Deep Echo State Networks with Multi-Span Features 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 topics in machine learning due to complex non-stationarity in the temporal dynamics. Many recurrent neural network models have been proposed for enhancing the prediction accuracy in time-series prediction tasks. Echo state networks (ESNs) are a variant of recurrent neural networks, which have great potential for addressing machine learning tasks with a very low learning cost. However, the existing ESN-based models have used only single-span features to our best knowledge. In this study, we propose two deep ESN models incorporating multi-span features to improve the prediction performance. We show that the two deep ESN models yield better prediction performance compared to the other state-of-the-art ESN-based methods in benchmark time-series prediction tasks with three models: the Lorenz system, the Mackey-Glass system, and the NARMA-10 system. Our analyses illustrate that deeper structures decrease the multicollinearity of the extracted features and thus contribute to improved performance. The presented results suggest that the proposed models contribute to the development of artificial intelligence for temporal information processing.**

***Index Terms*—machine learning, nonlinear time-series prediction, reservoir computing, deep echo state networks**

I. Introduction

Nonlinear time-series prediction [1] is one of the classical prediction tasks, which aims to predict the future of a nonlinear dynamical system from a given temporal data generated by the system. Since the Recurrent Neural Network (RNN) [2] demonstrated its outstanding ability in time-series prediction tasks, many RNN-based methods such as Long Short-Term Memory (LSTM) [3] and Gated Recurrent Unit (GRU) [4] have been proposed to deal with nonlinear time-series prediction. However, as the exploding gradient problem often occurs in the training process of the above mentioned RNNs, stable prediction performance cannot be easily ensured [5]. In addition, the Back Propagation Through Time (BPTT) algorithm [6] used for RNN training depends on long-term memory information, which is computationally expensive. Reservoir Computing (RC) [7]–[9] is a special framework of RNNs. As in the classical RNN framework, the RC model is composed of three parts: the input layer, the inner (reservoir) layer, and the readout layer. The merit of RC is that only the readout layer needs to be trained, whereas the input and the inner weights are fixed all the time. As a representative of RC models, Echo State Network (ESN) [10] has been widely studied for temporal data prediction and classification [11], [12] as well as time-series signal reconstruction [13], [14]. In this work, we focus on ESN-based methods for nonlinear time-series prediction tasks.

Many methods [15]–[18] based on standard architectures of ESN have so far been proposed for enhancing the prediction accuracy on nonlinear time-series prediction tasks. However, due to the limited representation ability caused by the single reservoir architecture of ESN, the prediction performance of the improved models was also limited.

With the development of deep learning [19], the concept of *stacking architecture* [20] has been introduced into RC models. Deep RC was first introduced in [21] where a deep RC model called DeepESN with stacking multiple ESNs was proposed. In [22], several ESN-unsupervised encoders are connected serially and the prediction performance is enhanced by reducing redundancy in the features. Mod-DeepESN proposed in [23] employs *wide* topology for enhancing richness of the features in the deep RC model.

However, since the current deep RC models (e.g. Deep-ESN, Deep-ESN, Mod-DeepESN) extract only single time-span states in each RC layer. For example, the responses to

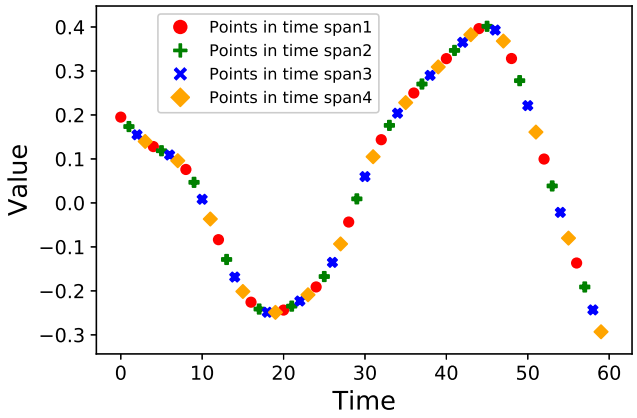


Fig. 1. Four different time-span points for the Macket-Glass system with $\varphi = 17$.

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

the t-th input and the $(t+1)$-th input are used for training without any changes in their order. This restriction inhibits leveraging multi-span states (e.g. the states sampled for different time spans.) which have been demonstrated to improve the prediction ability [24], [25]. Fig. 1 is a schematic diagram showing four different data point sequences corresponding to multi-span features for the Mackey-Glass system with $\varphi = 17$. The point sequences with different marks are sampled for four different types of time spans.

Therefore, to facilitate prediction performance using multi-span features, we develop deep ESN models as follows.

1) We provide two novel deep RC architectures: Deep Multi-Span ESN and Deep Multi-Temporal ESN. These two models can extract various time-span features by dividing input series of single time-span into that of various time-span. The final prediction results demonstrate that concatenated multi-span features in the proposed deep RC models can effectively enhance prediction performance for nonlinear and chaotic time series.
2) Through analyzing complexity of the two proposed models, we show that they have the same magnitude of computational complexity as the previous deep RC models. This ensures computational efficiency of our models.
3) We demonstrate that our models yield excellent prediction performance on the three benchmark nonlinear time-series prediction tasks with the Lorenz system, the Mackey-glass system, and the NARMA-10 system. In addition, we present the effect of the reservoir size, the number of layers, and the number of spans on the performance of the two proposed models. Moreover, we compare the multi-collinearity of the features in the two proposed models with that in the DeepESN model.

The rest parts of our paper are organized as follows. Section II contains the detailed description of the two proposed models. The analysis of computational complexity will be shown in Sec. III. The experimental results and analysis will be presented in Sec. IV. Discussion will be given in Sec. V. The conclusion will be provided in Sec. VI.

II. TWO ARCHITECTURES

A. Deep Multi-Span ESN

Before introducing our proposed architectures, we define the original input sequential data matrix $\mathbf{U}$ of length N_T, and the corresponding target sequential data matrix $\mathbf{Y}$ of length N_T as follows:

$$\mathbf{U} = \{\mathbf{u}(t=1), \mathbf{u}(t=2), ..., \mathbf{u}(t=N_T)\} \in \mathbb{R}^{N_U \times N_T}, \tag{1a}$$

$$\mathbf{Y} = \{\mathbf{y}(t=1), \mathbf{y}(t=2), ..., \mathbf{y}(t=N_T)\} \in \mathbb{R}^{N_Y \times N_T}, \tag{1b}$$

where N_U and N_Y are the dimensions of input and target data, respectively. t represents the time index of original sequential data. Our aim is to construct a machine learning model for supervised learning with the above data.

