A Multi-Reservoir Echo State Network with Multiple-Size Input Time Slices for Nonlinear Time-Series Prediction

Ziqiang Li[1](✉) and Gouhei Tanaka[1,2](✉)

[1] Department of Electrical Engineering and Information Systems, Graduate School of Engineering, The University of Tokyo, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{ziqiang-li,gtanaka}@g.ecc.u-tokyo.ac.jp

[2] International Research Center for Neurointelligence, The University of Tokyo, Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

Abstract. A novel multi-reservoir echo state network incorporating the scheme of extracting features from multiple-size input time slices is proposed in this paper. The proposed model, Multi-size Input Time Slices Echo State Network (MITSESN), uses multiple reservoirs, each of which extracts features from each of the multiple input time slices of different sizes. We compare the prediction performances of MITSESN with those of the standard echo state network and the grouped echo state network on three benchmark nonlinear time-series datasets to show the effectiveness of our proposed model. Moreover, we analyze the richness of reservoir dynamics of all the tested models and find that our proposed model can generate temporal features with less linear redundancies under the same parameter settings, which provides an explanation about why our proposed model can outperform the other models to be compared on the nonlinear time-series prediction tasks.

Keywords: Reservoir computing · Echo state network · Multi-size input time slices

1 Introduction

Nonlinear Time-series Prediction (NTP) [19] is one of the classical machine learning tasks. The goal of this task is to make predicted values close to the corresponding actual values. Recurrent Neural Networks (RNNs) [12] are a subset of Neural Networks (NNs) and have been widely used in nonlinear time-series prediction tasks. Many related works have reported that RNNs-based methods outperform other NNs-based methods on some prediction tasks [9,13]. However, the classical RNN model and its extended models such as Long Short-Term Memory (LSTM) [5] and Gated Recurrent Unit (GRU) [1] often suffer from expensive computational costs along with gradient explosion/vanishing problems in the training process.

Reservoir Computing (RC) [6,17] is an alternative computational framework which provides a remarkably efficient approach for training RNNs. The most

T. Mantoro et al. (Eds.): ICONIP 2021, LNCS 13109, pp. 28–39, 2021.
https://doi.org/10.1007/978-3-030-92270-2_3

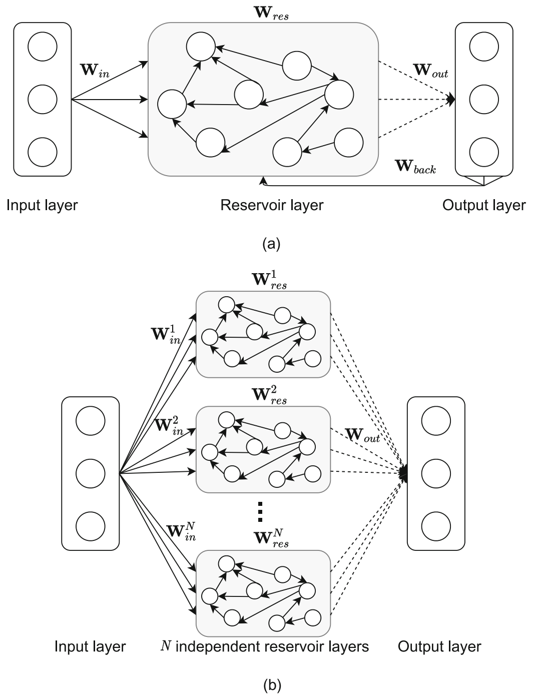

Fig. 1. (a) A standard ESN, (b) A standard GroupedESN.

important characteristic of this framework is that a predetermined non-linear system is used to map input data into a high-dimensional feature space. Based on this characteristic, a well-trained RNN can be built with relatively low computational costs.

