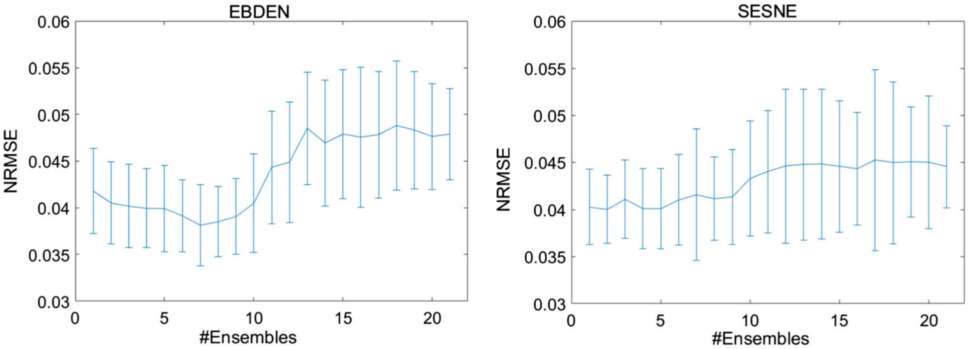
4 Results

All the computation models in this paper are implemented on a TITIAN X Nvidia graphic card with the dual-core Intel CPU processor in Windows environment. Moreover, EBDEN is applied to the classification and fitting problems by using SVM and RC to settle the issue of the readout weight according to Eq. (7). In order to ensure the strictness, all the experiments in Sects. 4.1, 4.2 and 4.3 are repeated 10 times. The critical difference diagram is depicted in Figs. 5, 6 and 7 to describe the statistically significant scores according to the corresponding probability between two comparisons [37]. The critical distance is controlled by the critical difference and number of comparisons.

4.1 Results for multivariate time-series datasets

When dealing with the MTS sets, the datasets comprise multiple time domain channels for each time step. In this study, three representative models for the MTS dataset are

Fig. 5 The performance comparison between EBDEN and SESNE



used: (1) feature-based models, (2) ensemble-based models and (3) deep-learning-based models.

With regard to the feature-based models, the hidden Markov model (HMM) [38] is used as the baseline model. The autoregressive kernel (AR) [39] utilizes the matrix normal-inverse Wishart prior to the measurement of the similarity between two MTS. The distances between the time series as the features are measured by DTW [40] and incorporated into machine learning. In terms of learned pattern similarity (LPS) [41], it models the dependency between the time stamps by local autopatterns. Time-series feature selection (TSFE) [40] computes thousands of time-series features before selecting some discriminative features by greedy forward selection technique with the linear classifier.

As for the deep-learning-based models, six contrasts, such as the temporal convolutional neural network [5], the time-series encoder (TSE) [6], the multi-channel deep convolutional neural network (MCDCNN) [7] and time-CNN [8], are mentioned in Sect. 1. The multi-scale convolutional neural network (MSCNN) [42] introduces the Window Slicing (WS) operator as the feature augmentation method and uses transformation, local convolution and full convolution stages to learn these features. Besides, multiple layer perceptron (MLP) [5] consists of 4 layers connected by full connection to model the time-series datasets.

To model the time-series, autoregressive forest [43] (AF) applies the autoregressive learning mechanism to the tree-based architecture regarding the ensemble-based models. The shapelet ensemble (SE) [44] captures the shapelet subsequence ensembles for time-series sets to measure the similarity between the series. Symbolic representation for multivariate time series (SMTS) [45] regards the code book as the word to label the leaf node trained by random forest.

The comparison of experiment results for EBDEN with feature-based models on 12 MTS sets is shown in Table 2. The contrasts contain HMM, AR, feature DTW, LPS and TSFE. From Fig. 6, it can be seen that EBDEN achieves 7 wins, while the HMM, AR, feature DTW, LPS and TSFE achieve 1, 3, 2, 3, 6 wins, respectively. TSFE holds the first rank of these models, with which EBDEN achieves a comparable performance. In our view, this is mainly due to the performance on UWave, in which TSFE visibly outperforms the EBDEN model.

