Application of next-generation reservoir computing for predicting chaotic systems from partial observations

Irmantas Ratas [*] and Kestutis Pyragas
Center for Physical Sciences and Technology, LT-10257 Vilnius, Lithuania

(Received 27 March 2024; accepted 6 June 2024; published 21 June 2024)

Next-generation reservoir computing is a machine-learning approach that has been recently proposed as an effective method for predicting the dynamics of chaotic systems. So far, this approach has been applied mainly under the assumption that all components of the state vector of dynamical systems are observable. Here we study the effectiveness of this method when only a scalar time series is available for observation. As illustrations, we use the time series of Rössler and Lorenz systems, as well as the chaotic time series generated by an electronic circuit. We found that prediction is only effective if the feature vector of a nonlinear autoregression algorithm contains monomials of a sufficiently high degree. Moreover, the prediction can be improved by replacing monomials with Chebyshev polynomials. Next-generation models, built on the basis of partial observations, are suitable not only for short-term forecasting, but are also capable of reproducing the long-term climate of chaotic systems. We demonstrate the reconstruction of the bifurcation diagram of the Rössler system and the return maps of the Lorenz and electronic circuit systems.

DOI: 10.1103/PhysRevE.109.064215

I. INTRODUCTION

Predicting the behavior of natural and human-built dynamical systems from their output signals is a challenging task with important practical applications. Significant progress has been made in addressing this problem through the development of various machine-learning algorithms based on artificial neural networks [1]. One of the leading candidates is echo state networks or reservoir computers (RCs) [2–7], which process information signals using the nonlinear responses of a high-dimensional dynamical system, known as a reservoir. In RCs, the weights connecting the input layer to the reservoir and the recurrent connection weights are assigned randomly and remain fixed. Only the readout weights are trained using a simple and efficient least squares method. Recently, Gauthier *et al.* [8] proposed next-generation RCs (NG-RCs), which avoids the inherent random nature of traditional RC and requires fewer metaparameters. The effectiveness of this approach has been demonstrated mainly for the case where all dynamic variables are available for observation [8–12]. However, in many experimental situations, especially when dealing with a natural, non-man-made system, only partial information about its dynamic variables can be extracted. The purpose of this paper is to test the effectiveness of NG-RC in such a situation.

The origins of RC research are mainly associated with two seminal papers published by Jaeger [2] and Maass*et al.* [3] in the early 2000s. Elegant in its simplicity, standard RC has proven to be very effective in predicting chaotic time series and reconstructing chaotic attractors [10,13–21]. RC can be used as an observer to infer unmeasured variables of a chaotic system from continuously measured ones, but this requires knowing the full state vector during the training phase [22]. The ability of RCs to reproduce long-term statistical properties of chaotic systems, known as climate replication [14,15], has recently been used to predict changes in system behavior as a function of a control parameter. This is achieved by incorporating an additional parameter-control channel into the standard RC. Then, by training RCs on time series at several parameter values, we can reconstruct the dynamics of the system for parameter values not included in the training set [23,24]. This method made it possible to reconstruct bifurcation diagrams of chaotic systems [24–27] even in the case of noisy [26] and real experimental [27] data. The standard RC has the advantage of being able to implement the reservoir on physical hardware and thus speed up computation [7,28–33]. The disadvantage of conventional RCs is that there is only a limited amount of theory about how they work. Theorists investigate the dependence of RC performance on various factors such as generalized synchronization [15,34,35], nodal dynamics [36,37], network topology [17,38,39] and size [40,41], readout methods [42], and others. Typically, to create a well-performing nonlinear reservoir, a number of metaparameters need to be optimized.

RC theory is greatly simplified under the assumption that the reservoir is a linear dynamical system and nonlinearity is present only in the output layer (feature vector). It was recently shown that a RC based on a linear reservoir and a feature vector constructed as a weighted sum of nonlinear functions of reservoir node values is a universal approximator of dynamical systems [43,44], and such a RC is equivalent to a nonlinear vector autoregression (NVAR) machine [45]. Inspired by these theoretical results, the authors of Ref. [8] developed a modified version of RCs called NG-RC. In NG-RC, there is no reservoir and fewer metaparameters need to be

[*]Contact author: irmantas.ratas@ftmc.lt

tuned. In addition, this approach requires less training data and shorter warmup time than conventional RCs [9]. A NG-RC uses the NVAR algorithm with a feature vector consisting of several time-delayed input signals and their nonlinear functions, which are usually taken in polynomial form.

The ability of NG-RCs to predict and reconstruct dynamical systems has recently been demonstrated using time series of reference low-dimensional chaotic systems [8–12], as well as a system exhibiting spatiotemporal chaos [46]. In the examples [8–12], a NG-RC was applied under the assumption that the full state vector of the system is observable (at least in the training phase). In this paper, we apply a NG-RC to two classical theoretical models of Rössler [47] and Lorenz [48] systems and to experimental time series of an electronic Rössler-like oscillator [49], assuming that only one scalar variable of the dynamical system is observable. This is a more complex task compared to one where all system variables are available. In the latter case, NG-RC works well with a fairly simple feature vector consisting of low-degree monomials of dynamic variables [8]. When a univariate time series is used, the algorithm becomes more complex. We now need to estimate the appropriate delay time and embedding dimension to reconstruct the system dynamics in time-delay coordinates [50]. In addition, good NG-RC performance is only achieved when high degree monomials are used. We show that performance can be improved by replacing the basis of monomial functions with an orthogonal basis of Chebyshev polynomials. We refer to this modification as the NG-RC-Ch algorithm.

The rest of the paper is organized as follows. In Sec. II, we formulate the problem and describe the NG-RC and NG-RC-Ch algorithms as applied to univariate time series. Section III presents the results of applying these algorithms to the Rössler [47] and Lorenz [48] systems, as well as to the chaotic time series generated by the electronic circuit [49]. The paper concludes with a discussion of the results in Sec. IV.

II. PROBLEM STATEMENT AND METHODS

We consider the problem of predicting the dynamics of chaotic systems described by autonomous nonlinear differential equations of the form

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, p), \tag{1}$$

where $\dot{\mathbf{x}}$ is the time derivative of the state vector $\mathbf{x}$, $\mathbf{F}(\mathbf{x}, p)$ defines the vector field of the system, and p is a parameter. We assume that the model Eq. (1) is unknown and the full state vector $\mathbf{x}(t)$ is unobservable, but the scalar variable $S(t)$, which is a function of the state vector $S(t) = h[\mathbf{x}(t)]$, can be measured at the output of the system. We also assume that the observed variable $S(t)$ is capable of accurately reconstructing the underlying attractor in time-delay coordinates.

Suppose we have an equidistantly measured time series $S_i = S(t_i)$ for $t_i = T_0 + (i-1)\Delta t$, where $i \geqslant 1$ is an integer, T_0 is the initial time, and Δt is the sampling time. We set $T_0 = -T_{\text{train}}$, where $T_{\text{train}} \equiv (n-1)\Delta t$ is the training time. Our goal is to learn the dynamics of S_i during the training phase $i = 1, \ldots, n$ $(-T_{\text{train}} \leqslant t \leqslant 0)$ and make a prediction of S_i for $i > n$. We denote the forecast time series as $\hat{S}_i$, $i > n$. Next, it is convenient to analyze the rescaled time series R_i with values from -1 to 1:

$$R_i = 2\frac{S_i - S_{\text{mn}}}{S_{\text{mx}} - S_{\text{mn}}} - 1. \tag{2}$$

Here, S_{mx} and S_{mn} denote the maximum and minimum values of the time series S_i determined before the prediction phase. The scaled predicted time series $\hat{R}_i$ is related to $\hat{S}_i$ in the same way as R_i and S_i.

