HP-ESN: Echo State Networks Combined with Hodrick-Prescott Filter for Nonlinear Time-Series Prediction

1st Ziqiang Li
Dept. of Electrical Engineering and Information Systems
Graduate School of Engineering
The University of Tokyo
Tokyo 113-8656, Japan
ziqiang_li@sat.t.u-tokyo.ac.jp

2nd Gouhei Tanaka
Dept. of Electrical Engineering and Information Systems
Graduate School of Engineering
The University of Tokyo
Tokyo 113-8656, Japan
gouhei@sat.t.u-tokyo.ac.jp

***Abstract*—Nonlinear time-series prediction is one of the challenging tasks in machine learning. Recurrent neural networks and their variants have been successful in such a task owing to its ability of storing past inputs in their dynamical states. Echo state networks (ESNs) are a special type of recurrent neural networks, which are capable of high-speed learning. To develop this computational scheme, we propose an HP-ESN method which combines ESNs with a preprocessing based on the Hodrick-Prescott (HP) filter. This filter extracts different components from a single time-series data. The extracted components are processed by ESNs. We show that the proposed method yields better prediction performance compared with other state-of-the-art ESN-based methods in prediction tasks with real-world time-series data. We also demonstrate that the computational performance depends on the setting of the smoothing parameter and the number of decompositions by the HP filter.**

***Index Terms*—machine learning, reservoir computing, time-series forecasting, HP-ESN**

I. Introduction

As one of the classical machine learning tasks, nonlinear time-series prediction aims at leveraging previous time-series data to make predictions as close to true values in the future as possible. In many previous studies, auto-regressive integrated moving average (ARIMA) based models [1]–[3] were regarded as the pragmatic paragons since they simply rely on linear equations. However, for real-world time-series data with high nonlinearity and fluctuations, linear relationship extracted by ARIMA-based models is often not sufficient.

In the field of machine learning, recurrent neural networks (RNNs) [4] have been widely and successfully used for nonlinear time-series prediction. As a result of recurrent calculation of hidden states, an RNN is able to own "memory" ability which means that the hidden states reflect the past input information as well as the current input. Several RNN-based models such as long short term memory (LSTM) [5] and gated recurrent unit (GRU) [6] have been proposed to further enhance model's memory capacity and temporal feature extraction ability. However, these methods can encounter some problems like exploding and vanishing gradients [7] in a learning phase and often require large training costs.

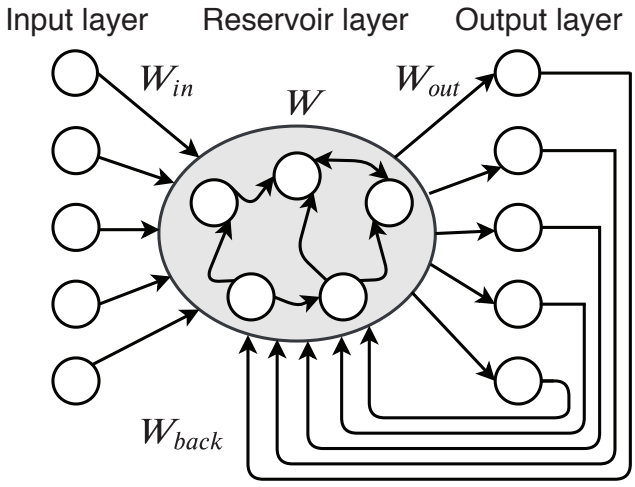


Fig. 1. The architecture of echo state network [9]

Reservoir computing (RC) [8] is a computational framework derived from special types of RNN models, which can avoid the above-mentioned problems in training RNNs. The echo state network (ESN) [9] is one of the representative models of reservoir computing as shown in Fig. 1. Three main parts, including an input layer, a reservoir layer, and an output layer, compose the basic architecture of ESN. The weight matrices, W_{in}, W and W_{out} are used to represent connection weights between input units and reservoir units, those between reservoir units themselves, and those between reservoir units and output units, respectively. The weight matrix W_{back} represents the weights on the feedback connections from the output layer to the reservoir layer. In the ESN, W_{in} and W are fixed and only W_{out} needs to be trained by a closed-form linear regression method, which not only reduces the learning cost compared with general RNNs but also simplifies the training process. In this paper, we focus on nonlinear time-series prediction by using ESN-based models.

Many ESN-based methods have been proposed to enhance

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

prediction accuracy in time-series forecasting tasks. The adaptive ESN [10] was proposed to adjust W_{out} by the online algorithm such that the prediction performance can be improved for non-stationary time-series data. The multi-step learning ESN [11] was proposed to reduce the prediction errors by using multiple ESN-based predictors and successively reducing the prediction errors. The Robust ESN [12] was proposed to leverage a Laplace likelihood function [13] in the readout layer instead of the standard linear regression. Even though those methods improve the prediction performance, their temporal feature extraction ability still has potentials to be enhanced. With the introduction of deep learning concepts into reservoir computing, ESN-based models with multiple reservoirs such as DeepESN [14], Deep-ESN [15], and Mod-deepesn [16] have been proposed. The performances of these deep-layer models outperform single-reservoir ESN models. However, the corresponding computational costs of deep reservoir models become larger when their architectures become deeper. All the above-mentioned methods follow the same direction in which the prediction performance is enhanced by strengthening the temporal feature extraction capability of the models themselves. This direction will make models become complex and poorly explainable.

In this work, we pay attention to another direction in which input data are preprocessed before being fed into the ESN. There are some existing methods in such a direction. For example, PCA-ESN [17] with a preprocessing by principal component analysis (PCA) was proposed for reducing input dimension before handling by an ESN. However, PCA will cause information loss which leads to inaccurate prediction. DBEN [18] using feature extraction by multi-layer restricted Boltzmann machines (RBMs) was proposed to enrich the information representation of the input data to the ESN. However, the training of DBN leads to extra training costs. WESN [19] using a wavelet transform method was proposed to separate original non-stationary signals into local spectral and temporal information. Nonetheless, the suitable scaling function is not easily determined depending on different kinds of time-series data.