Table 3 shows the comparison between EBDEN and deep-learning-based models. Thus, it can be seen that EBDEN wins 8 datasets out of the whole 12 datasets, while the other contrasts win 0, 9, 3, 0, 0, and 0 datasets, respectively. Hence, it can be said that EBDEN achieves the comparable performance with TCNN and is superior to other contrasts on MTS datasets, which is supported by the results of the critical difference diagram in Fig. 7. Moreover, the performance of EBDEN is comparable with TCNN.

Table 4 shows the comparison between the EBDEN and three ensemble-based models. The former achieves the 8 wins, while AF, SE and SMTS achieve 2, 5 and 3 wins, respectively. This data indicates the good performance of models SE and EBDEN on MTS sets. According to Fig. 8, EBDEN achieves the first rank among these contrasts and dramatically outperforms other models.

4.2 Results for chaotic series representation

To our best knowledge, chaotic series analyzation has been used in wide range fields, such as radar detection [46], chemical reaction [47] and EEG signal reaction [48]. Owing to the intricate mathematical formulations, Chaotic series are, however, always hard to be learnt. In this

Table 2 Performance of five feature-based time-series models and EBDEN on 12 datasets

Datasets	HMM	AR	Feature DTW	LPS	TSFE	EBDEN
AUSLAN	0.56	0.91	0.72	0.75	0.95	0.98
ArabicDigits	0.97	0.98	0.90	0.97	0.99	0.97
CMUsubject16	0.99	1.00	0.93	1.00	1.00	1.00
CharacterTrajectories	0.91	0.90	0.94	0.96	0.97	0.99
ECG	0.84	0.82	0.79	0.82	0.88	0.84
JapaneseVowels	0.95	0.98	0.96	0.95	0.97	0.99
KickvsPunch	0.64	0.92	0.60	0.90	1.00	0.93
Libras	0.91	0.95	0.88	0.90	0.89	0.90
NetFlow	0.91	0.89	0.97	0.96	0.96	0.97
UWave	0.50	0.90	0.91	0.98	0.97	0.90
Wafer	0.93	0.96	0.97	0.96	0.99	0.99
WalkvsRun	1.00	1.00	1.00	1.00	1.00	1.00
Total wins	1	3	2	3	6	7

Table 3 Performance of six deep-learning-based time-series models and EBDEN on 12 datasets

Datasets	MLP	TCNN	TSE	MSCNN	MCDCNN	Time-CNN	EBDEN
AUSLAN	0.93	0.98	0.94	0.01	0.85	0.72	0.98
ArabicDigits	0.97	0.99	0.98	0.10	0.96	0.96	0.97
CMUsubject16	0.63	1.00	0.98	0.53	0.52	0.98	1.00
CharacterTrajectories	0.97	0.99	0.97	0.05	0.93	0.96	0.99
ECG	0.75	0.87	0.87	0.67	0.53	0.84	0.84
JapaneseVowels	0.98	0.99	0.97	0.10	0.94	0.96	0.99
KickvsPunch	0.61	0.56	0.63	0.54	0.54	0.62	0.93
Libras	0.78	0.97	0.80	0.07	0.65	0.64	0.90
NetFlow	0.52	0.89	0.78	0.78	0.63	0.89	0.97
UWave	0.90	0.93	0.90	0.13	0.84	0.86	0.90
Wafer	0.89	0.98	0.99	0.89	0.66	0.94	0.99
WalkvsRun	0.70	1.00	1.00	0.75	0.5	1.00	1.00
Total wins	0	9	3	0	0	0	8

Table 4 Performance of three ensemble-based time-series models and EBDEN on 12 datasets

Datasets	AF	SE	SMTS	EBDEN
AUSLAN	0.93	0.95	0.94	0.98
ArabicDigits	0.95	0.95	0.96	0.97
CMUsubject16	1.00	1.00	0.99	1.00
CharacterTrajectories	0.92	0.97	0.99	0.99
ECG	0.78	0.88	0.81	0.84
JapaneseVowels	0.95	0.80	0.96	0.99
KickvsPunch	0.97	1.00	0.82	0.93
Libras	0.94	0.91	0.90	0.90
NetFlow	0.82	0.92	0.97	0.97
UWave	0.95	0.92	0.94	0.90
Wafer	0.93	0.95	0.96	0.99
WalkvsRun	1.00	1.00	1.00	1.00
Total wins	2	5	3	8

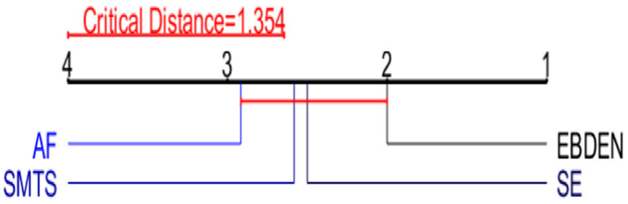


Fig. 8 The critical difference diagram for the comparison of EBDEN with three ensemble-based models

section, the performance of EBDEN is evaluated on three chaotic attractor datasets.

