IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 2, 2020, accepted May 9, 2020, date of publication May 14, 2020, date of current version May 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994773

Long-Short Term Echo State Network for Time Series Prediction

KAIHONG ZHENG[1], BIN QIAN[1], SEN LI[2], YONG XIAO[1], WANQING ZHUANG[2], AND QIANLI MA[2], (Member, IEEE)

[1]Electric Power Research Institute, China Southern Power Grid, Guangzhou 510080, China
[2]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

Corresponding author: Qianli Ma (qianlima@scut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872148, in part by the Natural Science Foundation of Guangdong Province under Grant 2017A030313355, Grant 2017A030313358, and Grant 2019A1515010768, and in part by the Guangzhou Science and Technology Planning Project under Grant 201704030051 and Grant 201902010020.

ABSTRACT The Echo State Networks (ESNs) is an efficient recurrent neural network consisting of a randomly generated reservoir (a large number of neurons with sparse random recurrent connections) and a trainable linear layer. It has received widespread attention for its simplicity and effectiveness, especially for time series prediction tasks. However, there is no explicit mechanism in ESNs to capture the inherent multi-scale characteristics of time series. To this end, we propose a model consisting of multi-reservoir structure named long-short term echo state networks (LS-ESNs) to capture the multi-scale temporal characteristics of time series. Specifically, LS-ESNs consists of three independent reservoirs, and each reservoir has recurrent connections of a specific time-scale to model the temporal dependencies of time series. The multi-scale echo states are then collected from each reservoir and concatenated together. Finally, the concatenated echo states representations are fed to the linear regression layer to obtain the results. Experiments on two time series prediction benchmark data sets and a real-world power load data sets demonstrate the effectiveness of the proposed LS-ESNs.

INDEX TERMS Time series prediction, echo state networks (ESNs), multi-scale temporal dependencies, long short term reservoir.

I. INTRODUCTION

In the past few years, time series analysis has become a very active research field. Commonly used techniques include time series prediction [1], [2], classification [3], clustering [4], outlier detection [5], and so on. Time series prediction is an attempt to predict the evolution of a system or phenomenon through previously observed values. It has a wide range of applications in agriculture, commerce, meteorology, military, and medical fields. Many methods have been proposed for time series prediction, such as fully connected feedforward neural network (FNN), support vector regression (SVR), recurrent neural networks (RNNs), etc. References [6]–[8]. Among these methods, RNNs show strong capabilities in dealing with non-linear time series because of the recurrent connections between neurons, which can approximate any non-linear system with arbitrary precision [9], [10].

Although RNNs have achieved excellent performance in many time series prediction tasks, it directly optimizes recurrent weights by the BPTT algorithm, which suffers from the problem of slow convergence, high computational cost, gradient vanishing/exploding, and local optimal solutions [11]–[13]. Thus, currently, the number of RNNs' hidden units is usually set to be relatively small, but this, in turn, reduces the expression ability of the model [14]. Recently, an efficient recurrent network model was proposed to solve the above problems by Jaeger [15], named echo state networks (ESNs). A typical ESNs consists of an input layer, a reservoir (a large number of sparsely connected neurons, typically 100 to 1000), and an output layer. The weights of input layer and the ones of the reservoir are fixed after initialization, and the closed-form solution directly obtains the output weights. Therefore, it does not suffer from the slow convergence and training process, and we can get the optimal global solution. It also avoids the gradient vanishing/exploding problem. With these properties and theoretical guarantee [16], [17], ESNs are widely used

The associate editor coordinating the review of this manuscript and approving it for publication was Aysegul Ucar.

in the time series prediction tasks [18]–[21]. For example, Song *et al.* [22] applied ESNs on real-time power load forecasting and achieved better performance of prediction than feed-forward neural networks. Li *et al.* [23] addressed the over-fitting problem existing in the pseudo-inverse algorithm and solved the output weights of ESNs with the bayesian framework, which further improved the performance of prediction. Deihimi *et al.* [24] adopt the wavelet transform to decompose the data and then input the decomposed components to the ESNs, which achieved better prediction performance. Chatzis and Demiris [25] proposed ESGP, combining the ESNs and Gaussian processes, to provide a confidence measure of the prediction values. To further improve the non-linearity in reservoirs, Gallicchio and Micheli [26] proposed φ-ESNs, adding a fixed random-projection layer of ELMs [27] to a reservoir. Moreover, Butcher *et al.* [28], [29] proposed R^2SP, adding two ELMs to encode inputs and reservoir states, respectively. Their results showed that employing ELMs as intermediate layers helps to improve the prediction performance.

Although ESNs have performed well on time series prediction, the existing work on ESNs has not explicitly captured the inherent multi-scale characteristics of time series yet [24], [30]. The traditional ESNs only have a single recurrent module, which makes it challenging to capture the multi-scale temporal features.

Recently, some literature proposed to model multi-scale features by stacking ESNs, such as [31], [32]. However, since the size of the reservoir is usually very large, stacking ESNs will inevitably pass some redundant information to the high-level and also impact the efficiency of the model [33]. Moreover, the stacking methods tend to model long-term features with the stacking of layers [34] and may ignore some essential short-term features [35].

To this end, we propose a long-short term echo state networks (LS-ESNs) to capture the multi-scale features of time series. LS-ESNs extracts the multi-scale temporal features with the multiple independent reservoirs that possess different recurrent connections. Specifically, LS-ESNs consists of three reservoirs, a long-term reservoir, a typical reservoir, and a short-term reservoir. The long-term reservoir is capable of capturing the long-term temporal dependency feature by skip connections. But this is the first time exploited in the RC approaches. The typical reservoir is consistent with the traditional ESNs based on the Markov hypothesis. And the short-term reservoir is capable of capturing the short-term temporal dependencies by ''Eliminating memory'', which only considers the impact of recent history memory. After that, the echo states from theses three reservoirs are concatenated together, forming the multi-scale temporal representations. Finally, the predictions are obtained by feeding the concatenated representations to the output layer. To verify the effectiveness, we conduct experiments on two time series prediction benchmark data sets and one real-world power load data set. We compared our model with other commonly used time series prediction methods. The experimental results show that the prediction performance of LS-ESNs is significantly better than other compared methods. The main contributions of this paper can be summarized as follows:

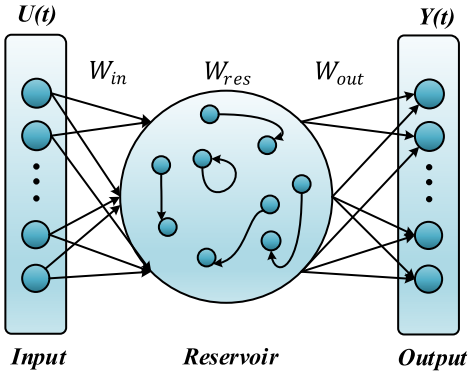


FIGURE 1. Echo state networks.

- We propose a novel multi-scale ESN-based network named LS-ESNs, which is composed of three independent reservoirs, and each reservoir has a different recurrent connection for capturing different temporal scale dependencies.
- With the training-free reservoirs and output weights obtained by the closed-form solution, the LS-ESNs possess high-computational efficiency for modeling complex temporal data.
- Compared with the several baseline methods, LS-ESNs achieves better performances on two prediction benchmark data sets and the real-word power load data set.

The remainder of this paper is organized as follows. In Section II, we briefly introduce the traditional ESNs. Section III introduces the details of the proposed method LS-ESNs. Section IV shows the experimental results and analysis of the well-known prediction benchmark and some real-world time series. Finally, we conclude in Section V.

II. ECHO STATE NETWORKS

A typical ESNs consists of an input layer, a recurrent layer, called reservoir (with a large number of sparsely connected neurons), and an output layer. The connection weights of the input layer and the reservoir layer are fixed after initialization, and the output weights are trainable and obtained by solving a linear regression problem. The structure of the echo state network is shown in Figure 1.

$u(t) \in \mathbb{R}^{D\times 1}$ denotes the input value at time t of the time series, $W_{in} \in \mathbb{R}^{N\times D}$ (D is the dimension of the input) represents the connection weights between the input layer and the hidden layer, $W_{res} \in \mathbb{R}^{N\times N}$ (N is the number of neurons) denotes input-to-hidden layer connections weights, and $W_{out} \in \mathbb{R}^{L\times N}$ (L is the dimension of the output) is the output weights from the hidden layer to the output layer. The state transition equation can be expressed by Equation (1):

$$x(t) = \gamma \cdot tanh(W_{in}u(t) + W_{res}x(t-1) + (1-\gamma)\cdot x(t-1)) \quad (1)$$

$$y(t) = W_{out}x(t) \quad (2)$$

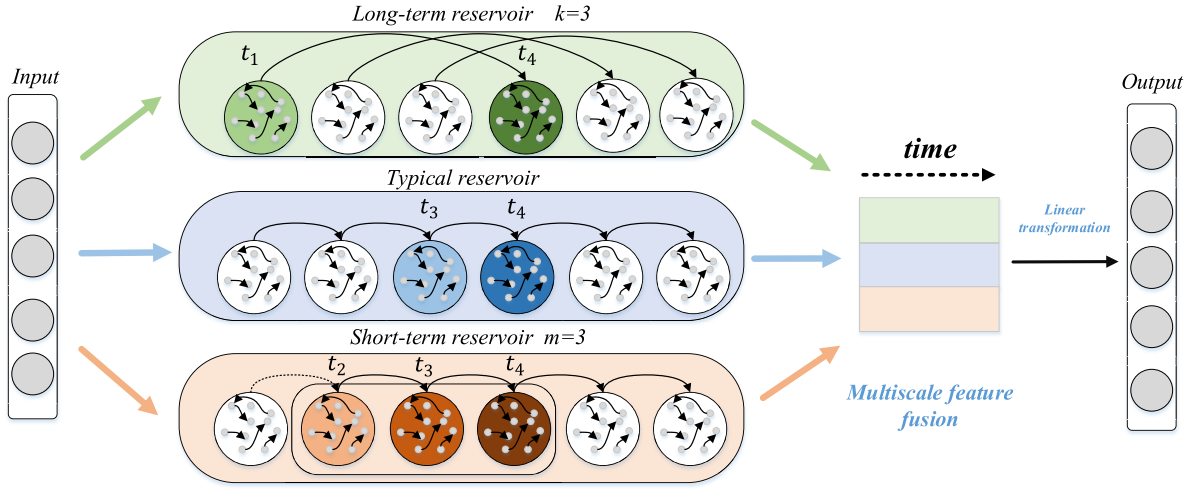


FIGURE 2. The general architecture of LS-ESNs. The LS-ESNs consist of multiple independent reservoirs, and each reservoir has a different recurrent connection. The long-term reservoir (top) is capable of capturing long-term temporal dependency feature with skip step (set skip step $k = 3$, then the state of t_4 directly dependent on the t_1). The typical reservoir (middle) bases on the Markov hypothesis (the state of t_4 directly dependent on the t_3). The short-term reservoir (bottom) is capable of capturing short-term temporal dependency feature (set short dependency scope $m = 3$, then the state of t_4 only dependent on t_2, t_3, wiping out the state outside this scope). Concatenating all the echo states from different reservoirs into the multi-scale temporal feature and then fed it to the output layer to get the final prediction.