As an important element of time-series processing technologies, time-series decomposition [20] has been widely used in the temporal data analysis. By the time-series decomposition, one complex time series can be divided into multiple time-series components. The Hodrick–Prescott filter (HP filter) [21] is one of the popular time-series decomposition methods, which decomposes a target time series into the trend and the cyclic components. In the HP filter, the sensitivity of the trend component to short-term fluctuations is simply adjusted by a smoothing parameter. These merits made the HP filter become a popular trend estimation method in the field of economics.

In this research, we propose a novel ESN-based method called the HP-ESN by combining the ESN with successive time series decompositions using the HP filter. In this hybrid model, the HP filter is adopted to disintegrate original time series into trend and cycle time-series components. The trend component is directly fed into an ESN for prediction, whereas the cyclic component is further decomposed by the HP filter. This process is carried out recursively. Each of the decomposed signals is used for prediction by an ESN and then the predicted results are integrated in the ensemble layer. The prediction results for two real-world time-series datasets, monthly sunspot series and daily minimum temperature in Melbourne, show that the prediction performances obtained by the proposed model are better than those of other ESN and deepESN based models. Further, we analyze the computational cost of the proposed model and show its high computational efficiency. Finally, we investigate the performances of the proposed model under variations in the number of decomposition processes, the value of smoothing parameters, and the size of reservoir.

The rest of this paper is organized as follows. The proposed method is described in Sec. II. The pseudocode and computational cost of the proposed method are provided in Sec. III. The details of numerical results are presented in Sec. IV. The discussion is provided in Sec. V. Conclusion and future works are given in Sec. VI.

II. Proposed method

Before introducing the proposed method, a schematic diagram of the proposed HP-ESN is shown in Fig. 2. This is the case where four time-series features are extracted through three decompositions using the HP filter. These extracted features are fed into four independent ESNs. The outputs of those ESNs are integrated in the ensemble layer to generate the final predicted time series. The three main parts in the proposed method, the HP filter, the ESN, and the ensemble layer, are described as below.

A. The HP filter

The HP filter is applied to a one-dimensional time-series data. In the proposed method, it is recursively used to decompose an original time-series data into multiple time-series components with different features. We assume that the target data at j-th decomposition ($j = 1, 2, \ldots, N_J$) is a one-dimensional time series denoted by

$$\mathbf{s}^{(j)} = [s^{(j)}(1), s^{(j)}(2), \ldots, s^{(j)}(N_T)]. \quad (1)$$

where N_T represents the length of data and $\mathbf{s}^{(1)}$ represents the original time-series data.

The target data at j-th decomposition is decomposed as follows:

$$\mathbf{s}^{(j)} = \mathbf{s}^{(j)}_{tre} + \mathbf{s}^{(j)}_{cyc}, \quad (2)$$

where the trend component is given by

$$\mathbf{s}^{(j)}_{tre} = [s^{(j)}_{tre}(1), s^{(j)}_{tre}(2), \ldots, s^{(j)}_{tre}(N_T)] \quad (3)$$

and the cyclic component is given by

$$\mathbf{s}^{(j)}_{cyc} = [s^{(j)}_{cyc}(1), s^{(j)}_{cyc}(2), \ldots, s^{(j)}_{cyc}(N_T)]. \quad (4)$$

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

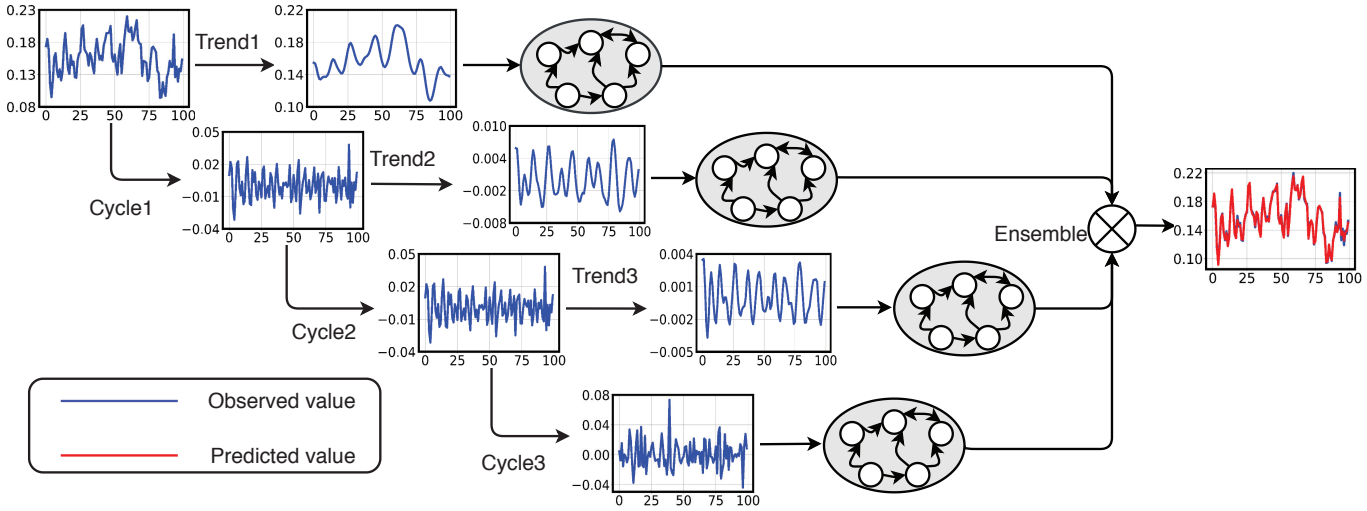


Fig. 2. An example of the proposed HP-ESN with 3 decompositions.