1) Group-wise reservoir layer: We propose ESN-based models with a serial combination of multiple reservoir layers as shown in Fig. 2. In this figure, the input data is given to the bottom nodes and the network output is obtained at the top nodes. The model states are expanded in time and the horizontal direction corresponds to the time axis. The time points are divided into several groups for extracting multi-span features.

Suppose that the input data at the time t in the g-th group of the l-th reservoir layer is defined as $\mathbf{u}_{(g)}^{(l)}(t) \in \mathbb{R}^{N_U}$, and the internal state at the time t corresponding to the g-th group in the l-th reservoir layer is denoted by $\mathbf{x}_{(g)}^{(l)}(t) \in \mathbb{R}^{N_R}$. Then, the internal state is updated as follows:

$$\widetilde{\mathbf{x}}_{(g)}^{(l)}(t) = \tanh\left(\mathbf{W}_{in(g)}^{(l)}\mathbf{u}_{(g)}^{(l)}(t) + \mathbf{W}_{(g)}^{(l)}\mathbf{x}_{(g)}^{(l)}(t-1)\right), \tag{2a}$$

$$\mathbf{x}_{(g)}^{(l)}(t) = \left(1-\alpha_{(g)}^{(l)}\right)\mathbf{x}_{(g)}^{(l)}(t-1) + \alpha_{(g)}^{(l)}\widetilde{\mathbf{x}}_{(g)}^{(l)}(t), \tag{2b}$$

where the parameter $\alpha_{(g)}^{(l)}$ denotes the leaking rate of the reservoir of the g-th group in the l-th layer, which controls the updating speed of reservoir dynamics. The matrices $\mathbf{W}_{in(g)}^{(l)} \in \mathbb{R}^{N_R \times N_U}$ and $\mathbf{W}_{(g)}^{(l)} \in \mathbb{R}^{N_R \times N_R}$ represent the input weight matrix and inner weight matrix of reservoir corresponding to the g-th group in the l-th layer, respectively. Typically, the elements of $\mathbf{W}_{in(g)}^{(l)}$ are generated randomly from the uniform distribution in the range of $[-1, 1]$. In order to expect the Echo State Property (ESP) [10] which is a requirement for the reservoir, the inner weights $\mathbf{W}_{(g)}^{(l)}$ should satisfy the following condition:

$$\max_{\substack{1\le l\le N_L \\ 1\le g\le N_G}} \rho\left(\left(1-a_{(g)}^{(l)}\right)\mathbf{E} + a_{(g)}^{(l)}\mathbf{W}_{(g)}^{(l)}\right) < 1, \tag{3}$$

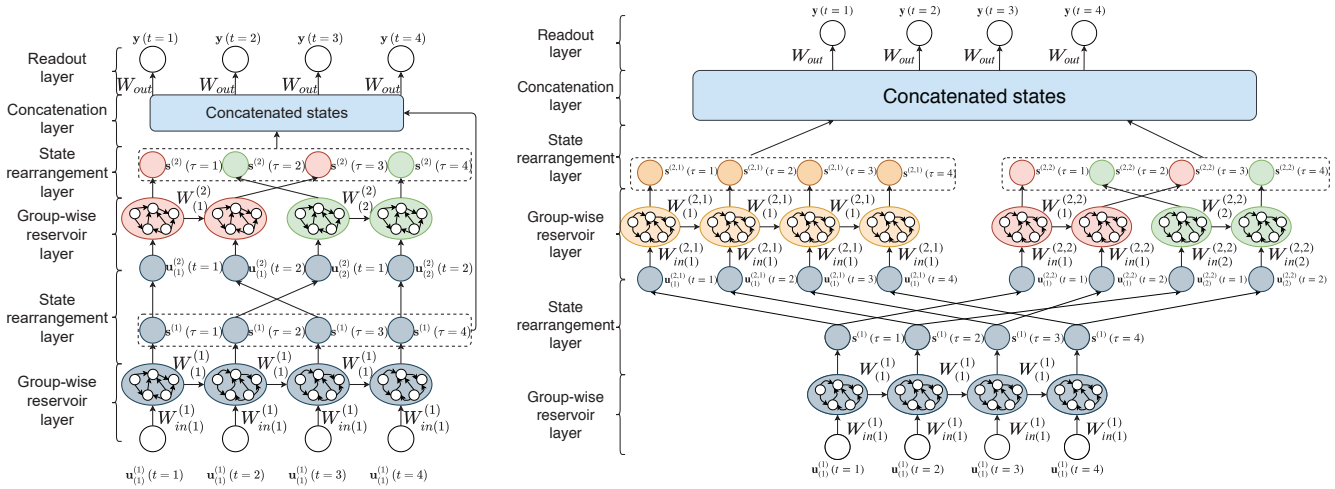
where $\rho(\cdot)$ represents the spectral radius of a matrix argument and $\mathbf{E} \in \mathbb{R}^{N_R \times N_R}$ denotes the identity matrix. With inequality (3), the reservoir can obtain "memory" ability like some traditional RNN models. Note that the ESP can be realized in the reservoir even when the left-hand side of inequality (3) equals to 1 or slightly larger than 1 in practice [26].

2) State rearrangement layer: A state rearrangement layer following the l-th reservoir layer is introduced to reorganize the reservoir states generated in the previous RC layer. In each state rearrangement layer, the states of the l-th reservoir layer are divided into $G^{(l)}$ groups, and the value of $G^{(l)}$ is given as follows:

$$G^{(l)} = 2^{l-1}. \tag{4}$$

Figure 2(a) shows an example of the two-layer Deep Multi-Span ESN where $G^{(1)} = 1$ and $G^{(2)} = 2$. There are two important processes in the state rearrangement layer: collection and distribution. Here, we define the τ-th time step state of the l-th state rearrangement layer as $\mathbf{s}^{(l)}(\tau)$. In the collection process, the states of the g-th reservoir group in the l-th state rearrangement layer $\mathbf{s}^{(l)}$ are collected as follows:

$$\mathbf{x}_{(g)}^{(l)}(t) \rightarrow \mathbf{s}^{(l)}(\tau), \tag{5}$$



(a) Deep Multi-Span ESN

(b) Deep Multi-Temporal ESN

Fig. 2. (a): A two-layer Deep Multi-Span ESN, (b) A two-layer Deep Multi-Temporal ESN.