where $u(t)$, $x(t) \in \mathbb{R}^{N\times 1}$ and $y(t)$ refer to the input, state of the reservoir and output, respectively. γ denotes the leaky rate, which integrates the previous state into the current one. $tanh(\cdot)$ and $\cdot$ denote the activation function in reservoir and multiplication, respectively.

The main hyper-parameters of an ESNs are the input scale IS, the spectral radius SR and sparsity α.

- IS is used for the initialization of the matrix W_{in}, and the elements of W_{in} are scaled between $-IS$ to IS.
- SR is the spectral radius of W_{res}, given by

$$W_{res} = SR \cdot \frac{W}{\lambda_{max}(W)} \tag{3}$$

  where $\lambda_{max}(W)$ is the largest eigenvalue of matrix W and elements of W are generated randomly in $[-0.5, 0.5]$.
- α denotes the proportion of non-zero elements in W_{res}.

The main characteristics of ESNs can be summarized as the following aspects:

- A multiple high-dimensional projection method is employed by ESNs to capture the dynamics of the input, which makes the reservoir provide strong nonlinear mapping capabilities.
- Echo State Property (ESP) [15], [36]: The ESP means that inputs with more similar short-term history will evoke closer echo states, which ensure the dynamical stability of the reservoir. ESP also provides the ESNs an important capability called "fading memory" or "short-term memory." With this short-term memory, the input history information from some time past will not easily fade away. This is guaranteed by limiting the spectral radius of recurrent connection weights to less than 1.
- None of the input and recurrent weights are trained, and the closed-form solution of linear regression can obtain the output weights.

III. LONG-SHORT TERM ECHO STATE NETWORKS

A. PROPOSED METHOD

To enhance the ability of ESNs to model multi-scale temporal features, we propose a long-short term echo state networks (LS-ESNs). LS-ESNs is composed of multiple independent reservoirs with different recurrent connections. Each reservoir is capable of capturing the different temporal scale dependency. The long-term reservoir captures the long-term dependency feature by skip connection, the typical reservoir is consistent with traditional ESNs based on the Markov hypothesis, and the short-term reservoir captures the short-term dependency feature by using the dependency scope m to only consider the impact of recent m history states. Then, the original time series is represented by the different temporal dependency scale echo states collected from these three reservoirs. Finally, the concatenated echo states are fed into the output layer to obtain the prediction result.

The model is shown in Figure 2, and it is composed of three different temporal dependency scale reservoirs, long-term reservoirs, typical reservoirs, short-term reservoirs. We will give the state transition equations of three different reservoirs, respectively. Note that, $u(t) \in \mathbb{R}^{D\times 1}$ denotes the input of time t-th.

1) THE LONG-TERM RESERVOIR

The long-term reservoirs state $x_{long}(t) \in \mathbb{R}^{N\times 1}$ (N is the size of the reservoirs) is updated by Equation (4) as follows:

$$x_{long}(t) = \gamma \cdot tanh(W_{in}u(t) + W_{res}x_{long}(t-k) + (1-\gamma)\cdot x_{long}(t-k)) \tag{4}$$

where k is the length of the skipped step, the bigger the value is, the longer temporal scale and a wider range of dependencies are. γ, W_{in} and W_{res} denote the leaky rate,

TABLE 1. The detailed hyper-parameter settings of LS-ESNs on the power load forecasting data sets and and the benchmark data sets.

No.	N	IS			SR			γ			k	m
		Long	Typical	Short	Long	Typical	Short	Long	Typical	Short		
1	600	0.108	0.533	0.593	0.102	0.143	0.736	0.039	0.316	0.913	4	2
2	600	0.081	0.063	0.088	0.537	0.791	0.851	0.649	0.143	0.562	4	4
3	300	0.334	0.433	0.587	0.14	0.933	0.154	0.511	0.599	0.559	8	2
4	900	0.25	0.375	0.785	0.616	0.151	0.699	0.975	0.358	0.26	4	6
5	900	0.922	0.69	0.17	0.406	0.288	0.677	0.881	0.37	0.733	4	6
6	900	0.429	0.676	0.593	0.446	0.861	0.8	0.416	0.287	0.547	4	4
7	600	0.3	0.453	1	0.123	0.52	0.511	0.071	0.637	0.581	8	4
8	600	0.093	0.77	0.375	0.16	0.849	0.375	0.857	0.937	0.299	8	8
9	600	0.213	0.964	0.006	0.032	0.323	0.291	0.363	0.886	0.018	10	2
10	900	0.25	0.147	1	0.625	0.552	0.866	0	1	0.866	8	2
Sunspots	300	0.350	0.996	0.062	0.114	0.783	0.391	0.470	0.998	0.240	10	2
Lorenz	600	0.899	0.907	0.485	0.863	0.468	0.194	0.990	0.647	0.953	4	2

connection weights from the input to the reservoir and the reservoir to itself, respectively.

Especially, the representations of the original time series at time step t depend on the state at time step $t-k$ and the input at t-th. Thus, Equation (4) explicitly models the more long-term dependency with a bigger value of skip step k.

2) THE TYPICAL RESERVOIR

The typical reservoirs state $x_{typical}(t) \in \mathbb{R}^{N\times 1}$ is updated by Equation (5) as follows:

$$x_{typical}(t) = \gamma \cdot tanh(W_{in}u(t) + W_{res}x_{typical}(t-1) + (1-\gamma)\cdot x_{typical}(t-1)) \quad (5)$$

where $x_{typical}(t-1)$ refers to the state of time $t-1$. Especially, the representations of the original time series at time step t is dependent on the previous time step $t-1$ and the input at t-th.

3) THE SHORT-TERM RESERVOIR

The short-term reservoirs state $x_{short}(t) \in \mathbb{R}^{N\times 1}$ aims to capture short-term dependencies and is updated by Equation (8) as follows:

$$x_{short}(t) = \gamma \cdot tanh(W_{in}u(t) + W_{res}x(t-1) + (1-\gamma)\cdot x(t-1)) \quad (6)$$

$$x(t-1) = \gamma \cdot tanh(W_{in}u(t-1) + W_{res}x(t-2) + (1-\gamma)\cdot x(t-2)) \quad (7)$$

$$\vdots$$

$$x(t-m+1) = \gamma \cdot tanh(W_{in}u(t-m+1) + W_{res}x(t-m) + (1-\gamma)\cdot x(t-m)) \quad (8)$$

where m represents the short-term dependency scope, $x(i), i = t-m+1, t-m, \ldots, t-1$ denote the intermediate iteration state. The short-term reservoirs capture the short-term dependency feature by considering only the history states from time step $t-m$ to time step $t-1$.

The short-term reservoir is very different from the typical one, because in the update formula of the typical reservoir, the time $t-1$ can contain relative long history information, while in the short-term reservoir, $t-1$ only considers the history information within the m time steps. In practice, when calculating the short-term reservoir state $x_{short}(t)$ at time step t, the state $x(t-m)$ at time step $t-m$ is re-initialized by sampling from the standard normal distribution, and loop m steps to obtain the state $x(t)$ at time step t, which is finally assigned to $x_{short}(t)$). $x(i), i = t-m+1, t-m, \ldots, t-1$ refers to the intermediate states and will not be collected in the echo state matrix. In contrast, $x_{short}(t)$ will be collected into the echo state matrix.

At each time step t, the representations of the original time series in different time scales are obtained by Equations (4), (5), and (8), respectively, denoted as $x_{long}(t), x_{typical}(t), x_{short}(t)$. Finally, the representations of different scales feature of the time series are concatenated as the multi-scale temporal representations denoted by $X(t) = [x_{long}(t), x_{typical}(t), x_{short}(t)] \in \mathbb{R}^{3N\times 1}$ and then are fed into the linear output layer as follows:

$$y(t+1) = f_{out}(W_{out}X(t)) \quad (9)$$

Rewriting Equation (9) in matrix form and setting f_{out} to be identify function, we have:

$$Y = W_{out}X \quad (10)$$

The weights of the output layer are trained by minimizing the loss function as follow:

$$L(W_{out}) = \|T - W_{out}X\|_2^2 + \lambda\|W_{out}\|_2^2 \quad (11)$$

where $T \in \mathbb{R}^{T\times 1}$ denotes the teacher signal, $W_{out} \in \mathbb{R}^{T\times 3N}$ represents the output layer weights, and λ is the regularization coefficient. The output layer weights can be solved by the closed-form solution of Equation (11) with the pseudo-inverse solution method [37], which is given by as follows:

$$W_{out} = (X^TX + \lambda I)^{-1}X^TT \quad (12)$$

B. ALGORITHM DESCRIPTION

In this section, we describe the details of our proposed method LS-ESNs.