We test our algorithms for short-term prediction and long-term reconstruction of attractor dynamics (climate replication). The accuracy of short-term prediction is evaluated by two measures: (i) the root mean square error

$$\text{RMSE} = \frac{\sqrt{1/L\sum_i(\hat{R}_i - R_i)^2}}{\sigma_R}, \tag{3}$$

where L is the number of terms in the prediction interval and σ_R is the standard deviation estimated before prediction; and (ii) the valid prediction time T_{vp} (see Ref. [21]). To determine T_{vp}, we first introduce the absolute error:

$$\varepsilon_i = |\hat{R}_i - R_i|/2. \tag{4}$$

The valid prediction time is defined as the time interval from the beginning of the prediction to the first point in time when the absolute value of the error ε_i exceeds some arbitrary threshold value ε_{th}. We choose $\varepsilon_{\text{th}} = 0.01$. To evaluate the ability of our algorithms to reproduce the long-term statistical properties (climate) of the original system, we use the largest Lyapunov exponent, return maps, and bifurcation diagrams as indicators.

A. NG-RC and NG-RC-Ch algorithms

Here we describe the NG-RC algorithm and its modified version NG-RC-Ch for a fixed value of the parameter p. When reconstructing a bifurcation diagram, we need to include in the algorithm a dependence on the parameter p. Such a modification is presented in the Appendix.

Given a scalar time series R_i, we reconstruct the dynamics of the system in a multidimensional state space using the time-delay coordinates. We define the delay time in these coordinates as τ and introduce the k-dimensional state vector as

$$\mathbf{R}_i = [R_i, R_{i-n_\tau}, \ldots, R_{i-(k-1)n_\tau}]^T, \tag{5}$$

where $n_\tau = \tau/\Delta t$ is assumed to be an integer. We estimate the embedding dimension k using the false nearest-neighbors algorithm [51] and evaluate the embedding delay time τ from the first minimum of average mutual information [52].

Following Refs. [8,45], we use the NVAR algorithm to approximate the system dynamics with a map

$$R_{i+1} = R_i + \Delta t \mathbf{W}_{\text{out}}\mathbf{O}(\mathbf{R}_i), \tag{6}$$

where $\mathbf{W}_{\text{out}}$ is the row vector of NG-RC output weights. It is determined at the training phase. The feature vector $\mathbf{O}(\mathbf{R}_i) \equiv \mathbf{O}_i$ is a function of the current state $\mathbf{R}_i$. The feature vector $\mathbf{O}_i$ is constructed from three parts: the bias c, the linear vector $\mathbf{O}_i^{(1)}$, and a set of nonlinear vectors $\mathbf{O}_i^{(l)}$, $l = 2, 3, \ldots$. Without loss of generality, we take $c = 1$. The linear part is simply the state vector:

$$\mathbf{O}_i^{(1)} = \mathbf{R}_i = [R_i, R_{i-n_\tau}, \dots, R_{i-(k-1)n_\tau}]^T. \tag{7}$$

The nonlinear vector $\mathbf{O}_i^{(l)}$ is the set of all unique monomials of degree l composed of all components of the state vector $\mathbf{R}_i$. For example, if we want to build a model in three-dimensional embedding space, $k = 3$, the second-order nonlinear vector $l = 2$ is as follows:

$$\mathbf{O}_i^{(2)} = \left[R_i^2, R_iR_{i-n_\tau}, R_iR_{i-2n_\tau}, R_{i-n_\tau}^2, R_{i-n_\tau}R_{i-2n_\tau}, R_{i-2n_\tau}^2\right]^T. \tag{8}$$

The total feature vector containing nonlinearities up to second order is of the form

$$\mathbf{O}_i = 1 \oplus \mathbf{O}_i^{(1)} \oplus \mathbf{O}_i^{(2)}. \tag{9}$$

Here $\oplus$ represents the vector concatenation operation. In this example, $k = 3$ and the length of the total vector $\mathbf{O}_i$ is 10. In general, the total vector containing all vectors $\mathbf{O}_i^{(l)}$ with degrees up to order m can be written as

$$\mathbf{O}_i = 1 \oplus \mathbf{O}_i^{(1)} \oplus \mathbf{O}_i^{(2)} \dots \oplus \mathbf{O}_i^{(m)}. \tag{10}$$

The length of the vector $\mathbf{O}_i^{(l)}$ is $d_l = \prod_{i=0}^{l-1}(k+i)/l!$, and the length of the total vector $\mathbf{O}_i$ is

$$d = 1 + \sum_{l=1}^{m} d_l. \tag{11}$$

For time series R_i of length n, there are $n - (k-1)n_\tau$ possible feature vectors $\mathbf{O}_i$.

The row vector $\mathbf{W}_{\text{out}}$ is determined in the training phase using the requirement that the map (6) provides the best approximation of the time series. The length of the vector $\mathbf{W}_{\text{out}}$ is d. Let us introduce a notation for the difference between two consecutive measurements:

$$Y_i = (R_{i+1} - R_i)/\Delta t = \mathbf{W}_{\text{out}}\mathbf{O}_i. \tag{12}$$

The values Y_i obtained at the training stage are collected into a row vector

$$\tilde{\mathbf{Y}} = [Y_{n-1}, Y_{n-2}, \dots, Y_{n-(k-1)n_\tau+1}] \tag{13}$$

of size $n - (k-1)n_\tau - 1$. The feature vectors are collected in a similar manner, resulting in a matrix

$$\tilde{\mathbf{O}} = [\mathbf{O}_{n-1}, \mathbf{O}_{n-2}, \dots, \mathbf{O}_{(k-1)n_\tau+1}] \tag{14}$$

of dimension $d \times [n - (k-1)n_\tau - 1]$. Now the regression problem reads

$$\tilde{\mathbf{Y}} = \mathbf{W}_{\text{out}}\tilde{\mathbf{O}}. \tag{15}$$

Using the least-squares method with Tikhonov regularization, we obtain

$$\mathbf{W}_{\text{out}} = \tilde{\mathbf{Y}}\tilde{\mathbf{O}}^T(\tilde{\mathbf{O}}\tilde{\mathbf{O}}^T + \beta\mathbf{I})^{-1}. \tag{16}$$

Here β is the regularization parameter, also known as ridge parameter, and $\mathbf{I}$ is the identity matrix.