The trend component $\mathbf{s}_{tre}$ is fed into an ESN model, while the cyclic component $\mathbf{s}_{cyc}$ is recursively decomposed by the HP filter by setting

$$\mathbf{s}^{(j+1)} = \mathbf{s}^{(j)}_{cyc} \quad \text{for } j = 1, 2, \ldots, N_J - 1. \tag{5}$$

The HP filter is characterized by a minimization of the objective function given by

$$\begin{aligned} e &= \lambda \sum_{t=3}^{N_T} \left(s^{(j)}_{tre}(t) - 2s^{(j)}_{tre}(t-1) + s^{(j)}_{tre}(t-2) \right)^2 \\ &\quad + \sum_{t=1}^{N_T} \left(s^{(j)}(t) - s^{(j)}_{tre}(t) \right)^2, \end{aligned} \tag{6}$$

where λ is a smoothing parameter which takes a positive value. The unique optimal solution minimizing the above objective function can be obtained as follows:

$$(\mathbf{s}^{(j)}_{tre})^T = (I + \lambda E)^{-1} (\mathbf{s}^{(j)})^T, \tag{7}$$

where $I \in \mathbb{R}^{N_T \times N_T}$ is the identity matrix and $E \in \mathbb{R}^{N_T \times N_T}$ is the band matrix given by

$$\mathbf{E} = \begin{pmatrix} 1 & -2 & 1 & \ldots & 0 & 0 \\ -2 & 4+1 & -2-2 & \ldots & 0 & 0 \\ 1 & -2-2 & 1+4+1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1+4 & -2 \\ 0 & 0 & 0 & \ldots & -2 & 1 \end{pmatrix}.$$

From the N_J decompositions, we obtain N_J trend time-series components and one cyclic time-series component from the original one-dimensional time-series data. We denote the N_J+1 time-series components as follows:

$$\begin{bmatrix} \mathbf{u}^{(1)} \\ \mathbf{u}^{(2)} \\ \vdots \\ \mathbf{u}^{(N_J)} \\ \mathbf{u}^{(N_J+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{s}^{(1)}_{tre} \\ \mathbf{s}^{(2)}_{tre} \\ \vdots \\ \mathbf{s}^{(N_J)}_{tre} \\ \mathbf{s}^{(N_J)}_{cyc} \end{bmatrix} \in \mathbb{R}^{(N_J+1) \times N_T} \tag{8}$$

where $\mathbf{u}^{(k)} := \left[u^{(k)}(1), \ldots, u^{(k)}(N_T) \right]$ is the input time series to the k-th ESN for $k = 1, \ldots, N_J + 1$. The target time series for the k-th ESN can be represented as follows:

$$\mathbf{d}^{(k)} := \left[d^{(k)}(1), d^{(k)}(2), ..., d^{(k)}(N_T) \right]. \tag{9}$$

The target time-series data $\mathbf{d}^{(k)}$ is obtained by shifting $\mathbf{u}^{(k)}$ by one time step. To compute corresponding time-series prediction data, these generated data will be used for the processing with ESNs as described in the following subsection.

B. The ESN

Suppose that the N_R-dimensional internal state of the k-th ESN at the time t is defined as $\mathbf{x}^{(k)}(t) \in \mathbb{R}^{N_R}$. The internal state is updated as follows:

$$\mathbf{x}^{(k)}(t) = (1 - \alpha)\, \widehat{\mathbf{x}}^{(k)}(t) + \alpha \mathbf{x}^{(k)}(t-1), \tag{10a}$$

$$\widehat{\mathbf{x}}^{(k)}(t) = \tanh\left(\mathbf{W}^{(k)}_{in} \mathbf{u}^{(k)}(t) + \mathbf{W}^{(k)} \mathbf{x}^{(k)}(t-1) + \mathbf{b}^{(k)} \right), \tag{10b}$$

where the matrices $\mathbf{W}^{(k)}_{in} \in \mathbb{R}^{N_R \times N_U}$ and $\mathbf{W}^{(k)} \in \mathbb{R}^{N_R \times N_R}$ denote the input and internal connection weight matrices. The vector $\mathbf{b}^{(k)} \in \mathbb{R}^{N_R}$ represents the bias. The parameter α symbolizes the leaking rate which is used to control the updating speed of reservoir dynamics. The input matrix $\mathbf{W}^{(k)}_{in}$ is randomly assigned from the uniform distribution in the range of $[-1, 1]$, and re-scaled by the input scaling θ. The internal matrix $\mathbf{W}^{(k)}$ is randomly initialized from the uniform

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

distribution in the range of $[-1, 1]$, and then the fraction of non-zero elements of $\mathbf{W}^{(k)}$ is controlled by the density η. The reservoir is required to satisfy an asymptotic stability property called Echo State Property (ESP) [9]. In order to expect ESP, the internal weight is set to satisfy the following condition:

$$\rho\left((1-\alpha)\,\mathbf{I}+\alpha\mathbf{W}^{(k)}\right)<1, \tag{11}$$

where $\rho(\cdot)$ represents the spectral radius of a matrix argument and $\mathbf{I}\in\mathbb{R}^{N_R\times N_R}$ denotes the identity matrix. By this condition, the echo state network can obtain a short-term memory as the finely-tuned recurrent neural network.

In the readout layer, the output at time t can be computed by a linear operation as follows:

$$\mathbf{y}^{(k)}(t)=\mathbf{W}_{out}^{(k)}\mathbf{x}^{(k)}(t), \tag{12}$$

where $\mathbf{W}_{out}^{(k)}\in\mathbb{R}^{N_Y\times N_R}$ denotes the output weight matrix, $\mathbf{y}^{(k)}(t)\in\mathbb{R}^{N_Y}$ represents the output of the k-th ESN at time t, and N_Y is the dimension of output. In order to avoid an ill-conditioned problem, we introduce the Tikhonov regularization [22] to calculate the output weight matrix as follows:

$$\mathbf{W}_{out}^{(k)}=\mathbf{d}^{(k)}\left(\mathbf{X}^{(k)}\right)^{T}\left(\mathbf{X}^{(k)}\left(\mathbf{X}^{(k)}\right)^{T}+\beta\mathbf{I}\right)^{-1}, \tag{13}$$

where $\mathbf{X}^{(k)}=\left[\mathbf{x}^{(k)}(1),\mathbf{x}^{(k)}(2),...,\mathbf{x}^{(k)}(N_T)\right]\in\mathbb{R}^{N_R\times N_T}$. The regularization parameter β should be non-negative.

C. The Ensemble layer

Since each ESN only predicts a portion of final output, all the generated prediction will be ensembled to be the final output $\mathbf{y}$ in the ensemble layer as follows:

$$\mathbf{y}=\sum_{k=1}^{N_J+1}\mathbf{y}^{(k)}, \tag{14}$$

where $\mathbf{y}^{(k)}:=\left[y^{(k)}(1),y^{(k)}(2),...,y^{(k)}(N_T)\right]$. In Fig. 2, we can clearly find that there are three decompositions in the HP-ESN, and four generated time series are fed into the ensemble layer. This strategy ensures that each decomposed time-series data is extracted by each ESN separately and these independent predicted results are integrated to produce the final prediction.