1) DATA PREPROCESSING AND NETWORK INITIALIZATION

- (1) The original data $M = [m(1), , m(t), , m(T)]$ is normalized by the Equation (13). The time series after processing is represented as $U = [u(1), , u(t), , u(T)]$. Then

we divided U into three parts with the ratio of 7:1:2: the training set U_{Train}, the validation set U_{Val} and the test set U_{Test}. $U_{Train}[1 : T - step]$ and $U_{Train}[step + 1 : T]$ are taken from U_{Train} as input signal U_{input} and teacher signal Y_{target} in the training phase respectively, Also, $U_{Test}[1 : T - step]$ and $U_{Test}[step + 1 : T]$ are taken from U_{Test} as input signal and teacher signal in the test phase. *Step* represents the prediction horizon. The validation set U_{Val} is used for tune hyper-parameters.

$$u(i) = \frac{m(i) - min(M)}{max(M) - min(M)} \tag{13}$$

- (2) Set the size of reservoirs N, spectral radius SR, input scale IS and other hyper-parameters. The input layer weights W_{in} and the recurrent connection weights W_{res} are sampled from the standard normal distribution and fixed during the training phase.

2) THE TRAINING ALGORITHM OF LS-ESNs

Note that only the output layer weights are trainable while we fix the weights of the input layer and the ones of the reservoir with a spectral radius less than 1 after initialization, which is the necessary condition stability of ESNs and ESP.

- (1) The training data set U_{train} is fed to the long-term reservoirs, the typical reservoir, and the short-term reservoir, respectively. Then we can obtain the different scale features of original time series by Eq. (4), (5), and (8) step by step. We collect the echo states of each reservoir to get the long-term, typical, and short-term temporal features, denoted by x_{long}, $x_{typical}$ and x_{short} respectively.
- (2) Concatenate all the features obtained by step (1) as the multi-scale representations of time series denoted by $X_{train} = [x_{train\_long}, x_{train\_typical}, x_{train\_short}] \in \mathbb{R}^{3N\times T}$. The weights of output layer can be obtained by Equation (12).

3) THE PREDICTION ALGORITHM OF LS-ESNs

- (1) Select all the optimal hyper-parameters according to the performance on the validation sets.
- (2) The testing data set $U_{Test}[1 : T - step]$ is fed into the long-term reservoir, the typical reservoir and the short-term reservoir respectively. Then we can obtain the different time-scale features of original time series by Eq. (4), (5), and (8) step by step.
- (3) Concatenate all the features obtained by step (2) as the multi-scale representations of time series denoted by $X_{test} = [x_{test\_long}, x_{test\_typical}, x_{test\_short}] \in \mathbb{R}^{3N\times T}$. Input the concatenated features into the well-trained output layer to obtain the prediction $Y_{pre} = W_{out}X_{test}$.

The complete algorithm of LS-ESNs is given by Algorithm 1.

Algorithm 1 Long-Short Term Echo State Networks (LS-ESNs)

Input: U_{train} :input in training phase; Y : teacher signal;
U_{test} : input in testing phase; N : size of reservoir;
IS : input scale; SR : spectral radius;
γ : leaky rate; k : skip step;
m : short dependency scope;
λ : regularization coefficient;

Output:
Y_{pre} : prediction result.

for $t = 0 : T_{trian} - 1$ **do**
  compute $x_{train\_long}(t)$ according to Eq. (4)
  compute $x_{train\_typical}(t)$ according to Eq. (5)
  compute $x_{train\_short}(t)$ according to Eq. (8)
end for
$X = concat(x_{train\_long}, x_{train\_typical}, x_{train\_short})$
$W_{out} = (X^T X + \lambda I)^{-1} X^T Y$
for $t = 0 : T_{test} - 1$ **do**
  compute $x_{test\_long}(t)$ according to Eq. (4)
  compute $x_{test\_typical}(t)$ according to Eq. (5)
  compute $x_{test\_short}(t)$ according to Eq. (8)
end for
$X_{test} = concat(x_{test\_long}, x_{test\_typical}, x_{test\_short})$
$Y_{pre} = W_{out}X_{test}$

C. COMPUTATIONAL COMPLEXITY ANALYSIS

In this section, we analyze the computational complexity of our proposed LS-ESNs. Assuming an LS-ESNs has 3 reservoirs, where sizes of the reservoirs are all fixed by N. Given T-length D-dimensional input time series, we can analyze the computational complexity of LS-ESNs as follows.

For the high-dimensional projection of input and the update step of echo state in long-term reservoir computing Equation (4), its complexity can be computed by

$$C_{long} = \mathcal{O}(\alpha TN^2 + TND) \tag{14}$$

where α represents the sparsity of the reservoir and is usually very small (set to 0.05 in our experiment).

For the typical reservoir computing Equation (5), its complexity can be computed by

$$C_{typical} = \mathcal{O}(\alpha TN^2 + TND) \tag{15}$$

For the short-term reservoir computing Equation (8), its complexity can be computed by

$$C_{short} = \mathcal{O}(\alpha mTN^2 + mTND) \tag{16}$$

where m represents the scope of short-term dependency.

The dimension of LS-ESNs is formed by concatenating the echo states of long-term, typical, and short-term reservoir, where its size is $3N$. In this way, the complexity of solving the regression problem in Eq. (12) can be computed by

$$C_{reg} = \mathcal{O}((3N)^3 + 2(3N)^2T + (3N)^2 + 3NTD) \tag{17}$$

Thus in total, the computational complexity of LS-ESNs can be given by

$$\begin{aligned} C_{LS-ESNs} &= C_{long} + C_{typical} + C_{short} + C_{reg} \\ &= \mathcal{O}(2\alpha TN^2 + 2TND + \alpha mTN^2 + mTND \\ &\quad + (3N)^3 + 2(3N)^2T + (3N)^2 + 3NTD) \end{aligned} \tag{18}$$

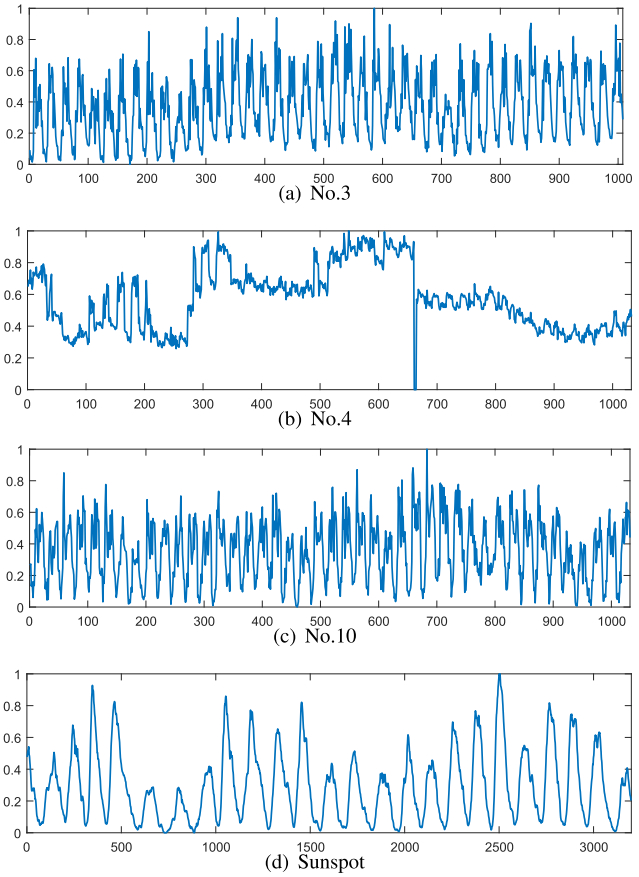


FIGURE 3. Visualization of the data set. (a) the customer No.3, (b) the customer No.4, (c) the customer No.10, and (d) Sunspot data set.

Since the size of reservoir N is typically 100 to 1000 and the dimension of the input time series D is usually much smaller than N, we can assume that $N \gg D$. Moreover, if T is much larger than N, then we have $T \gg N$, and m is a constant. The computational complexity of LS-ESNs is

$$C_{LS-ESNs} \approx \mathcal{O}(TN^2) \tag{19}$$

A traditional ESN's computational complexity can be given by

$$\begin{aligned} C_{ESNs} &= C_{typical} + C_{reg} \\ &\approx \mathcal{O}(TN^2) \end{aligned} \tag{20}$$

We can see that the complexity of LS-ESNs is basically the same as that of traditional ESNs. Therefore, LS-ESNs remains the high computational efficiency of traditional reservoir computing networks.

IV. EXPERIMENTS

In this section, we first verify the proposed method LS-ESNs on two time series prediction benchmark data sets, and then we apply it to a read-world power load prediction. The two time series prediction benchmark data sets are Monthly Sunspot [38] and Lorenz [39], respectively. The real-world power load data is collected from large customers (large factories) in a particular area of China on our own.

To evaluate the effectiveness of proposed long-short term echo state networks (LS-ESNs), we compare with nine baseline models. These methods can be divided into two categories: 1) the traditional methods, including Fully connected feedforward neural network(FNN) [40], support vector regression (SVR) [41], RNN [8], LSTM [42]. 2) the ESN-based methods, inlcuding φ-ESNs [26], R^{2}SP [28], [29], MESM [32], deepESN [31] and the constructed ESN-based method, multiple echo state networks (M-ESNs).

Compared With Traditional Methods: We first compare our model with SVR, FNN, RNN, and LSTM. SVR and FNN that usually use sliding windows for time series prediction. For example, if setting the sliding window size to 4, the input is $\{u_{t-4}, u_{t-3}, u_{t-2}, u_{t-1}\}$ and the target output is u_t (u_t denotes input value at time step t of the time series). For a fair comparison, we also apply the mechanism of the sliding window to make a prediction for RNN and our model LS-ESNs. For all of these methods, we search the window size from $\{12, 24, 48\}$. For SVR, the kernel parameters are searched from$\{0.01, 0.1, 1, 10\}$. For the FNN, the batch-size are searched from$\{10, 20, 30, 40\}$ and the hidden size are searched from$\{10, 20, 30\}$. For the RNN and LSTM, the hidden size is searched from $\{10, 20, 30\}$ and the batch-size are searched from $\{10, 20, 30, 40\}$.