Having determined the output weights $\mathbf{W}_{\text{out}}$ during the training phase $i \leqslant n$ ($t \leqslant 0$), we can now use it to predict the dynamics of the system for $t > 0$, i.e., generate forecast time series $\hat{R}_i$:

$$\hat{R}_{i+1} = \hat{R}_i + \Delta t\mathbf{W}_{\text{out}}\mathbf{O}(\hat{\mathbf{R}}_i), \ i \geqslant n, \tag{17}$$

with the initial conditions $\hat{R}_i = R_i$ for $i = n - n_{\text{w}}, \dots, n$ where $n_{\text{w}} = (k-1)n_\tau$. The initial conditions are taken from the interval $t \in [-t_{\text{w}}, 0]$ of the length $t_{\text{w}} = (k-1)\tau$ known as the warm-up time.

It is worth noting that in the time-continuous limit $\Delta t \to 0$, Eq. (17) is transformed into a scalar nonlinear delay differential equation (DDE),

$$d\hat{R}/dt = \mathbf{W}_{\text{out}}\mathbf{O}[\hat{R}(t), \hat{R}(t-\tau), \dots, \hat{R}(t-(k-1)\tau)], \quad t > 0, \tag{18}$$

with initial conditions $\hat{R}(t) = R(t)$ given on the time interval $t \in [-t_{\text{w}}, 0]$. Thus, in the time-continuous limit, the NG-RC algorithm produces a surrogate prediction model in the form of a nonlinear DDE that is trained by fitting its solution to a given scalar signal.

The above algorithm is a direct replication of NG-RC to the case of scalar time series. In addition, we present a modified version of this algorithm, which we called NG-RC-Ch. The main idea of the modification is to replace the basis of monomial functions with an orthogonal basis of Chebyshev polynomials of the first kind. More precisely, monomials of type $x^{j_1}y^{j_2}z^{j_3}$ in nonlinear feature vectors are replaced by expressions $T_{j_1}(x)T_{j_2}(y)T_{j_3}(z)$, where T_{j_k} is a Chebyshev polynomial of degree j_k. For example, a modification of the nonlinear feature vector Eq. (8) has the following form:

$$\mathbf{O}_i^{(2)} = [T_2(R_i),\ T_1(R_i)T_1(R_{i-n_\tau}),\ T_1(R_i)T_1(R_{i-2n_\tau}),\ T_2(R_{i-n_\tau}),\ T_1(R_{i-n_\tau})T_1(R_{i-2n_\tau}),\ T_2(R_{i-2n_\tau})]^T. \tag{19}$$

In the next section, we will present a comparison of the performance of both algorithms using specific examples.

III. RESULTS

We apply the above algorithms to scalar time series generated by two classical chaotic systems Rössler [47] and Lorenz [48], as well as to experimental data generated by an electronic Rössler-like oscillator [49]. Using these systems as examples, we compare the performance of the NG-RC and NG-RC-Ch algorithms depending on two parameters: sampling time Δt and the maximum degree m of monomials or Chebyshev polynomials utilized in the feature vectors. We call the parameter m the degree of nonlinearity of the algorithm or trained model. In the figures below, we use the labels Mn (monomials) and Ch (Chebyshev polynomials) followed by the number m to denote the results of the NG-RC and NG-RC-Ch algorithms with a given degree of nonlinearity, respectively. The values of characteristic parameters of time series for different systems are summarized in Table I.

A. Rössler system

We'll start our analysis with the Rössler system [47],

$$\dot{x} = -(y+z), \tag{20a}$$

$$\dot{y} = x + ay, \tag{20b}$$

$$\dot{z} = b + z(x-c), \tag{20c}$$

TABLE I. Values of characteristic parameters of time series for different systems. Here $\Delta t = 33.3\,\mu s$ is the sampling time of the chaotic time series of the electronic oscillator [49].

Parameter	Rössler	Lorenz	Electronic oscillator
Observed variable	y	x	v_2
Embedding dimension	3	3	4
Embedding delay time	1.5	0.15	$7\Delta t$
Lyapunov time	14.08	1.10	$90\Delta t$

assuming standard parameter values $a = b = 0.2$, $c = 5.7$. These parameters provide a chaotic regime with the largest Lyapunov exponent $\Lambda = 0.071$ or Lyapunov time $1/\Lambda = 14.08$. Let us imagine that the full state vector $\mathbf{x}(t) = [x(t), y(t), z(t)]^T$ is unobservable and only the variable $y(t)$ is available for observation, that is, $S(t) = h[\mathbf{x}(t)] = y(t)$. We reconstruct the dynamics of the system in time delay coordinates using the scalar variable $y(t)$. A false nearest-neighbor analysis [51] of the variable $y(t)$ shows that the appropriate embedding dimension is $k = 3$ and the delay time estimated from the average mutual information function [52] is $\tau = 1.5$. We use these parameter values in both the NG-RC and NG-RC-Ch algorithms.

Figure 1 shows an example of a prediction made by the NG-RC-Ch [Fig. 1(a)] and NG-RC [Fig. 1(b)] algorithms with degree of nonlinearity $m = 8$. The time is normalized by the largest Lyapunov exponent. The predicted signals (red dashed curves) are compared to the original signals (blue solid curves). The blue and red curves in Fig. 1(c) show the absolute errors of the NG-RC-Ch and NG-RC algorithms, respectively. The sampling time $\Delta t = 0.1$ and the training time $T_{\text{train}} = 1000$ are taken to be the same for both algorithms. The order of the regularization parameter β for each algorithm was chosen by trial and error to obtain the best result. This example shows that Chebyshev polynomials provide better prediction than monomials. The valid prediction time of NG-RC-Ch is approximately two Lyapunov time units longer than that of NG-RC.

The characteristic prediction time depends weakly on the length of the training interval T_{train} if it is chosen large enough. However, the prediction time is inhomogeneous on the attractor; the prediction result depends on the choice of the starting point on the strange attractor. Therefore, for a more accurate comparison of the NG-RC-Ch and NG-RC algorithms, the statistical measures of performance over the initial conditions should be used. In Fig. 2, the colored bars show the median of valid prediction time med(T_{vp}) and the error bars indicate the first and third quartiles. We generated $N = 1000$ trajectories with different initial conditions on the Rössler attractor and used them to train the output weights $\mathbf{W}_{\text{out}}$ and obtain valid prediction times T_{vp} for both algorithms. Then, we found quartiles of T_{vp} for each of the algorithms. The vertical axis of Fig. 2 is normalized by the largest Lyapunov exponent. The results are presented for four different values of the sampling time Δt, marked on the horizontal axis. The bars of different colors, labeled Mn (monomials) and Ch (Chebyshev polynomials), correspond to the NG-RC and NG-RC-Ch algorithms, respectively. The numbers following these labels indicate the