III. Analysis

In this section, the pseudocode and the computational cost of the proposed HP-ESN are given.

A. Pseudocode

The pseudocode of the proposed HP-ESN is shown in Algorithm 1. It is obvious that the original time-series data $\mathbf{s}^{(1)}$ is decomposed by the HP filter recursively by the processes from 1 to 8. Each ESN is used for predicting corresponding decomposed time-series data as presented in the processes from 9 to 12. At the end, the predicted result is composed at process 13.

Algorithm 1 HP-ESN

Input: original time-series data $\mathbf{s}^{(1)}$, smoothing parameter λ, density of internal weights η, size of reservoir N_R, leaking rate α, regularization parameter β, spectral radius ρ, number of decompositions N_J.

Output: predicted time-series data $\mathbf{y}$

Initialisation : input weight $\mathbf{W}_{in}^{(k)}$, internal weight $\mathbf{W}^{(k)}$, parameter matrix $\mathbf{E}$.

1: **for** $j=1$ to N_J **do**
2: **if** $(j==1)$ **then**
3: $\mathbf{s}^{(j)}=\mathbf{s}^{(1)}$
4: **else**
5: $\mathbf{s}^{(j)}=\mathbf{s}_{cyc}^{(j-1)}$
6: **end if**
7: Applying the HP filter recursively descried in Sec. II-A
8: **end for**
9: **for** $k=1$ to N_J+1 **do**
10: Gathering input and target data for the k-th ESN.
11: Processing with the ESN descried in Sec. II-B
12: **end for**
13: $\mathbf{y}=\mathbf{y}^{(1)}+\mathbf{y}^{(2)}+,...,+\mathbf{y}^{(N_J+1)}$
14: **return** $\mathbf{y}$

B. Computational complexity

We suppose N_J decompositions and (N_J+1) ESNs in the proposed model. The computational complexity of each decomposition of HP filter, C_{HP}, can be formulated as follows:

$$C_{HP}=O(N_T). \tag{15}$$

The computational complexity of reservoir part in each ESN can be formulated as follows:

$$C_{RES}=O\left(N_TN_R+N_T(N_R)^2\right). \tag{16}$$

Since the readout part adopts the Tikhonov regularization for calculating output weights, the computational costs of calculating $\mathbf{d}^{(k)}\left(\mathbf{X}^{(k)}\right)^T$ and $\left(\mathbf{X}^{(k)}\left(\mathbf{X}^{(k)}\right)^T+\beta\mathbf{I}\right)^{-1}$ are $O(N_TN_R)$ and $O\left(N_T(N_R)^2+(N_R)^3\right)$, respectively. The calculation of multiplying $\mathbf{d}^{(k)}\left(\mathbf{X}^{(k)}\right)^T$ with $\left(\mathbf{X}^{(k)}\left(\mathbf{X}^{(k)}\right)^T+\beta\mathbf{I}\right)^{-1}$ costs $O\left((N_R)^2\right)$. Note that $N_R\ll N_T$. In summary, the total computational cost of the proposed HP-ESN can be summarized as follows:

$$\begin{aligned}C_{total}&=N_J(C_{HP})+(N_J+1)(C_{RES}+C_{REG})\\&\approx(N_J+1)(C_{RES}+C_{REG})\\&\approx O\left((N_J+1)\left(N_TN_R+N_T(N_R)^2\right)\right).\end{aligned} \tag{17}$$

From Eq. (17), we can see that the computational cost of ESN dominates that of the proposed HP-ESN. Because of the recursive processes, the computational cost of the proposed model is higher than those of ESN-based models [8], [9], [23]. Since each ESN is independent, the input dimension of each ESN in our proposed model is the same as the dimension

TABLE I
DATA PARTITION FOR MONTHLY SUNSPOT AND DAILY MINIMUM TEMPERATURE IN MELBOURNE

	Training	Validation	Testing	Washout
Monthly sunspot	2046	553	640	100
DMTM	2336	584	730	100

of $\mathbf{s}^{(1)}$ whereas those of some deepESN based model [14], [16] are enlarged to N_R from the second reservoir layer, which suggests that our proposed model is more time-efficient than some of deepESN-based models. In practice, each ESN is implemented parallelly following the HP filter, which can greatly enhance computational efficiency.

IV. NUMERICAL EXPERIMENTS

In this section, we will provide comprehensive experimental results about evaluating the proposed model on two real-world benchmark time-series prediction tasks: monthly sunspot prediction and daily minimum temperature prediction.

A. Dataset description

1) Monthly sunspot number series: Sunspots are dynamically formed by strong magnetic field on the sun surface. It has been widely demonstrated that the changes in the number of sunspots directly affect the climate of earth [24] and the series shows a high degree of non-linearity [25]. We collected monthly smoothed sunspot series from January, 1749 to November, 2019 provided by Sunspot Index and Long-term Solar Observations (SILSO)[1]. The time series is normalized by 1000 as shown in Fig. 3(a).

2) Daily minimum temperatures in Melbourne: It is well known that a temperature series can show chaotic behavior, which makes them difficult to be forecasted accurately. We adopt a benchmark temperatures dataset called Daily Minimum Temperatures in Melbourne (DMTM)[2] to evaluate the prediction performance of our proposed model. There are 3650 minimum temperature points in Melbourne collected from January 1st, 1981 to December 31th, 1990. The time series is normalized by 10 as shown in Fig. 3(b).

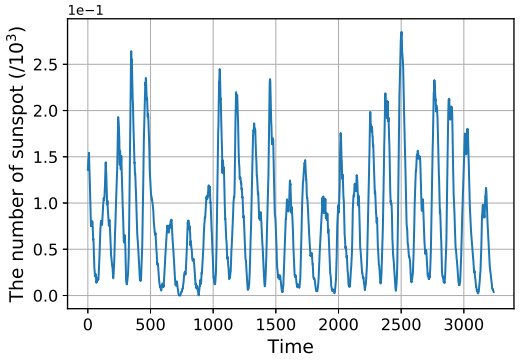
The partition of training set, validation set, testing set, and washout on the above-mentioned two time-series data are listed in Table I. For fair comparison, we kept the same split ratio for each dataset as reported in the study on DeepESN [15].