Compared With ESN-Based Methods: To be consistent with the settings of the existing ESN literature rather than using sliding windows to make predictions, we adopt the values of previous time-steps as the inputs and the ones at the next time-step as the output. For example, the inputs are $\{u_{t-4}, u_{t-3}, u_{t-2}, u_{t-1}\}$ and the target output are $u_{t-3}, u_{t-2}, u_{t-1}, u_t$, respectively. Furthermore, we construct a baseline model named multiple echo state network (M-ESN) with three traditional independent reservoirs to verify the effectiveness of the various recurrent connections in LS-ESNs. In other words, we replace the long-term reservoir, and the short-term reservoir by two typical reservoirs and Equation (5) updates the states of reservoirs in the M-ESNs. The sizes of the reservoirs in φ-ESNs, R^{2}SP, MESM and M-ESNs are consistent with the LS-ESNs and the values are searched from$\{300, 600, 900\}$ for all of these methods.

We here focus on the one-step-ahead prediction. Thus, all of the methods are required to learn the mapping from the current input $u(t)$ to the target output $y(t)$, where $y(t) = u(t+1)$. We experimented ten times independently and reported the average results and standard deviations.

The commonly used strategies grid search for hyper-parameters setting can not apply for LS-ESNs due to its large parameter space. Therefore, we use Genetic Algorithms (GA), which is a heuristic optimization method that generates high-quality solutions for optimization and search problems [43]. In the following experiment, we adopt the metric of MSE to optimize hyper-parameter, including IS, SR, and γ. Note that, the GA is also applied to optimize all the ESN-based method (i.e., φ-ESNs, R^{2}SP, MESM,

TABLE 2. Genetic Algorithms (GA) search settings.

GA search settings	
Input scale IS	[0,1]
Spectral radius SR	[0,1]
Leaky rate γ	[0,1]
Population size	20
Generations	20

TABLE 3. Average results with standard deviations of one-step-ahead prediction for Monthly Sunspot data set.

Methods	MSE	SMAPE	NRMSE
φ-ESNs	2.19E-06± 1.12E-07	2.15E-02±8.82E-04	2.17E-02±5.59E-04
R^2SP	1.92E-06± 4.11E-07	1.95E-02±1.95E-02	2.10E-02±2.10E-02
MESM	1.95E-06 ±3.23E-07	2.03E-02±1.30E-03	2.15E-02±1.76E-03
deepESN	1.92E-06±5.37E-08	1.99E-02± 3.72E-04	**2.04E-02± 3.35E-04**
M-ESNs	2.05E-06±3.55E-07	1.92E-02±1.50E-03	2.17E-02±1.86E-03
LS-ESNs	**1.87E-06± 2.31E-07**	**1.91E-02±1.07E-03**	2.08E-02±1.25E-03

TABLE 4. Average results with standard deviations of one-step-ahead prediction for Lorenz data set.

Methods	MSE	SMAPE	NRMSE
φ-ESNs	1.38E-08± 1.12E-07	6.02E-04±8.82E-04	2.53E-04±5.59E-04
R^2SP	5.73E-10±2.57E-10	9.70E-05±2.30E-05	5.25E-05± 6.46E-06
MESM	4.22E-09±1.24E-08	1.04E-04±9.01E-05	8.67E-05±1.14E-04
deepESN	1.12E-09±6.43E-11	1.28E-04±1.82E-05	7.17E-05±2.07E-06
M-ESNs	5.32E-09±1.48E-08	1.22E-04 ±1.01E-04	9.74E-05±1.34E-04
LS-ESNs	**5.49E-10±8.83E-12**	**8.26E-05±1.58E-05**	**5.14E-05± 4.12E-07**

deepESN and M-ESNs). Moreover, we select the optimal hyper-parameters (skip connection step k, dependent scope m, regularization coefficient λ, and sparsity α) according to the performances on the validation sets. The λ are searched from $\{10^{-4}, 10^{-2}, 1, 10^{2}\}$. The k and m are both searched from $\{2, 4, 6, 8, 10, 12\}$. The impacts of skip step k and dependent scope m will be discussed in the section of hyper-parameter analysis. The details of the GA parameters settings are given in Table 2.

A. EVALUATION METRIC

The performance of all methods is evaluated by three widely used metrics: the mean squared error (MSE), the normalized root mean squared error (NRMSE), and the symmetric mean absolute percentage error (SMAPE). They are used by most of the ESN-based methods and can be formulated as follows.

$$\mathrm{MSE} = \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2 \tag{21}$$

$$\mathrm{NRMSE} = \sqrt{\frac{\sum_{t=1}^{T}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{T}(y_t - \bar{y})^2}} \tag{22}$$

$$\mathrm{SMAPE} = \frac{1}{T}\sum_{t=1}^{T}\frac{|y_t - \hat{y}_t|}{(|y_t| + |\hat{y}_t|)/2} \times 100\% \tag{23}$$

where T denotes the length of the target signals. y_t, $\hat{y}_t$ and $\bar{y}$ are the ground truth, prediction and mean value, respectively.

B. TIME SERIES PREDICTION BENCHMARK DATA SETS

The descriptions of two classic time series prediction benchmark data sets are as follows:

- Monthly Sunspot [38]: The sunspot number is a dynamic manifestation of the strong magnetic field in the Sun's outer regions. The study found that the number of sunspots has a very close statistical relationship with solar activity. Due to the complexity of potential solar activity and high non-linearity of the sunspot series, forecasting and analyzing these series is a challenging task, and it is often used to evaluate time series prediction models capability. The World Data Center SILSO provides an open-source 13-month smooth monthly sunspot series. From July 1749 to November 2016, there were a total of 3,209 sample points. Since the last 11 points are still temporary and may be revised, we remove these points using only the remaining 3198 observation points for a one-step prediction task [33].
- Lorenz [39]: The Lorenz system is one of the most typical benchmark tasks for time series prediction, which is described as follows:

$$\begin{cases} \dfrac{dx}{dt} = -ax(t) + ay(t) \\ \dfrac{dy}{dt} = bx(t) - y(t) - x(t)z(t) \\ \dfrac{dz}{dt} = x(t)y(t) - cz(t) \end{cases} \tag{24}$$

  where $a = 10$, $b = 28$, and $c = 8/3$. We use the Runge-Kutta method to generate a Lorenz time series sample of length 2500 from the initial conditions (12, 2, 9), usually with a step size of 0.01. The generated time series is then normalized to $[-1, 1]$. Here we select the data of the x-axis for one-step time series prediction task.

As shown in Tables 3 and 4, our proposed LS-ESNs outperform all the baselines in terms of the lower metric of MSE, SMAPE, and NRMSE, demonstrating that the multi-scale temporal dependency feature does improve the prediction performance.

Moreover, we also plot the prediction curve of the baselines on the Monthly Sunspot data set, removing the methods SVR and FNN due to the poor prediction performance. As shown in Figure 4, in most smooth regions, most of the methods can obtain excellent predictive performance. However, on non-smooth, complex regions (circled by the blue box in the figures), our proposed LS-ESNs predict more accurately than others, which indicates that the multi-scale temporal dependency features help to describe complex local patterns, improving the prediction performance.

TABLE 5. Average results of MSE with standard deviations of one-step-ahead prediction for real-world power load forecasting data sets.

NO.	R^2SP	φ-ESNs	MESM	deepESN	M-ESNs	LS-ESNs
1	4.78E-03±1.22E-05	5.14E-03±1.31E-04	4.24E-03±1.55E-04	4.30E-03±3.80E-04	4.74E-03 ±1.89E-04	**3.61E-03±8.36E-05**
2	1.89E-02±1.31E-04	**1.63E-02±1.31E-04**	1.80E-02±7.52E-05	1.73E-02±5.85E-04	1.81E-02±1.89E-04	1.66E-02±8.36E-05
3	9.73E-03±6.71E-04	9.02E-03±1.55E-04	8.00E-03±7.54E-04	9.60E-03±8.28E-04	9.37E-03±2.19E-04	**7.80E-03±1.30E-04**
4	9.28E-04±1.29E-04	2.62E-03±8.54E-05	1.09E-03±3.56E-04	7.49E-04±5.46E-05	8.80E-04 ±1.76E-04	**7.24E-04±4.63E-05**
5	5.99E-03±1.29E-04	6.27E-03±1.26E-04	5.21E-03±4.20E-04	**4.90E-03±4.31E-04**	6.23E-03 ±2.28E-03	5.27E-03±1.61E-04
6	2.81E-03±2.51E-04	**2.60E-03±5.08E-04**	2.90E-03±5.62E-04	3.00E-03±3.96E-04	3.21E-03±7.51E-05	2.82E-03±2.65E-04
7	1.48E-04±7.08E-05	6.40E-04±1.58E-05	1.06E-04±2.18E-05	**4.25E-05±1.02E-04**	2.67E-04±5.72E-05	6.65E-05±1.44E-05
8	2.20E-02±7.08E-05	2.70E-02±1.17E-03	2.13E-02±2.01E-04	2.33E-02±1.20E-03	2.26E-02 ±2.91E-04	**2.12E-02±1.04E-07**
9	2.76E-03±7.06E-05	2.84E-03± 1.17E-03	2.90E-03±1.15E-03	3.60E-03±2.22E-04	3.22E-03±1.24E-04	**2.50E-03±1.08E-04**
10	7.29E-03±7.06E-05	7.28E-03 ±2.40E-04	7.25E-03 ±2.81E-04	6.60E-03±8.40E-04	8.44E-03±1.61E-03	**6.45E-03±1.32E-04**

TABLE 6. Average results of SMAPE with standard deviations of one-step-ahead prediction for real-world power load forecasting data sets.