As one of the important implementations of RC, Echo State Network (ESN), was first proposed in Ref. [6] and has been widely used to handle NTP tasks [8,14]. The standard architecture of ESN, including an input layer, a reservoir layer, and an output layer, is shown in Fig. 1(a). We can see that an input weight matrix, $\mathbf{W}_{in}$, represents the connection weights between the input layer and the reservoir layer. Moreover, a reservoir weight matrix denoted by $\mathbf{W}_{res}$ represents the connection weights between neurons inside the reservoir layer. The readout weight matrix, $\mathbf{W}_{out}$, represents the connection weights between the reservoir layer and the output layer. A feedback matrix from the output layer to the reservoir layer is denoted by $\mathbf{W}_{back}$. Typically, the element values of three matrices, $\mathbf{W}_{in}$, $\mathbf{W}_{res}$, and $\mathbf{W}_{back}$, are randomly drawn from certain uniform distributions and are kept fixed. Only $\mathbf{W}_{out}$ (dash lines) need to be trained by the linear regression.

Based on the above introduced ESN, we can quickly obtain a well-trained RNN. However, this simple architecture leads to a huge limitation in enhancing its representation ability and further makes the corresponding prediction performances on the NTP task hard to be improved. An effective remedy proposed in Ref. [4] is to feed the input time series into N independent randomly generated reservoirs for producing N different reservoir states, and then combine them to enrich the features used for training the output weight matrix. The authors of Ref. [4] called this architecture "GroupedESN" and reported that the prediction performances obtained by their proposed model is much better than those obtained by the standard ESN on some tasks. We show a schematic diagram of GroupedESN in Fig. 1(b).

The purpose of multi-reservoir ESN, including GroupedESN, inherited from the standard ESN is to extract features from each "data point" in a time series. In most of related works [4,15], we found that they only used each sampling point in a time series as the "data point" and extracted the corresponding temporal feature from them in each reservoir. In fact, the scheme of extracting features from inseparable sampling points can capture the most fine-grained temporal dependency from the input series. However, this monotonous scheme unavoidably ignores some useful temporal information in the "time slices" composed of a period of continuous sampling points [20]. Figure 2 shows an example of transforming the original sampling points of a time series into several time slices of size two.

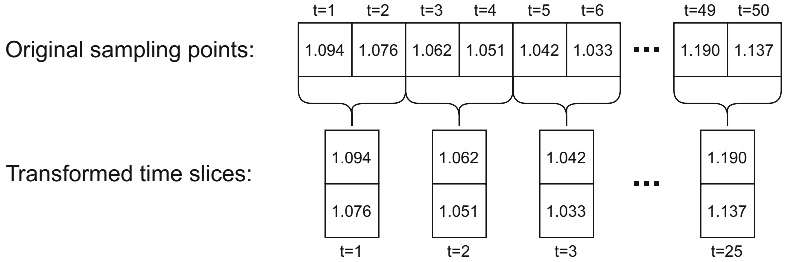

Fig. 2. An example of transforming original sampling points of a time series into time slices of size two.

In this paper, we propose a novel multi-reservoir ESN model, Multi-size Input Time Slice Echo State Network (MITSESN), which can extract various temporal features corresponding to input time slices of different sizes. We compare the proposed model with the standard ESN and the GroupedESN on the three NTP benchmark datasets and demonstrate the effectiveness of our proposed MITSESN. We provide an empirical analysis of richness in the reservoir-state dynamics to explain why our proposed model performs better than the other tested models on the NTP tasks.

The rest of this paper is organized as follows: We describe the details of the proposed model in Sect. 2. We report the experimental results, including results on three NTP benchmark datasets and corresponding analyses of richness in Sect. 3. We conclude this work in Sect. 4.