Two contrasts of TSE and MCDCNN are employed as the contrasts in this section. Furthermore, testing error is used as the metric for chaotic series representation. Figures 9 and 10 show the testing outputs and errors, respectively. Since the activation time length is too long, simply the X-axis of these attractors and 100 out of 2500 samples are captured in the figures.

From Figs. 9 and 10, it can be seen that the TSE model outputs the curves with highest testing errors on Lorenz and Mackey–Glass attractors. And MCDCNN model outputs the curve with highest testing errors on Rossler attractor. But for the EBDEN model, it outputs the curves with small testing errors and achieves high goodness-of-fit performance on all three chaotic attractors, which prominently surpasses TSE and MCDCNN. (The testing errors of EBDEN approach 0 in Fig. 10.)

4.3 Results for Dansgaard–Oeschger component estimation

The Dansgaard–Oeschger is the gradual cooling current when there are abrupt increases in the North Atlantic region's surface temperature of up to 15 over a few decades [49]. The Dansgaard–Oeschger effect is exploited by the recording of the time series of the Greenland ice cores collected by the INTIMATE project. Accordingly, the Ca^{2+} and $\delta^{18}O$ are two essential elements to influence the Dansgaard–Oeschger events, the tendency of which is thus predicted by EBDEN when 2000 time-series sets are used as the training dataset and 2500 and 2000 time-series sets as the corresponding testing datasets. Moreover, the DeepESN [19] which uses the recursive least square is deemed as the contrast model in this experiment.

As shown in Fig. 11, EBDEN achieves the comparable and even relatively better performance than DeepESN. From Fig. 11a, b, when meeting with the irregular points like extremum points, the performance of EBDEN is better than that of DeepESN.

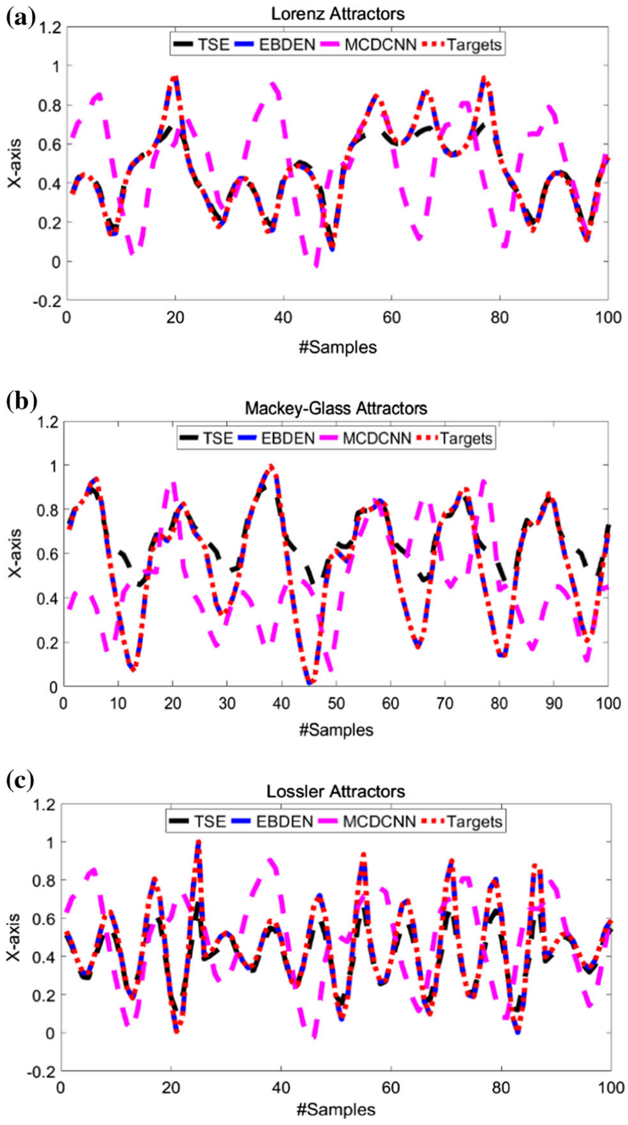


Fig. 9 Three representation performances of chaotic attractor models. **a** The performance of three models on Lorenz attractors. **b** The performance of three models on Mackey–Glass attractors. **c** The performance of three models on Rossler attractors