NO.	R^2SP	φ-ESNs	MESM	deepESN	M-ESNs	LS-ESNs
1	3.29E-01±3.30E-07	3.10E-01±4.68E-03	3.06E-01±5.77E-03	3.04E-01±2.61E-02	3.36E-01±9.12E-04	**2.67E-01±5.08E-03**
2	3.35E-01±1.31E-04	3.29E-01±2.16E-03	3.42E-01±8.24E-04	3.32E-01±9.40E-03	3.29E-01±1.26E-03	**3.23E-01±1.81E-03**
3	1.89E-01±1.14E-02	1.82E-01±4.03E-03	1.64E-01±8.09E-03	1.65E-01±2.18E-02	1.78E-01±4.12E-03	**1.54E-01±4.85E-03**
4	**6.58E-02±7.24E-03**	8.90E-02±6.18E-04	8.90E-02 ±8.09E-03	6.96E-02±2.18E-02	7.90E-02±1.10E-02	6.85E-02±9.61E-02
5	8.02E-01±8.06E-01	**6.56E-01± 2.34E-02**	8.02E-01 ±5.32E-02	7.94E-01±7.85E-01	8.01E-01±2.03E-01	7.79E-01±9.61E-02
6	**1.56E-01±3.00E-03**	1.61E-01±1.53E-02	1.61E-01±9.35E-03	1.57E-01±2.21E-01	1.64E-01±2.76E-03	1.61E-01±1.72E-02
7	1.32E+00±3.48E-02	1.29E+00±3.11E-02	1.28E+00±2.87E-02	**1.24E+00±3.95E+00**	1.36E+00±7.17E-02	1.27E+00±3.07E-02
8	4.35E-01±3.48E-02	3.76E-01±6.05E-03	3.61E-01± 4.57E-03	3.36E-01±4.57E-03	4.34E-01±4.68E-03	**3.27E-01±3.74E-07**
9	2.65E-01±6.66E-02	2.64E-01±6.05E-03	2.64E-01±1.83E-02	3.48E-01±5.45E-01	6.29E-01±6.05E-03	**2.61E-01±5.56E-01**
10	3.20E-01 ±2.24E+00	3.07E-01±1.78E-02	3.00E-01±1.66E-02	3.13E-01±1.24E-01	3.45E-01±2.10E-02	**2.87E-01±6.29E-02**

TABLE 7. Average results of NRMSE with standard deviations of one-step-ahead prediction for real-world power load forecasting data sets.

NO.	R^2SP	φ-ESNs	MESM	deepESN	M-ESNs	LS-ESNs
1	6.72E-01±3.99E-07	6.87E-01±6.83E-03	6.29E-01±1.12E-02	6.50E-01 ±2.73E-02	6.23E-01±3.65E-03	**3.66E-01±3.72E-03**
2	5.66E-01±2.79E-04	5.34E-01±3.27E-03	5.03E-01±1.62E-03	4.00E-01±1.32E-02	5.55E-01±4.36E-03	**3.68E-01±1.98E-03**
3	5.08E-01±1.76E-02	4.88E-01±4.16E-03	5.09E-01±1.99E-02	4.60E-01±1.30E-02	4.84E-01±5.78E-03	**4.43E-01±3.87E-03**
4	4.66E-01±2.86E-02	5.34E-01±4.16E-03	4.35E-01±6.18E-02	3.72E-01±1.34E-02	4.05E-01±4.01E-02	**3.65E-01±1.17E-02**
5	4.89E-01±5.99E-03	**3.98E-01 ±4.37E-03**	5.06E-01±1.84E-02	4.81E-01±2.08E-02	5.41E-01 ±8.20E-02	4.97E-01±7.57E-03
6	3.46E-01 ±9.99E-03	3.36E-01 ±1.71E-02	2.49E-01 ±1.59E-02	3.47E-01 ±1.74E-02	3.63E-01±2.93E-03	**2.46E-01±1.05E-02**
7	2.21E+00±5.15E-01	1.96E+00±5.49E-02	1.56E+00±2.87E-02	**1.21E+00±7.48E-01**	1.76E+00±3.38E-01	1.51E+00±1.58E-01
8	7.62E-01± 5.15E-01	7.57E-01±1.23E-02	6.58E-01±3.10E-03	6.81E-01±1.77E-02	6.59E-01±4.33E-03	**6.49E-01±1.59E-06**
9	2.19E-01±2.77E-03	2.20E-01±1.23E-02	2.37E-01±3.64E-02	2.19E-01±7.80E-03	2.26E-01±4.55E-03	**2.08E-01±4.52E-03**
10	4.64E-01±2.84E-03	4.66E-01±5.99E-03	4.76E-01 ±8.92E-03	4.58E-01±2.72E-02	5.16E-01±4.31E-02	**4.55E-01±6.29E-02**

C. REAL-WORLD POWER LOAD FORECASTING

The expansion of modern power grids and the rapid spread of smart grids have increased requirements for power operations and accurate dispatching. Among them, power load forecasting is a critical way to improve the stability of the power grid system and achieve accurate power dispatching. Accurate power load forecasting can help save energy and reduce production costs, and increase the competitiveness of the enterprise market.

Therefore, we apply LS-ESNs to the power load forecasting data set collected from large customers (factories) in a particular area of China on our own. It records the hourly power load of large customers from January 2017 to December 2018. Since the original data contains missing values, for each large customer, we selected the longest completed continuous sub-sequence as the new data set. The statistical information of the new data set is shown in Table 8. We visualize the load data of the large customer NO.3, No.4 and No.10, as shown in Figure 3, NO.3 and No.10 both present strong nonlinearity and seasonality and No.4 exhibits the violently fluctuating characteristics. We conducted the experiments on these sampled data sets. Besides, the hyper-parameter settings of LS-ESNs on power load forecasting are searched by GA again, and shown in Table 1.

TABLE 8. Statistics of the used real-world power load data set.

NO.	Start date	End date	Length
1	2017-04-14	2017-05-28	1080
2	2017-10-31	2017-12-12	1032
3	2017-05-25	2017-06-30	1008
4	2018-09-14	2018-11-03	1224
5	2018-10-10	2018-12-03	1320
6	2017-10-31	2018-01-31	2184
7	2018-11-09	2018-12-31	1032
8	2017-11-07	2017-12-31	1080
9	2017-06-18	2017-09-23	2352
10	2018-11-07	2018-12-29	1032

LS-ESNs are compared with the two types of methods mentioned before separately on the real-world power load data sets.

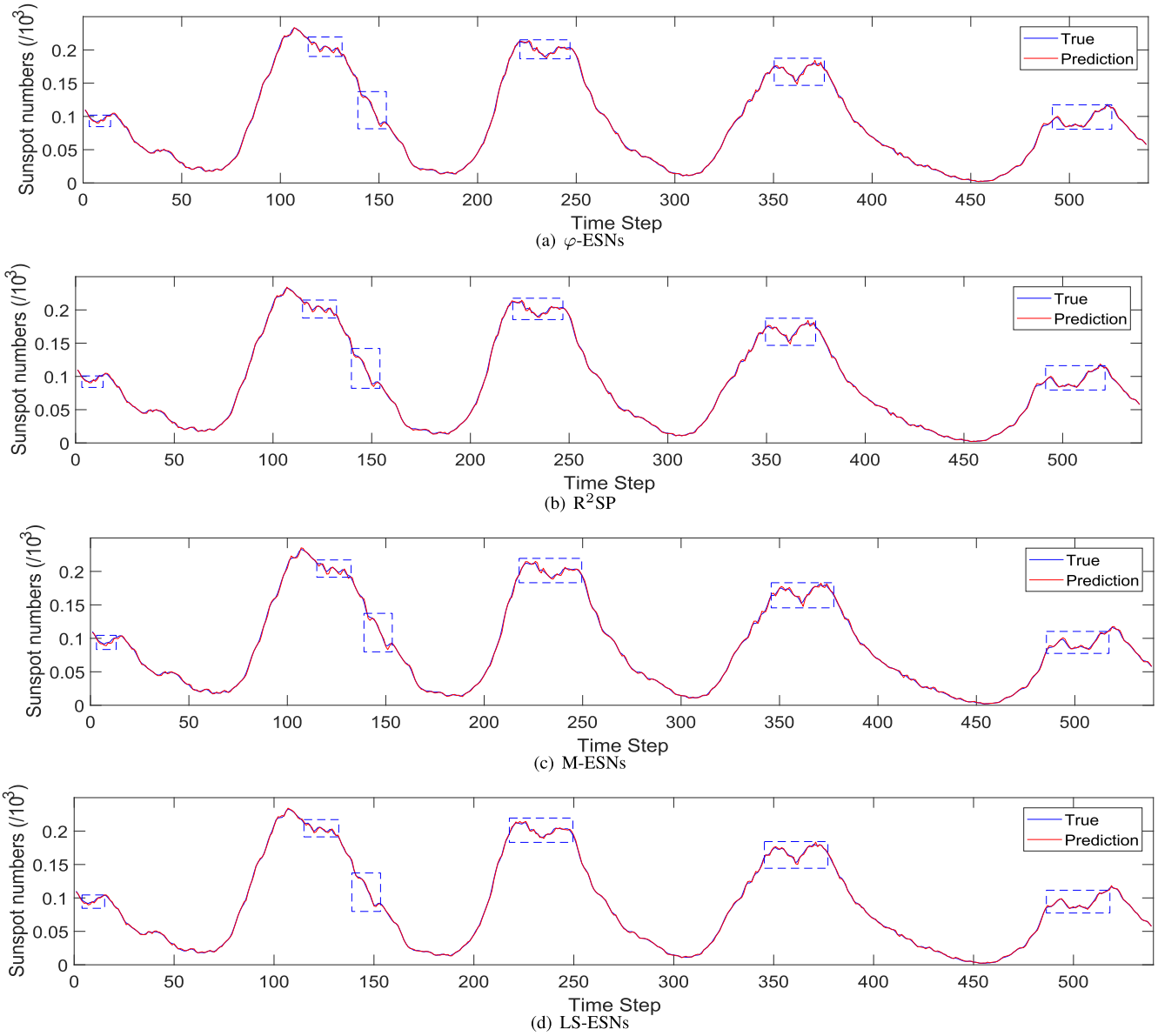


FIGURE 4. The plot of the prediction on the Monthly Sunspot data set. (a) One-step prediction of the φ-ESNs. (b) One-step prediction of the R^2SP. (c) One-step prediction of the M-ESNs. (d) One-step prediction of the proposed LS-ESNs.