2 The Proposed Model: MITSESN

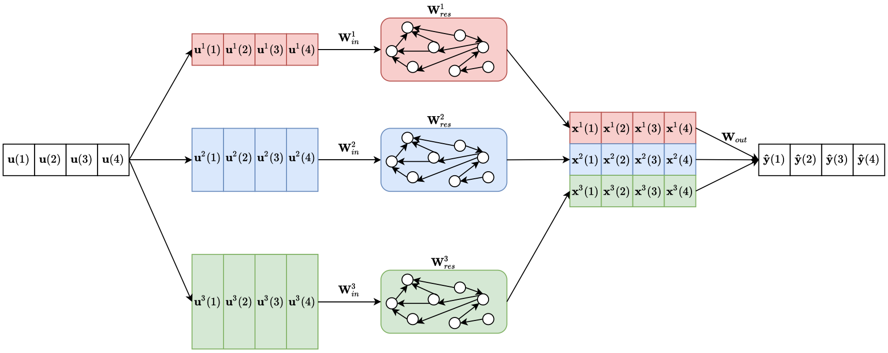
$\mathbf{W}^1_{res}$
$\mathbf{W}^2_{res}$
$\mathbf{W}^3_{res}$
$\mathbf{W}^1_{in}$
$\mathbf{W}^2_{in}$
$\mathbf{W}^3_{in}$
$\mathbf{W}_{out}$
$\mathbf{u}(1)$ $\mathbf{u}(2)$ $\mathbf{u}(3)$ $\mathbf{u}(4)$
$\mathbf{u}^1(1)$ $\mathbf{u}^1(2)$ $\mathbf{u}^1(3)$ $\mathbf{u}^1(4)$
$\mathbf{u}^2(1)$ $\mathbf{u}^2(2)$ $\mathbf{u}^2(3)$ $\mathbf{u}^2(4)$
$\mathbf{u}^3(1)$ $\mathbf{u}^3(2)$ $\mathbf{u}^3(3)$ $\mathbf{u}^3(4)$
$\mathbf{x}^1(1)$ $\mathbf{x}^1(2)$ $\mathbf{x}^1(3)$ $\mathbf{x}^1(4)$
$\mathbf{x}^2(1)$ $\mathbf{x}^2(2)$ $\mathbf{x}^2(3)$ $\mathbf{x}^2(4)$
$\mathbf{x}^3(1)$ $\mathbf{x}^3(2)$ $\mathbf{x}^3(3)$ $\mathbf{x}^3(4)$
$\hat{\mathbf{y}}(1)$ $\hat{\mathbf{y}}(2)$ $\hat{\mathbf{y}}(3)$ $\hat{\mathbf{y}}(4)$

Fig. 3. An example of the proposed MITSESN with three independent reservoirs.

A schematic diagram of our proposed MITSESN is shown in Fig. 3. This is a case where an original input time series with length four is fed into the proposed MITSESN with three independent reservoirs. The original input time series is transformed into three time slices of different sizes. Then, each time slice is fed into the corresponding reservoir and the generated reservoir states are concatenated together. Finally, the concatenated state matrix is decoded to the desired values. Based on the above introduction, we can observe that our proposed MITSESN can be divided into three parts: the series-to-slice transformer, the multi-reservoir encoder, and the decoder. We introduce the details of these parts as below.

2.1 Series-to-Slice Transformer

We define the input vector and the target vector at time t as $\mathbf{u}(t) \in \mathbb{R}^{N_U}$ and $\mathbf{y}(t) \in \mathbb{R}^{N_Y}$, respectively. The length of input series and that of target series are denoted by N_T.

To formulate the transformation from the original input time-series points into input time slices of different sizes, we define the maximal size of the input slice used in the MITSESN as M. In our model, the maximal size of the input slice is equivalent to the number of different sizes. We denote the size of input slice by m, where $1 \leq m \leq M$. In order to keep the length of the transformed input time slice the same as those of the original input time series, we add zero paddings of length $(m-1)$ into the beginning of the original input series, which can be formulated as follows:

$$\mathbf{U}^m_{zp} = [\underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{m-1}, \mathbf{u}(1), \mathbf{u}(2), \ldots, \mathbf{u}(N_T)], \tag{1}$$

where $\mathbf{U}_{zp}^{m} \in \mathbb{R}^{N_U \times (N_T+m-1)}$ is the zero-padded input matrix. Based on the above settings, we can obtain the transformed input matrix corresponding to input time slices of size m as follows:

$$\mathbf{U}^{m} = \left[\mathbf{u}^{m}(1), \mathbf{u}^{m}(2), \ldots, \mathbf{u}^{m}(N_T)\right], \tag{2}$$

where $\mathbf{U}^{m} \in \mathbb{R}^{mN_U \times N_T}$ and $\mathbf{u}^{m}(t)$ is composed of the vertical concatenation of vectors from the t-th column to the $(t+m-1)$-th column in $\mathbf{U}_{zp}^{m}$. We show an example of $\mathbf{U}^{m}$ when $m = 3$ as follows:

$$\begin{aligned}\mathbf{U}^{3} &= \left[\mathbf{u}^{3}(1), \mathbf{u}^{3}(2), \ldots, \mathbf{u}^{3}(N_T)\right] \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \ldots & \mathbf{u}(N_T-2) \\ \mathbf{0} & \mathbf{u}(1) & \ldots & \mathbf{u}(N_T-1) \\ \mathbf{u}(1) & \mathbf{u}(2) & \ldots & \mathbf{u}(N_T) \end{bmatrix}.\end{aligned} \tag{3}$$