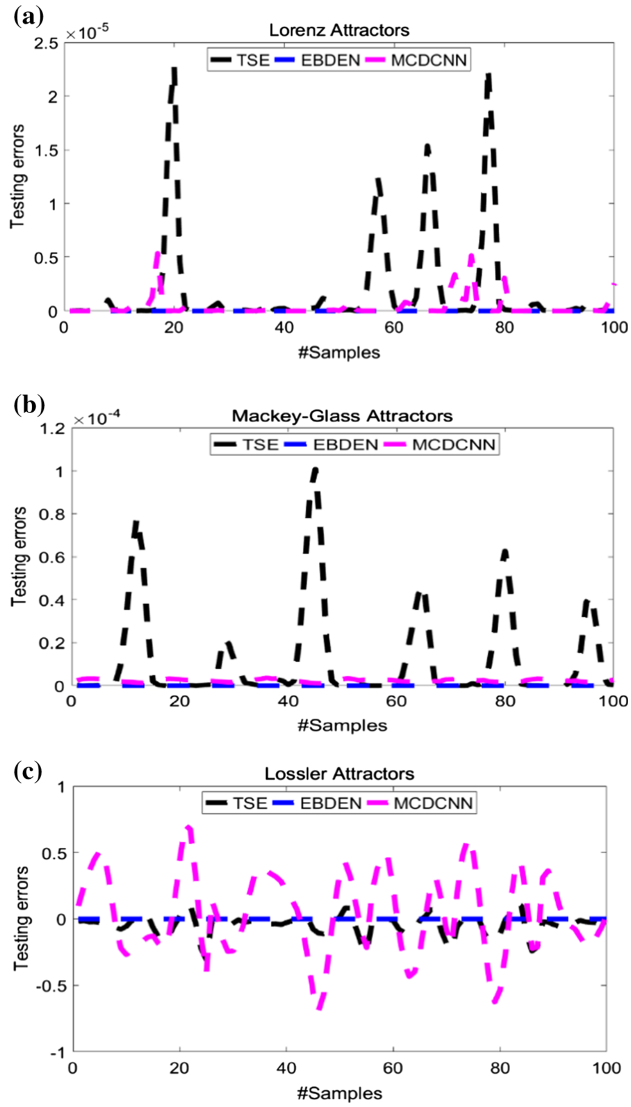


Fig. 10 Three testing error performances of chaotic attractor models. **a** The performance of three models on Lorenz attractors. **b** The performance of three models on Mackey–Glass attractors. **c** The performance of three models on Rossler attractors

5 Conclusions

The brain can give inspiration for echo state models to code various time-series datasets, such as multivariate and chaotic time series. In this paper, a novel echo state computation framework, called ensemble Bayesian deep echo network, is proposed and applied to model the time-series datasets in this paper.

There are three key contributions for this paper. First, the bidirectional multiple-scale reservoirs across the time step are fused to construct the deep architecture of the ensemble Bayesian deep echo network, which is demonstrated by the contraction mapping analyzation in this paper to own the higher memory capacity than that for the shallow reservoirs. The second contribution is Bayesian optimization used in the network to select the suitable hyper-parameters, which can activate the network to achieve great performance. Different from traditional echo state networks, the hyper-parameters are not kept fixed in the ensemble Bayesian deep echo network. Third, the ensemble Bayesian deep echo network can avoid redundant computing when encountering with multiple channels of multivariate time series by the ensemble mechanism.