where $\tau = \left((t-1)\cdot G^{(l)}+1\right)+(g-1) \bmod G^{(l)}$ and "$\rightarrow$" symbolizes the collection process. In the distribution process, $\mathbf{s}^{(l)}(\tau)$ is distributed to the g-th group of the $(l+1)$-th reservoir layer as input $\mathbf{u}_{(g)}^{(l+1)}(t)$, which can be formulated as follows:

$$\mathbf{s}^{(l)}(\tau) \rightarrow \mathbf{u}_{(g)}^{(l+1)}(t), \quad \text{where} \quad \begin{cases} g = (\tau-1) \bmod G^{(l+1)} + 1 \\ t = (\tau-1)/G^{(l+1)} + 1 \end{cases}. \tag{6}$$

Note that the group number of the first reservoir layer is $G^{(1)} = 1$ and $\mathbf{u}_{(1)}^{(1)}(t) = \mathbf{u}(t)$. The state rearrangement layer in Fig. 2(a) shows the collection and distribution processes between two adjacent reservoir layers. Through the state rearrangement layer, Deep Multi-Span ESN can capture different time-span features. Our experiments demonstrate that the multi-span features facilitate the performance in nonlinear time-series prediction tasks, as described in Sec. IV.

3) Concatenation layer: In the concatenation layer, all the states corresponding to the τ-th time-step, collected in the state rearrangement layer, are vertically concatenated together. Hence, the concatenated state vector of τ-th time-step, symbolized as $\mathbf{s}(\tau)$, is represented as follows:

$$\mathbf{s}(\tau) = \left\{\mathbf{s}^{(1)}(\tau); \mathbf{s}^{(2)}(\tau); ...; \mathbf{s}^{(N_L)}(\tau)\right\} \in \mathbb{R}^{N_L N_R}. \tag{7}$$

Unlike Deep-ESN and Mod-DeepESN, the original inputs are not contained in the concatenated state $\mathbf{s}(\tau)$ since we found that the original inputs have no benefits in our proposed models.

4) Readout layer: For the sake of low-computational cost, we inherited the basic idea of the readout layer of Deep-ESN [22] and employed the closed-form linear-regression for computing output weights $W_{out} \in \mathbb{R}^{N_Y \times N_L N_R}$. However, since various time-span features are collected as a high-dimensional concatenated state matrix, it unavoidably leads to an increase of its *multicollinearity* which would greatly affect the prediction performance. Therefore, we leverage the ridge regression with Tikhonov regularization [27] for computing W_{out} as follows:

$$W_{out} = \mathbf{Y}\mathbf{S}^{\mathbf{T}}\left(\mathbf{S}\mathbf{S}^{\mathbf{T}} + \lambda\mathbf{E}\right)^{-1}, \tag{8}$$

where $\mathbf{S} = \{\mathbf{s}(\tau=1), \mathbf{s}(\tau=2), ..., \mathbf{s}(\tau=N_T)\} \in \mathbb{R}^{N_L N_R \times N_T}$. The parameter λ symbolizes the Tikhonov regularization factor. The matrix $\mathbf{E} \in \mathbb{R}^{N_L N_R \times N_L N_R}$ denotes the identity matrix. A study [28] reported that the calculation of $\mathbf{S}^{\mathbf{T}}\left(\mathbf{S}\mathbf{S}^{\mathbf{T}} + \lambda\mathbf{E}\right)^{-1}$ with the singular value decomposition (SVD) [29] will be beneficial for improving the accuracy of regression. Therefore, we leverage SVD for computing W_{out} in the whole experiments.

B. A Variation: Deep Multi-Temporal ESN

Figure 2(b) shows a modified model of Deep Multi-Span ESN, called Deep Multi-Temporal ESN. It is obvious that Deep Multi-Temporal ESN uses similarly the group-wise reservoir layer proposed in Sec. II-A1.

However, unlike the Deep Multi-Span ESN, various time-span features are extracted in the last group-wise reservoir layer. We slightly modified the state rearrangement layer proposed in Sec. II-A2. The states collected from the (N_L-1)-th reservoir layer are copied as D duplicated distributions. The input to the g-th group corresponding to the d-th duplicated states in the last reservoir layer is represented as $\mathbf{u}_{(g)}^{(N_L,d)}(t)$. The d-th duplication of states are divided into $G^{(d)} = 2^{d-1}$ groups. The distribution process of the (N_L-1)-th state rearrangement layer can be re-formulated as follows:

$$\mathbf{s}^{(N_L-1)}(\tau) \rightarrow \mathbf{u}_{(g)}^{(N_L,d)}(t), \quad \text{where} \quad \begin{cases} g = (\tau-1) \bmod G^{(d)} + 1 \\ t = (\tau-1)/G^{(d)} + 1 \end{cases}. \tag{9}$$

In the collection process of the last state rearrangement layer, the states corresponding to the d-th duplication $\mathbf{s}_{(g)}^{(N_L,d)}(\tau)$ are collected as follows:

$$\mathbf{x}_{(g)}^{(N_L,d)}(t) \rightarrow \mathbf{s}^{(N_L,d)}(\tau), \tag{10}$$

where $\tau = \left((t-1)\cdot G^{(d)}+1\right)+(g-1) \bmod G^{(d)}$. Further, for the concatenation layer of Deep Multi-Temporal ESN, only the states in the last state rearrangement layer are vertically concatenated, which can be formulated as follows:

$$\mathbf{s}(\tau) = \left\{\mathbf{s}^{(N_L,1)}(\tau); \mathbf{s}^{(N_L,2)}(\tau); ...; \mathbf{s}^{(N_L,D)}(\tau)\right\} \in \mathbb{R}^{DN_R}. \tag{11}$$

The readout layer of Deep Multi-Temporal ESN is the same as that of Deep Multi-Span ESN described in Sec. II-A4. Note that the feature diversity of Deep Multi-Temporal ESN only relies on the number of duplications D. Therefore, unlike Deep Multi-Span ESN, various temporal features can be extracted at the last reservoir layer in the total model.