2.2 Multi-reservoir Encoder

We adopt the basic architecture of GroupedESN in Fig. 1(b) to build the multi-reservoir encoder. However, the feeding strategy of the multi-reservoir encoder is different from that of GroupedESN. We assume that input time slices of size m are fed into the m-th reservoir. Therefore, there are totally M reservoirs in the multi-reservoir encoder. For the m-th reservoir, we define the input weight matrix and the reservoir weight matrix as $\mathbf{W}_{in}^{m} \in \mathbb{R}^{N_R^m \times mN_U}$ and $\mathbf{W}_{res}^{m} \in \mathbb{R}^{N_R^m \times N_R^m}$, respectively, where N_R^m represents the size of the m-th reservoir. The state of the m-th reservoir at time t, $\mathbf{x}^{m}(t)$, is calculated as follows:

$$\mathbf{x}^{m}(t) = (1-\alpha)\,\mathbf{x}^{m}(t-1) + \alpha \tanh\left(\mathbf{W}_{in}^{m}\mathbf{u}^{m}(t) + \mathbf{W}_{res}^{m}\mathbf{x}^{m}(t-1)\right), \tag{4}$$

where the element values of $\mathbf{W}_{in}^{m}$ are randomly drawn from the uniform distribution of the range $[-\theta, \theta]$. The parameter θ is the input scaling. The element values of $\mathbf{W}_{res}^{m}$ are randomly chosen from the uniform distribution of the range $[-1, 1]$. To ensure the "Echo State Property" (ESP) [6], $\mathbf{W}_{res}^{m}$ should satisfy the condition described as follows:

$$\rho\left((1-\alpha)\,\mathbf{E} + \alpha \mathbf{W}_{res}^{m}\right) < 1, \tag{5}$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix argument, the parameter α represents the leaking rate which is set in the range $(0, 1]$, and $\mathbf{E} \in \mathbb{R}^{N_R^m \times N_R^m}$ is the identity matrix. Moreover, we use the parameter η to denote the sparsity of $\mathbf{W}_{res}^{m}$.

We denote the reservoir-state matrix composed of N_T state vectors corresponding to the m-th reservoir as $\mathbf{X}^{m} \in \mathbb{R}^{N_R^m \times N_T}$. By concatenating M reservoir-state matrices in the vertical direction, we obtain a concatenated state matrix, $\mathbf{X} \in \mathbb{R}^{\sum_{m=1}^{M} N_R^m \times N_T}$, which can be written as follows:

$$\mathbf{X} = \left[\mathbf{X}^{1}; \mathbf{X}^{2}; \ldots; \mathbf{X}^{M}\right]. \tag{6}$$

2.3 Decoder

We use the linear regression for converting the concatenated state matrix into the output matrix, which can be formulated as follows:

$$\hat{\mathbf{Y}} = \mathbf{W}_{out}\mathbf{X}, \tag{7}$$

where $\hat{\mathbf{Y}} \in \mathbb{R}^{N_Y \times N_T}$ is the output matrix. The readout matrix $\mathbf{W}_{out}$ is given by the closed-form solution as follows:

$$\mathbf{W}_{out} = \mathbf{Y}\mathbf{X}^{\mathrm{T}}\left(\mathbf{X}\mathbf{X}^{\mathrm{T}} + \lambda\mathbf{I}\right)^{-1}, \tag{8}$$

where $\mathbf{Y} \in \mathbb{R}^{N_Y \times N_T}$ represents the target matrix, $\mathbf{I} \in \mathbb{R}^{\sum_{m=1}^{M} N_R^m \times \sum_{m=1}^{M} N_R^m}$ is an identity matrix, and the parameter λ symbolizes the Tikhonov regularization factor [18].