Due to these contributions, the ensemble Bayesian deep echo network can be used as a high-performance brain-like computational framework for solving realistic problems, such as multivariate time-series classification, chaotic attractors-based time-series representation and Dansgaard–

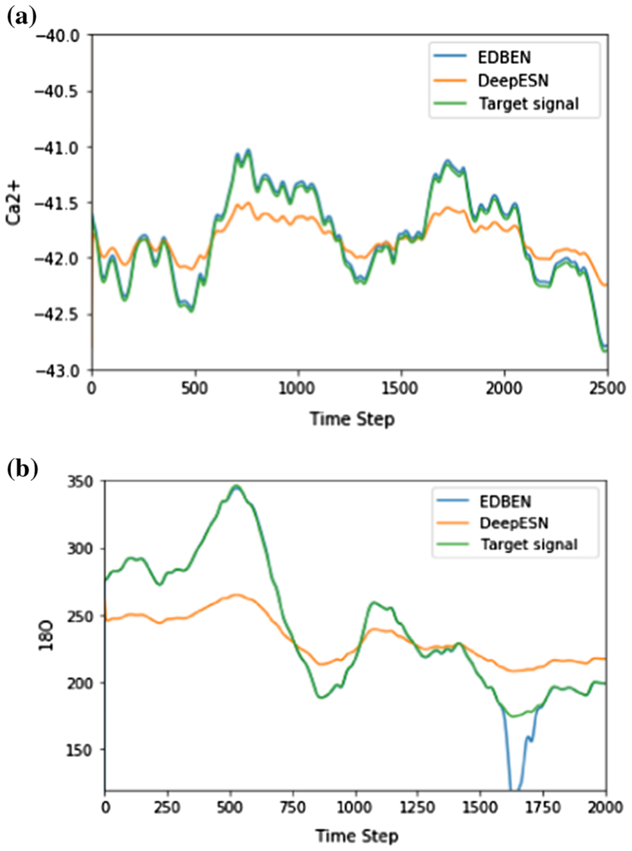


Fig. 11 Part of testing results for Ca^{2+} and δ^{18}O from EBDEN and DeepESN

Oeschger component estimation tasks. Besides, it can bridge the gap between bio-inspired networks and conventional neural network models.

Acknowledgements This work was supported by the National Science Foundation for Young Scientists of China (61801338), the National Natural Science Foundation of China (61874079 and 61574102), the Wuhan Research Program of Application Foundation (2018010401011289) and the Luojia Young Scholars Program.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest.