III. ANALYSIS

A. Computational Complexity

In this section, analyses of computational complexity for Deep Multi-Span ESN and Deep Multi-Temporal ESN are given.

We assume that there are N_L reservoir layers in the two proposed deep models. The size of each group-wise reservoir is N_R and the time length of N_U-dimensional input data is N_T. Hence, the computational cost for the group-wise reservoir layer is given as follows:

$$C_{res}^{(l)} = O\left(N_T N_R N_U / G^{(l)} + N_T \left(N_R\right)^2 / G^{(l)}\right). \tag{12}$$

There is a loop in distribution and collection processes in the state rearrangement layer. Therefore, the computational cost for the state rearrangement layer can be represented as $C_{rea} = O(2N_T)$. The computational cost for the concatenation layer is given by $C_{con} = O(N_T)$. In the readout layer, the computational cost for calculating $\mathbf{YS}^T$ and $\left(\mathbf{SS^T}+\lambda\mathbf{I}\right)^{-1}$ are $O(N_Y N_R N_L N_T)$ and $O\left(\left(N_R N_L\right)^2 N_T + \left(N_R N_L\right)^3\right)$, respectively. Multiplying $\mathbf{YS}^T$ with $\left(\mathbf{SS^T}+\lambda\mathbf{I}\right)^{-1}$ costs $O\left(N_Y \left(N_R N_L\right)^2\right)$. Note that $N_U \ll N_R$, $N_Y \ll N_R$, and $N_L \ll N_R$. Therefore, the total computational complexity of Deep Multi-Span ESN can be formulated as follows:

$$\begin{aligned} C_{total} &= \sum_{l=1}^{N_L} G^{(l)} C_{res}^{(l)} + N_L C_{rea} + C_{con} + C_{reg} \\ &\approx O\left(\sum_{l=1}^{N_L}\left(N_T \left(N_R\right)^2\right) + N_T \left(N_R N_L\right)^2\right) \\ &\approx O\left(N_L N_T \left(N_R\right)^2\right). \end{aligned} \tag{13}$$

For Deep Multi-Temporal ESN, the computational cost for the first (N_L-1) reservoir layers is $C_{res} = (N_L-1)(N_T\left(N_R\right)^2 + N_T N_R N_U)$. The computational cost for the d-th group-wise reservoir in the last reservoir layer is given as follows:

$$C_{res}^{(N_L)}(d) = O\left(N_T N_R N_U / G^{(d)} + N_T \left(N_R\right)^2 / G^{(d)}\right). \tag{14}$$

The computational costs for the state rearrangement layer, the concatenation layer, and the readout layer are the same as those of Deep Multi-Span ESN. In summary, the total cost of Deep Multi-Temporal ESN can be formulated as follows:

$$\begin{aligned} C_{total} &= C_{res} + \sum_{d=1}^{D} G^{(d)} C_{res}^{(N_L)}(d) + N_L C_{rea} + C_{con} + C_{reg} \\ &\approx O\left(\left(N_L + D - 1\right)\left(N_T \left(N_R\right)^2 + N_T N_R N_U\right)\right) \\ &\approx O\left(\left(N_L + D - 1\right) N_T \left(N_R\right)^2\right). \end{aligned} \tag{15}$$

From Eq. (13) and Eq. (15), multiplying reservoir states with inner weights dominates the total computational complexity in the Deep Multi-Span ESN and Deep Mutli-Temporal ESN. Note that computational complexities given by Eq. (13) and Eq. (15) have the same magnitude as those reported for Deep-ESN [21] and Deep-ESN [22], which indicates our proposed models can provide multi-temporal features without extra computational complexity added. Moreover, the calculations of the group states in the last layer of Deep Multi-Temporal ESN are mutually independent, and therefore, parallel computation can be applied for deceasing the operation time in practice.

B. Relationship with Existing ESN Models

Although our proposed models are able to extract multi-span features from input sequences, there is a very close relationship with existing ESN models. Deep Multi-Temporal ESN can be transformed into the DeepESN when D is fixed at 1. Deep Multi-Temporal ESN with $D = 1$ and $N_L = 1$ and Deep Multi-Span ESN with $N_L = 1$ are reduced to the standard ESN.

On the other hand, since the basic element in each group-wise reservoir layer of the proposed models is the standard ESN, the other improvement method for ESNs can be used for boosting the prediction performance of our proposed two models as well (see Sect.IV-D2).

IV. NUMERICAL EXPERIMENTS

A. Task Description

Benchmark prediction tasks were conducted for evaluating our proposed models using the Lorenz system , the Mackey-Glass system, and the 10-th order nonlinear auto-regressive moving average (NARMA-10) system.

1) Lorenz system: Lorenz system [30] is a chaotic system of ordinary differential equations which can be represented as follows:

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma(y-x), \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= x(\rho-z)-y, \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy-\beta z, \end{aligned} \tag{16}$$

TABLE I
DATA PARTITION FOR LORENZ SYSTEM, MGS-17, AND NARMA-10 TASKS

	Training	Validation	Testing	Washout
Lorenz	3000	1000	1000	300
MGS-17	6400	1600	2000	300
NARMA-10	2560	640	800	120

TABLE II
THE PARAMETER SETTINGS OF DEEP MULTI-SPAN ESN, DEEP MULTI-TEMPORAL ESN, AND DEEPESN

Parameters	Symbol	Value
Input scaling	θ	0.1
Leaking rate	$\alpha_{(g)}^{(l)}$	0.9
Density of inner weights	η	0.1
Regularizing factor	λ	1e-10
Reservoir size	N_R	$[100, 100, 1000]$
Spectral radius	ρ	$[0.60, 0.05, 1.10]$

with the standard parameter setting: $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$. The initial state was set at $(x(0), y(0), z(0)) = (12, 2, 9)$. The time-series data were collected with sampling interval $\Delta t = 0.02$ and re-scaled by scaling factor 0.1. In this task, we employed the x values to predict six-step-ahead y values, which can be represented as $\mathbf{u}(t) = x(t)$ and $\mathbf{y}(t) = y(t+6)$.