B. Evaluation metrics

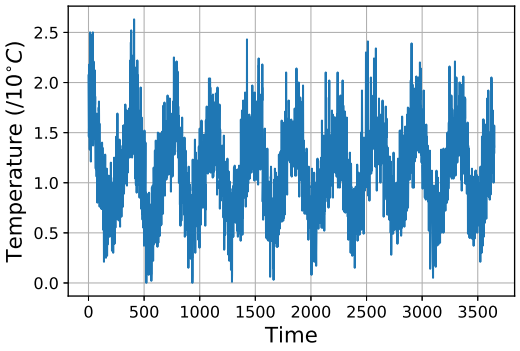
We evaluated our proposed model and all the compared models by using three metrics, the root mean square error (RMSE), the normalized root mean square error (NRMSE),

[1]Sunspot were downloaded from http://www.sidc.be/silso/datafiles.

[2]Data downloaded from https://www.kaggle.com/paulbrabban/daily-minimum-temperatures-in-melbourne.



(a) Monthly sunspot series



(b) Daily minimum temperature in Melbourne

Fig. 3. Two real-world time-series datasets.

and the mean absolute percentage error (MAPE). They are defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_T}\sum_{t=1}^{N_T}\left(\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right)^2}, \tag{18}$$

$$\mathrm{NRMSE} = \frac{\mathrm{RMSE}}{\sqrt{\frac{1}{N_T}\sum_{t=1}^{N_T}\left(\mathbf{y}\left(t\right) - \overline{\mathbf{y}}\right)^2}}, \tag{19}$$

$$\mathrm{MAPE} = \frac{1}{N_T}\sum_{t=1}^{N_T}\frac{|\mathbf{y}(t) - \hat{\mathbf{y}}(t)|}{\mathbf{y}(t)}, \tag{20}$$

where $\mathbf{y}\left(t\right)$ indicates the t-th observation value in the N_T-length label data, $\hat{\mathbf{y}}\left(t\right)$ represents the t-th predicted value in the N_T-length prediction data, and $\overline{\mathbf{y}}\left(t\right)$ denotes the mean value of the N_T-length label data. In order to avoid zero value in the data, a small bias 0.001 was added to the values of each data point.

C. Parameter settings

In the following experiments, the parameter setting of the HP-ESN is listed in Table II. The input scaling θ, spectral radius ρ, density of internal weights η, and regularizing factor β were set at 0.1, 0.95, 0.1, and 1e-6, respectively. For each dataset, we tested the reservoir size $N_R \in [100, 200, \ldots, 1000]$, the number of decompositions $N_J \in [1, 2, \ldots, 9]$, the leaking rate $\alpha \in [0.1, 0.2, \ldots, 1]$. The

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

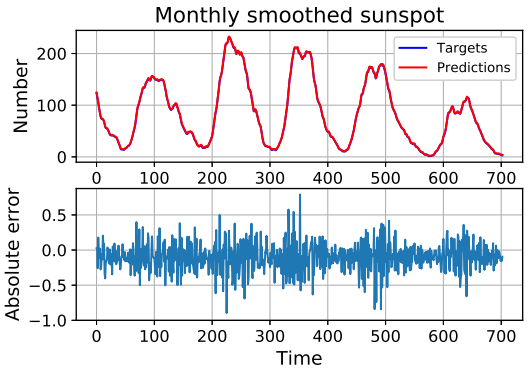
smoothing factor λ for each decomposition was searched empirically in the set of $[0.1, 1, 10, 50, 100, 200, 500, 1000, 1600]$. The grid search strategy was applied for finding the best parameter combination in the experiment. We repeated 20 independent trials for each parameter setting.

TABLE II
THE PARAMETER SETTINGS OF HP-ESN

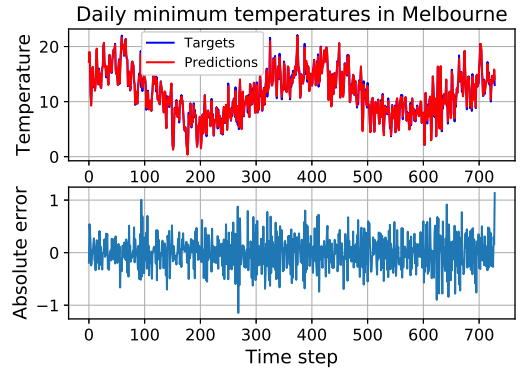
Parameters	Symbol	Value
Input scaling	θ	0.1
Density of internal weights	η	0.1
Spectral radius	ρ	0.95
Regularizing factor	β	1e-6
Leaking rate	α	$[0.1, 0.1, 1]$
Reservoir size	N_R	$[100, 100, 1000]$
Number of decompositions	N_J	$[1, 1, 9]$
Smoothing factor	λ	Listed in Set.IV-C

D. Simulation results

1) Monthly sunspot: The best prediction performance of our proposed model and those reported in [15] are listed in Table III. The best prediction performance of our proposed method was obtained under the condition that $N_J = 9$, $\alpha = 0.9$, $N_R = 1000$, and $\lambda = 10$. This is about five times more accurate than the performance of Deep-ESN using deep architecture. The predicted time series and corresponding absolute errors are shown in Fig. 4(a).



(a) Monthly smoothed sunspot



(b) Daily minimum temperature

Fig. 4. Prediction performance on monthly smoothed sunspot and daily minimum temperature in Melbourne.

2) Daily minimum temperatures in Melbourne: The best prediction performance of our proposed model and those reported in [15] are listed in Table IV. The best results of the proposed method were obtained under the parameter conditions: $N_J = 9$, $\alpha = 0.3$, $N_R = 1000$, and $\lambda = 10$. The predicted performance of the proposed method outperforms those of the compared models. The prediction time series and corresponding absolute errors are presented in Fig. 4(b). These simulation results illustrate the effectiveness our proposed model.

V. DISCUSSION

In order to make a thorough investigation about how hyper-parameters of our proposed method affect its prediction performance, we evaluated the prediction performance of the HP-ESN by changing the smoothing factors, the reservoir size, and the number of decompositions. In this experiment, the parameter settings for each dataset were inherited from the best case reported in Sec. IV-D.