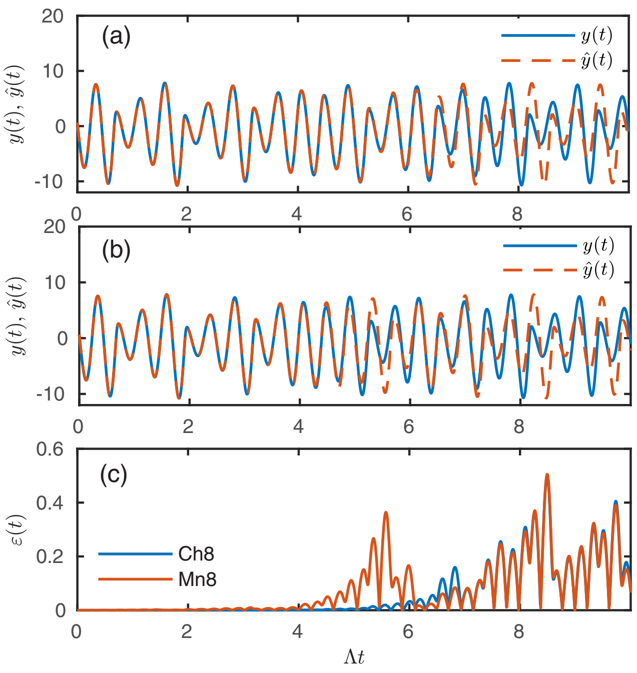

FIG. 1. An example of predicting the dynamics of the Rössler system by the NG-RC-Ch and NG-RC algorithms. The blue continuous curves in (a) and (b) show the original time series $y(t)$. The time series $\hat{y}(t)$ predicted by the NG-RC-Ch (a) and NG-RC (b) algorithms are shown by the red dashed curves. The blue and red curves in (c) show the absolute errors Eq. (4) of the scaled time series for the NG-RC-Ch and NG-RC algorithms, respectively. The parameters $\Delta t = 0.1$, $T_{\text{train}} = 1000$, $k = 3$, $\tau = 1.5$, and $m = 8$ are the same for both algorithms. The regularization parameter β is 10^{-7} for NG-RC-Ch and 10^{-3} for NG-RC. The time axis is normalized by the largest Lyapunov exponent.

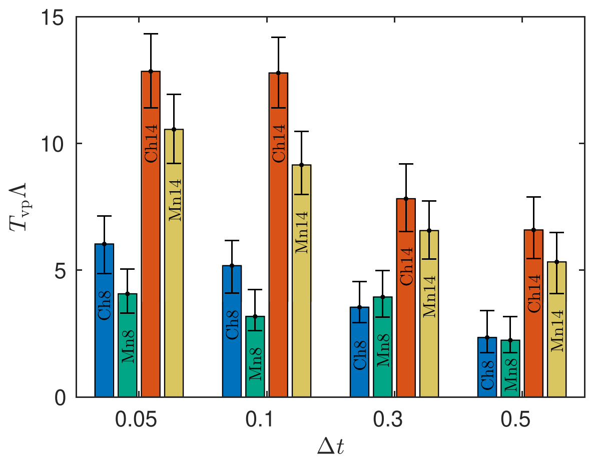

FIG. 2. The medians of valid prediction time of the Rössler system as a function of sampling time Δt and degree of nonlinearity m. The bars labeled Mn and ChM followed by the number m correspond to the NG-RC and NG-RC-Ch algorithms, respectively. Error bars show the first and third quartiles. The values of the parameters $T_{\text{train}} = 2000$, $k = 3$, and $\tau = 1.5$ are the same for both algorithms. The value of the ridge parameter β was chosen from a set $(10^{-3}, 10^{-4}, \ldots, 10^{-9})$ for every bar individually to maximize med(T_{vp}).

degree of nonlinearity m used in the algorithm. The results are presented for $m = 8$ and $m = 14$. The value of the ridge parameter β was chosen for every bar individually to maximize med(T_{vp}). This figure shows that for a high degree of nonlinearity, $m = 14$, NG-RC-Ch outperforms NG-RC for all Δt. For a lower degree of nonlinearity, $m = 8$, NG-RC-Ch also provides a better result for almost all Δt, except for $\Delta t = 0.3$, where med(T_{vp}) of NG-RC is slightly larger.

In many practical applications, continuous prediction of system behavior is required. That is, knowing the history of the output signal $S(t)$ up to time t, we need to predict its values $\hat{S}(t + T_{\text{pred}})$ for varying t and fixed prediction time T_{pred}. We implement such an algorithm using the following steps:

(1) Choose an initial point on the strange attractor and generate the output time series $R(t_i)$.

(2) In the training time interval $[-T_{\text{train}}, 0]$, compute the output weights $\mathbf{W}_{\text{out}}$ and keep it fixed for the whole subsequent prediction procedure. The initial time at which the prediction starts is $t = 0$.

(3) Iterating Eq. (17) with initial conditions $\hat{R}(t_i) = R(t_i)$ on the time interval $[-t_w, 0]$, find the predicted value of $\hat{R}(T_{\text{pred}})$ at time T_{pred}.

(4) Shift the initial moment of prediction in Eq. (17) by one sampling period ahead $t \to t + \Delta t$ and, using initial conditions $\hat{R}(t_i) = R(t_i)$ on the time interval $[t - t_w, t]$, find the prediction $\hat{R}(t + T_{\text{pred}})$.

(5) Repeat step 4 until the time t reaches the end T_{end} of the prediction interval $[0, T_{\text{end}}]$ and compute the RMSE of the prediction at that interval.

Note that in some segments of the prediction interval, the predicted values of $\hat{R}(t_i)$ may go beyond the theoretically permissible values of $|\hat{R}| \leqslant 1$. Such events occur more often at larger T_{pred}. When estimating RMSE, we take $\hat{R}(t_i) = \text{sgn}(\hat{R}(t_i))$ for $|\hat{R}(t_i)| > 1$.

The results of continuous prediction of the Rössler system by the NG-RC and NG-RC-Ch algorithms are presented in Fig. 3. Calculations were carried out using 24 trajectories with different initial conditions on a strange attractor. For each trajectory, RMSE was estimated over a prediction time interval of length $T_{\text{end}} = 560$. The horizontal axis represents the prediction time T_{pred} normalized by the largest Lyapunov exponent. The vertical axis shows the parameters of statistical averaging of RMSE over the trajectories. The quartiles of the RMSE distribution are indicated by the three horizontal bars, with the middle bar showing the median. The red and blue colors correspond to the NG-RC and NG-RC-Ch algorithms, respectively. Both algorithms have the same degree of nonlinetity $m = 14$. This figure shows a clear advantage of NG-RC-Ch over NG-RC. NG-RC-Ch predicts the behavior of a Rössler system up to eight Lyapunov times with RMSE less than 1%, while NG-RC gives more or less adequate forecast for only six Lyapunov times.

Finally, we turn to the problem of long-term climate prediction. In long-term forecasting, the signal $\hat{R}(t_i)$ generated by the trained model (17) need not necessarily coincide with the original signal $R(t_i)$. This model should reproduce only the statistical properties of the original system so typical characteristics of a strange attractor, such as Lyapunov exponents, return maps, bifurcation diagrams, etc., can be recovered.