TABLE 9. Average results of MSE with standard deviations of one-step-ahead prediction for real-world power load forecasting data sets.

NO.	SVR	FNN	RNN	LSTM	LS-ESNs
1	1.30E-02±0.00E-00	9.88E-03±2.50E-03	1.84E-02±3.26E-03	1.84E-02 ±5.93E-04	**5.91E-03±2.84E-02**
2	2.93E-02±0.00E-00	2.03E-02±3.22E-03	1.92E-02± 6.81E-04	1.71E-02±1.62E-04	**1.59E-02±1.58E-04**
3	3.98E-02±0.00E-00	1.54E-02± 4.06E-03	1.52E-02±1.42E-03	1.38E-02±3.17E-04	**6.01E-03±1.46E-04**
4	1.70E-02±0.00E-00	9.74E-03± 9.84E-03	1.55E-02±7.37E-04	1.34E-03±2.08E-04	**1.11E-03±7.53E-05**
5	2.43E-02±0.00E-00	**2.31E-02±9.25E-03**	4.74E-03±2.52E-04	4.55E-03±3.76E-04	**2.31E-03±4.82E-05**
6	1.52E-02±0.00E-00	5.82E-03±1.36E-03	6.02E-05±1.15E-02	1.12E-04±3.60E-03	**5.71E-05±9.51E-05**
7	1.14E-02±0.00E-00	6.28E-03±8.49E-03	7.51E-05±6.73E-05	9.43E-05±1.49E-05	**2.87E-05±5.04E-05**
8	1.61E-02±0.00E-00	**1.51E-02±4.56E-03**	1.72E-02±3.34E-04	1.75E-02±1.19E-03	1.63E-02±3.73E-04
9	2.63E-02±0.00E-00	1.64E-02±2.40E-03	6.94E-03±3.34E-04	6.71E-03±3.44E-04	**2.69E-03±1.37E-04**
10	3.22E-02±0.00E-00	1.71E-02±4.30E-03	1.60E-02±1.95E-03	1.58E-02±4.98E-04	**8.22E-03±1.31E-04**

Compared With ESN-Based Methods: As shown in Tables 5, 6 and 7, our proposed LS-ESNs outperform all the baselines again in terms of the lower metric of MSE, SMAPE, and NRMSE. The results further demonstrate the effectiveness of our proposed LS-ESNs and the benefit of the multi-scale temporal dependency features. To further

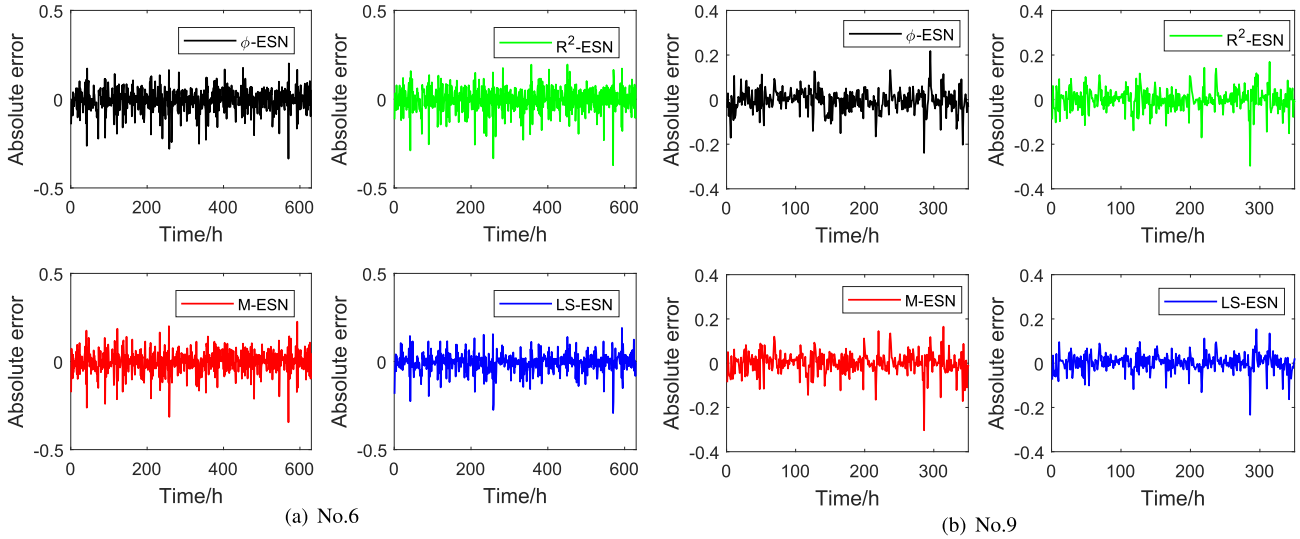


FIGURE 5. Absolute error plot on power load large customer data sets, The left plot is the absolute error curve of each model on the large customer No.6, and the right is the large customer No.9.

TABLE 10. Average results of SMAPE metric with standard deviations of one-step-ahead prediction for real-world power load forecasting data sets.

NO.	SVR	FNN	RNN	LSTM	LS-ESNs
1	6.34E-01±0.00E-00	4.40E-01±3.59E-02	1.02E+00±4.52E-02	1.05E+00±1.11E-01	**3.10E-01±8.91E-01**
2	4.64E-01±0.00E-00	3.37E-01±3.59E-02	**3.09E-01±1.09E-02**	5.69E-01±1.65E-03	3.32E-01±1.58E-01
3	6.80E-01±0.00E-00	6.05E-01± 5.00E-02	5.26E-01±3.76E-03	5.25E-01±5.61E-03	**1.76E-01±1.31E-03**
4	2.02E-01±0.00E-00	1.23E-01±4.54E-02	2.28E-01± 3.68E-03	2.22E-01±1.47E-03	**9.38E-02±3.24E-03**
5	1.02E+00±0.00E-00	1.04E+00±7.44E-02	9.76E-01±2.19E-02	9.67E-01±1.49E-02	**9.22E-01±1.55E-01**
6	5.16E-01±0.00E-00	3.45E-01±2.73E-02	9.87E-01±3.96E-02	9.83E-01±1.84E-02	**1.89E-01±4.75E-03**
7	1.87E+00±0.00E-00	1.45E+00 ±5.17E-01	**5.11E-01±2.13E-01**	9.98E-01±4.51E-02	1.21E+00±4.42E-02
8	8.06E-01±0.00E-00	5.50E-01±4.08E-02	5.27E-01±5.88E-03	5.11E-01±1.43E-02	**3.98E-01±2.17E-02**
9	9.53E-01±0.00E-00	9.05E-01±2.57E-02	8.23E-01±5.88E-03	8.22E-01±7.59E-03	**3.10E-01±1.13E-02**
10	6.42E-01 ±0.00E-00	6.30E-01±8.05E-02	5.22E-01±1.07E-02	5.16E-01±6.81E-03	**2.33E-01±3.62E-03**

TABLE 11. Average results of NRMSE metric with standard deviations of one-step-ahead prediction for real-world power load forecasting data sets.

NO.	SVR	FNN	RNN	LSTM	LS-ESNs
1	9.84E-01±0.00E-00	8.44E-01± 5.47E-02	8.01E-01±4.70E-02	7.97E-01±1.60E-02	**7.04E-01±6.09E-01**
2	**5.45E-01±0.00E-00**	7.49E-01±5.47E-02	7.51E-01±1.23E-02	7.11E-01±4.27E-02	6.49E-01±1.88E-03
3	9.88E-01±0.00E-00	5.95E-01±6.73E-02	6.25E-01±2.79E-02	5.98E-01±2.68E-02	**4.08E-01±3.49E-03**
4	6.32E-01±0.00E-00	4.69E-01±1.80E-01	4.66E-01±2.91E-03	**4.05E-01±2.54E-03**	4.61E-01±1.42E-02
5	7.66E-01±0.00E-00	8.43E-01±1.44E-01	3.65E-01 ±1.72E-02	3.61E-01±1.63E-02	**2.41E-01±1.77E-03**
6	5.83E-01±0.00E-00	3.55E-01± 2.81E-02	5.44E-01±2.13E-02	5.47E-01±1.35E-02	**2.86E-01±3.53E-03**
7	1.56E+00±0.00E-00	1.11E+00±5.17E-01	9.72E-01±3.75E-01	**6.04E-01±2.94E-01**	1.33E+00±4.34E-01
8	9.91E-01±0.00E-00	6.98E-01± 9.38E-02	**3.33E-01±1.52E-02**	7.18E-01±1.54E-02	6.76E-01±6.31E-03
9	6.88E-01±0.00E-00	5.62E-01±3.63E-02	3.33E-01±1.52E-02	3.29E-01±1.10E-02	**2.57E-01±4.33E-03**
10	6.13E-01±0.00E-00	6.90E-01± 5.19E-02	6.11E-01±2.33E-02	5.81E-01±6.20E-02	**5.22E-01±3.59E-03**

analyze the performance, we conduct a pairwise comparison for each ESN-based method against LS-ESNs. Specifically, we conduct the Nemenyi non-parametric statistical rank order test [44] on MSE, SMAPE, and NRMSE average rank. And the results are shown in Figure 6. Horizontal lines join the clustering methods that are not statistically significantly different. The critical difference is 2.850, which means that the two methods are not significantly different at the $p < 0.05$ level when their rank difference is less than 2.850. LS-ESNs achieve the best average ranks on all measures. LS-ESNs are significantly superior to M-ESNs at the $p < 0.05$ level on all measures and significantly better than R^2SP at the $p < 0.05$ level on NRMSE measure. Although LS-ESNs is not statistically significantly better than φ-ESNs, deepESN and MESM, it is numerically superior to them in the average rank of MSE, SMAPE, and NRMSE.