Figures 5(a) and 5(b) show the prediction performances plotted against the reservoir size N_R for different values of the smoothing parameter λ in the tasks with the monthly sunspot and the daily temperature dataset, respectively. Based on the results shown in Fig. 5, the best prediction performances is obtained for smoothing parameter $\lambda = 10$. In addition, a larger reservoir size yields better prediction performances except the case of $\lambda = 0.1$ as shown in Fig.5(b). We tested the case of $\lambda = 1600$ which has been widely used in many literatures about financial analysis [29]–[31]. However, in our work, $\lambda = 1600$ yields the second worst prediction performance. Based on the above results, it can be concluded that the smoothing parameter has a big impact on the performance of our proposed model and should be carefully adjusted to obtain better prediction results.

Moreover, we investigated the effects of the number of decompositions in the proposed model. In this investigation, we focus on three different cases: N_R = 100, 500, and 1000. Figure 6 shows the prediction performance of the proposed model for different number of decompositions from 1 to 9 on the two time-series datasets. In Fig. 6(a), it is clearly found that RMSEs decay with an increase in the number of decompositions from 1 to 8 for the monthly sunspots data. However, when the number of decompositions is increased to 9, the RMSEs for $N_R = 100$ and $N_R = 500$ turn to increase whereas that for N_R=1000 continues to decrease. In Fig. 6(b), the RMSEs for $N_R = 100, 500$ and 1000 are monotonically decreasing as the number of decompositions increases.

In order to figure out the reason why the RMSEs for $N_R = 100$ and $N_R = 500$ are increased in Fig. 6(a) when the number of decompositions is increased to 9. We show the prediction performances of the trend and the cycle at different decomposition for training and test sets of two datasets in Fig. 7. There are only very small differences between training errors and test errors on the trend and cycle of two datasets, and therefore, it can be found that over-fitting does not occur. Also, in Fig. 7(a), we can find that all the three compared

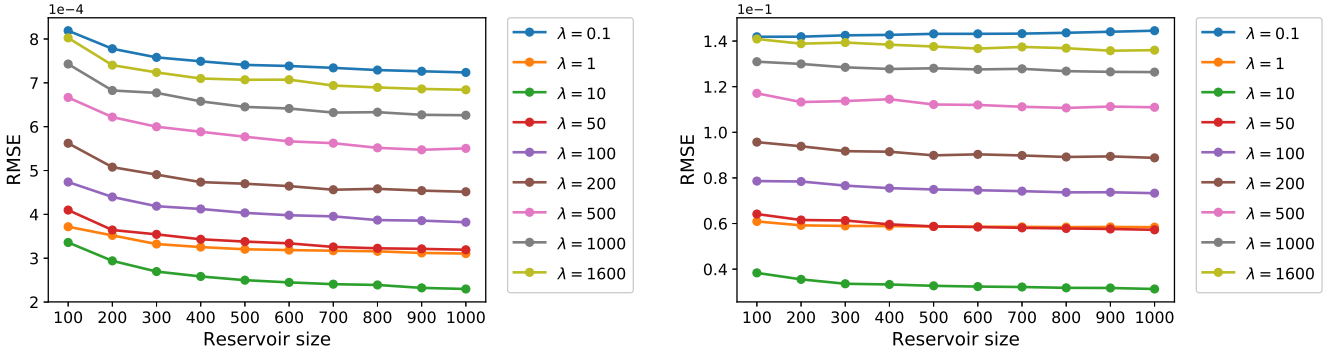
978-1-7281-6926-2/20/$31.00 ©2020 IEEE

TABLE III
COMPARISON OF AVERAGE RESULTS ON THE ONE-STEP-AHEAD PREDICTION FOR MONTHLY SUNSPOT SERIES.

Models	RMSE	NRMSE	MAPE	Layers
ESN [23]	1.30E-03 ± (7.43E-06)	2.08E-02 ± (1.16E-04)	4.96E-03 ± (8.74E-06)	1
φ-ESN [26]	1.25E-03 ± (2.28E-05)	1.93E-02 ± (3.51E-04)	4.77E-03 ± (4.00E-05)	2
R^2 SP [27]	1.27E-03 ± (2.44E-05)	1.98E-02 ± (3.81E-04)	4.98E-03 ± (1.23E-04)	2
MESN [28]	1.26E-03 ± (3.08E-05)	1.94E-02 ± (4.72E-04)	4.87E-03 ± (9.15E-05)	3
Deep-ESN [15]	1.22E-03 ± (1.24E-09)	1.87E-02 ± (1.89E-04)	4.76E-03 ± (2.83E-05)	3
HP-ESN (best)	**2.29E-04 ± (2.73E-06)**	**3.62E-03 ± (4.31E-05)**	**4.11E-03 ± (6.91E-05)**	1

TABLE IV
COMPARISON OF AVERAGE RESULTS ON THE ONE-STEP-AHEAD PREDICTION FOR DAILY MINIMUM TEMPERATURE IN MELBOURNE.