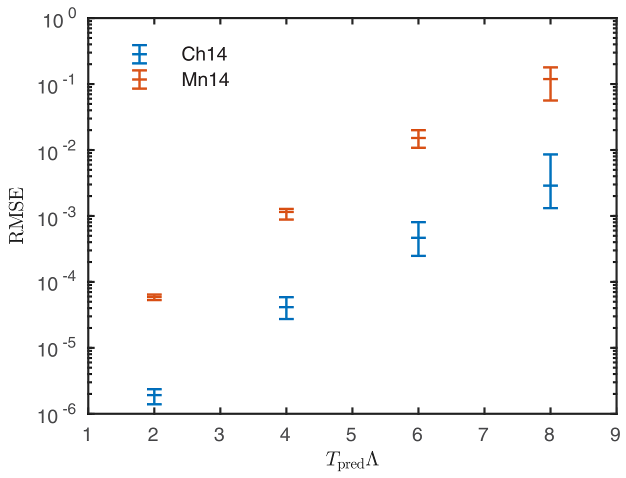

FIG. 3. The results of continuous prediction of the Rössler system. The root mean square prediction errors for the NG-RC (red) and NG-RC-Ch (blue) algorithms are shown as a function of the prediction time T_{pred} normalized to the largest Lyapunov exponent. The values of the parameters $T_{\text{train}} = 2000$, $T_{\text{end}} = 560$, $\Delta t = 0.1$, $k = 3$, $\tau = 1.5$, and $m = 14$ are the same for both algorithms. The value of the ridge parameter β was chosen from the set $(10^{-3}, 10^{-4}, \ldots, 10^{-9})$ for different T_{pred} and different algorithms separately to minimize the RMSE.

Typically, Eq. (17) generates a bounded signal over a large time interval. However, this equation does not guarantee the asymptotic stability of the solution, and on fairly large timesscales the trajectory may escape from the strange attractor and even diverge to infinity. Instability of trained models is observed for both RC and NG-RC algorithms and is generally a major open problem [5,15,53]. In the examples below, our trained models generate stable time series over sufficiently large time intervals so we can recover the statistical characteristics of the strange attractor.

As a first test for long-term prediction, we estimated the largest Lyapunov exponent of the Rössler system using a trained model based on the NG-RC-Ch algorithm with degree of nonlinearity $m = 8$. Training was performed on a time series with sampling time $\Delta t = 0.1$ over an interval of length $T_{\text{train}} = 1000$. We then generated a time series of length 2800 using the trained model. Applying the Rosenstein [54] algorithm to this time series, we estimated the largest Lyapunov exponent. We repeated this procedure for 700 different initial conditions on the strange attractor and obtained an average value of the largest Lyapunov exponent equal to $\Lambda = 0.060$. The same algorithm for the original time series gave the result $\Lambda = 0.062$. Thus, the difference between the Lyapunov exponents estimated from the original time series and the time series generated by the trained model is about 3%.

As the next example of climate prediction, we consider the problem of recovering the bifurcation diagram of the Rössler system from the scalar observable $y(t)$. We fixed the parameter values $a = 0.2$ and $c = 5.7$ as before and chose b as the bifurcation parameter. This parameter becomes an additional component in the state vector $\mathbf{R}_i$ (see Appendix for details). Figure 4(a) shows the original bifurcation diagram obtained from the Rössler Eqs. (20). The dots show the local maxima

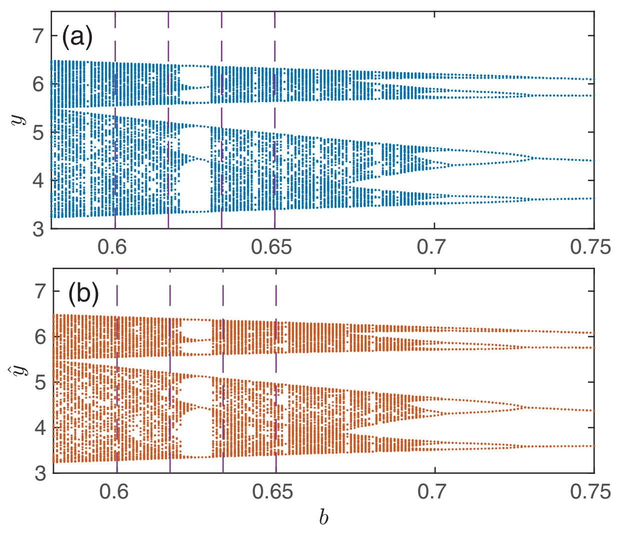

FIG. 4. Reconstructing the bifurcation diagram of the Rössler system from a scalar observable. The dots show the local maxima of the variable $y(t)$ obtained (a) from the original system Eqs. (20) and (b) from the trained prediction model. The parameter values $a = 0.2$ and $c = 5.7$ of the Rössler system are fixed as before and b is the bifurcation parameter. The four vertical dashed lines indicate the values of the bifurcation parameter at which time series of the variable $y(t)$ were recorded for training the prediction model based on the NG-RC-Ch algorithm. The parameters are $T_{\text{train}} = 1000$, $\Delta t = 0.1$, $k = 4$, $\tau = 1.5$, $m = 7$, and $\beta = 10^{-3}$.

of the variable $y(t)$. The four vertical dashed lines indicate the values of the bifurcation parameter from which the bifurcation diagram shown in Fig. 4(b) was recovered. At these parameter values, four time series of the variable $y(t)$ with length $T_{\text{train}} = 1000$ were generated to train a prediction model based on the NG-RC-Ch algorithm with a degree of nonlinearity $m = 7$. The trained model made it possible to generate time series for values of the bifurcation parameter that were not used in the training process. We see that the bifurcation diagram reconstructed from several time series of the scalar observable generally replicates well the original bifurcation diagram. Although some details, such as narrow periodic windows, are not always captured in the reconstructed diagram. The results can be improved by increasing the number of reference parameter values. However, there is no guarantee to obtain a good extrapolation of the bifurcation diagram far beyond the range of reference points [12].

B. Lorenz system

Next, we compare the performance of the NG-RC and NG-RC-Ch algorithms using the time series of the Lorenz system [48]:

$$\dot{x} = \sigma(y - x), \tag{21a}$$

$$\dot{y} = x(\rho - z) - y, \tag{21b}$$

$$\dot{z} = xy - \beta z. \tag{21c}$$

We use the standard parameter values $\sigma = 10$, $r = 28$, $b = 8/3$, and assume that $x(t)$ is a scalar observable. The largest Lyapunov exponents at these parameters is $\Lambda = 0.91$. When

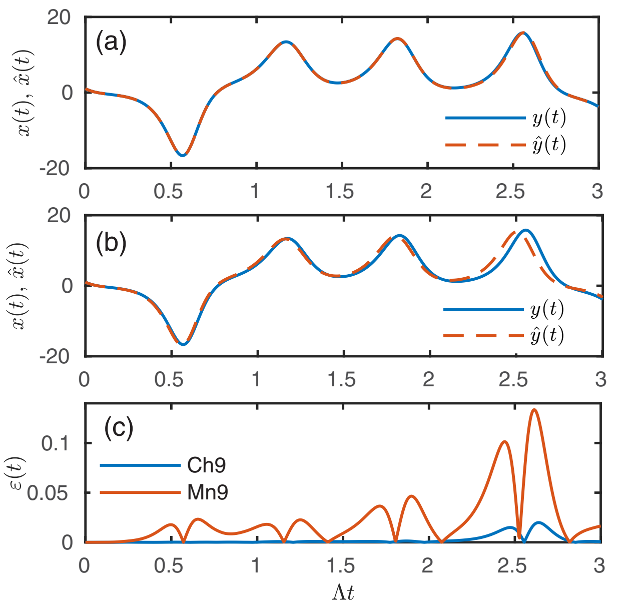

FIG. 5. An example of predicting the dynamics of the Lorenz system by the NG-RC-Ch and NG-RC algorithms. The blue continuous curves in (a) and (b) show the original time series $x(t)$. The time series $\hat{x}(t)$ predicted by the NG-RC-Ch (a) and NG-RC (b) algorithms are shown by the red dashed curves. The blue and red curves in (c) show the absolute errors Eq. (4) of the scaled time series for the NG-RC-Ch and NG-RC algorithms, respectively. The parameters $\Delta t = 0.005$, $T_{\text{train}} = 100$, $k = 3$, $\tau = 1.5$, $m = 9$, and $\beta = 10^{-9}$ are the same for both algorithms.