2) Mackey-Glass system: The Mackey-Glass System (MGS) [31] can be represented as follows:

$$y(t+1) = y(t) + \delta \cdot \left(a \frac{y(t-\varphi/\delta)}{1 + y(t-\varphi/\delta)^n} - by(t) \right), \quad (17)$$

where a, b, n, and δ are fixed at 0.2, -0.1, 10, and 0.1, respectively. The MGS shows chaotic behavior when $\varphi > 16.8$. We followed the setting in previous works [7], [22], [32] and tested our models by setting $\varphi = 17$ (MGS-17). The task is to predict 84-step-ahead time-series data, which can be represented as $\mathbf{u}(t) = y(t)$ and $\mathbf{y}(t) = y(t+84)$.

3) NARMA-10 system: The NARMA-n system [33] is a nonlinear dynamical system which is represented as follows:

$$y(t+1) = \alpha \cdot y(t) + \beta \cdot y(t) \cdot \sum_{i=1}^{n} y(t-i) + \gamma \cdot u(t-n) \cdot u(t) + \sigma, \quad (18)$$

where n, α, β, γ, and σ are fixed at 10, 0.3, 0.05, 1.5, and 0.1, respectively. The input $u(t)$ is sampled randomly from uniform distribution in the range of $[0, 0.5]$. This challenging prediction task is widely used for evaluating RC models since long temporal dependency and rich randomness are included in the time sequence. In this task, we set $\mathbf{u}(t) = u(t)$ and $\mathbf{y}(t) = y(t)$.

The partition of training set, validation set, testing set, and washout on the above three time-series data are listed in Table I. For fair comparison, the partition of training set, validation set, and testing set is the same as that in [22], [23].

B. Evaluation Metrics

Three metrics, root-mean-square error (RMSE), normalized root mean square error (NRMSE), and mean absolute percentage error (MAPE) were leveraged in the experiments. They are defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_T} \sum_{\tau=1}^{N_T} (\mathbf{y}(t) - \hat{\mathbf{y}}(t))^2}, \quad (19)$$

$$\mathrm{NRMSE} = \frac{\mathrm{RMSE}}{\sqrt{\frac{1}{N_T} \sum_{t=1}^{N_T} (\mathbf{y}(t) - \overline{\mathbf{y}})^2}}, \quad (20)$$

$$\mathrm{MAPE} = \frac{1}{N_T} \sum_{t=1}^{N_T} \frac{|\mathbf{y}(t) - \hat{\mathbf{y}}(t)|}{\mathbf{y}(t)}, \quad (21)$$

where $\mathbf{y}(t)$ indicates the t-th observation value in the N_T-length label data, $\hat{\mathbf{y}}(t)$ represents the t-th observation value in the N_T-length prediction data, and $\overline{\mathbf{y}}(t)$ denotes the mean value of the N_T-length target data.

C. Simulation Setting

In the following experiments, the parameter setting of Deep Multi-Span ESN, Deep Multi-Temporal ESN, and the baseline counterpart DeepESN were set as listed in Table II. The input scaling for input data θ, the leaking rate $\alpha_{(g)}^{(l)}$, the density of inner weights η, and the Tikhonov regularization factor λ are fixed at 0.1, 0.9, 0.1, and 1e-10, respectively. The reservoir size was varied in the range of $[100, 1000]$ with the interval of 100. The spectral radius of inner weights was adjusted to be in the range of $[0.60, 1.10]$ with the interval of 0.05. We repeated 20 independent trials for each parameter setting.

For simulations, Pytorch 1.10 was employed for implementing the proposed Deep Multi-Span ESN and Deep Multi-Temporal ESN. The module of ridge regression in Scikit-learn 0.21.3 was used for implementing the readout layer of the two proposed models.

D. Results

1) Lorenz system: In this section, we compare the prediction performance of our proposed methods with that of DeepESN [21]. Table III shows the prediction performance of the three models under the best settings: $N_R = 900$, $\rho = 0.95$, and $D = 2$. Our proposed methods outperform the baseline DeepESN and in particular Deep Multi-Temporal ESN produces the best performance. Further, the corresponding predicted time series and the absolute error of Deep Multi-Temporal ESN under the best settings are shown in Fig. 3(a).

2) MGS-17: In this section, we compare the prediction performance of our proposed methods with those reported in [22]. The research [37] reported that Intrinsic Plasticity (IP) [38] can improve the prediction results of RC model on MGS-17 by maximizing output information of each reservoir neuron. This effective way of information transformation relies on minimizing the KL-divergence between Gaussian-distribution

TABLE III
COMPARISON OF AVERAGE RESULTS ON THE SIX-STEP-AHEAD PREDICTION FOR LORENZ SYSTEM.

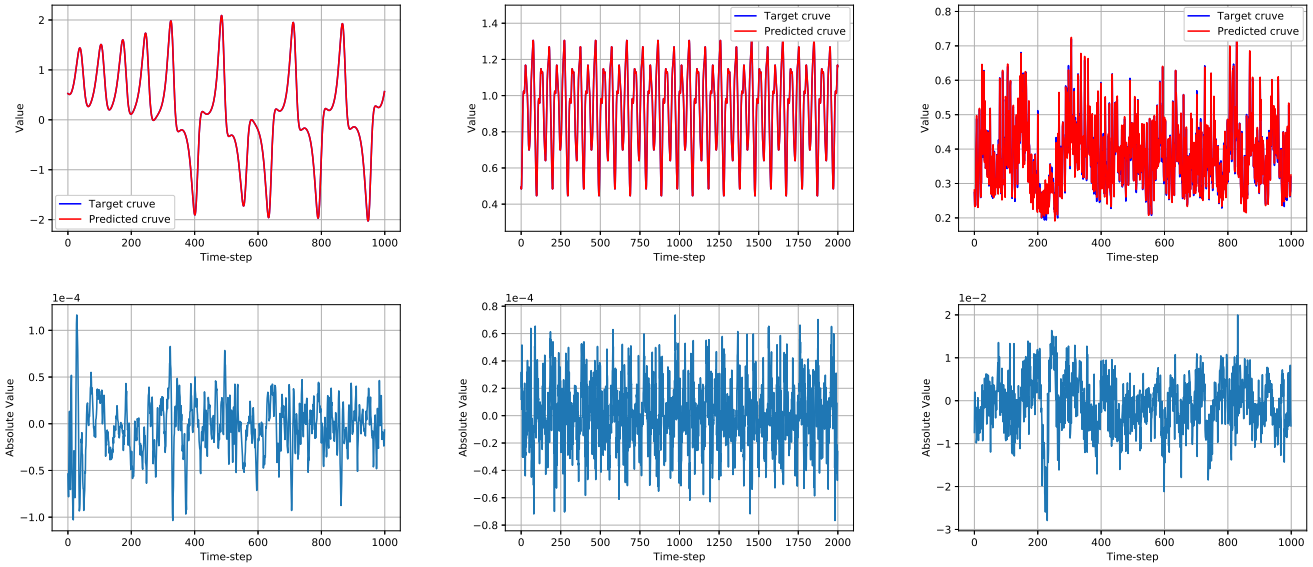
	RMSE(STD)	NRMSE(STD)	MAPE(STD)	Layers
DeepESN [21]	1.23E-04±(5.84E-05)	1.40E-04±(6.63E-05)	1.81E-03±(1.10E-03)	3
Deep Multi-Span ESN	1.19E-04±(3.82E-05)	1.35E-04±(4.33E-05)	7.35E-04±(4.56E-04)	3
Deep Multi-Temporal ESN	**8.66E-05±(2.28E-05)**	**9.82E-05±(1.73E-05)**	**6.10E-04±(3.57E-04)**	3