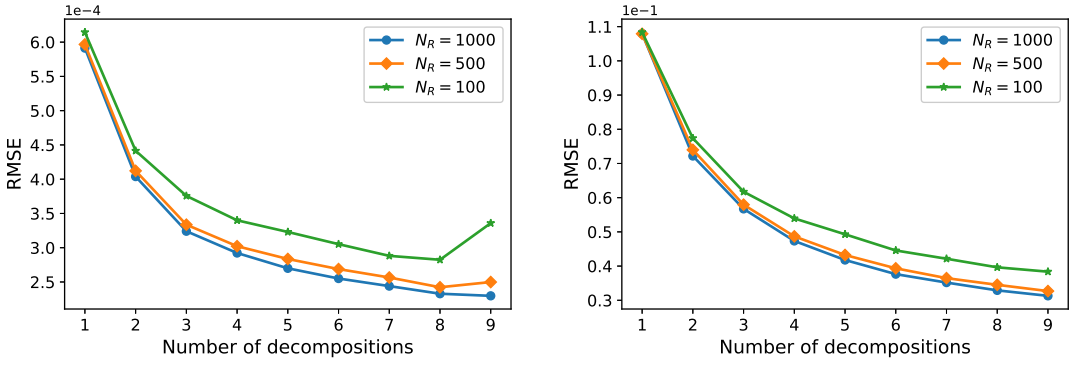
Models	RMSE	NRMSE	MAPE	Layers
ESN [23]	5.01E-01 ± (3.70E-03)	1.39E-01 ± (1.02E-03)	3.95E-02 ± (2.37E-04)	1
φ-ESN [26]	4.93E-01 ± (3.86E-03)	1.41E-01 ± (1.10E-03)	3.96E-02 ± (3.74E-04)	2
R^2 SP [27]	4.95E-01 ± (3.55E-03)	1.37E-01 ± (9.82E-04)	3.93E-02 ± (4.34E-04)	2
MESN [28]	4.78E-01 ± (3.39E-03)	1.36E-01 ± (9.67E-04)	3.77E-02 ± (3.36E-04)	2
Deep-ESN [15]	4.73E-01 ± (2.77E-03)	1.35E-01 ± (7.91E-04)	3.70E-02 ± (2.14E-04)	2
Mod-deepesn [16]	4.59E-01 ± (-)	1.32E-01 ± (-)	3.71E-02 ± (-)	4
HP-ESN (best)	**3.31E-02 ± (4.66E-03)**	**7.62E-02 ± (1.13E-03)**	**2.52E-02 ± (4.72E-04)**	1



(a) The case of monthly sunspots

(b) The case of daily minimum temperature

Fig. 5. The prediction performance of nine decompositions under different smoothing parameters λ and reservoir size N_R on two different real-world time-series datasets.



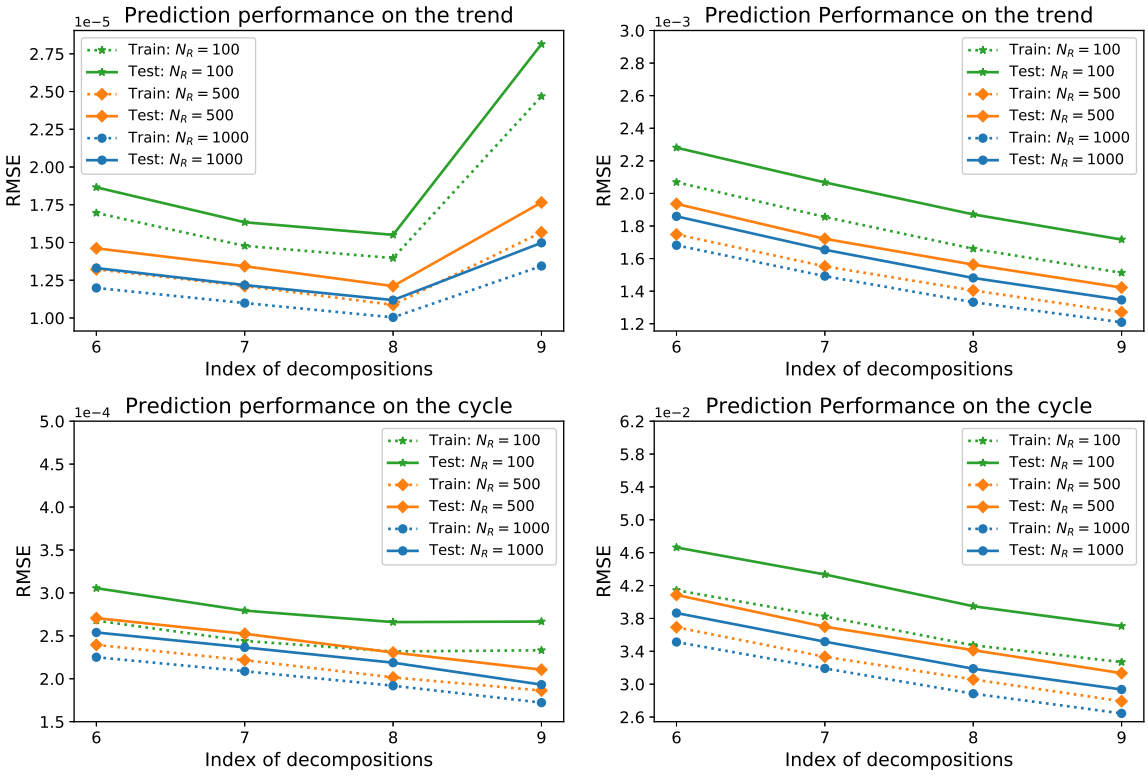
(a) Monthly sunspots

(b) Daily minimum temperature

Fig. 6. The prediction performance of proposed method with reservoir size $N_R = 100, 500$ and 1000 by changing different number of decompositions from 1 to 9.

cases show similar trends with a change in the number of decompositions for the monthly sunspots dataset. Figure 8 presents the visualization of the trends generated from the 8-th decomposition and the 9-th decomposition. We can find that the trend at the 9-th decomposition is less stable than that at the 8-th decomposition, which would increase the difficulty

978-1-7281-6926-2/20/$31.00 ©2020 IEEE



(a) Monthly sunspots (b) Daily minimum temperature

Fig. 7. The training and test errors of the proposed method with the case of reservoir size $N_R = 100$, $N_R = 500$ and $N_R = 1000$ by changing the number of decompositions from 6 to 9.

in forecasting accurately. In general, an ESN with a larger reservoir size has a better temporal feature extraction ability. Therefore, the reason for the increase of RMSEs for the cases of $N_R = 100$ and $N_R = 500$ shown in Fig. 6(a) is concluded that the size of reservoir is not sufficiently large. Extending to the case of our proposed model, we should appropriately increase the size of the reservoir for better performance.

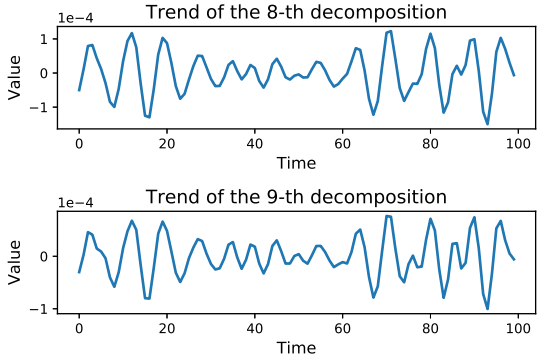


Fig. 8. Visualization of the trends of the 8-th decomposition (upper) and the 9-th decomposition (lower) in the HP-ESN with 9 decompositions.