Compared With Traditional Methods: As shown in Tables 9, 10 and 11, LS-ESNs achieve nine bests on MSE, eight bests on SMAPE, six bests on NRMSE which are the best result among all comparison methods. Also, LS-ESNs achieves the best average rank. The results further demonstrate the effectiveness of our proposed LS-ESNs and the

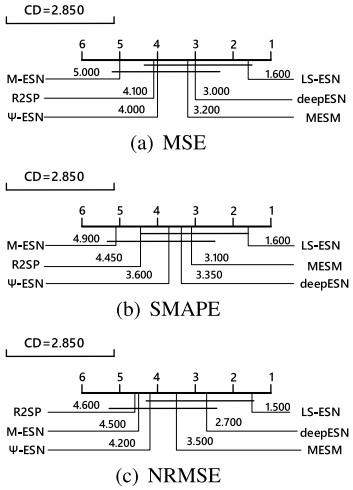


FIGURE 6. Critical difference diagram over the average rank of (a) MSE, (b) SMAPE, (c) NRMSE of LS-ESNs and five ESN-based methods on the power load data sets. The methods connected in one group are not significantly different at $p < 0.05$ significance level.

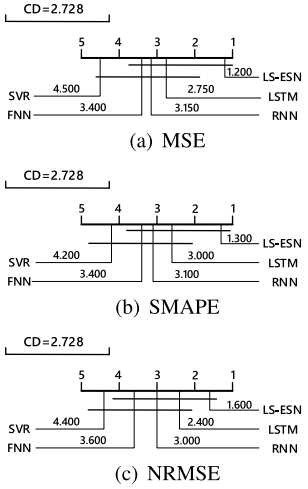


FIGURE 7. Critical difference diagram over the average rank of MSE(a), SMAPE(b), NRMSE(c) of LS-ESNs and five traditional methods on the power load data sets. The methods connected in one group are not significantly different at $p < 0.05$ significance level.

benefit of the multi-scale temporal dependency features. To further analyze the performance, we conduct a pairwise comparison for each traditional method against LS-ESN. Specifically, we conducted the Nemenyi non-parametric statistical rank order test [44] on MSE, SMAPE, and NRMSE average rank. And the results are shown in Figure 7. Horizontal lines join the clustering methods that are not statistically significantly different. The critical difference is 2.728, which means that the two methods are not significantly different at the $p < 0.05$ level when their rank difference is less than 2.728. LS-ESNs again achieve the best average ranks on all measure. LS-ESNs is significantly superior to SVR at the $p < 0.05$ level under all measures. Although LS-ESNs is not statistically significantly better than LSTM, RNN, FNN, it is numerically superior to them in the average rank of MSE, SMAPE, and NRMSE.

We randomly selected 2 large customers to plot the absolute error curve. As shown in Figure 5, our proposed method LS-ESNs achieve the best performance with minimum error.

TABLE 12. Memory capacity results (higher is better) achieved by deepESN, LS-ESNs and conventional ESNs.

Methods	MC score
deepESN	41.25
LS-ESNs(k=0, conventional ESNs)	39.60
LS-ESNs(k=2)	46.35
LS-ESNs(k=4)	51.29
LS-ESNs(k=8)	53.02
LS-ESNs(k=16)	51.17

D. MEMORY CAPACITY TASK

In this section, we will analyze the memory capability of LS-ESNs. This task provides a measure of the short-term memory capacity of RC networks by evaluating how well it is possible to echo delayed versions of the inputs [31]. The data used by the task is a univariate time series, and the value of each time setp is uniformly sampled from $[-0.8, 0.8]$. The task aims to reconstruct the history signals. For each time step t, we consider the target values $y_k(t) = u(t-k), k = 0, 1, \ldots, \infty$. The score of MC is defined as:

$$MC = \sum_{k=0}^{\infty} r^2(u(t-k), y_k(t)) \tag{25}$$

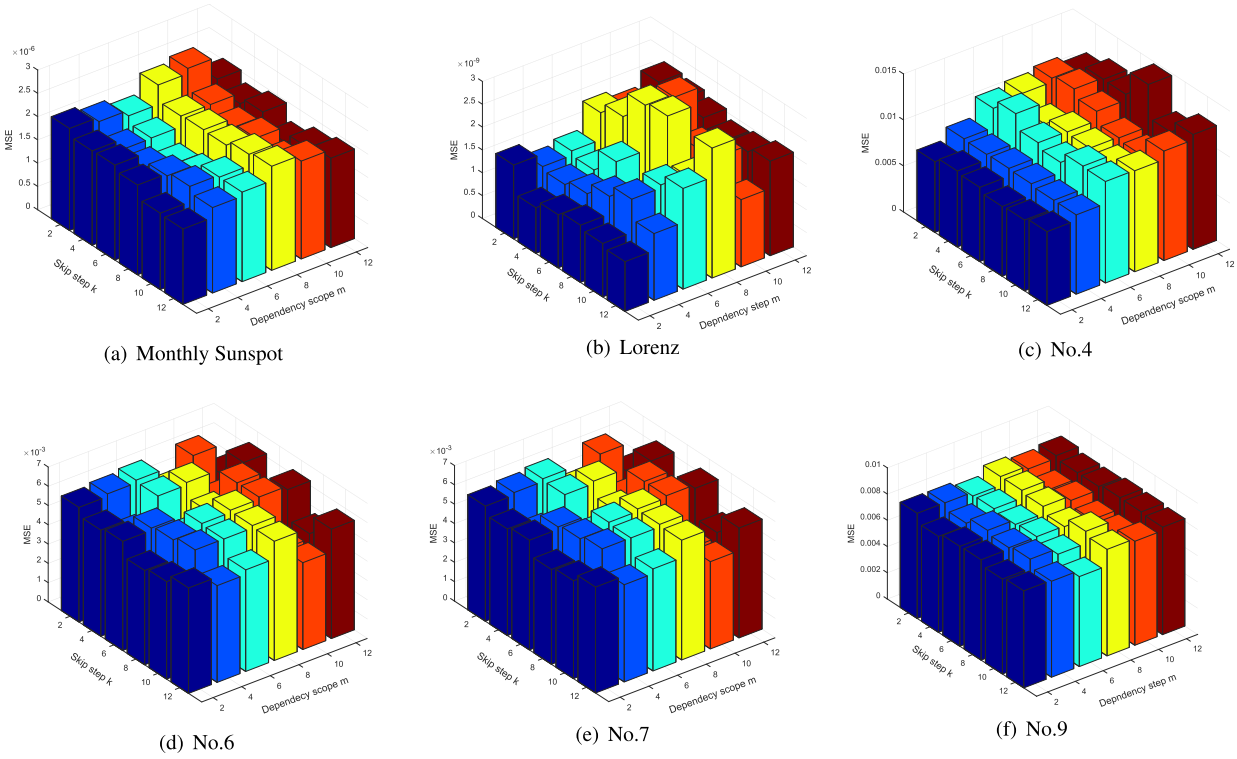
where $r^2(u(t-k), y_k(t))$ refer to the squared correlation coefficient between the input with delay k and the corresponding reconstructed value $y_k(t)$ (higher is better). In practice, the MC score can be computed by considering only a finite number of delayed signals.

In this paper, we set up an MC task similar to [31], by considering a number of delays equal to 200. The input signal contained 6000 steps, 5000 of which used for training, and the remaining 1000 for the test. We independently repeat the experiment ten times to report the mean MC score.

The MC score on the test set achieved by deepESN, LS-ESNs and conventional ESNs are reported in Table 12. As shown in Table 12, LS-ESNs achieves the best MC score among all the comparison methods, which indicates the skipping connections can enhance the memory ability of conventional ESN. In particular, LS-ESNs obtain the best MC value of 53.01 with an improvement of 28% compared to the value achieved by deepESN and 33% compared to the value achieved by conventional ESNs.

E. ABLATION STUDY

The long-term reservoir with the larger value of skip step k in LS-ESNs can capture the more long-term trends of time series, and the short-term reservoir can capture the more local patterns of time series, making LS-ESNs capable of capturing



(a) Monthly Sunspot (b) Lorenz (c) No.4

(d) No.6 (e) No.7 (f) No.9

FIGURE 8. The prediction performance of LS-ESNs for the skip step k and short dependency scope m (measured by the MSE).

multi-scale temporal dependency features. Hence, we remove one or both of long/short-term reservoirs and compare them with the full model LS-ESNs to verify the effect of each reservoir.

The ablation study results are shown in Figure 9. We performed an ablation study on the power load data of two randomly selected large factories, resulting in two sub-graphs. ESN represents the typical reservoir, ESN+Short represents the combination of the traditional ESNs and the short-term reservoir, ESN+Long represents the combination of the traditional ESNs and the long-term reservoir, and LS-ESNs represents the full model.

As shown in Figure 9, the best prediction performance is obtained by including long-term reservoir, typical reservoir, and short-term reservoir. Containing either the long-term or short-term reservoir can significantly improve the prediction performance. Compared with ESN+Short and ESN+Long, LS-ESNs achieve about 30% and 15% error reduction, indicating that the long-term temporal dependency feature is more important for time series prediction.

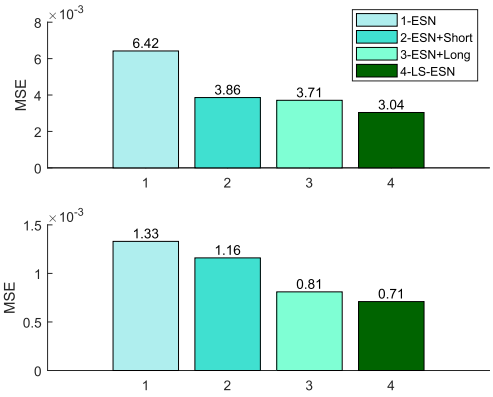


FIGURE 9. Ablation study on power load large customers data sets. The top plot is the ablation study on large customer No.2, and the bottom one is large customer No.6.