3 Numerical Simulations

In this section, we report the details and results of simulations. Specifically, three benchmark nonlinear time-series datasets and the corresponding task settings are described in Sec. 3.1, the evaluation metrics are listed in Sec. 3.2, the tested models and parameter settings are described in Sec. 3.3, the corresponding simulation results are presented in Sec. 3.4. The analyses of richness for all the tested models are given in Sec. 3.5.

3.1 Datasets Descriptions and Task Settings

We leverage three nonlinear time-series datasets, including the Lorenz system, MGS-17, and KU Leuven datasets, to evaluate the prediction performances of our proposed model. Glimpses of the above datasets are shown in Fig. 4. The partitions of the training set, the validation set, the testing set, and the initial transient set are listed in Table 1. We introduce the details of these datasets and task settings as below.

Lorenz System. The equation of Lorenz system [10] is formulated as follows:

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma(y - x), \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= x(\delta - z) - y, \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy - \beta z. \end{aligned} \tag{9}$$

When $\delta = 28$, $\sigma = 10$, and $\beta = 8/3$, the system exhibits a chaotic behavior. In our evaluation, we used the chaotic Lorenz system and set the initial condition at $(x(0), y(0), z(0)) = (12, 2, 9)$. We adopted the sampling interval $\Delta t = 0.02$ and rescaled by the scaling factor 0.1, which is the same as those reported in [7]. We set a six-step-ahead prediction task on x values, which can be represented as $\mathbf{u}(t) = x(t)$ and $\mathbf{y}(t) = x(t + 6)$.

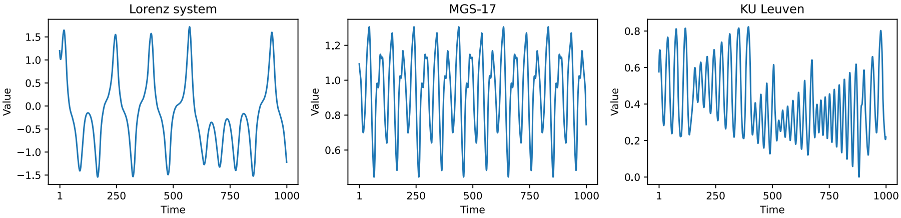

Fig. 4. Examples of three nonlinear time-series datasets.

MGS-17. The equation of Mackey-Glass system [11] is formulated as follows:

$$z(t+1) = z(t) + \delta \cdot \left(a \frac{z(t-\varphi/\delta)}{1+z(t-\varphi/\delta)^n} - bz(t) \right), \tag{10}$$

where a, b, n, and δ are fixed at 0.2, -0.1, 10, and 0.1, respectively. The Mackey-Glass system exhibits a chaotic behavior when $\varphi > 16.8$. We kept the value of φ equal to 17 (MGS-17). The task on MGS-17 is to predict the 84-step-ahead value of z [7], which can be represented as $\mathbf{u}(t) = z(t)$ and $\mathbf{y}(t) = z(t+84)$.

KU Leuven. KU Leuven dataset was first proposed in a time-series prediction competition held at KU Leuven, Belgium [16]. We set an one-step-ahead prediction task on this dataset for the evaluation.

Table 1. The partitions of Lorenz system, MGS-17, and KU Leuven datasets.

	Training set	Valiation set	Testing set	Initial transient set
Lorenz system	3000	1000	1000	500
MGS-17	3000	1000	1000	500
KU Leuven	2000	500	500	200

3.2 Evaluation Metrics

We use two evaluation metrics in this work, including Normalized Root Mean Square Error (NRMSE) and Symmetric Mean Absolute Percentage Error (SMAPE), to evaluate the prediction performances. These two evaluation metrics are formulated as follows:

$$\mathrm{NRMSE} = \frac{\sqrt{\frac{1}{N_T}\sum_{t=1}^{N_T}\left(\hat{\mathbf{y}}(t) - \mathbf{y}(t)\right)^2}}{\sqrt{\frac{1}{N_T}\sum_{t=1}^{N_T}\left(\mathbf{y}\left(t\right) - \bar{\mathbf{y}}\right)^2}}, \tag{11}$$

$$\mathrm{SMAPE} = \frac{1}{N_T}\sum_{t=1}^{N_T}\frac{|\hat{\mathbf{y}}(t) - \mathbf{y}(t)|}{\left(|\hat{\mathbf{y}}(t)| + |\mathbf{y}(t)|\right)/2}, \tag{12}$$

where $\bar{\mathbf{y}}$ denotes the mean of data values of $\mathbf{y}(t)$ from $t = 1$ to N_T.