TABLE IV
COMPARISON OF AVERAGE RESULTS ON THE 84-STEP-AHEAD PREDICTION FOR MGS-17.

Models	RMSE(STD)	NRMSE(STD)	MAPE(STD)	Layers
ESN [32]	4.37E-02±(6.31E-03)	2.01E-01±(2.91E-02)	7.03E-01±(1.27E-01)	1
φ-ESN [34]	8.60E-03±(1.63E-03)	3.96E-02±(7.49E-03)	1.00E-01±(2.13E-02)	2
R^2 SP [35]	2.72E-02±(4.27E-03)	1.25E-01±(1.96E-02)	1.00E-01±(2.13E-02)	2
MESN [36]	1.27E-03±(2.50E-03)	5.86E-02±(1.15E-02)	1.91E-01±(4.22E-02)	7
Mod-DeepESN [23]	7.22E-03±(*)	2.75E-02±(*)	5.55E-01±(*)	3
Deep-ESN [22]	1.12E-03±(1.87E-04)	5.17E-03±(8.61E-04)	1.51E-02±(3.06E-03)	8
Deep Multi-Span ESN ($N_R = 300$)	1.47E-04±(4.65E-05)	6.59E-04±(2.07E-04)	1.36E-04±(4.41E-05)	3
Deep Multi-Temporal ESN ($N_R = 300$)	1.36E-04±(5.48E-05)	6.11E-04±(2.45E-04)	1.26E-04±(5.11E-05)	3
Deep Multi-Span ESN (best)	2.91E-05±(1.02E-05)	1.30E-04±(4.60E-05)	2.66E-05±(9.34E-06)	3
Deep Multi-Temporal ESN (best)	**1.93E-05±(5.48E-06)**	**8.63E-05±(2.45E-05)**	**1.51E-05±(4.99E-06)**	2

TABLE V
COMPARISON OF AVERAGE RESULTS ON THE NARMA-10 TIME-SERIES PREDICTION

Models	RMSE(STD)	NRMSE(STD)	MAPE(STD)	Layers
ESN [32]	2.76E-02±(2.25E-03)	2.45E-01±(2.00E-02)	5.72E-02±(5.01E-03)	1
φ-ESN [34]	1.92E-02±(2.00E-03)	1.69E-01±(1.75E-02)	3.94E-02±(4.13E-03)	2
R^2 SP [35]	2.05E-02±(2.38E-03)	1.81E-01±(2.21E-02)	4.30E-02±(5.43E-03)	2
MESN [36]	1.91E-02±(2.73E-03)	1.68E-01±(2.40E-02)	4.07E-02±(5.59E-03)	2
Deep-ESN [22]	1.39E-02±(1.33E-03)	1.21E-01±(1.16E-02)	2.19E-02±(2.48E-03)	4
Deep Multi-Span ESN ($N_R = 300$)	6.96E-03±(2.53E-04)	6.92E-02±(2.52E-03)	1.51E-02±(6.62E-04)	3
Deep Multi-Temporal ESN ($N_R = 300$)	7.07E-03±(2.38E-04)	7.03E-02±(2.36E-03)	1.57E-02±(5.33E-04)	3
Deep Multi-Span ESN ($N_R = 400$)	**6.61E-03±(2.31E-04)**	**6.58E-02±(2.30E-03)**	**1.46E-02±(4.74E-04)**	3
Deep Multi-Temporal ESN ($N_R = 400$)	6.84E-03±(2.28E-04)	6.81E-02±(2.57E-03)	1.51E-02±(5.75E-04)	3



(a) Lorenz (b) MGS-17 (c) NARMA-10

Fig. 3. The prediction curve and the absolute error on Lorenz system, MGS-17 and NARMA-10

and empirical output distribution. The IP rule can be briefly described as follows:

$$\mathbf{x}_{out} = \tanh\left(\mathbf{g}\mathbf{x}_{in} + \mathbf{b}\right), \tag{22}$$

where $\mathbf{x}_{in} \in \mathbb{R}^{N_R}$ and $\mathbf{x}_{out} \in \mathbb{R}^{N_R}$ are the input vector and the output vector of the reservoir units, respectively and $\mathbf{g}$ and $\mathbf{b}$ denote the gain vector and the bias vector of the nonlinear tangent hyperbolic activation, respectively. The gradient-decent algorithm was employed to compute the gradients of $\mathbf{g}$ and $\mathbf{b}$ as follows:

$$\Delta\mathbf{b} = -\epsilon(\left(\mathbf{x}_{out}/\sigma^2\right)\left(2\sigma^2 + 1 - \mathbf{x}_{out}^2 + \mu\mathbf{x}_{out}\right) - \left(\mu/\sigma^2\right)), \tag{23}$$

$$\Delta\mathbf{g} = \epsilon\mathbf{g} + \Delta\mathbf{b}\mathbf{x}_{in}, \tag{24}$$

where μ and σ denote the mean and standard deviation of Gaussian-distribution, respectively, and ϵ means the learning rate. We set the parameters of IP rules at $\mu = 0$, $\sigma = 0.1$, and $\epsilon = 0.0005$, the updating epoch at $N_{epoch} = 10$, and the threshold of the updating process at $\phi = 0.1$. Table IV shows the best prediction performance of our proposed models in comparison with those reported in [22]. The best performance of Deep Multi-Temporal ESN was obtained under the best parameter setting: $N_R = 900$, $\rho = 1$, and $D = 3$. Deep Multi-Span ESN obtained the best performances under the best parameter setting: $N_R = 800$, and $\rho = 1$. Since the reported performances of Deep-ESN are obtained by using eight layers of 300 reservoir neurons, we also show those of our proposed models with settings: $N_R = 300, \rho = 1$, and $D = 3$. The Deep Multi-Temporal ESN yields the overwhelming results even with two reservoir layers. It is shown that better prediction performances were obtained by using the same reservoir size in each layer of our proposed models as those in Deep-ESN. We can see that The predicted time series and the absolute error curve corresponding to the best results of Deep Multi-Temporal ESN are shown in Fig. 3(b).