F. HYPER-PARAMETER ANALYSIS

Here, we provided the analysis on the hyper-parameter, skip step k, and short dependency scope m. As shown in Figure 3(d), The sunspot data set exhibits obvious long-term characteristics of periodicity and trend, which requires a large k to capture this long-term dependence. As shown in Figure 8(a), when the value of k is relatively large, a better MSE is obtained. However, increasing k does not always significantly help. Because when the value of k is too small, it is not enough to model long-term dependent features. If the value of k is too large, it may lead to ignoring too much information. Therefore, there is a relatively suitable value of k for each parameter setting. On the other hand, when the value of m is relatively small, LS-ESN obtains a better MSE, as shown in Figure 8(c). This is because No.4 (Figure 3(b)) shows a phenomenon of violent fluctuations, and there is no obvious long-term trend and periodicity. Thus, we need to model short-term features. The Lorenz dataset also exhibits

similar behavior. The other datasets in Figure 8 tend to have relatively large k and achieve lower MSE.

Summarizing the above analysis reveals the long-term reservoir is ideal for capturing long-scale temporal dependency features and short-term reservoir is suitable for capturing short-scale temporal dependency features. In addition, Figure 8 also illustrates that LS-ESN is sensitive to the skip step k and short dependency scope m to some extent.

V. CONCLUSION

In this paper, the long-short term echo state networks (LS-ESNs) are proposed to effectively capture multi-scale temporal features by the multiple independent reservoirs with different recurrent connections. The experiments on the two time series prediction benchmark data sets and a real-world power load data set demonstrate the effectiveness of the proposed LS-ESNs. Moreover, we analyzed the effect of each reservoir on the performance of the LS-ESNs and discussed the impact of skip step k and short dependency scope m.

LS-ESNs currently only can capture the multi-scale temporal dependency features of univariate time series and cannot model the relationship between the variables that exist in multivariate time series. The future work will consider enhancing LS-ESNs to model the multiple variable dependencies and apply them to multivariate time series prediction.

REFERENCES

[1] O. Anava, E. Hazan, and A. Zeevi, "Online time series prediction with missing data," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2191–2199.

[2] J. Zhao, Z. Han, W. Pedrycz, and W. Wang, "Granular model of long-term prediction for energy system in steel industry," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 388–400, Feb. 2016.

[3] J. Mei, M. Liu, Y.-F. Wang, and H. Gao, "Learning a mahalanobis distance-based dynamic time warping measure for multivariate time series classification," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1363–1374, Jun. 2016.

[4] J. Hensman, M. Rattray, and N. D. Lawrence, "Fast nonparametric clustering of structured time-series," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 383–393, Feb. 2015.

[5] H. Wang, M. Tang, Y. Park, and C. E. Priebe, "Locality statistics for anomaly detection in time series of graphs," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 703–717, Feb. 2014.

[6] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[7] C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004.

[8] P. Tino, C. Schittenkopf, and G. Dorffner, "Financial volatility trading using recurrent neural networks," *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 865–874, Jul. 2001.

[9] H. Zhang, Z. Wang, and D. Liu, "Robust stability analysis for interval Cohen–Grossberg neural networks with unknown time-varying delays," *IEEE Trans. Neural Netw.*, vol. 19, no. 11, pp. 1942–1955, Nov. 2008.

[10] H. Zhang, D. Liu, Y. Luo, and D. Wang, *Adaptive Dynamic Programming for Control: Algorithms and Stability*. Springer, 2012.

[11] S. Haykin and J. Principe, "Making sense of a complex world [chaotic events modeling]," *IEEE Signal Process. Mag.*, vol. 15, no. 3, pp. 66–81, May 1998.

[12] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[13] T. Lin, B. G. Horne, P. Tino, and C. L. Giles, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1329–1338, 1996.

[14] G. Deco and B. Schürmann, "Neural learning of chaotic dynamics," *Neural Process. Lett.*, vol. 2, no. 2, pp. 23–26, Mar. 1995.

[15] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks-with an erratum note," Germany Nat. Res. Center Inf. Technol., Bonn, Germany, GMD Tech. Rep. 148, 2001, no. 34, p. 13.

[16] M. Massar and S. Massar, "Mean-field theory of echo state networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 87, no. 4, Apr. 2013, Art. no. 042809.

[17] G. Wainrib and M. N. Galtier, "A local echo state property through the largest Lyapunov exponent," *Neural Netw.*, vol. 76, pp. 39–45, Apr. 2016.

[18] H. Cui, C. Feng, Y. Chai, R. P. Liu, and Y. Liu, "Effect of hybrid circle reservoir injected with wavelet-neurons on performance of echo state network," *Neural Netw.*, vol. 57, pp. 141–151, Sep. 2014.

[19] S. Yuenyong and A. Nishihara, "Evolutionary pre-training for CRJ-type reservoir of echo state networks," *Neurocomputing*, vol. 149, pp. 1324–1329, Feb. 2015.

[20] M. Lukoševičius, H. Jaeger, and B. Schrauwen, "Reservoir computing trends," *KI-Künstliche Intelligenz*, vol. 26, no. 4, pp. 365–371, Nov. 2012.

[21] D. Li, M. Han, and J. Wang, "Chaotic time series prediction based on a novel robust echo state network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 787–799, May 2012.

[22] Q. Song, X. Zhao, Z. Feng, Y. An, and B. Song, "Hourly electric load forecasting algorithm based on echo state neural network," in *Proc. Chin. Control Decis. Conf. (CCDC)*, May 2011, pp. 3893–3897.

[23] G. Li, B.-J. Li, X.-G. Yu, and C.-T. Cheng, "Echo state network with Bayesian regularization for forecasting short-term power production of small hydropower plants," *Energies*, vol. 8, no. 10, pp. 12228–12241, 2015.

[24] A. Deihimi, O. Orang, and H. Showkati, "Short-term electric load and temperature forecasting using wavelet echo state networks with neural reconstruction," *Energy*, vol. 57, pp. 382–401, Aug. 2013.

[25] S. P. Chatzis and Y. Demiris, "Echo state Gaussian process," *IEEE Trans. Neural Netw.*, vol. 22, no. 9, pp. 1435–1445, Sep. 2011.

[26] C. Gallicchio and A. Micheli, "Architectural and Markovian factors of echo state networks," *Neural Netw.*, vol. 24, no. 5, pp. 440–456, Jun. 2011.

[27] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Netw.*, vol. 61, pp. 32–48, Jan. 2015.

[28] J. Butcher, D. Verstraeten, B. Schrauwen, C. Day, and P. Haycock, "Extending reservoir computing with random static projections: A hybrid between extreme learning and RC," in *Proc. 18th Eur. Symp. Artif. Neural Netw. (ESANN)*, 2010, pp. 303–308.

[29] J. B. Butcher, D. Verstraeten, B. Schrauwen, C. R. Day, and P. W. Haycock, "Reservoir computing and extreme learning machines for non-linear time-series data analysis," *Neural Netw.*, vol. 38, pp. 76–89, Feb. 2013.

[30] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 190–198.

[31] C. Gallicchio, A. Micheli, and L. Pedrelli, "Design of deep echo state networks," *Neural Netw.*, vol. 108, pp. 33–47, Dec. 2018.

[32] Z. K. Malik, A. Hussain, and Q. J. Wu, "Multilayered echo state machine: A novel architecture and algorithm," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 946–959, Apr. 2017.

[33] Q. Ma, L. Shen, and G. W. Cottrell, "DeePr-ESN: A deep projection-encoding echo-state network," *Inf. Sci.*, vol. 511, pp. 152–171, Feb. 2020.

[34] S. E. Hihi, M. Q. Hc-J, and Y. Bengio, "Hierarchical recurrent neural networks for long-term dependencies," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 8, 199, pp. 493–499.

[35] Y. Li and Y. Zhao, "Recognizing emotions in speech using short-term and long-term features," in *Proc. Int. Conf. Spoken Lang. Process.*, 1998, pp. 1–4.

[36] Q. Ma, L. Shen, E. Chen, S. Tian, J. Wang, and G. W. Cottrell, "WALKING WALKing walking: Action recognition from action echoes," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2457–2463.

[37] R. A. Willoughby, "Solutions of Ill-posed problems (A. N. Tikhonov and V. Y. Arsenin)," *SIAM Rev.*, vol. 21, no. 2, pp. 266–267, 1979.

[38] R. J. Bray and R. E. Loughhead, *Sunspots*. London, U.K.: Chapman & Hall, 1964.

[39] E. N. Lorenz, "Deterministic nonperiodic flow," *J. Atmos. Sci.*, vol. 20, no. 2, pp. 130–141, 1963.

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[41] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 155–161.
[42] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Netw. (ICANN)*, 1999, pp. 850–855.
[43] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.
[44] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

KAIHONG ZHENG received the master's degree from Zhejiang University, in 2014.

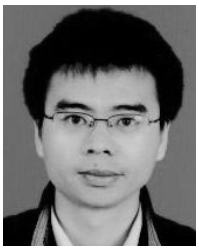
He is currently an Engineer with the Electric Power Research Institute, China Southern Power Grid. His research interests include electric energy measurement, electrical energy measurement automation systems, electricity information acquisition systems, and electricity technology.

BIN QIAN received the master's degree from the Huazhong University of Science and Technology, in 2014.

He is currently an Engineer with the Electric Power Research Institute, China Southern Power Grid. His research interests include electric energy measurement, electrical energy measurement automation systems, electricity information acquisition systems, and electricity technology.

SEN LI is currently pursuing the master's degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

His current research interests include machine learning and deep learning.



YONG XIAO received the Ph.D. degree from Wuhan University, in 2016.

He is currently an Engineer with the Electric Power Research Institute, China Southern Power Grid. His research interests include electric energy measurement, electrical energy measurement automation systems, electricity information acquisition systems, and electricity technology.

WANQING ZHUANG is currently pursuing the master's degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

His current research interests include machine learning and deep learning.

QIANLI MA (Member, IEEE) received the Ph.D. degree in computer science from the South China University of Technology, Guangzhou, China, in 2008.

From 2016 to 2017, he was a Visiting Scholar with the University of California at San Diego, La Jolla, CA, USA. He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology. His current research interests include machine-learning algorithms, data-mining methodologies, and time-series modeling and their applications.