3.3 Tested Models and Parameter Settings

In our simulation, we compared the prediction performances of our proposed model with those of ESN and GroupedESN. We denote the overall reservoir size $N_R = \sum_{m=1}^{M} N_R^m$ for all the models. Two architectures with $M = 2$ and $M = 3$ for GroupedESN and MITSESN were considered. We represent the architecture with M reservoirs as $N_R^1 - N_R^2 - \cdots - N_R^M$.

To make a fair comparison, we set N_R the same for each model. For simplicity, the size of each reservoir in the GroupedESN and the proposed MITSESN was kept the same. The parameter settings for all the tested models are listed in Table 2. The spectral radius, the sparsity of reservoir weights, and the Tikhonov regularization were set at 0.95, 90%, and 1E-06, respectively. The input scaling, the leaking rate and the overall reservoir size were searched in the ranges of [0.01, 0.1, 1], $[0.1, 0.2, \ldots, 1]$, and $[150, 300, \ldots, 900]$, respectively. For each setting, we averaged the results over 20 realizations.

Table 2. The parameter settings for all the tested models

Parameter	Symbol	Value
Spectral radius	ρ	0.95
Sparsity of reservoir weights	η	90%
Tikhonov regularization	λ	1E-06
Input scaling	θ	[0.01, 0.1, 1]
Leaking rate	α	$[0.1, 0.2, \ldots, 1]$
Overall reservoir size	N_R	$[150, 300, \ldots, 900]$

3.4 Simulation Results

We report the averaged prediction performances on the three datasets in Tables 3, 4 and 5. It is obvious that our proposed MITSESN with three reservoirs obtains the smallest NRMSE and SMAPE among all the tested models with the same overall reservoir size. By comparing the prediction performances of the GroupedESN with those of our proposed MITSESN, we can clearly find that the strategy of extracting temporal features from multi-size input time slices can significantly improve the prediction performances. Moreover, with the increase of the size of input time slices, the performance is obviously improved. Especially, our simulation results on MGS-17 show that only adding more reservoirs is not a universally effective method to improve prediction performances for the GroupedESN. Lastly, we observe that the best prediction performances of all the tested models are obtained under the maximal values in the searching range of input scaling and reservoir size, which indicates that all the models benefit from high richness [3]. We investigate how this important characteristic changes under different N_R for all the tested models in the following section.

Table 3. Average performances of the six-step-ahead prediction task on the Lorenz system.

Models	Architecture	NRMSE	SMAPE	Best parameters
ESN	900	7.44E-05	1.21E-04	$N_R = 900$, $\theta = 1$, $\alpha = 0.4$
GroupedESN ($M = 2$)	450-450	7.10E-05	1.04E-04	$N_R = 900$, $\theta = 1$, $\alpha = 0.4$
GroupedESN ($M = 3$)	300-300-300	7.04E-05	1.13E-04	$N_R = 900$, $\theta = 1$, $\alpha = 0.4$
MITSESN ($M = 2$)	450-450	6.29E-05	1.16E-04	$N_R = 900$, $\theta = 1$, $\alpha = 0.5$
MITSESN ($M = 3$)	300-300-300	**5.58E-05**	**9.23E-05**	$N_R = 900$, $\theta = 1$, $\alpha = 0.4$

Table 4. Average performances of the 84-step-ahead prediction task on the MGS-17.

Models	Architecture	NRMSE	SMAPE	Best parameters
ESN	900	1.87E-02	3.94E-03	$N_R = 900$, $\theta = 1$, $\alpha = 0.2$
GroupedESN ($M = 2$)	450-450	2.05E-02	4,14E-03	$N_R = 900$, $\theta = 1$, $\alpha = 0.3$
GroupedESN ($M = 3$)	300-300-300	2.16E-02	4.48E-03	$N_R = 900$, $\theta = 1$, $\alpha = 0.3$
MITSESN ($M = 2$)	450-450	1.49E-02	3.12E-03	$N_R = 900$, $\theta = 1$, $\alpha = 0.2$
MITSESN ($M = 3$)	300-300-300	**1.24E-02**	**2.41E-03**	$N_R = 900$, $\theta = 1$, $\alpha = 0.3$

3.5 Analysis of Richness

The richness is a desirable characteristic in the reservoir state as suggested by Ref. [2]. Typically the higher richness indicates the less redundancy held in the reservoir state. We leverage the Uncoupled Dynamics (UD) proposed in [3] to measure the richness of $\mathbf{X}$ for all the tested models. The UD of $\mathbf{X}$ is calculated as follows:

$$\arg\min_{d} \left\{ \sum_{k=1}^{d} R_k \mid \sum_{k=1}^{d} R_k \geq \mathcal{A} \right\}, \tag{13}$$

where $\mathcal{A}$ is in the range of $(0, 1]$ and represents the desired ratio of explained variability in the concatenated state matrix. We kept $\mathcal{A} = 0.9$ in the following evaluation. R_k denotes the normalized relevance of the i-th principal component, which can be formulated as follows:

$$R_i = \frac{\sigma_i}{\sum_{j=1}^{N_R} \sigma_j}, \tag{14}$$

where σ_i denotes the i-th singular value in the decreasing order. The higher the value of UD in Eq. (13) is, the less linear redundancy held in the concatenated state matrix $\mathbf{X}$ is. For the evaluation settings, we used a univariable time series of length 5000 and we randomly chose each value from the uniform distribution of the range $[-0.8, 0.8]$. We fixed the leaking rate $\alpha = 1$ and input scaling $\theta = 1$ in all the models.

The average UDs of all the tested models when varying N_R from 150 to 900 are shown in Fig. 5. It is obvious that the MITSESN (M=3) outperforms the other models when varying N_R from 300 to 900. With the increase of N_R (from

Table 5. Average performances of the one-step-ahead prediction task on the KU Leuven dataset.

Models	Architecture	NRMSE	SMAPE	Best parameters
ESN	900	2.31E-02	6.94E-03	$N_R = 900$, $\theta = 1$, $\alpha = 0.5$
GroupedESN ($M = 2$)	450-450	2.29E-02	7.18E-03	$N_R = 900$, $\theta = 1$, $\alpha = 0.4$
GroupedESN ($M = 3$)	300-300-300	2.28E-02	7.11E-03	$N_R = 900$, $\theta = 1$, $\alpha = 0.4$
MITSESN ($M = 2$)	450-450	2.25E-02	7.01E-03	$N_R = 900$, $\theta = 1$, $\alpha = 0.4$
MITSESN ($M = 3$)	300-300-300	**2.24E-02**	**6.64E-03**	$N_R = 900$, $\theta = 1$, $\alpha = 0.5$

$N_R = 450$), differences between UDs of our proposed MITSESN with those of ESN and GroupedESNs ($M = 2$ and 3) gradually become larger and larger, which indicates that our proposed MITSESN can generate less linear redundancy in the concatenated state matrix than the ESN and the GroupedESN under the case of the larger N_R. Moreover, we find that the larger size of input time slices is, the less linear redundancy in the concatenated state matrix of MITSESN is. The above analyses explain the reasons why our proposed MITSESN outperforms the ESN and the GroupedESNs, and the MITSESN ($M = 3$) has the best performances on the three prediction tasks.

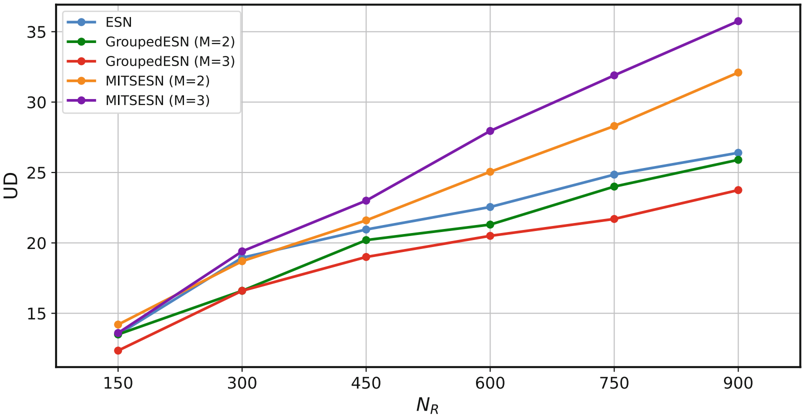

Fig. 5. UDs of all the tested models varying N_R from 150 to 900.