constructing the feature vector, we suppose that information about the symmetry of the system is known in advance. Specifically, knowing that the Lorenz equations are symmetric under the transformation $(x, y, z) = (-x, -y, z)$, we compose the feature vector for the observable x from only odd degree monomials or only odd degree Chebyshev polynomials. Also, to preserve this symmetry in scaled time series, we set $S_{\text{mx}} = -S_{\text{mn}} = \max(|S_i|)$ in Eq. (2).

Figure 5 shows an example of prediction performed by the NG-RC-Ch and NG-RC algorithms with a degree of nonlinearity $m = 9$. The sampling time $\Delta t = 0.005$ and the training time $T_{\text{train}} = 100$ is the same for both algorithms. The optimal value of the regularization parameter for both algorithms is $\beta = 10^{-9}$. The advantage of NG-RC-Ch over NG-RC is clearly seen from the error dynamics shown in Fig. 5(c). NG-RC-Ch provides good predictions up to two Lyapunov times, while the signal predicted by NG-RC deviates significantly from the original signal already at half the Lyapunov time.

A more detailed comparison of the NG-RC-Ch and NG-RC algorithms for the Lorenz system is presented in Fig. 6. As for the Rössler system, the advantage of NG-RC-Ch over NG-RC is more pronounced when the degree of nonlinearity m is larger. When $m = 21$, the median of valid prediction time of NG-RC-Ch, depending on the sampling time Δt, ranges between 2.7 and 3.3 Lyapunov times. Whereas for NG-RC, the valid prediction is only between 1.5 and 1.9 Lyapunov times.

Figure 7 shows the performance of the NG-RC-Ch and NG-RC algorithms for the Lorenz system in the case of

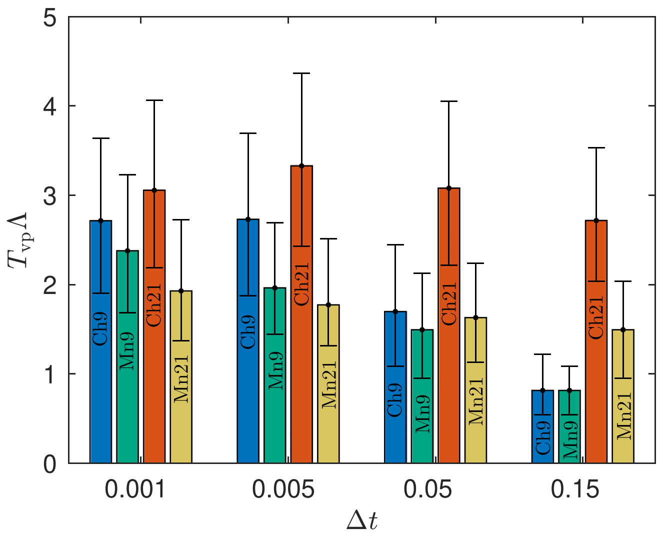

FIG. 6. The medians of valid prediction time of the Lorenz system as a function of sampling time Δt and degree of nonlinearity m. The designations are the same as in Fig. 2. The parameters are $T_{\rm train} = 200$, $k = 3$, and $\tau = 0.15$. The value of the ridge parameter β was chosen as described in the caption of Fig. 2. The valid prediction time is averaged over $N = 1000$ trajectories with different initial conditions on the Lorenz attractor.

continuous prediction. Results are presented for degree of nonlinearity $m = 21$ and sampling time $\Delta t = 0.05$. Again, we see that NG-RC-Ch provides better prediction accuracy than NG-RC for any chosen value of prediction time $T_{\rm pred}$.

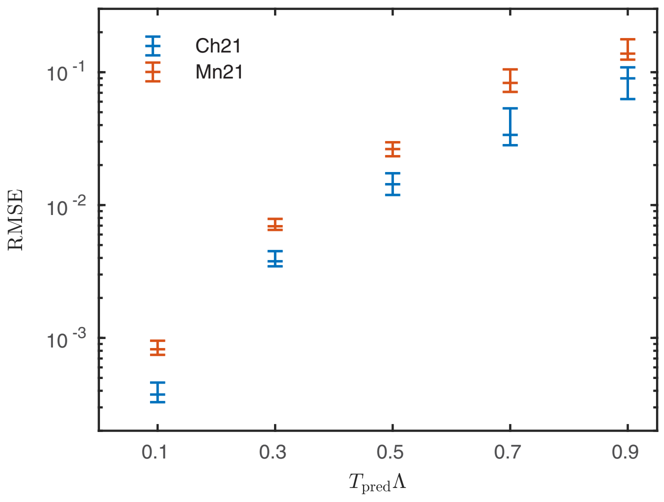

FIG. 7. The results of continuous prediction of the Lorenz system. The root mean square prediction errors for the NG-RC (red) and NG-RC-Ch (blue) algorithms are shown as a function of the prediction time $T_{\rm pred}$ normalized to the largest Lyapunov exponent. The designations are the same as in Fig. 3. The parameters are $T_{\rm train} = 1000$, $\Delta t = 0.05$, $k = 3$, $\tau = 0.15$, and $m = 21$. The value of the ridge parameter β was chosen as described in the caption of Fig. 3. The RMSE was evaluated on a prediction time interval of length $T_{\rm end} = 400$, using 24 trajectories with different initial conditions on a strange attractor.

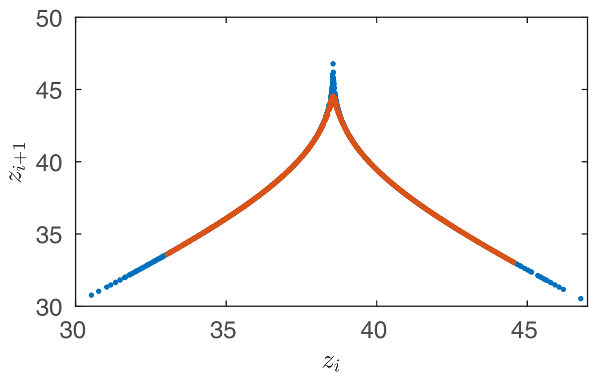

FIG. 8. Reconstruction of the Lorenz map from a scalar observable using the NG-RC-Ch algorithm. Return map of the actual and predicted variable z of the Lorenz system. This plot is made with a time series of length 1100, where the blue dots refer to the actual Lorenz system and the red dots overlaying the blue dots refer to the prediction. The parameters are $T_{\rm train} = 100$, $\Delta t = 0.001$, $k = 3$, $\tau = 0.1$, $m = 9$, and $\beta = 10^{-5}$.