References

1. Lin P, Chang S, Wang H, Huang Q, He J (2018) SpikeCD: a parameter-insensitive spiking neural network with clustering degeneracy strategy. Neural Comput Appl 5786:1–13. https://doi.org/10.1007/s00521-017-3336-6
2. Hu R, Chang S, Wang H, He J, Huang Q (2018) Efficient multispike learning for spiking neural networks using probability-modulated timing method. IEEE Trans Neural Netw Learn Syst 99:1–14. https://doi.org/10.1109/TNNLS.2018.2875471
3. Sheng P, Han J, Hua W, Hathal A, Yu Z, Mazrouei SM (2018) Modulation classification based on signal constellation diagrams and deep learning. IEEE Trans Neural Netw Learn Syst 30:718–727. https://doi.org/10.1109/TNNLS.2018.2850703
4. Tang ZR, Chang S, Ma QM, Zhu RH, He J, Wang H, Huang QJ (2018) A hardware friendly unsupervised memristive neural network with weight sharing mechanism. Neurocomputing 332:193–202. https://doi.org/10.1016/j.neucom.2018.12.049
5. Wang Z, Yan W, Oates T (2017) Time-series classification from scratch with deep neural networks: a strong baseline. In: Proceedings IJCNN, pp 2161–2161-8
6. Serra J, Pascual S, Karatzoglou A (2018) Towards a universal neural network encoder for time series. In: International conference of the Catalan Association for Artificial Intelligence, pp 120–129
7. Zheng Y, Liu Q, Chen E, Ge Y, Zhao JL (2016) Exploiting multichannels deep convolutional neural networks for multivariate time series classification. Front Comput Sci 10:96–112. https://doi.org/10.1007/s11704-015-4478-2
8. Zhao B, Lu H, Chen S, Liu J, Wu D (2017) Convolutional neural networks for time series classification. Syst Eng Electron 28:162–169. https://doi.org/10.1007/978-3-319-59060-8_57
9. Karim F, Majumdar S, Darabi H, Chen S (2018) LSTM fully convolutional networks for time series classification. IEEE Access 6:1662–1669. https://doi.org/10.1109/ACCESS.2017.2779939
10. Ibrahim AO, Shamsuddin SM, Abraham A (2012) Adaptive memetic method of multi-objective genetic evolutionary algorithm for backpropagation neural network. Neural Comput Appl 31:4945–4962. https://doi.org/10.1007/s00521-018-03990-0
11. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: Proceedings ECCV, pp 630–645
12. Yang C, Qiao J, Wang L (2018) Dynamical regularized echo state network for time series prediction. Neural Comput Appl 31:6781–6794. https://doi.org/10.1007/s00521-018-3488-z
13. Hu R, Huang Q, Wang H, Chang S (2019) Monitor-based spiking recurrent network for the representation of complex dynamic patterns. Int J Neural Syst 29:1950006–1950023. https://doi.org/10.1142/s0129065719500060
14. Jaeger H, Haas H (2004) Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science 304:78–80. https://doi.org/10.1126/science.1091277
15. Li Q, Wu Z, Zhang H (2020) Spatio-temporal modeling with enhanced flexibility and robustness of solar irradiance prediction: a chain-structure echo state network approach. J Clean Prod 261:1–10. https://doi.org/10.1016/j.jclepro.2020.121151
16. Wu Z, Li Q, Xia X (2020) Multi-timescale forecast of solar irradiance based on multi-task learning and echo state network approaches. IEEE Trans Ind Inf. https://doi.org/10.1109/TII.2020.2987096
17. Gallicchio C, Micheli A (2017) Echo state property of deep reservoir computing networks. Cognit Comput 9:337–350. https://doi.org/10.1007/s12559-017-9461-9
18. Chen S, Chen M (2013) Addressing the advantages of using ensemble probabilistic models in estimation of distribution algorithms for scheduling problems. Int J Prod Econ 141:24–33. https://doi.org/10.1016/j.ijpe.2012.05.010
19. Qiao J, Li F, Han H, Li W (2017) Growing echo-state network with multiple subreservoirs. IEEE Trans Neural Netw Learn Syst 28:391–404. https://doi.org/10.1109/TNNLS.2016.2514275
20. Li Z, Zheng Z, Outbib R (2019) Adaptive prognostic of fuel cells by implementing ensemble echo state networks in time-varying model space. IEEE Trans Ind Electron 67:379–389. https://doi.org/10.1109/TIE.2019.2893827
21. Bacic B (2016) Echo state network ensemble for human motion data temporal phasing: a case study on tennis phasing: a case

study on tennis forehands. Int Conf Neural Inf Process. https://doi.org/10.1007/978-3-319-46681-1_2