3) NARMA-10: The best prediction performance of our proposed models and those reported in [22] are listed in Table V. The Deep Multi-Temporal ESN kept $D = 3$. It is obvious that our proposed methods give the best prediction results under the parameter setting: $N_R = 400$ and $\rho = 0.80$. Note that we also list the performance of our proposed methods with the same reservoir size $N_R = 300$ as that in Deep-ESN. For the best results obtained by Deep Multi-Span ESN with $N_L = 3$, the corresponding predicted time series and the absolute error are shown in Fig. 3(c).

V. DISCUSSION

In order to clarify the effect of the number of layers, the reservoir size, and the number of multi-span groups on the performance in nonlinear time-series prediction, we investigated the prediction performances of Deep Multi-Span ESN, Deep Multi-Temporal ESN, and DeepESN under the best parameter settings by varying the reservoir size N_R from 100 to 1000. Figure 4 presents RMSE values plotted against reservoir size for different parameter settings in the three time-series prediction tasks. We find that the overall RMSE of the two proposed models are better than those based on DeepESN when more reservoir layers are added. For example, DeepESN can achieve the lowest RMSE with the two reservoir layers on the NARMA-10 dataset. However, the Deep Multi-Span ESN and Deep Multi-Temporal ESN outperform DeepESN in the case of three reservoir layers on all the time-series datasets. In addition, we observed that excessive time-span groups will not lead to better results as shown in Figs. 4(a)-4(b). Therefore, the number of groups, the reservoir size, and the number of layers, should be appropriately determined depending on different time-series data.

Multicollinearity [39] of features is an important factor affecting regression performance. A high multicollinearity tends to lead a bad prediction performance. Therefore, we compared multicollinearity in each layer (group) of DeepESN, Deep Multi-Span ESN, and Deep Multi-Temporal ESN. Here, the condition number is used for measuring the degree of multicollinearity, which can be formulated as follows:

$$Cond\left(\mathbf{S}\right) = \frac{\sigma_{max}\left(\mathbf{S}\right)}{\sigma_{min}\left(\mathbf{S}\right)}, \tag{25}$$

where $\mathbf{S}$ is the feature matrix, and σ_{max} and σ_{min} are maximal and minimal singular values of $\mathbf{S}$, respectively. Commonly, a higher condition number represents more collinearity in the target matrix. For DeepESN and Deep Multi-Span ESN, $\mathbf{S}^l \in \mathbb{R}^{N_R \times N_T}$ represents collection matrix of features in the l-th reservoir layer. For Deep Multi-Temporal ESN, $\mathbf{S}^d \in \mathbb{R}^{N_R \times N_T}$ is the collection matrix of the d-th group in the last reservoir layer. The condition numbers for DeepESN, Deep Multi-Span ESN, and Deep Multi-Temporal ESN for the Lorenz system, MGS-17, and NARMA-10 are shown in Fig. 5. It is obvious that multi-span features extracted by our proposed models lead to lower multicollinearity than the same time-span features extracted by DeepESN, which may be the major reason for the higher prediction performance of our models compared with those of other ESN-based models.

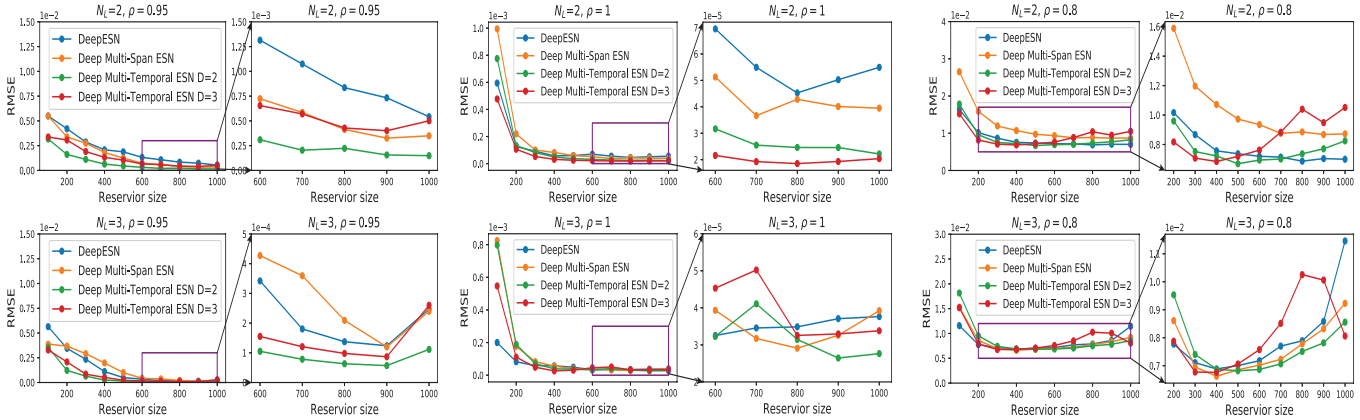
VI. CONCLUSION

In this paper, two novel deep RC models with the ability to extract multi time-span features, Deep Multi-Span ESN and Deep multi-Temporal ESN, have been proposed. The analysis of the computational costs of the two proposed models have shown the computational efficiency. The prediction performances on the Lorenz system, MGS-17 and NARMA-10 have shown significant effectiveness of the proposed models for nonlinear time-series prediction tasks. Also, we have evaluated the performance of the two proposed models by varying the number of reservoir layers, the reservoir size, and the number of time-span groups. Finally, the investigation of multicollinearity in the various layers (groups) have been given.

As future works, we will continue to study how the proposed models are applied to other temporal processing tasks such as time series classification tasks.