We now turn to the long-term prediction of the Lorenz system. First, we evaluate the largest Lyapunov exponent using the NG-RC-Ch model with degree of nonlinearity $m = 9$. We train this model on a time series of variable $x(t)$ of length $T_{\rm train} = 100$ with sampling time $\Delta t = 0.005$. Then, using the trained model, we generate a time series of length 110 and estimate the largest Lyapunov exponent. Repeating this procedure for 1000 different initial conditions on the strange attractor, we obtain an average value of the largest Lyapunov exponent equal to $\Lambda = 0.94$. The same algorithm for the original time series results in $\Lambda = 0.87$. As a next test of long-term forecasting, we reconstruct the Lorenz map. The classical Lorenz map is constructed based on the variable $z(t)$ of the Lorenz system, so we now assume that this variable is a scalar observable. The NG-RC-Ch model with degree of nonlinearity $m = 9$ was trained on a time series of length $T_{\rm train} = 100$ with a sampling step $\Delta t = 0.001$. Since there is no symmetry in the dynamics of the variable $z(t)$ of the Lorenz system, the feature vector was composed using Chebyshev polynomials of both even and odd degrees. The results are presented in Fig. 8. We followed Lorenz's procedure to plot the return map of successive maxima of $z(t)$. We first obtain $z(t)$ for a long period of time, $0 < t < 1100$, for both the actual and the predicted time series. We then arrange all local maxima of the actual and predicted $z(t)$ in time order and label them $[z_1, z_2, \ldots, z_M]$. Then, we plot successive pairs of these maxima $[z_i, z_{i+1}]$ as dots in Fig. 8. The blue dots are from the actual Lorenz system, while the red dots printed over the blue dots are from the trained model. The red dots generated by the trained model fall on top of the blue dots throughout the entire running time ($0 < t < 1100$). Note, however, that the blue dots corresponding to the large and small values of the local maxima of the actual time series $z(t)$ remain uncovered. This is due to the fact that such maxima are rare events in the time series $z(t)$ of the actual Lorenz system. They are not picked up by the reconstructed model during the training phase.

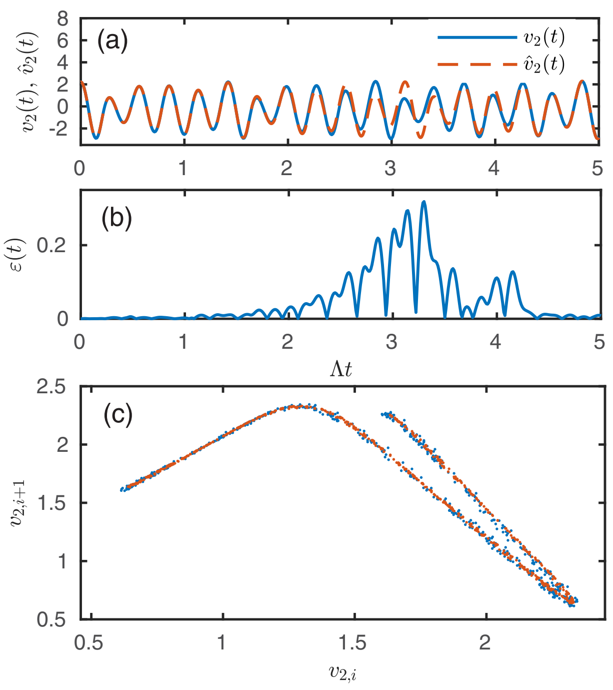

FIG. 9. Prediction of experimental time series of the electronic Rössler-like oscillator. The prediction is performed by the NG-RC-Ch algorithm with degree of nonlinearity $m = 9$. (a) Dynamics of actual (blue curve) and predicted (red dashed curve) time series. (b) Dynamics of the prediction error Eq. (4) obtained from a scaled time series. (c) Return maps constructed from the original (blue dots) and predicted (red dots) time series. The parameters in all panels are $T_{\text{train}} = 15\,000\Delta t$, $k = 4$, $\tau = 7\Delta t$, and $\beta = 10^{-9}$.

C. Experimental time series of the electronic Rössler-like oscillator

Finally, we applied our algorithm to realistic time series in which noise perturbations, measurement errors, and other imperfections are inevitable. Specifically, we used the experimental data sets of the electronic Rössler-like oscillator provided by Vera-Ávila *et al.* [49]. In this work, the authors recorded signals from a network of 28 identical electronic oscillators. The recordings were carried out at different values of the coupling strength between the oscillators, including the case of uncoupled (free) oscillators. Just such a case is analyzed in our paper. We use the time series of one free oscillator, designated in Ref. [49] as the first oscillator. The state of the oscillator is determined by three dynamic variables: voltages v_1, v_2, and v_3. In our analysis, we use the time series of the variable $v_2(t)$ that was recorded with a sampling rate of 30 KS/s or sampling time $\Delta t = 33.3$ µs. The waveform of the electronic oscillator resembles that of the standard Rössler oscillator, although the dynamic equations governing these systems are significantly different. The nonlinearity in the standard Rössler equations is determined by the product of the dynamic variables, whereas in the dynamic equations of the electronic circuit given in [49], the nonlinearity is determined by a piecewise linear function.

The results of our analysis are presented in Fig. 9. A false nearest-neighbor analysis [51] of the time series $v_2(t)$ shows that the appropriate embedding dimension is $k = 4$, and the delay time estimated from the average mutual information function [52] is $\tau = 7\Delta t$. Rosenstein's algorithm [54] gives the value of the largest Lyapunov exponent $\Lambda = 1/(90\Delta t)$. We demonstrate the prediction of the experimental time series using NG-RC-Ch algorithm with degree of nonlinearity $m = 9$. The NG-RC-Ch model was trained on a time interval of length $T_{\text{train}} = 15\,000\Delta t$. The time series predicted by the trained model (red dashed curve) is compared with the actual time series (blue curve) in Fig. 9(a). Figure 9(b) shows the dynamics of the prediction error. A fairly good prediction is seen up to two Lyapunov times.

To demonstrate long-term prediction, we reconstructed the return map. As in the case of the Lorenz map, we found local maxima $[v_{2,1}, v_{2,2}, \dots, v_{2,M}]$ of the original and predicted time series $v_2(t)$. We then plotted successive pairs of these maxima $[v_{2,i}, v_{2,i+1}]$ as dots in Fig. 9(c). The blue dots correspond to the actual time series, while the red dots printed over the blue dots correspond to the trained model. The red dots are consistently near the blue dots, indicating that the trained model reproduces the climate in the long term. Note that the red dots fall on an imaginary thin continuous curve, while the blue dots are scattered (presumably due to noise present in the experimental time series) in the neighborhood of this curve. The effect of less scattering of red dots corresponding to the reconstructed map can be explained by the ability of the next-generation RC algorithm to filter out noise. This explanation is consistent with the claim of a recent publication [27] that a conventional reservoir computer can be used as a noise filter to recover system dynamics from noisy signals.

IV. DISCUSSION

The aim of this paper was to investigate the effectiveness of the NG-RC approach for predicting the dynamics of chaotic systems when only partial observations are available. We focused on the most difficult case, when a prediction needs to be made based on the dynamics of a scalar observable. In our paper, we used time series of chaotic Rössler [47] and Lorenz [48] systems as well as chaotic time series generated by an electronic circuit [49].