22. IbanezSoria D, SoriaFrisch A, GarciaOjalvo J, Ruffini G (2018) Echo state networks ensemble for SSVEP dynamical online detection. https://doi.org/10.1101/268581
23. Jaeger H, Lukoševičius M, Popovici D, Siewert U (2007) Optimization and applications of echo state networks with leaky-integrator neurons. Neural Netw 20:335–352. https://doi.org/10.1016/j.neunet.2007.04.016
24. Xiang K, Nan LB, Zhang L, Pang M, Wang M, Li X (2009) Regularized Taylor echo state networks for predictive control of partially observed system. IEEE Access 4:3300–3309. https://doi.org/10.1109/ACCESS.2016.2582478
25. Chatzis SP, Demiris Y (2011) Echo state Gaussian process. IEEE Trans Neural Netw 22(9):1435–1445. https://doi.org/10.1109/TNN.2011.2162109
26. Rodan A, Faris H (2015) Echo state network with SVM-readout for customer churn prediction. IEEE Jordan Conf Appl Electr Eng Comput Technol. https://doi.org/10.1109/AEECT.2015.7360579
27. Graves A, Schmidhuber J (2005) Frame phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw 18:602–610. https://doi.org/10.1016/j.neunet.2005.06.042
28. Pan WJ, Dibeklioglu H, Tax D, Maaten L (2018) Multivariate time series classification using the hidden unit logistic model. IEEE Trans Neural Netw Learn Syst 29:920–931. https://doi.org/10.1109/TNNLS.2017.2651018
29. Hu R, Huang Q, Chang S, Wang H (2019) The MBPEP: a deep ensemble pruning algorithm providing high quality uncertainty prediction. Appl Intell 49:2942–2955. https://doi.org/10.1007/s10489-019-01421-8
30. Baydogan MG. Multivariate time series classification datasets. www.mustafabaydogan.com. Accessed 2015
31. Weigend S, Morgan M, Srivastava AN (1995) Nonlinear gated experts for time series: discovering regimes and avoiding overfitting. Int J Neural Syst 6:373–399. https://doi.org/10.1142/s0129065795000251
32. Yue Y, Cheng X, Gao S (2017) Data driven identification and control of nonlinear systems using multiple NARMA-L2 models. Int J Robust Nonlinear Control. https://doi.org/10.1002/rnc.3818
33. Vishik IM (2001) Attractors for equations of mathematical physics. Am Math Soc Colloq Publ Am Math Soc 49:363. https://doi.org/10.1007/s10489-019-01421-8
34. Gallicchio C, Micheli A (2011) Architectural and markovian factors of echo state networks. Neural Netw 24:440–456. https://doi.org/10.1016/j.neunet.2011.02.002
35. Gallicchio C, Micheli A, Pedrelli L (2017) Deep reservoir computing: a critical experimental analysis. Neurocomputing 268:87–99. https://doi.org/10.1016/j.neucom.2016.12.089
36. Grigoryeva L, Henriques J, Larger L, Ortega JP (2016) Nonlinear memory capacity of parallel time-delay reservoir computers in the processing of multidimensional signals 28:1411–1451. https://doi.org/10.1162/NECO_a_00845
37. Demsar J (2006) Statistical comparisons of classifiers over multiple datasets. J Mach Learn Res 7:1–30. https://doi.org/10.1007/s10846-005-9016-2
38. Petropoulous A, Chatzis SP, Xanthopoulous S (2017) A hidden markov model with dependence jumps for predictive modeling of multidimensional time-series. Inf Sci 412:50–66. https://doi.org/10.1016/j.ins.2017.05.038
39. Yu P, Li W, Ng F (2017) The generalized conditional autoregessive Wishart model for multivariate realized volatility. J Bus Econ Stat 35:1–41. https://doi.org/10.1080/07350015.2015.1096788
40. Kate RJ (2016) Using dynamic time warping distances as features for improved time series classification. Data Min Knowl Disc 30:283–312. https://doi.org/10.1007/s10618-015-0418-x
41. Baydogan M, Runger G (2016) Time series representation and similarity based on local autopatterns. Data Min Knowl Disc 30:476–509. https://doi.org/10.1007/s10618-015-0425-y
42. Cui Z, Chen W, Chen Y (2016) Multi-scale convolutional neural network for time series classification. arXiv: 1603.06995
43. Tuncel KS, Baydogan MG (2018) Autoregressive forests for time series modeling. Pattern Recognit 73:202–215. https://doi.org/10.1016/j.patcog.2017.08.016
44. Hills J, Lines J, Baranauskas E, Mapp J, Bagnall A (2014) Classification of time series by shapelet transformation. Data Min Knowl Disc 28(4):851–881. https://doi.org/10.1007/s10618-013-0322-1
45. Baydogan M, Runger G (2015) Learning a symbolic representation for multivariate time series classification. Data Min Knowl Disc 29:400–422. https://doi.org/10.1007/s10618-014-0349-y
46. Shen D, Zhang LR, Liu X, Liu N (2013) A novel method of using chaotic sequences in MIMI radar for multiple targets detection. In: Proceedings IEEE ICCT, pp 1–5. https://doi.org/10.1109/ICCT.2012.6511328
47. Sawyers DR, Sen M, Chang HC (1996) Effect of chaotic interfacial stretching on bimolecular chemical reaction in helical-coil reactors. Chem Eng J 64:129–139. https://doi.org/10.1016/S0923-0467(96)03132-6
48. Liu WH, Huang QJ, Chang S, Wang H, He J (2018) Multiple feature branch convolutional neural network for myocardial infarction diagnosis using electrocardiogram. Biomed Signal Process Control 45:22–32. https://doi.org/10.1016/j.bspc.2018.05.013
49. Lohmann J, Ditlevsen PD (2018) Random and externally controlled occurrences of Dansgaard–Oeschger events. Clim Past 14:609–617. https://doi.org/10.5194/cp-14-609-2018