VI. Conclusion

In this paper, a novel hybrid ESN-based method, HP-ESN, has been proposed for nonlinear time-series prediction tasks. The prediction performance of the proposed model evaluated for the two real-world time-series datasets, monthly smoothed sunspot and daily minimum temperatures in Melbourne, has shown its effectiveness. Our analysis of the computational cost has shown that the proposed model has lower computational costs than some of deepESN-based models. The investigation of the effects of various hyper-parameters, such as the smoothing parameter, the number of decompositions and the reservoir size on the computational performance, have shown the importance of finding a suitable value of the smoothing parameter and adding more decompositions with large reservoir size, which can improve the prediction performance.

We will continually study how to process multi-variate time-series prediction tasks by the proposed model. Further, effects of the other hyper-parameters on the performance of the HP-ESN should be evaluated, such as the spectral radius and the leaking rate.

Acknowledgements

This work was partially supported by JST-Mirai Program Grant Number JPMJMI19B1, Japan (GT) and partially based on results obtained from a project (No. 18102285-0) subsidized by the New Energy and Industrial Technology Development Organization (NEDO) (GT).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

REFERENCES

[1] D. Ö. Faruk, "A Hybrid Neural Network and ARIMA Model for Water Quality Time Series Prediction," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 586–594, 2010.

[2] O. Valenzuela, I. Rojas, F. Rojas, H. Pomares, L. J. Herrera, A. Guillén, L. Marquez, and M. Pasadas, "Hybridization of Intelligent Techniques and ARIMA Models for Time Series Prediction," *Fuzzy Sets and Systems*, vol. 159, no. 7, pp. 821–845, 2008.

[3] D. Zeng, J. Xu, J. Gu, L. Liu, and G. Xu, "Short Term Traffic Flow Prediction Using Hybrid ARIMA and ANN Models," in *2008 Workshop on Power Electronics and Intelligent Transportation System*. IEEE, 2008, pp. 621–625.

[4] J. L. Elman, "Finding Structure in Time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[5] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[7] R. Grosse, "Lecture 15: Exploding and Vanishing Gradients," *University of Toronto Computer Science*, 2017.

[8] H. Jaeger and H. Haas, "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[9] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13, 2001.

[10] H. Jaeger, "Adaptive nonlinear system identification with echo state networks," in *Advances in Neural Information Processing Systems*, 2003, pp. 609–616.

[11] T. Akiyama and G. Tanaka, "Analysis on Characteristics of Multi-step Learning Echo State Networks for Nonlinear Time Series Prediction," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[12] D. Li, M. Han, and J. Wang, "Chaotic Time Series Prediction Based on a Novel Robust Echo State Network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 5, pp. 787–799, 2012.

[13] L. Tierney, R. E. Kass, and J. B. Kadane, "Fully Exponential Laplace Approximations to Expectations and Variances of Nonpositive Functions," *Journal of the American Statistical Association*, vol. 84, no. 407, pp. 710–716, 1989.

[14] C. Gallicchio, A. Micheli, and L. Pedrelli, "Deep Reservoir Computing: A Critical Experimental Analysis," *Neurocomputing*, vol. 268, pp. 87–99, 2017.

[15] Q. Ma, L. Shen, and G. W. Cottrell, "Deep-esn: A Multiple Projection-encoding Hierarchical Reservoir Computing Framework," *arXiv preprint arXiv:1711.05255*, 2017.

[16] Z. Carmichael, H. Syed, S. Burtner, and D. Kudithipudi, "Mod-DeepESN: Modular Deep Echo State Network," *arXiv preprint arXiv:1808.00523*, 2018.

[17] F. M. Bianchi, E. De Santis, A. Rizzi, and A. Sadeghian, "Short-term Electric Load Forecasting Using Echo State Networks and PCA Decomposition," *IEEE Access*, vol. 3, pp. 1931–1943, 2015.

[18] X. Sun, T. Li, Q. Li, Y. Huang, and Y. Li, "Deep Belief Echo-state Network and Its Application to Time Series Prediction," *Knowledge-Based Systems*, vol. 130, pp. 17–29, 2017.

[19] 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, 2013.

[20] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The Empirical Mode Decomposition and The Hilbert Spectrum for Nonlinear and Non-stationary Time Series Analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[21] R. J. Hodrick and E. C. Prescott, "Postwar US Business Cycles: An Empirical Investigation," *Journal of Money, Credit, and Banking*, pp. 1–16, 1997.

[22] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[23] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and Applications of Echo State Networks with Leaky-integrator Neurons," *Neural Networks*, vol. 20, no. 3, pp. 335–352, 2007.

[24] B. Geerts and E. Linacre, "Sunspots and Climate," Website, 1997, http://www-das.uwyo.edu/ geerts/cwx/notes/chap02/sunspots.html.

[25] V. Suyal, A. Prasad, and H. P. Singh, "Nonlinear Time Series Analysis of Sunspot Data," *Solar Physics*, vol. 260, no. 2, pp. 441–449, 2009.

[26] C. Gallicchio and A. Micheli, "Architectural and Markovian Factors of Echo State Networks," *Neural Networks*, vol. 24, no. 5, pp. 440–456, 2011.

[27] 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 Networks*, vol. 38, pp. 76–89, 2013.

[28] Z. K. Malik, A. Hussain, and Q. J. Wu, "Multilayered Echo State Machine: A Novel Architecture and Algorithm," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 946–959, 2016.

[29] H. Ahumada and M. L. Garegnani, "Hodrick-Prescott Filter in Practice," in *IV Jornadas de Economía Monetaria e Internacional (La Plata, 1999)*, 1999.

[30] A. Harvey and T. Trimbur, "Trend Estimation and the Hodrick-Prescott Filter," *Journal of the Japan Statistical Society*, vol. 38, no. 1, pp. 41–49, 2008.

[31] J. D. Hamilton, "Why You Should Never Use the Hodrick-Prescott Filter," *Review of Economics and Statistics*, vol. 100, no. 5, pp. 831–843, 2018.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE