Delayed self-feedback echo state network for long-term dynamics of hyperchaotic systems

Xu Xu,[1,*] Jianming Liu,[1] and Eric Li[2]
[1]College of Mathematics, Jilin University, 2699 Qianjin Street, Changchun 130012, China
[2]School of Computing, Engineering & Digital Technologies, Teesside University, Middlesbrough TS1 3BX, United Kingdom

Analyzing the long-term behavior of hyperchaotic systems poses formidable challenges in the field of nonlinear science. This paper proposes a data-driven model called the delayed self-feedback echo state network (self-ESN) specifically designed for the evolution behavior of hyperchaotic systems. Self-ESN incorporates a delayed self-feedback term into the dynamic equation of a reservoir to reflect the finite transmission speed of neuron signals. Delayed self-feedback establishes a connection between the current and previous m time steps of the reservoir state and provides an effective means to capture the dynamic characteristics of the system, thereby significantly improving memory performance. In addition, the concept of local echo state property (ESP) is introduced to relax the conventional ESP condition, and theoretical analysis is conducted on guiding the selection of feedback delay and gain to ensure the local ESP. The judicious selection of feedback gain and delay in self-ESN improves prediction accuracy and overcomes the challenges associated with obtaining optimal parameters of the reservoir in conventional ESN models. Numerical experiments are conducted to assess the long-term prediction capabilities of the self-ESN across various scenarios, including a 4D hyperchaotic system, a hyperchaotic network, and an infinite-dimensional delayed chaotic system. The experiments involve reconstructing bifurcation diagrams, predicting the chaotic synchronization, examining spatiotemporal evolution patterns, and uncovering the hidden attractors. The results underscore the capability of the proposed self-ESN as a strategy for long-term prediction and analysis of the complex systems.

I. INTRODUCTION

Chaos is recognized for its intriguing nonlinear characteristics, exhibiting complexity, ergodicity, and sensitivity to initial conditions [1,2]. Over the past few decades, chaos has found widespread applications in diverse fields, including electrical power systems, astronomy, finance, and network traffic [3–8]. As a subset of chaotic systems, the hyperchaos is defined by having at least two positive Lyapunov exponents, and the minimal dimension of the phase space embedded in the hyperchaotic attractor is greater than three [1]. Compared with conventional chaos, hyperchaos exhibits more complex dynamics featuring higher dimensions, increased randomness, and heightened unpredictability [9,10]. Consequently, understanding the overall system dynamics of hyperchaotic systems poses a particularly challenging problem.

In recent years, there has been a significant surge in the application of machine learning for the model-free analysis of chaotic systems [11–15]. Reservoir computing (RC) is a paradigm of machine learning in which the information-processing capabilities of dynamical systems are exploited for solving temporal tasks; it derived from recurrent neural networks but with the major advantage of low training cost and fast learning. It was independently proposed around the same time as an echo state network (ESN) by Jaeger [16] using conventional artificial neurons, and as a liquid state machine (LSM) by Maass *et al.* [17] using more biologically plausible spiking neurons. Reservoir computing is a promising architecture for a variety of time series prediction tasks that rely on short-term associations. Since RC exploits generic dynamical systems for computing, the concept of RC has been successfully extended into the quantum regime, and quantum reservoir computing [18,19] was proposed to exploit dynamical systems to solve nonlinear and temporal tasks [20–23].

As a particular technique of reservoir computing, echo state networks (ESNs) have been widely used for modeling and analyzing complex systems [24–26]. Notably, ESNs operate by being driven by input data, generating output data through a readout function. In contrast to conventional RNNs, ESNs only train the readout weights, while all other parameters remain fixed during the initial construction. This streamlined and rapid training process effectively addresses challenges associated with gradient explosion and disappearance in conventional RNNs trained using gradient descent algorithms [25]. Consequently, it becomes feasible to significantly reduce the computational cost of learning. ESN models have found widespread applications in addressing complex problems, including time series prediction, classification, nonlinear system control, and dynamic pattern recognition [27–34]. Thus it has become a valuable tool for model-free analysis of chaotic systems, such as attractor reconstruction, computing Lyapunov exponents, and generalized synchronization of the chaotic systems [35–37].

However, ESNs face limitations in predicting the long-term evolution of chaotic systems that heavily rely on initial

[*] Contact author: xuxu@jlu.edu.cn

conditions, especially in hyperchaotic and large spatiotemporal chaotic systems [12,38,39]. This becomes a critical concern for practical applications like secure communication and encryption/decryption, where precise and prolonged chaotic solutions are imperative [40,41]. The challenges stem from the sensitivity of the ESN reservoir to various hyperparameters, requiring numerous trial-and-error attempts for construction, and the computational load in determining optimal parameters [24]. Additionally, the insufficient memory capacity of ESN models and their emphasis on short-term memory constrain their ability to effectively address long-term evolution challenges in complex systems.

To address these challenges, several ESN models have been proposed to enhance memory performance. These include the deep time-delay reservoir computing model [42], hierarchical delay-memory echo state network [43], single-node delay-based reservoir computer [44], variable memory length echo state network [45], and leaky ESNs [46]. These advancements effectively enhance the memory capability of ESNs by integrating past information. However, existing models lack comprehensive theoretical analyses for selecting appropriate parameters and understanding their impact on performance.

This paper introduces a solution, the delayed self-feedback echo state network (self-ESN), to address the issues outlined above. Its primary objective is to devise an effective technique for achieving long-term behavior analysis of the future evolution of hyperchaotic systems based on available data. The key contributions are outlined as follows:

(1) *Incorporation of Delayed Self-Feedback:* We integrate a delayed self-feedback mechanism into the ESN model, enhancing its ability to simulate the delayed effects of neurons within the reservoir.

(2) *Explicit Use of Previous States:* The self-ESN introduces the past state of the reservoir to emphasize the impact of previous inputs on the current state, thereby improving the memory performance of the ESN model.

(3) *Theoretical Analysis:* This paper includes a comprehensive theoretical analysis that explores how time delay and feedback gain affect the system dynamics, providing valuable insights into the behavior of the self-ESN.

(4) *Improved Prediction Technique:* The self-ESN introduces a technique for achieving enhanced predictive performance, addressing challenges encountered in conventional ESNs in determining optimal reservoir parameters.

Through these contributions, the self-ESN offers a promising solution to the challenges associated with hyperchaotic systems, facilitating improved long-term predictions and contributing to a deeper understanding of the underlying theoretical aspects of the model.

II. ECHO STATE NETWORKS (ESN)

An echo state network is a specialized neural network architecture with an input layer, output layer, and reservoir layer consisting of a large number of discrete-time artificial neurons [24], as shown in Fig. 1. When there is no feedback from the output to reservoir, the reservoir states at time step t can be described as follows [12,24]:

$$\mathrm{x}_{t+1} = f(W_{\mathrm{in}}\mathrm{u}_{t+1} + W_{\mathrm{res}}\mathrm{x}_t), \tag{1}$$

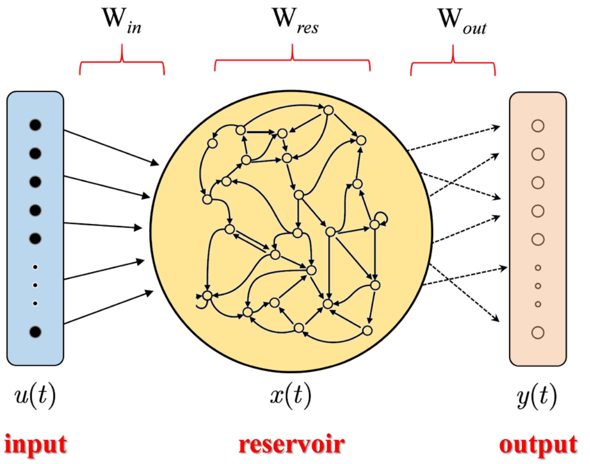


FIG. 1. An echo state network: only the readout weight W_{out} is trained, while all other parameters remain fixed at the initial construction.

where $\mathrm{x}_t \in \mathbb{R}^N$ is the reservoir state, $f(\cdot)$ is the activation function of the reservoir nodes satisfying $0 \leqslant |f(\cdot)| < 1$, $W_{\mathrm{in}} \in \mathbb{R}^{N_{\mathrm{res}}\times D}$ and $W_{\mathrm{res}} \in \mathbb{R}^{N_{\mathrm{res}}\times N_{\mathrm{res}}}$ are, respectively, the matrices for the input-reservoir and reservoir-reservoir connections, and $\mathrm{u}(t) \in \mathbb{R}^D$ denotes the external input.

The output of the network is described by $\mathrm{y}(t) = f_{\mathrm{out}}(W_{\mathrm{out}}[\mathrm{x}(t); u(t)])$, where W_{out} is the reservoir-output weight matrix, which is trained using a simple rule such as linear regression.

Remark 1. Only the readout weights W_{out} are trained, and other parameters are fixed at the initial construction.

The reservoir needs to have the echo state property to eliminate the effects of the initial conditions of the reservoir [12]. For an ESN with 1-Lipschitz activation function $f(\cdot)$, a necessary condition for the ESP is $\rho(W_{\mathrm{res}}) < 1$, where ρ is the spectral radius of W_{res}. In the real computation, W_{res} is usually rescaled by $W_{\mathrm{res}} = s/\rho W_{\mathrm{res}}$ with $0 < s < 1$.

III. DELAYED SELF-FEEDBACK ESN (SELF-ESN)

Due to scaling of the spectral radius of W_{res} to be slightly below 1, the network has a fading "echo" of the previous input. In order to improve memory performance, this paper proposes a delayed self-feedback ESN (self-ESN) by introducing time-delay state feedback of reservoirs, as shown in

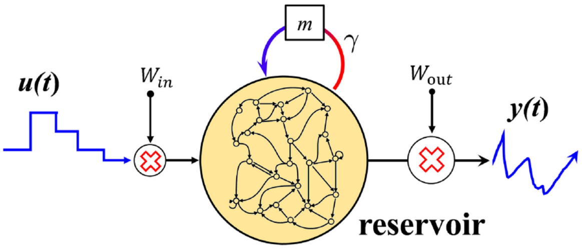


FIG. 2. The self-ESN model by introducing the delayed state feedback (with delay m and gain γ) of the reservoir; when $\gamma = 0$, it is degenerated to the conventional ESN.

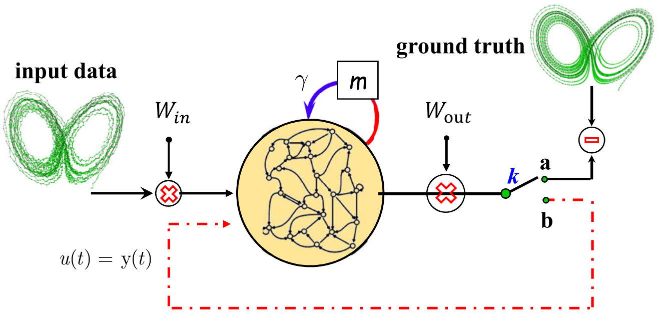


FIG. 3. Self-ESN for free-running prediction: solid black line (k connected to a) for training, dashed red loop (k connected to b) for prediction.

Fig. 2. Network dynamics is described by

$$\begin{aligned} x_{t+1} &= f(W_{\text{in}}u_{t+1} + W_{\text{res}}x_t + \gamma x_{t-m}) \\ \mathrm{y}_t &= f_{\text{out}}(W_{\text{out}}[\mathrm{x}_t; u_t]), \end{aligned} \tag{2}$$

where γx_{t-m} is the delayed self-feedback with gain γ and delay $m \geqslant 0$. The proposed self-ESN differs from the conventional ESN in several key aspects:

(1) The introduction of delayed self-feedback allows for a more accurate simulation of the interaction between neurons because the transmission speed of signals within neurons is limited.

(2) Delayed self-feedback establishes a connection between the current and previous m time steps of the reservoir state, offering an effective mean to capture the dynamical characteristics of the system. This enhancement contributes to improved memory performance in ESNs.

(3) In the absence of self-feedback ($\gamma = 0$), the self-ESN reverts to the conventional ESN, providing flexibility in model configuration.

(4) By judiciously selecting feedback gain γ and delay m, the self-ESN may enable better predictions, and thus mitigates the challenges in determining optimal reservoir parameters in conventional ESN models.

During training, $\mathrm{x}(t)$ are stacked into a state collection $\mathbf{X}$, and the desired output $\mathbf{y}_*(t)$ are stacked into a vector $\mathbf{Y}$. When the activation function of the output units is the identity function, W_{out} can be calculated by $\mathbf{W}_{\text{out}} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \xi\mathbf{I})^{-1}$, where $\mathbf{I}$ is the identity matrix, ξ is Tikhonov regularization parameter [12].

For time series prediction, the training target corresponds to the value of the series one step ahead $x(t+1)$. After undergoing sufficient training steps to achieve a low training loss, the output $y(t)$ is fed back into the reservoir input for a free-running prediction as shown in Fig. 3.

IV. LOCAL ECHO STATE PROPERTY

Roughly speaking, the echo state property (ESP) means that the state of the network is primarily influenced by the input data and is independent of the initial conditions of the reservoir [47]. Essentially, the ESP implies that the network should exhibit replicable and robust global attractors irrespective of the initial conditions of the reservoir. However, in practical applications, a crucial aspect of assessing the stability of network trajectories involves quantifying the impact of small perturbations. In this section, a localized version of ESP is defined as follows.

Definition 1. (Local ESP [47]). A dynamical system has the local ESP if for a small perturbation $\tilde{x}(t_0)$ of $x(t_0)$ at time $t_0 : \tilde{x}(t_0) = x(t_0) + \delta$, $\| \tilde{x}(t) - x(t) \| \rightarrow 0$ holds true for a sufficiently small δ.

Definition 1 ensures the trajectory's resilience to adequately small perturbations. This is a distinction from the conventional ESP, which guarantees the existence of a unique globally stable attractor. In practical applications, when training and prediction are aimed at the same attractor, the local ESP proves to be a satisfactory criterion [47].

Denote $\tilde{x}(t,\kappa) = W_{\text{in}}u_t + W_{\text{res}}x_t + \gamma x_{t-\kappa}$, then from Eq. (2) we have $x_{t+1} = f(\tilde{x}(t,\kappa))$. Denote $\delta t > 0$ as the time step, and then we have

$$(\delta t)^{-1}[x(t+1) - x(t)] = (\delta t)^{-1}[-x(t) + f(\tilde{x}(t,\kappa))].$$

If we denote $\beta = (\delta t)^{-1}$, the evolution of Eq. (2) is the forward Euler approximation to the system of the relation equation

$$\beta^{-1}\dot{x}(t) = -x(t) + f(\tilde{x}(t,\tau)), \tag{3}$$

where $\dot{x}(t) = dx(t)/dt$, $\tau = m\delta t$ is the time delay.

Let $e(t) = x(t) - z(t)$, where $x(t)$ and $z(t)$ are two reservoir states. Using Eq. (3) we have

$$\beta^{-1}\dot{e}(t) = -e(t) + f(\tilde{x}(t,\tau)) - f(\tilde{z}(t,\tau)). \tag{4}$$

Then the local asymptotical stability of zero equilibrium of Eq. (4) implies the local ESP of Eq. (2). We have the following results.

Theorem 1. If nonlinear activation function of the reservoir neurons satisfies $|f(\cdot)| \leqslant 1$, $f'(0) = 1$ and $\rho < 1$, then self-ESN has local echo state property if one of the following conditions is met:

(i) $\rho + |\gamma| < 1, \forall\tau \geqslant 0$

(ii) $\rho + |\gamma| > 1$, and γ is within the boundary $\chi(\tau)$ as shown in Fig. 4.

The proof of Theorem 1 and stability boundary $\chi(\tau)$ can be found in the Appendix.

For any $\tau \geqslant 0$, the delay-independent stable boundary $\chi(\tau)$ is a circle of radius $\kappa = 1 - \rho$ centered at the origin as shown in the red star curve in Fig. 4. At a finite delay, the stable boundary showing as the teardrop-shaped closed curve crosses the real axis in the positive half-plane at κ and in the negative half-plane at a delay-dependent value γ_*. Only the value of γ inside the stability boundary (the teardrop-shaped closed curve) is actually stable.

Remark 2. Condition (i) in Theorem 1 implies that self-ESN has the ESP for any given the time delay $\tau \geqslant 0$. It is the same ESP condition as that in the conventional ESP. However, the delay-dependent local ESP condition (ii) relaxes the conventional ESP condition.

Remark 3. From Fig. 4, it is clear that the larger the value of the time delay, the smaller the feedback coefficient value ensuring the local ESP.

Remark 4. Negative feedback coefficient γ can also ensure ESP. This is very different from the leaky ESN [46] and the hierarchical delay-memory ESN [43], where only the positive parameters are used.

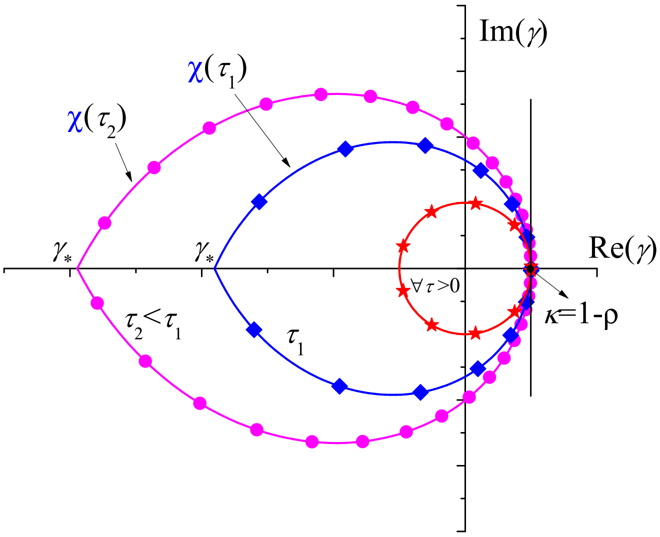


FIG. 4. Stability boundary $\chi(\tau)$ for feedback gain γ in the complex plane, where the red star curve is the delay-independent boundary, teardrop-shaped closed curves are the delay-dependent boundaries, and purple circle curve and blue square curves represent the stability boundaries for two time delays τ_1 and τ_2 with $\tau_2 < \tau_1$.

V. MEMORY PERFORMANCE

From Eq. (2), using the fact that activation function $|f(\cdot)|$ is bounded between 0 and 1, we have

$$x_{t+1} \leqslant \|W_{\text{in}}u_t\| + \rho\|x_t\| + |\gamma|\cdot\left\|x_{t-m}\right\|. \tag{5}$$

Applying Eq. (5) repeatedly yields

$$\begin{aligned} x_{t+1} &\leqslant \|W_{\text{in}}u_t\| + \rho\|W_{\text{in}}u_{t-1}\| + |\gamma|\|W_{\text{in}}u_{t-2}\| + \cdots \\ &= \sum_{l=0}^{+\infty}\sum_{k=0}^{l} C_l^k\rho^{l-k}|\gamma|^k\|W_{\text{in}}u_{t-(km+l)}\|. \end{aligned} \tag{6}$$

Obviously, the output of the reservoir depends on all inputs before the current time. In addition, if $0 < \rho < 1$ and $|\gamma| < 1$, then there exists an appropriately positive integer L such that $\rho^j \to 0$ and $|\gamma|^j \to 0$ when $j > L$. Therefore, we have

$$x_{t+1} \leqslant \sum_{l=0}^{L}\sum_{k=0}^{l} C_l^k\rho^{l-k}|\gamma|^k\|W_{\text{in}}u_{t-(km+l)}\| + o(\rho^L + |\gamma|^L),$$

where $o(\cdot)$ represents the higher order infinitesimal.

So we have

$$x_{t+1} \leqslant \rho^L\|W_{\text{in}}u_{t-L}\| + |\gamma|^L\|W_{\text{in}}u_{t-L(m+1)}\| + \cdots.$$

In the above equation, the first term on the right side indicates that the reservoir in the conventional ESN model ($\gamma = 0$) can effectively memorize input samples that are L steps away from the current time. However, the second term on the right side indicates self-ESN ($\gamma \neq 0$) can effectively memorize input samples that are $L(m+1)$ steps away from the current time as shown in Fig. 5. Obviously, the introduction of the delayed feedback improves the memory performance of ESN.

Jaeger used the memory capacity (MC) to evaluate the capabilities of ESN for recalling the history of the input data [16]. MC is defined by $MC = \sum_{k=1}^{\infty} MC_k$, where

$$MC_k = \frac{\text{cov}^2(u(n-K), y_k(n))}{\sigma^2(u(n))\sigma^2(y_k(n))}$$

is the squared correlation coefficient between the input $u(n-K)$ and the reconstructed value $y_k(n)$, "cov" denotes covariance, σ^2 is the variance, and K is the delay time steps.

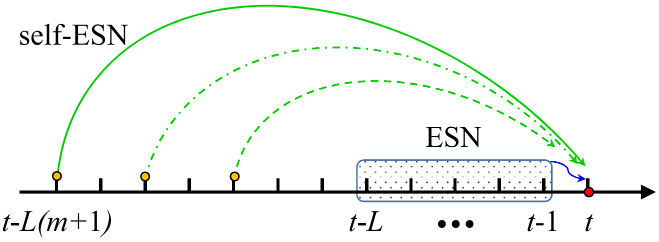


FIG. 5. Comparison of the memory performance between ESN and self-ESN.

We consider the input data whose elements satisfy the uniform distribution over $[-1, 1]$. The input contained 5000 data with the first 500 data for warm-up, the following 3500 for training, and the remaining 1000 for the test. It is noted that Eq. (2) is a discrete dynamic system, and the state $x(t)$ of neurons in the reservoir is affected by their initial value x_0. In order to eliminate the influence of the initial state of the reservoir, warm-up processing is required, which means discarding a certain amount of data from the beginning of the training process.

In particular, we considered a number of input delays up to 100, and $\rho = 0.9$, $N_{\text{res}} = 500, 800$, $\gamma = -0.4$, and $m = 4$. Figure 6 shows the "forgetting curve," a plot of the memory capacities of K delay input. Obviously, traditional ESNs exhibit almost 100% recall ability for input delays up to 13, followed by steep slopes descending in a clif-like manner, as shown by the black star curve. This clearly indicates the short-term memory capacity of the ESN. However, the

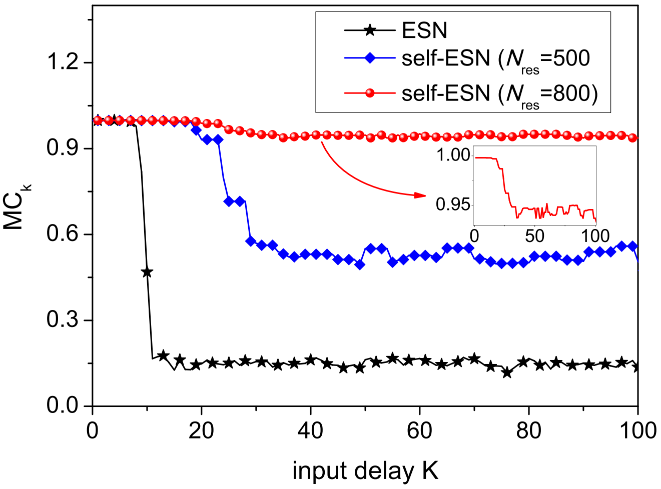


FIG. 6. The "forgetting" curve of the trained network in ESN (black star curve) and self-ESN (blue square curve for $N_{\text{res}} = 500$, and red circle curve for $N_{\text{res}} = 800$). This indicates that the self-ESN has excellent memory performance.

self-ESN exhibits a close-to-100 recall for delays up to 18, which is higher than conventional ESN. As the input delay increases, its memory capacity gradually decreases. However, unlike ESN, the MC value of self-ESN does not decrease in a clif-like manner, but exhibits a gradually decreasing behavior in a stepped manner as shown by the blue square curve in Fig. 6. Ultimately, the memory capacity stabilizes around $MC_\infty \approx 0.55$, which is much greater than the $MC_\infty \approx 0.15$ in ESN. Additionally, from this figure, when the number of nodes in the reservoir is 800, the value of MC_k is almost equal to 1 for input delay up to 100, as shown in the inset in Fig. 6. This indicates that the self-ESN has excellent memory performance.

VI. NUMERICAL EXPERIMENTS

Several numerical experiments are conducted to assess the effectiveness of the self-ESN. In each of the cases, the input $\mathrm{u}(t)$ and desired data $\mathrm{y}_*(t)$ are both normalized to have mean 0 and standard deviation 1. A reservoir network with nodes $N_{\mathrm{res}} = 500$ is considered, where the elements in W_{in} and $\tilde{W}_{\mathrm{res}}$ were obtained from a random uniform distribution with values between -0.5 and 0.5. The symmetric reservoir matrix is then obtained by $W_{\mathrm{res}} = 0.5(\tilde{W}_{\mathrm{res}} + \tilde{W}_{\mathrm{res}}^{\mathrm{T}})$ to guarantee the conditions of Theorem 1. The sparsity is $\eta = 0.1$, and the regularization factor is $\xi = 10^{-8}$. In order to provide fair and objective results, the experimental result is the average of 10 random reservoir initialization attempts. In the experiments, the following root-mean-square error (RMSE)is used to measure the performance of the models:

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum\nolimits_{t=1}^{T}[y_*(t) - y(t)]^2},$$

where $y_*(t)$ and $y(t)$ are, respectively, the desired and predicted values, and T is the prediction time steps.

A. A 4D hyperchaotic system

We consider the following hyperchaotic system [12]:

$$\begin{aligned}\dot{x} &= a(y - x)\\ \dot{y} &= ax - az^2 - u\\ \dot{z} &= x^2 - bz\\ \dot{w} &= my,\end{aligned} \tag{7}$$

where $a = 20$, $b = 3$, $m = 36$. The Lyapunov exponents of the system are $\lambda_1 = 0.89$, $\lambda_2 = 0.40$, $\lambda_3 = 0$, and $\lambda_4 = -24.2$ [12]. So the system is in a hyperchaotic state as there are two positive Lyapunov exponents, and the sum of all exponents is less than 0.

The Runge-Kutta method [12] is used to solve Eq. (7) and obtain the training and test data. The effectiveness of the Runge-Kutta method in solving complex differential systems has been proven by a large amount of work [12], even for a large time span. In this experiments, we uses the classical four-order Runge-Kutta algorithm to integrate chaotic system (7), where the relative tolerance error is 1.0×10^{-8}, the absolute tolerance error is 1.0×10^{-10}, the initial step size is 0.01, and the maximum step size is 0.1, and the Dormand-Prince algorithm is used to adaptively select the variable step size [12]. We generate the time series containing 10 000 data points, where the first 1000 samples are used for warm-up, 4000 data are as the input $\mathrm{u}(t)$ for training the reservoir, and the following 5000 data as the desired output $\mathrm{y}_*(t)$. Both input and output are four-dimensional data (x_k, y_k, z_k, w_k), =1, 2,...,10 000.

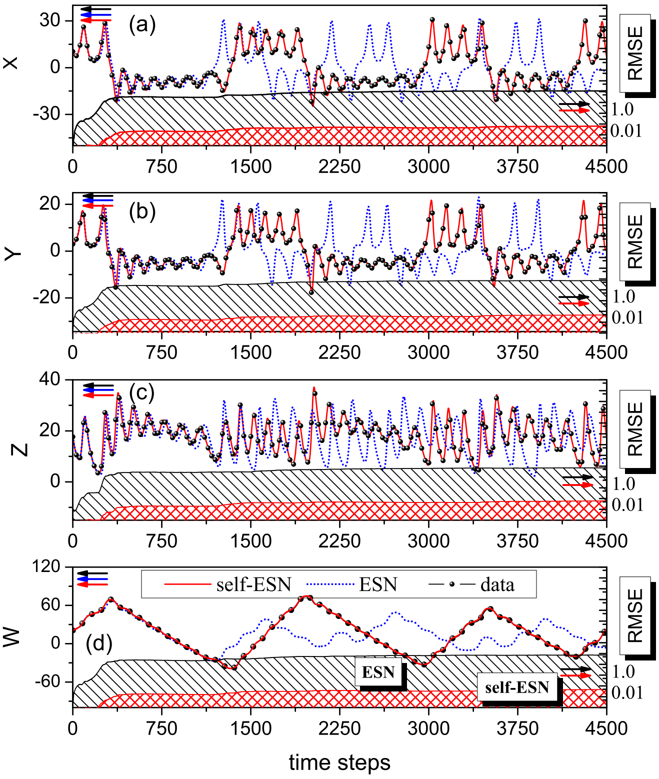


FIG. 7. Prediction curve in the x, y, z, and w directions. The black slash and red grid lines represent the prediction errors of ESN and self-ESN, respectively. Obviously, self-ESN can exhibit the long-term dynamical behavior as actual chaotic systems.

By choosing $\rho = 0.55$, $\gamma = -0.4$, and $m = 1$, we illustrate the prediction curve and RMSE using self-ESN and traditional ESN as shown in Fig. 7. Each subgraph in Fig. 7 has two vertical axes, with the left axis representing the system's state and the right axis representing the system's RMSE. Due to the sensitivity of chaotic system to initial conditions, ESN produces good predictions (RMSE < 0.1) when the discrete time step is less than about 500, but diverges from the actual data as time evolves. However, the predicted trajectory from self-ESN can accurately track the actual chaotic data with RMSE $< 10^{-2}$ when the discrete time step is less than about 4500. This example indicates that self-ESN can exhibit the same long-term dynamical behavior as actual chaotic systems. In addition, a comparison of predicting results between ESN and self-ESN in the phase plane (see Fig. 8) further confirms the superior long-term predictive abilities of self-ESN.

To illustrate the impact of the time delay m and gain γ on prediction performance, the predicted errors in the m-γ plane are depicted in Fig. 9, where the color map represents the RMSE, with bluer colors indicating smaller RMSE values. The observations from the figure are as follows: (1) self-ESN consistently yields better prediction results across

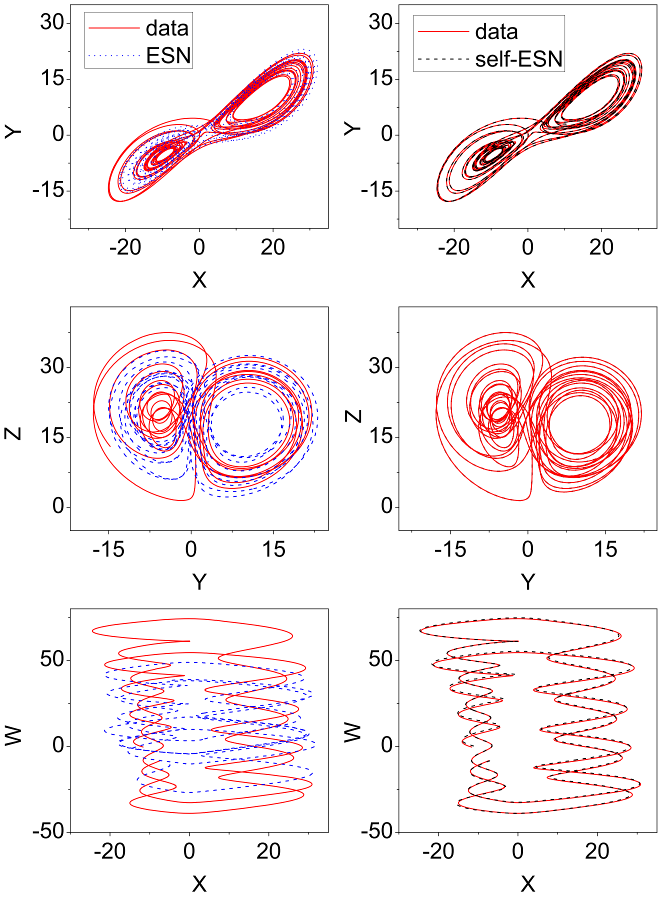


FIG. 8. Comparisons of the prediction results in the phase plane. The results in the left column are obtained from the ESN, while the right column are from self-ESN. Obviously, self-ESN exhibits the same long-term dynamic behavior as actual chaotic systems, while ESN cannot.

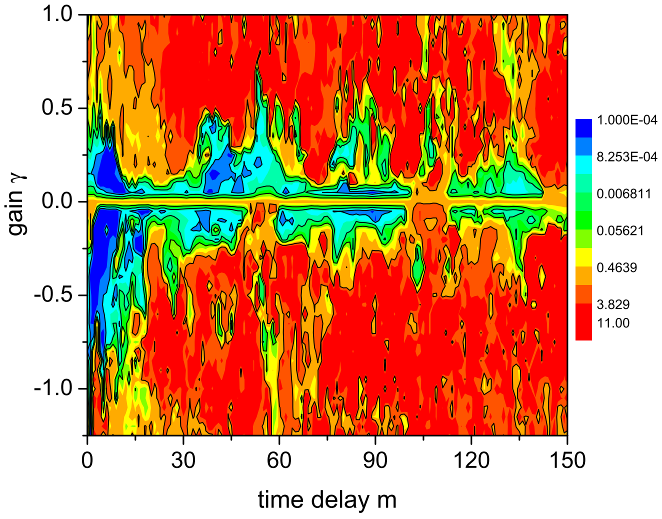


FIG. 9. Color maps of the predicted errors in the m-γ plane: the darker (bluer) the color, the smaller the RMSE.

a wide range of parameters (γ, m), evident in the green and blue regions where RMSE is below 0.1. This suggests that selecting parameters (γ, m) for optimal prediction results is more achievable in self-ESN. (2) With increasing delay, regions with higher prediction accuracy (blue regions) gradually decrease, indicating that a larger delay negatively impacts prediction accuracy. (3) In some areas with negative feedback gain, self-ESN has better prediction accuracy. This is different from some existing methods, such as the leaky ESN [46] and the hierarchical delay-memory ESN [43], where the parameters used are usually positive. (4) The better RMSE is often located at the regions with small delay and small feedback gain.

Remark 5. In real computation, we mesh the parameters m and γ by $m_i = i(m_{\max} - m_{\min})/M$ and $\gamma_j = j(\gamma_{\max} - \gamma_{\min})/M$ where $i, j = 1, 2, \ldots, M$, and get the grid points (m_i, γ_j) on the m-γ plane. By computing the short-term RMSE on the grid (m_i, γ_j), we know that a better RMSE implies the optimal parameters (m, γ). Using these optimal parameters, we can predict the long-term evolutionary behavior of the system. So the self-ESN offers a straightforward approach to improve accuracy by simply adjusting the parameters m and γ. This contrasts sharply with traditional ESN methods ($\gamma = 0$) that achieving optimal parameters involves high computational costs due to the ESN's performance being highly sensitive to reservoir parameters, such as spectral radius.

B. A hyperchaotic network with disturbed data

A small world network with six nodes is considered for the synchronization analysis. The motion equation of ith node is described by

$$\dot{u}_i(t) = f_i(u_i) + \sum_{j=1}^{N} a_{ij} u_j(t) + k(u_i - u_0), \qquad (8)$$

where $u_i(t) = (u_{i,x}, u_{i,y}, u_{i,z})^T \in \mathbb{R}^3$ is the state vector of node i; $i \in Z_+ \equiv \{1, 2, \ldots, 6\}$, nodes 1 and 4 are set as the Chua oscillator, and nodes 2, 3, 5, 6 are the Lorenz oscillator, u_0 is the solution of a separated Lorenz oscillator [7]. The topology and parameters of the network are shown in Fig. 10(a).

According to Ref. [7], two Chua oscillators 1 and 4 are synchronized, while the other four Lorenz nodes are also synchronized. The synchronization manifolds are shown in Fig. 10(b). In addition, by simple computation, we can get two sets of Lyapunov exponents: 0.0, −0.199, 0.312 for node 1 and 4, and −0.476, 0.0, 0.151 for nodes 2, 3, 5, and 6. Therefore, there are six positive Lyapunov exponents. So the system is in the hyperchaotic state.

From Eq. (8), we generate time series containing 12 000 data points using the Runge-Kutta method, the details of which can be found in experiment A. When the discrete time step is smaller than 4000 steps, we applied a disturbed chaotic signal $u_0(t)$ on the nodes 2 and 4 in the Z direction to investigate whether self-ESN method can reconstruct chaotic synchronization from input data with such disturbed signals [see Fig. 10(c)]. The purpose of adding noise to the training data is to consider whether the proposed method can learn the inherent dynamics of systems with noise. The first 3000 samples are used for warm-up, the subsequent 6000 data points

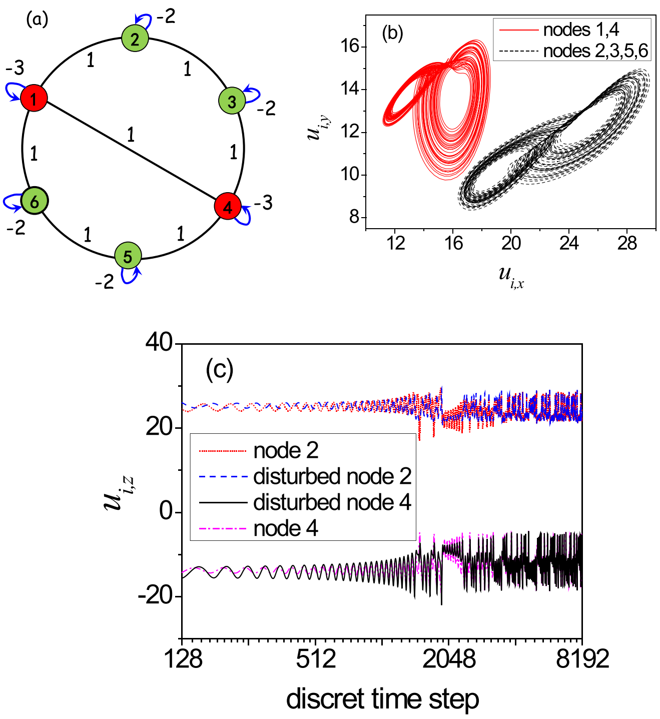


FIG. 10. A small world network: (a) schematic; (b) two attractors; (c) disturbed training data on nodes 2 and 4.

serve as input $u(t)$ for training the reservoir, and the remaining 1000 data points are utilized as the desired output to evaluate prediction performance.

By choosing $\rho = 0.95$, $\gamma = -0.2$, and $m = 5$, Figs. 11(a)–11(c) illustrate the system's evolution pattern, where the horizontal axis signifies the evolution time, the vertical axis represents the spatial position of the nodes, and colors indicate predicted values. Figure 12 visualizes the error between the predicted and actual solutions, with greener areas denoting smaller errors. For discrete time steps less than 200, ESN provides accurate predictions, as evidenced by the green regions in Fig. 12(a), but diverges from the actual trajectory as time evolves. However, the predicted trajectory from self-ESN accurately tracks the actual chaotic curve even with discrete time steps less than about 700 as shown in Fig. 12(b). Moreover, even with time steps exceeding 700, synchronization behavior can still be exhibited using self-ESN, as nodes 1 and 4, 2 and 5, and 3 and 6 maintain the same pattern [see Fig. 11(a)]. In contrast, ESN fails to predict synchronization behavior, as nodes 1 and 4 and 2 and 5 exhibit markedly different behaviors after time step 250 [see Fig. 11(c)]. Clearly, the proposed self-ESN demonstrates superior predictive ability for synchronization in complex systems with disturbed signals.

In a similar way to Fig. 9, the predicted errors of the chaotic network in the m-γ plane are depicted in Fig. 13, where the color map represents the RMSE, with bluer colors indicating smaller RMSE values. The observations from the figure indicate that self-ESN consistently yields better prediction results across a wide range of parameters (γ, m) as shown in the blue regions where RMSE is below 0.1. This also indicates that selecting parameters (γ, m) is more achievable in self-ESN for optimal prediction results. This transforms the difficulty of

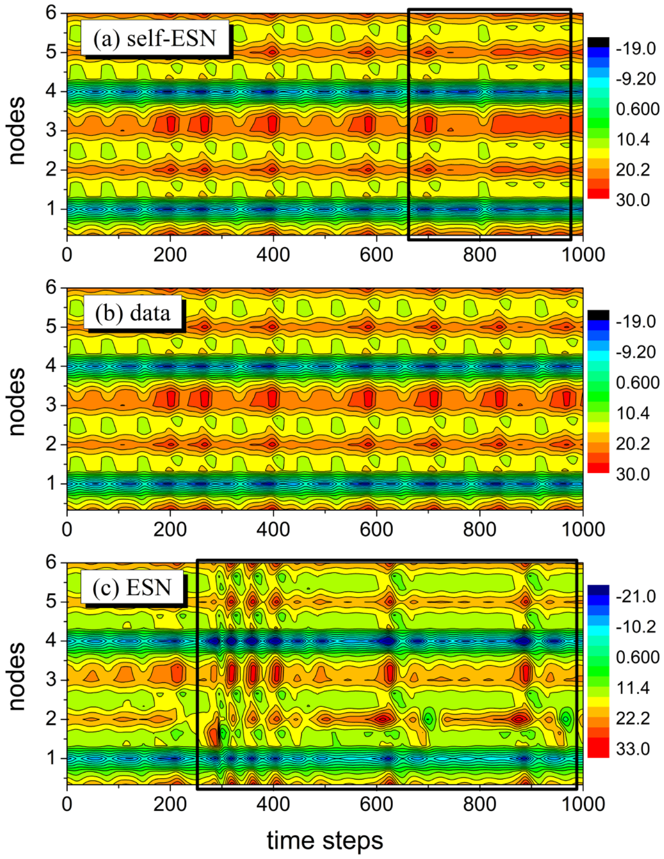


FIG. 11. Prediction of synchronization pattern of the complex networked systems: (a) self-ESN, (b) actual data, and (c) ESN. The black rectangles in (a) and (c) show the difference between the ESN and self-ESN with the actual data. Obviously, the self-ESN can exhibit synchronous behavior and better prediction over a long period of time but the ESN cannot.

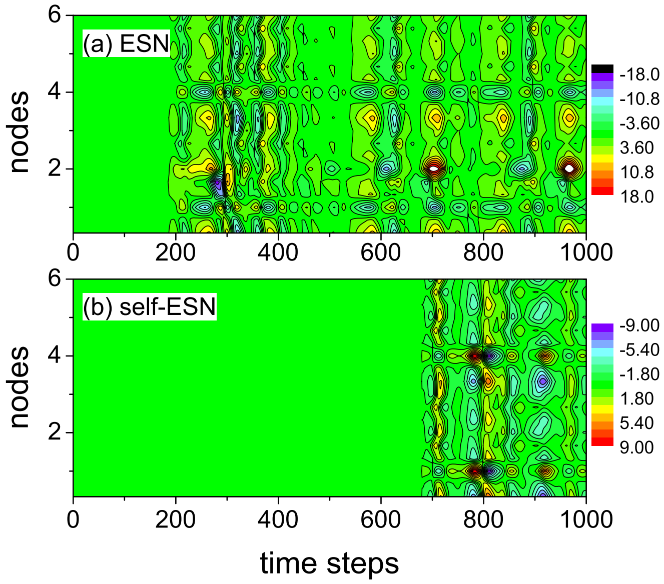


FIG. 12. Error between the predicted and actual solutions, with lighter (greener) areas denoting smaller errors.

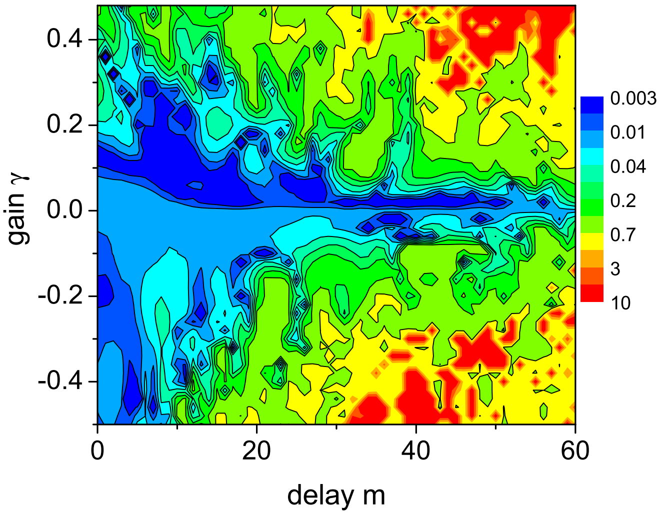


FIG. 13. Predicted errors of the hyperchaotic network in the m-γ plane. The colors indicate the predicted values from self-ESN with the darker (bluer) the color indicating the smaller the RMSE.

obtaining optimal reservoir parameters in conventional ESN into the problem of searching for parameters (γ, m). It is clear from Fig. 9 and Fig. 13 that there are large continuous regions on the γ-m plane where self-ESN has better prediction accuracy. Therefore, it is not difficult to obtain the optimal parameters (γ, m) through a grid search technique as discussed in Fig. 9. In addition, due to a larger delay negatively impacting prediction accuracy, the range of empirical search parameters is $m \in [0, 30]$ and $\gamma \in [-0.5, 0.5]$.

C. An infinite-dimensional chaotic system

We consider a nonlinear spring-mass system with negative damping and the delayed feedback as described by [48]:

$$\begin{aligned} &\ddot{x}(t) + cx_\tau + kx_\tau + bx_\tau^3 + Ax_T x(\theta) = x_0(t), \\ &\dot{x}(\theta) = \tilde{x}_0(\theta), \theta \in [-\max(\tau, T), 0], \end{aligned} \quad (9)$$

where $x(t)$ denotes displacement, $x_\tau = x(t-\tau)$, $k > 0$ is the stiff, $c \leqslant 0$ is the negative damping parameter, $|b| << k$, $\tau \geqslant 0$ is the inner delay, A and $T \geqslant 0$ are the feedback gain and delay, respectively, and $x_0(t)$ and $\tilde{x}_0(t)$ are initial functions defined in the interval $[-\max(\tau, T), 0]$.

The delayed system (9) is characterized as an infinite-dimensional system due to the initial condition being a continuous function defined in the $[-\max(\tau, T), 0]$. Consequently, the number of positive Lyapunov exponents is no longer constrained by the dimensionality of the system; rather, it becomes uncountable because the initial function is infinite. This attribute highlights that delayed chaotic systems can significantly augment the complexity of a system without necessitating an increase in its dimensionality.

D. Reconstruction of bifurcation diagram

From Ref. [48], when $k = 3, c = -0.03, \tau = 0.45, b = 0.1, T = 0.83$, system (9) exhibits period-doubling bifurcation as A varies from $A_b = 3.9$ to $A_e = 4.6$. The bifurcation diagram from the actual data is shown in Fig. 14(a) [48].

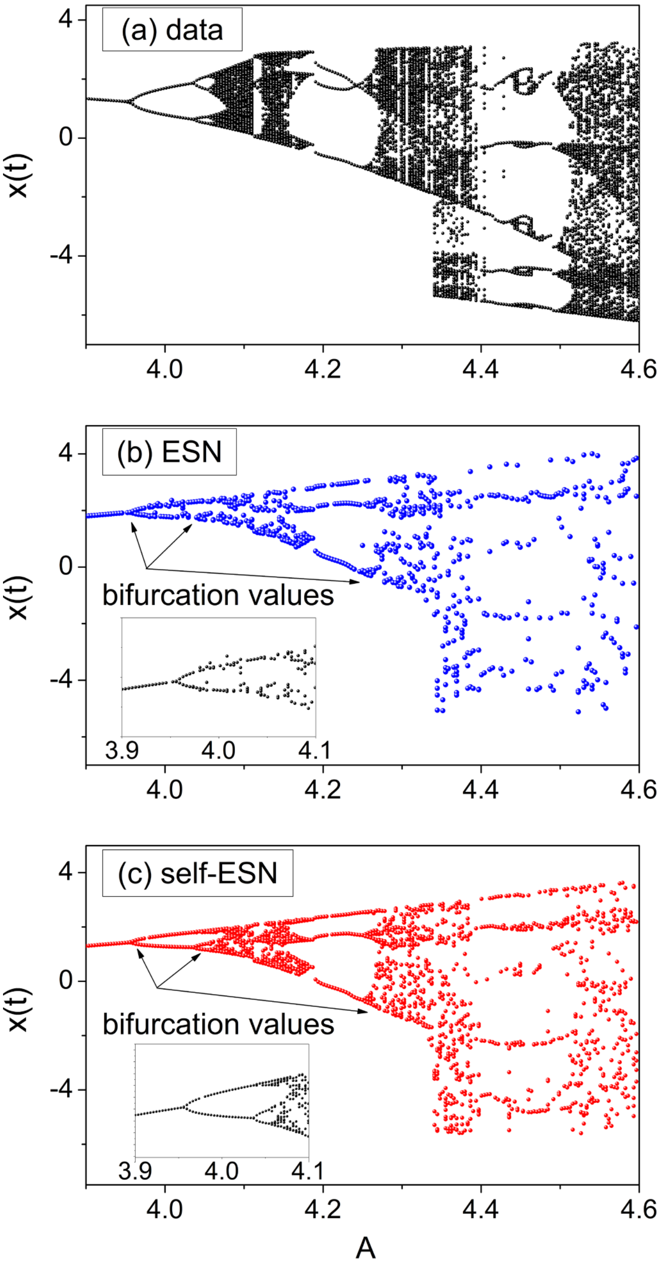


FIG. 14. Bifurcation diagram of system (9) on the Poincaré section $\sum = \{(A, x) : (\dot{x} = 0,\ \ddot{x} > 0)\}$.

We use the self-ESN to reconstruct the period-doubling bifurcation diagram, and make a comparison with conventional ESNs. We first choose grid spacing $A_i = i(A_e - A_b)/(M-1)$. In each A_i $(i = 1, 2, \dots, M)$, using the Runge-Kutta method, we generate time series containing 12 000 data points from the initial condition $(x_0, y_0) = (0.1, 0, 01)$. The initial 1000 samples are designated for warm-up, while the subsequent 3000 data points serve as input for training. The ensuing data are employed as the desired output to evaluate prediction performance. The parameters in self-ESN models are $\rho = 0.55$, $\gamma = 0.2$, and $m = 6$, and all others are the same as those at the beginning of Sec. VI.

In Figs. 14(b) and 14(c), the predicted bifurcation diagram from ESN and self-ESN are depicted. Clearly, both ESN and

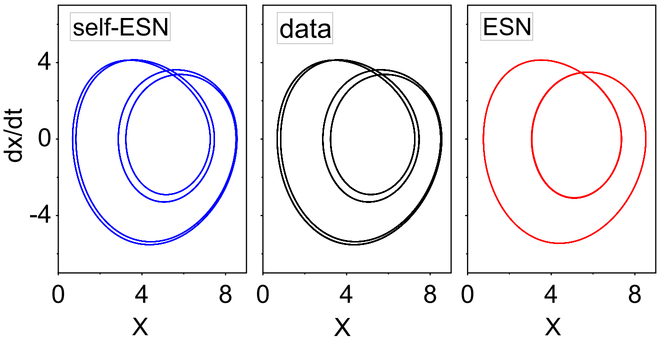


FIG. 15. Prediction of period-4 solution when $A = 4.04$, where self-ESN can predict the period-4 solution exhibited by the actual data, but ESN predicts the period-2 solution.

self-ESN successfully reconstruct period-doubling bifurcation diagrams. However, ESN falls short in accurately predicting bifurcation values as shown in the inset in Fig. 14(b), potentially hindering its ability to faithfully represent the dynamical behaviors of the actual system near bifurcation points, for instance, in the vicinity of the bifurcation point $A = 4.04$, where Eq. (9) exhibits a period-4 solution, but ESN predicts only a period-2 solution, as shown in Fig. 15. A similar discrepancy occurs at $A = 4.275$, where Eq. (9) displays chaotic behavior that ESN fails to capture, as illustrated in Fig. 16. In contrast, self-ESN demonstrates the capability to precisely predict dynamical behaviors near bifurcation points, as detailed in Fig. 15 and Fig. 16. This underscores the superior performance of the proposed model in bifurcation diagram reconstruction.

For values of A at 4.0, 4.05, 4.2, and 4.3, Eq. (9) exhibits period-2, period-4, period-3, and chaotic solutions [48]. Employing the parameters in Fig. 14, we visualize the spatiotemporal patterns of the systems onto the t-index plane using the self-ESN, as depicted in Fig. 17, where the horizontal axis signifies the evolution time, and the vertical axis represents the index of $(x(t), dx/dt)$ under different parameters A. For example, the three consecutive scales on the vertical axis represent the index of $(A_i, x_i(t), dx_i/dt)$. The black lines denote contour lines representing the values of $(A, x(t), \mathrm{d}x/dt)$ generated by the actual data, and the color map illustrates the results generated by the self-ESN. It clearly reveals the period-2, period-4, and period-3 solutions (as shown in the black circles) and the chaotic solution (as shown in the black rectangle).

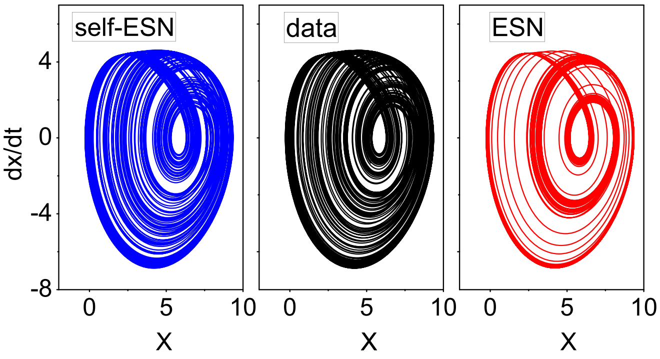


FIG. 16. Prediction of chaotic solution when $A = 4.275$, where self-ESN can predict the same chaotic solution with the actual data but ESN cannot.

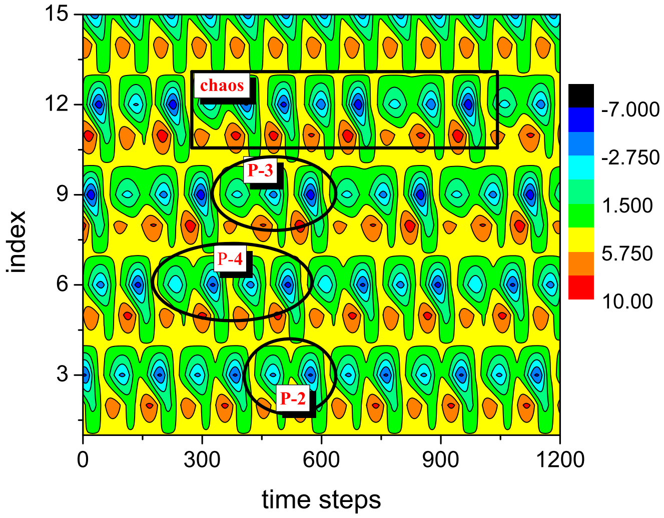


FIG. 17. Spatiotemporal pattern of the systems using self-ESN: the period-2, period-4, period-3, and chaotic behavior are clearly shown in the circles and rectangle.

In the following, we check the chaotic motion by estimating the maximal conditional Lyapunov exponent (CLE). The CLE measures the convergence or divergence rate of the system. If its maximal value is positive, the system exhibits the chaotic behavior. We denote the solutions of two identical systems under the different initial conditions as $v(t, \varphi_0)$ and $\tilde{v}(t, \tilde{\varphi}_0)$. Then the distance between them is $d(t, \varphi_0) = \sup \|v(t, \varphi_0) - \tilde{v}(t, \tilde{\varphi}_0)\| \approx c \exp(\lambda_{\max} t)$, where $\lambda_{\max}$ determines the numerically approximated CLE.

It is noted that the delayed system (9) is an infinite-dimensional systems, and the number of positive Lyapunov exponents is infinite. Figure 18 illustrates the maximal CLE of the predicted time series $x(t)$ at initial conditions (0.1, 0.01),

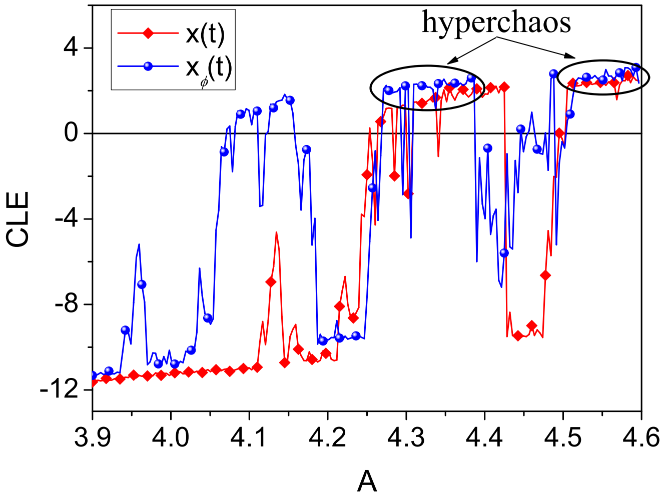


FIG. 18. Numerical estimations of conditional Lyapunov exponents using self-ESN.

and $x_{\varphi}(t)$ at another initial condition $(\sin(\theta), -\cos(\theta))$ with $\theta \in [-\max(\tau, T), 0]$. The red square curve and blue circle curve represent the CLEs from $x(t)$ and $x_{\varphi}(t)$, respectively. It is clear that the system exhibits the hyperchaotic behavior since there are two positive CLE when $4.25 < A < 4.35$ and $4.5 < A < 4.6$ as shown by the black circle in the figure.

E. Uncovering hidden attractors

Subsequent to obtaining W_{out} from the input data at the specified initial conditions $(0.1, 0.01)$, we investigate the feasibility of using this weight to predict the hidden dynamical behavior. The prediction of hidden attractors is carried out using the following equation:

$$\begin{aligned} x_{t+1} &= f(W_{\text{in}} \mathrm{y}_t + W_{\text{res}} x_t + \gamma x_{t-m}) \\ \mathrm{y}_{t+1} &= W_{\text{out}} \mathrm{x}_{t+1}. \end{aligned} \tag{10}$$

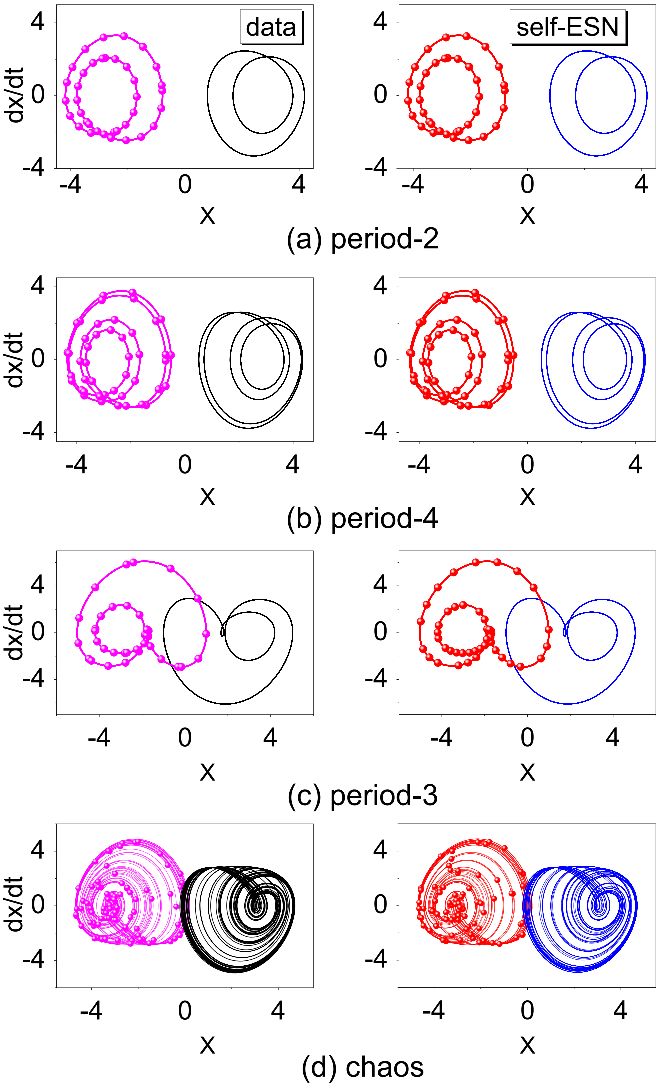
Unlike Eq. (2), the first equation of Eq. (10) uses y_t for the free run prediction instead of u_{t+1}, where $\mathrm{y}_0(\theta) = (A, \varphi_0, \hat{\varphi}_0)$ with $\theta \in [-\max(\tau, T), 0]$ and $(\varphi_0, \hat{\varphi}_0)$ is the initial condition $(-0.5, 1.4)$.

When $A = 4.0, 4.05, 4.15, 4.25$, using Eq. (10), we predict the hidden period-2, period-4, period-3 and a chaotic attractor as shown in Figs. 19(a)–19(d). In each subfigure, the left column depicts the attractors derived from the actual data, while the right column shows the predicted results. The blue solid attractors represent the predictions derived from training data under the original condition $(0.1, 0.01)$, and the red dotted ones denote the hidden attractors under the original condition $(0.5, -1.4)$. Remarkably, the predicted hidden attractors closely align with the actual data, demonstrating a high level of accuracy in tracking the dynamics.

Utilizing the self-ESN, Fig. 20 presents the predicted hidden bifurcation diagram on the Poincaré section as A varies from 3.9 to 4.6. Notably, two distinct period-doubling bifurcation processes emerge for different initial conditions. For parameter $A < 4.25$, these bifurcation processes lack wrapping regions, leading to the coexistence of multiperiod solutions and two separate single-scroll-like chaotic attractors, as illustrated in Fig. 19. When $A > 4.25$, two period-doubling processes develop wrapping regions, resulting in the merging of two initially separated single-scroll-like attractors into a double-scroll-like attractor, as depicted in Fig. 21, where Fig. 21(a) shows the phase plane diagram, and Fig. 21(b) presents the Poincaré section diagram $\sum = \{(x(t-\tau), x) : \dot{x} = 0, \ddot{x} > 0\}$. Importantly, this highlights the capability of self-ESN to uncover more intricate hidden dynamics. The hidden dynamics observed align with those of the actual system discussed in Ref. [48].

Furthermore, due to the chaotic system's sensitivity to initial conditions, ESN produces accurate predictions only when the discrete time step is less than about 400. In contrast, the predicted trajectory from self-ESN accurately tracks the actual chaotic motion even with about 1200 time steps as shown in Fig. 22. This example underscores that the proposed self-ESN can reconstruct the same dynamic behavior with the more complex chaotic attractor, even over the long term.

Remark 6. We only compared self-ESN with conventional ESN model for the sake of simplification. In fact, delayed state self-feedback can be introduced into any improved ESN



(a) period-2

(b) period-4

(c) period-3

(d) chaos

FIG. 19. Predicted hidden attractors, where the left column is from the actual data and the right column is from self-ESN where the blue-solid attractors represent the predictions derived from actual data under the original condition $(0.1, 0.01)$, and the red circle ones denote the uncovered hidden attractors under the original condition $(0.5, -1.4)$.

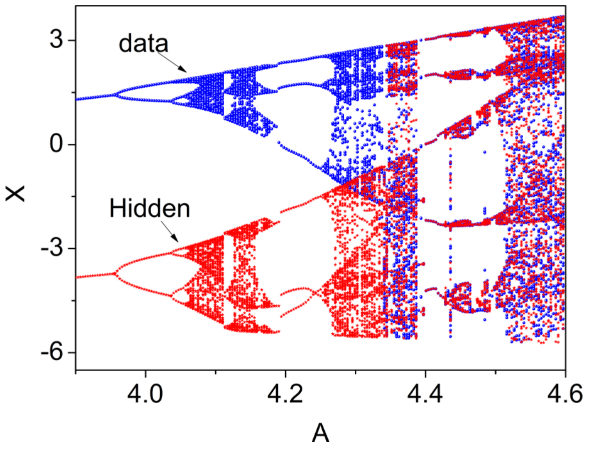
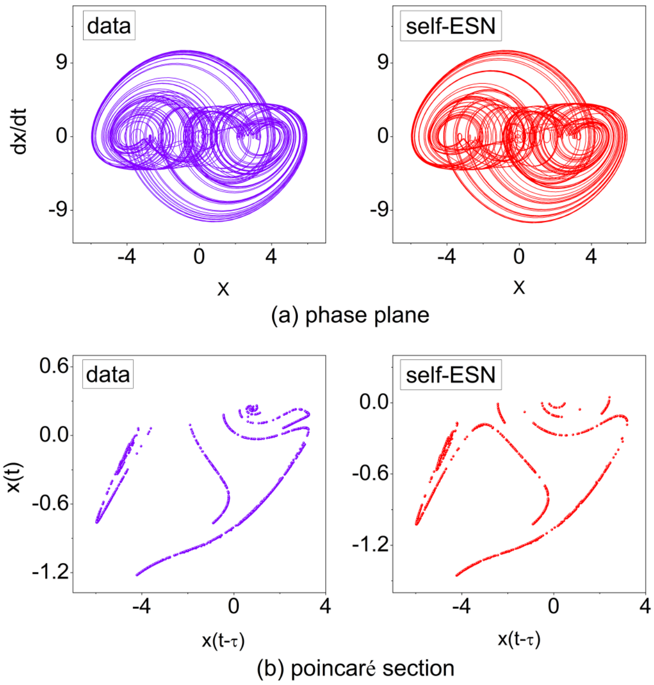


FIG. 20. Coexisting bifurcation diagrams on the Poincaré section $\sum = \{(A, x) : (\dot{x} = 0,\ \ddot{x} > 0)\}$.



(a) phase plane

(b) poincaré section

FIG. 21. Hidden double-scroll-attractor using self-ESN.

models to fully demonstrate the dynamic and memory performance of actual systems.

VII. CONCLUSION

This paper develops a self-ESN model for the long-term evolution behaviors of the hyperchaotic systems. By incorporating delay state feedback into the reservoir, self-ESN employs an explicit representation of the previous state of the reservoir to emphasize the impact of past inputs on the current state, significantly improving memory performance and potentially obtaining long-term evolutionary behavior of the system. In addition, the concept of the local echo state property is introduced, and theoretical analysis is conducted on how to select feedback gain and delay for ensuring local ESP. The self-ESN thus offers a technique for achieving superior prediction accuracy by selecting feedback delay and gain, which sidesteps the challenges in obtaining optimal parameters in conventional ESN models. The effectiveness of ESNs has been validated in a hyperchaotic system, the complex chaotic networks, and an infinite-dimensional delayed chaotic systems. The results indicate that self-ESN effectively analyzes the long-term evolutionary behavior of hyperchaotic systems and has potential application in the challenging field of complex system. Future research efforts will concentrate on two main areas: (1) developing a strategy for the automatic selection of the parameters γ and m and (2) applying the proposed self-ESN to analyze large spatiotemporal chaotic systems and complex real-world systems.

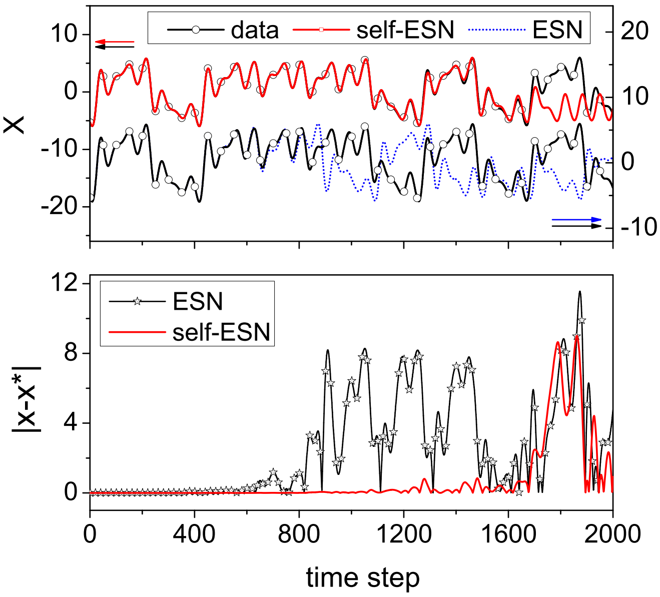


FIG. 22. Time history and RMSE of the hidden double-scroll-attractor using self-ESN.

ACKNOWLEDGMENTS

This work was supported in part by the National Science and Technology Major Project under Grant No. 2021ZD0112500, the National Natural Science Foundation of China (NSFC) under Grant No. 12072128, and the State Key Laboratory of Mechanics and Control for Aerospace Structures at Nanjing University of Aeronautics and Astronautics under Grant No. MCAS-E-0224K01. X.X. thanks Prof. Zaihua Wang from Nanjing University of Aeronautics and Astronautics in China for his valuable comments and suggestions.

APPENDIX: PROOF OF THEOREM 1

To investigate the local ESP, for the very close initial value $x(0)$ and $z(0)$, linearizing Eq. (4) around the zero equilibrium yields

$$\beta^{-1}\dot{e}(t) = -e(t) + W_{\text{res}}e(t) + \gamma e(t-\tau). \tag{A1}$$

Assumption 1. All the eigenvalues of W_{res} are simple real roots.

Under the Assumption 1, we have the characteristic equation of Eq. (A1) as

$$\det(\lambda) = \prod_{k=1}^{n} q_k(\lambda) = 0, \tag{A2}$$

where

$$q_k(\lambda) = \lambda+\kappa - z\exp(-\lambda\tau) \tag{A3}$$

with $\kappa = \beta(1-\rho_k)$, ρ_k is the kth eigenvalue of W_{res}, and $z = \beta\gamma$ can be considered as a complex variable. If the spectral radius of the reservoir matrix is adjusted to be smaller than 1, then we have $\kappa > 0$.

For a given time delay τ, denote $\upsilon = r(\theta)\exp(i\theta)$, where $0 \leqslant \theta < 2\pi$, and $r(\theta) > 0$ is the radial distance away from the origin $\upsilon = 0$. It is from Ref. [5] that the subsets of parameters space defined by $\lambda = 0$ and $\lambda = \mathrm{i}\omega$ form the boundary of the stable region. Substituting $\lambda = 0$ into Eq. (A3) we get the restriction $|\upsilon| = \kappa$. Substituting $\lambda = \mathrm{i}\omega$ $(\omega > 0)$ into

Eq. (A3) gives that $\kappa + i\omega = r\exp(\theta - \omega\tau)i$, where $i^2 = -1$. Separating its real and imaginary parts yields

$$r\cos(\theta - \omega\tau) = \kappa, r\sin(\theta - \omega\tau) = \omega \tag{A4}$$

and thus

$$r^2 = \omega^2 + \kappa^2, \tag{A5}$$

$$\kappa\tan(\omega\tau - \theta) + \omega = 0. \tag{A6}$$

For given κ, θ, τ, one can get many roots by solving Eq. (A6) and $\rho_m(\theta)$ by Eq. (A5) for $m = 1, 2, \ldots$. We now show how the number of positive real parts for the eigenvalues changes when r crosses each critical value $r_m(\theta)$. From Eq. (A3), the derivative of eigenvalue λ with respect to r at the critical values ω_m and $r_m(\theta)$ is

$$\mathrm{sgn}\Bigg[\mathrm{Re}\frac{d\lambda}{dr}\quad_{\omega_m, r_m} = \mathrm{sgn}\Bigg[\frac{\kappa}{r_m} + r_m\tau \quad > 0. \tag{A7}$$

Equation (A7) indicates that the system added a pair of eigenvalues with a positive real part for each crossing of critical value $r_m(\theta)$ for a given θ. We are interested in the smallest values $\omega_{\min}$ and $r_{\min}$, and thus $v_{\min}(\theta) = r_{\min}(\theta)\exp(i\theta)$. When $0 \leqslant \theta < 2\pi$, $v_{\min}(\theta)$ forms the stability boundary $\chi(\tau)$. Large roots ω_m produced large values of r by Eq. (A5), which lie outside of the stability region shown in Fig. 4. Applying the above discussion to Eq. (A2), we have the results of Theorem 1.

[1] E. Ott, *Chaos in Dynamical Systems* (Cambridge University Press, Cambridge, 2002).

[2] H. Nagashima and Y. Baba, *Introduction to Chaos: Physics and Mathematics of Chaotic Phenomena* (CRC Press, New York, 2018).

[3] S. H. Strogatz, *Nonlinear Dynamics and Chaos with Applications to Physics, Biology, Chemistry, and Engineering* (Taylor & Francis, London, 2018).

[4] X. Xu and J. Liu, Stability, bifurcation and dynamics in a network with delays, Int. J. Bifurcation Chaos **34**, 2450025 (2024).

[5] X. Xu, D. Yu, and Z. Wang, Inter-layer synchronization of periodic solutions in two coupled rings with time delay, Physica D **396**, 1 (2019).

[6] D. Yu, X. Xu, J. Zhou, and E. Li, Stability and instability of a neuron network with excitatory and inhibitory small-world connections, Neural Netw. **89**, 50 (2017).

[7] X. Xu, Z. Li, T. Yan, and E. Li, A novel control approach for the fixed-time synchronization of the complex networks, IEEE Trans. Circuits Syst. II **69**, 2882 (2022).

[8] T. Yan, X. Xu, and E. Li, A subsystem-based analysis approach for fixed-time consensus of multi-agent systems with local pinning strategy, Automatica **142**, 110372 (2022).

[9] S. Wang, A novel hyperchaotic system with fast and slow attractors, AIP Adv. **12**, 105220 (2022).

[10] H. Gu, C. Li, Y. Li, X. Ge, and T. Lei, Various patterns of coexisting attractors in a hyperchaotic map, Nonlinear Dyn. **111**, 7807 (2023).

[11] Y. Tang, J. Kurths, W. Lin, E. Ott, and L. Kocarev, Introduction to focus issue: When machine learning meets complex systems: Networks, chaos, and nonlinear dynamics, Chaos **30**, 063151 (2020).

[12] liu Jianming, X. Xu, and E. Li, A minimum complexity interaction echo state network, Neural Comput. Appl. **36**, 4013 (2024).

[13] D. J. Gauthier, E. Bollt, A. Griffith, and W. A. S. Barbosa, Next generation reservoir computing, Nat. Commun. **12**, 5564 (2021).

[14] V. Gupta, L. K. B. Li, S. Chen, and M. Wan, Model-free forecasting of partially observable spatiotemporally chaotic systems, Neural Netw. **160**, 297 (2023).

[15] L. W. Kong, H. W. Fan, C. Grebogi, and Y. C. Lai, Machine learning prediction of critical transition and system collapse, Phys. Rev. Res. **3**, 013090 (2021).

[16] H. Jaeger, The 'echo state' approach to analysing and training recurrent neural networks—With an erratum note, GMD Technical Report No. 148 (German National Research Center for Information Technology, Bonn, Germany, 2001), p. 13.

[17] W. Maass, T. Natschläger, and H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Comput. **14**, 2531 (2002).

[18] K. Fujii and K. Nakajima, Harnessing disordered-ensemble quantum dynamics for machine learning, Phys. Rev. Appl. **8**, 024030 (2017).

[19] P. Mujal, R. Martínez-Peña, J. Nokkala, J. García-Beni, G. L. Giorgi, M. C. Soriano, and R. Zambrini, Opportunities in quantum reservoir computing and extreme learning machines, Adv. Quantum Technol. **4**, 2100027 (2021).

[20] L. Domingo, G. Carlo, and F. Borondo, Taking advantage of noise in quantum reservoir computing, Sci. Rep. **13**, 8790 (2023).

[21] R. Martínez-Peña, G. L. Giorgi, J. Nokkala, M. C. Soriano, and R. Zambrini, Dynamical phase transitions in quantum reservoir computing, Phys. Rev. Lett. **127**, 100502 (2021).

[22] J. García-Beni, G. L. Giorgi, M. C. Soriano, and R. Zambrini, Scalable photonic platform for real-time quantum reservoir computing, Phys. Rev. Appl. **20**, 014051 (2023).

[23] L. Domingo, M. Grande, F. Borondo, and J. Borondo, Anticipating food price crises by reservoir computing, Chaos Solitons Fractals **174**, 113854 (2023).

[24] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science **304**, 78 (2004).

[25] G. Tanaka, T. Yamane, J. B. Heroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, Recent advances in physical reservoir computing: A review, Neural Netw. **115**, 100 (2019).

[26] R. Soltani, E. Benmohamed, and H. Ltifi, Echo state network optimization: A systematic literature review, Neural Process. Lett. **55**, 10251 (2023).

[27] Z. Gong, H. Chen, B. Yuan, and X. Yao, Multiobjective learning in the model space for time series classification, IEEE Trans. Cybern. **49**, 918 (2019).

[28] F. M. Bianchi, S. Scardapane, S. Lokse, and R. Jenssen, Reservoir computing approaches for representation and classification

of multivariate time series, IEEE Trans. Neural Netw. Learn. Syst. **32**, 2169 (2021).

[29] S. Basterrech and G. Rubino, Evolutionary echo state network: A neuroevolutionary framework for time series prediction, Appl. Soft. Comput. **144**, 110463 (2023).

[30] H. Wang, J. Wu, D. Wang, J. Xin, Y. Yang, and K. Yu, Echo state network with a global reversible autoencoder for time series classification, Inf. Sci. **570**, 744 (2021).

[31] J. Liu, X. Xu, and E. Li, An echo state network with interacting reservoirs for modeling and analysis of nonlinear systems, Nonlinear Dyn. **112**, 8341 (2024).

[32] A. Micheli and D. Tortorella, Addressing heterophily in node classification with graph echo state networks, Neurocomputing **550**, 126506 (2023).

[33] H. Wang, Y. Liu, D. Wang, Y. Luo, C. Tong, and Z. Lv, Discriminative and regularized echo state network for time series classification, Pattern Recognit. **130**, 108811 (2022).

[34] M. Roy, S. Mandal, C. Hens, A. Prasad, N. V. Kuznetsov, and M. Dev Shrimali, Model-free prediction of multistability using echo state network, Chaos **32**, 101104 (2022).

[35] Z. Lu, B. R. Hunt, and E. Ott, Attractor reconstruction by machine learning, Chaos **28**, 061104 (2018).

[36] D. Ibáñez-Soria, J. Garcia-Ojalvo, A. Soria-Frisch, and G. Ruffini, Detection of generalized synchronization using echo state networks, Chaos **28**, 033118 (2018).

[37] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data, Chaos **27**, 121102 (2017).

[38] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach, Phys. Rev. Lett. **120**, 024102 (2018).

[39] H. Zhang, H. Fan, L. Wang, and X. Wang, Learning Hamiltonian dynamics with reservoir computing, Phys. Rev. E **104**, 024205 (2021).

[40] Z. Man, J. Li, X. Di, Y. Sheng, and Z. Liu, Double image encryption algorithm based on neural network and chaos, Chaos Solitons Fractals **152**, 111318 (2021).

[41] L. Jin, Z. Liu, A. Guan, Z. Wang, R. Xue, and L. Li, A high efficient next generation reservoir computing to predict and generate chaos with application for secure communication, IET Commun. **17**, 489 (2023).

[42] M. Goldmann, F. Koester, K. Luedge, and S. Yanchuk, Deep time-delay reservoir computing: Dynamics and memory capacity, Chaos **30**, 093124 (2020).

[43] X. Na, W. Ren, and X. Xu, Hierarchical delay-memory echo state network: A model designed for multi-step chaotic time series prediction, Eng. Appl. Artif. Intell. **102**, 104229 (2021).

[44] D. W. Liedji, J. H. T. Mbe, and G. Kenne, Chaos recognition using a single nonlinear node delay-based reservoir computer, Eur. Phys. J. B **95**, 18 (2022).

[45] S. Lun, X. Yao, and H. Hu, A new echo state network with variable memory length, Inf. Sci. **370–371**, 103 (2016).

[46] H. Jaeger, M. Lukosevicius, D. Popovici, and U. Siewert, Optimization and applications of echo state networks with leaky-integrator neurons, Neural Netw. **20**, 335 (2007).

[47] G. Wainrib and M. N. Galtier, A local echo state property through the largest Lyapunov exponent, Neural Netw. **76**, 39 (2016).

[48] X. Xu, H. Y. Hu, and H. L. Wang, Stability, bifurcation and chaos of a delayed oscillator with negative damping and delayed feedback control, Nonlinear Dyn. **49**, 117 (2007).