4 Conclusion

In this paper, we proposed a novel multi-reservoir echo state network, MITSESN, for nonlinear time-series prediction tasks. Our proposed MITSESN can extract various temporal features from multi-size input time slices. The prediction performances on three benchmark nonlinear time-series datasets empirically demonstrate the effectiveness of our proposed model. We provided an empirical analysis from the prospective of reservoir-state richness to show the superiority of MITSESN.

As future works, we will continue to evaluate the performances of the proposed model on the other temporal tasks such as time series classification tasks.

Acknowledgements. This work was partly supported by JSPS KAKENHI Grant Number 20K11882 and JST-Mirai Program Grant Number JPMJMI19B1, Japan (GT), and partly based on results obtained from Project No. JPNP16007, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

1. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
2. Gallicchio, C., Micheli, A.: A Markovian characterization of redundancy in echo state networks by PCA. In: ESANN. Citeseer (2010)
3. Gallicchio, C., Micheli, A.: Richness of deep echo state network dynamics. In: Rojas, I., Joya, G., Catala, A. (eds.) IWANN 2019. LNCS, vol. 11506, pp. 480–491. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20521-8_40
4. Gallicchio, C., Micheli, A., Pedrelli, L.: Deep reservoir computing: a critical experimental analysis. Neurocomputing **268**, 87–99 (2017)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
6. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. Ger. Natl. Res. Cent. Inf. Technol. GMD Tech. Rep. **148**(34), 13 (2001)
7. Li, Z., Tanaka, G.: Deep echo state networks with multi-span features for nonlinear time series prediction. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–9. IEEE (2020)
8. Li, Z., Tanaka, G.: HP-ESN: echo state networks combined with hodrick-prescott filter for nonlinear time-series prediction. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–9. IEEE (2020)
9. Liu, Y., Gong, C., Yang, L., Chen, Y.: DSTP-RNN: a dual-stage two-phase attentionbased recurrent neural network for long-term and multivariate time series prediction. Expert Syst. Appl. **143**, 113082 (2020)
10. Lorenz, E.N.: Deterministic nonperiodic flow. J. Atmos. Sci. **20**(2), 130–141 (1963)
11. Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control systems. Science **197**(4300), 287–289 (1977)
12. Medsker, L., Jain, L.C.: Recurrent Neural Networks: Design and Applications. CRC Press, Boca Raton (1999)
13. Menezes, J.M., Barreto, G.A.: A new look at nonlinear time series prediction with narx recurrent neural network. In: 2006 Ninth Brazilian Symposium on Neural Networks (SBRN 2006), pp. 160–165. IEEE (2006)
14. Shen, L., Chen, J., Zeng, Z., Yang, J., Jin, J.: A novel echo state network for multivariate and nonlinear time series prediction. Appl. Soft Comput. **62**, 524–535 (2018)
15. Song, Z., Wu, K., Shao, J.: Destination prediction using deep echo state network. Neurocomputing **406**, 343–353 (2020)

16. Suykens, J.A., Vandewalle, J.: The KU leuven time series prediction competition. In: Suykens J.A.K., Vandewalle J. (eds) Nonlinear Modeling, pp. 241–253. Springer, Boston (1998). https://doi.org/10.1007/978-1-4615-5703-6_9
17. Tanaka, G., et al.: Recent advances in physical reservoir computing: a review. Neural Netw. **115**, 100–123 (2019)
18. Tikhonov, A.N., Goncharsky, A., Stepanov, V., Yagola, A.G.: Numerical Methods for the Solution of III-Posed Problems, vol. 328. Springer Science & Business Media, Berlin (2013)
19. Weigend, A.S.: Time Series Prediction: Forecasting The Future And Understanding The Past Routledge, Abingdon-on-Thames (2018)
20. Yu, Z., Liu, G.: Sliced recurrent neural networks. arXiv preprint arXiv:1807.02291 (2018)