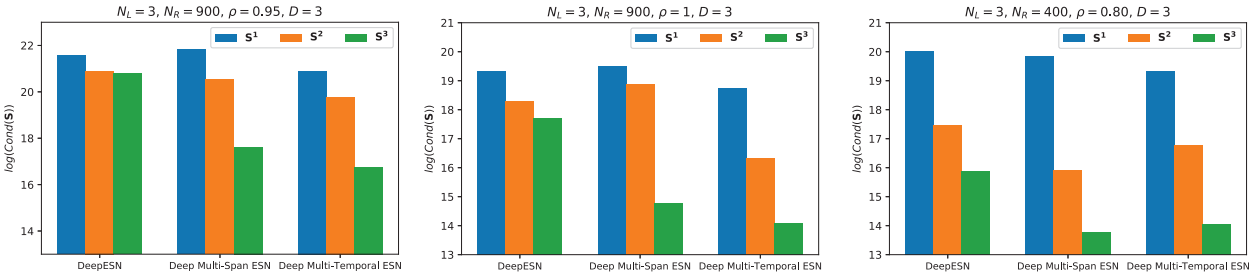
ACKNOWLEDGEMENTS

This work was partially supported by JST-Mirai Program Grant Number JPMJMI19B1, Japan (GT) and partially based



(a) Lorenz system (b) MGS-17 (c) NARMA-10

Fig. 4. The average RMSE of DeepESN, Deep Multi-Span ESN and Deep Multi-Temporal ESN on Lorenz system, MGS-17 and NARMA-10 with variations of the reservoir size from 100 to 1000.



(a) Lorenz system (b) MGS-17 (c) NARMA-10

Fig. 5. The average condition number of DeepESN, Deep Multi-Span ESN and Deep Multi-Temporal ESN on Lorenz system, MGS-17 and NARMA-10.

on results obtained from a project (No. 18102285-0) subsidized by the New Energy and Industrial Technology Development Organization (NEDO) (GT).

REFERENCES

[1] M. Casdagli, "Nonlinear prediction of chaotic time series," *Physica D: Nonlinear Phenomena*, vol. 35, no. 3, pp. 335–356, 1989.

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

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

[4] 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.

[5] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[6] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[7] 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.

[8] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[9] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Networks*, 2019.

[10] 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.

[11] K. G. Boroojeni, M. H. Amini, S. Bahrami, S. S. Iyengar, A. I. Sarwat, and O. Karabasoglu, "A novel multi-time-scale modeling for electric power demand forecasting: From short-term to medium-term horizon," *Electric Power Systems Research*, vol. 142, pp. 58–73, 2017.

[12] M. D. Skowronski and J. G. Harris, "Automatic speech recognition using a predictive echo state network classifier," *Neural Networks*, vol. 20, no. 3, pp. 414–423, 2007.

[13] D. Liu, J. Wang, and H. Wang, "Short-term wind speed forecasting based on spectral clustering and optimised echo state networks," *Renewable Energy*, vol. 78, pp. 599–608, 2015.

[14] Y. Xia, B. Jelfs, M. M. Van Hulle, J. C. Príncipe, and D. P. Mandic, "An augmented echo state network for nonlinear adaptive filtering of complex noncircular signals," *IEEE Transactions on Neural Networks*, vol. 22, no. 1, pp. 74–83, 2010.

[15] J. P. Donate and P. Cortez, "Evolutionary optimization of sparsely connected and time-lagged neural networks for time series forecasting," *Applied Soft Computing*, vol. 23, pp. 432–443, 2014.

[16] 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.

[17] 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.

[18] H. Wang and X. Yan, "Optimizing the echo state network with a binary particle swarm optimization algorithm," *Knowledge-Based Systems*, vol. 86, pp. 182–193, 2015.

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[20] L. Deng, D. Yu, and J. Platt, "Scalable stacking and learning for building deep architectures," in *2012 IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP)*. IEEE, 2012, pp. 2133–2136.

[21] C. Gallicchio and A. Micheli, "Deep Reservoir Computing: A Critical Analysis." in *ESANN*, 2016.

[22] Q. Ma, L. Shen, and G. W. Cottrell, "Deep-esn: A multiple projection-encoding hierarchical reservoir computing framework," *arXiv preprint arXiv:1711.05255*, 2017.

[23] Z. Carmichael, H. Syed, and D. Kudithipudi, "Analysis of Wide and Deep Echo State Networks for Multiscale Spatiotemporal Time Series Forecasting," *arXiv preprint arXiv:1908.08380*, 2019.

[24] M. H. Loorak, C. Perin, N. Kamal, M. Hill, and S. Carpendale, "Timespan: Using visualization to explore temporal multi-dimensional data of stroke patients," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 409–418, 2015.

[25] J. R. Bellegarda, "Large vocabulary speech recognition with multispan statistical language models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 76–84, 2000.

[26] T. Strauss, W. Wustlich, and R. Labahn, "Design strategies for weight matrices of echo state networks," *Neural Computation*, vol. 24, no. 12, pp. 3246–3276, 2012.

[27] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[28] W. N. van Wieringen, "Lecture notes on ridge regression," *arXiv preprint arXiv:1509.09169*, 2015.

[29] G. H. Golub and C. Reinsch, *Singular value decomposition and least squares solutions*. Springer, 1971, pp. 134–151.

[30] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[31] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[32] 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.

[33] A. F. Atiya and A. G. Parlos, "New results on recurrent network training: unifying the algorithms and accelerating convergence," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 697–709, 2000.

[34] C. Gallicchio and A. Micheli, "Architectural and markovian factors of echo state networks," *Neural Networks*, vol. 24, no. 5, pp. 440–456, 2011.

[35] 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.

[36] 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.

[37] M.-H. Yusoff, J. Chrol-Cannon, and Y. Jin, "Modeling neural plasticity in echo state networks for classification and regression," *Information Sciences*, vol. 364, pp. 184–196, 2016.

[38] B. Schrauwen, M. Wardermann, D. Verstraeten, J. J. Steil, and D. Stroobandt, "Improving reservoirs using intrinsic plasticity," *Neurocomputing*, vol. 71, no. 7-9, pp. 1159–1171, 2008.

[39] E. R. Mansfield and B. P. Helms, "Detecting multicollinearity," *The American Statistician*, vol. 36, no. 3a, pp. 158–160, 1982.