Previous works [8–12] have shown that NG-RC is very effective when all components of the state vector of a chaotic system are used for prediction. Good results were achieved using a fairly simple feature vector consisting of only low-degree monomials. In the case of univariate time series, we used time-delay coordinates to reconstruct the dynamics of the system in a high-dimensional state space. It turns out that NG-RC in these coordinates only works when high-degree monomials are taken into account. To improve the performance of the NG-RC algorithm, we replaced the basis of monomial functions with an orthogonal basis of Chebyshev polynomials of the first kind. We called this modification the NG-RC-Ch algorithm. The advantage of NG-RC-Ch over NG-RC is confirmed by a detailed comparison of their characteristics on time series of the Rössler and Lorenz systems. For the same degree of nonlinearity, the modified algorithm increases the valid prediction time interval by about two units of Lyapunov time. The benefit of the modification is more pronounced at higher degrees of nonlinearity.

Next-generation RC models, built on the basis of partial observations, are suitable not only for short-term forecasting but are also capable of reproducing the long-term climate of chaotic systems. The value of the largest Lyapunov exponent estimated from the time series generated by the trained NG-RC-Ch model is close to the value estimated from the original time series. Using the time series of only one component of the state vector of the Rössler system, we were able to reconstruct the bifurcation diagram. Having trained the NG-RC-Ch model on four time series corresponding to different values of the bifurcation parameter, we reproduced the dynamics of the system at values of the bifurcation parameter not included in the training set. A long-term prediction of the Lorenz system was demonstrated by the successful reconstruction of the return map.

To test the performance of the NG-RC-Ch algorithm in a real experimental situation, where noise and measurement errors are inevitable, we applied it to the time series generated by an electronic Rösler-like oscillator [49]. The reconstructed return map obtained from the trained model looks less noisy than the return map of the original time series. This is consistent with the results of a recent publication showing that a conventional RC can be used as a noise filter [27].

The choice of Chebyshev polynomials in our modified NG-RC-Ch algorithm was motivated by the fact that these polynomials have a unique and important minimax property. They provide a near-best uniform approximation of a function defined on a finite interval by a polynomial of a given degree [55]. An interesting question is how the performance of the NG-RC algorithm changes if we replace Chebyshev polynomials with other types of orthogonal polynomials, and how do these results depend on the particular time series we want to predict? We leave the analysis of these issues for future work.

The main unsolved problem of both conventional and next-generation RCs is related to the stability of the autonomous dynamics of the reconstructed model [5,15,53]. The RC model can be well fitted to the training time series, but due to instability its trajectory may asymptotically deviate from the strange attractor of the original system and produce a false climate. The question of how to construct an RC model that autonomously generates asymptotically stable trajectories close to a true strange attractor remains open.

ACKNOWLEDGMENTS

This work was supported by the Research Council of Lithuania through Grant No. P-MIP-24-29.

APPENDIX: PARAMETER-DEPENDENT ALGORITHM

Suppose we have n_p scalar time series measured for different values of parameter p: $\{p_j\}_{j=1,\dots,n_p}$. For simplicity, we assume that all time series have the same length n and the same embedding delay time can be used for each of them. Our goal is to incorporate all these time series into the training procedure and modify the algorithm so it can predict the system dynamics for parameter values not included in the training set.

We denote the scaled values of the time series measured at parameter value p_j as $R_{i|j}$, where $i = 1, \dots, n$ and $j = 1, \dots, n_p$. The parameter is incorporated in the algorithm as an additional component of the state vector, extending its dimension by one. Specifically, instead of the state vector defined by the Eq. (5), we now write

$$\mathbf{R}_{i|j} = [R_{i|j}, R_{i-n_\tau|j}, \dots, R_{i-(k-1)n_\tau|j}, p_j]^T. \tag{A1}$$

Using the components of this state vector, we compose the feature vectors in the same way as described in Sec. II A. The linear part of the feature vector now has the form

$$\mathbf{O}^{(1)}_{i|j} = \mathbf{R}_{i|j}. \tag{A2}$$

For the NG-RC algorithm, the nonlinear vector $\mathbf{O}^{(l)}_{i|j}$ is the set of all unique monomials of degree l composed of all components of the state vector $\mathbf{R}_{i|j}$. For example, in the case of two-dimensional embedding space, $k = 2$, the second-order nonlinear vector $l = 2$ is as follows:

$$\mathbf{O}^{(2)}_{i|j} = \left[R^2_{i|j}, R_{i|j}R_{i-n_\tau|j}, R_{i|j}p_j, R^2_{i-n_\tau|j}, R_{i-n_\tau|j}p_j, p_j^2\right]^T. \tag{A3}$$

For the NG-RC-Ch algorithm, the monomials should be replaced by Chebyshev polynomials, and the above example looks like this:

$$\mathbf{O}^{(2)}_{i|j} = [T_2(R_{i|j}),\ T_1(R_{i|j})T_1(R_{i-n_\tau|j}),\ T_1(R_{i|j})T_1(p_j),$$
$$T_2(R_{i-n_\tau|j}),\ T_1(R_{i-n_\tau|j})T_1(p_j),\ T_2(p_j)]^T. \tag{A4}$$

Note that in this case, the parameter p must be scaled so its scaled value is in the interval $[-1, 1]$ throughout the whole range of values to be studied. In the general case, the length of the vector $\mathbf{O}^{(l)}_{i|j}$ is equal to $d_l = \prod_{i=0}^{l-1}(k+1+i)/l!$. The total feature vector including all vectors $\mathbf{O}^{(l)}_{i,j}$ with degrees up to mth order can be written as

$$\mathbf{O}_{i|j} = 1 \oplus \mathbf{O}^{(1)}_{i|j} \oplus \mathbf{O}^{(2)}_{i|j} \dots \oplus \mathbf{O}^{(m)}_{i|j}. \tag{A5}$$

As before, we introduce a notation for the difference between two consecutive measurements, for a fixed parameter p_j:

$$Y_{i|j} = (R_{i+1|j} - R_{i|j})/\Delta t. \tag{A6}$$

Now we combine all differences $Y_{i|j}$ for the jth training time series into a row vector

$$\tilde{\mathbf{Y}}_j = [Y_{n-1|j},\ Y_{n-2|j},\ \dots,\ Y_{n-(k-1)n_\tau+1|j}] \tag{A7}$$

and merge all vectors $\tilde{\mathbf{Y}}_j$ into the total row vector:

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{Y}}_1 \oplus \tilde{\mathbf{Y}}_2 \dots \oplus \tilde{\mathbf{Y}}_{n_p}. \tag{A8}$$

Accordingly, we first combine the feature vectors of the time series corresponding to the jth parameter value:

$$\tilde{\mathbf{O}}_j = [\mathbf{O}_{n-1|j},\ \mathbf{O}_{n-2|j}, \dots,\ \mathbf{O}_{(k-1)n_\tau+1|j}]. \tag{A9}$$

Note that $\mathbf{O}_{i|j}$ is a column vector, while $\tilde{\mathbf{O}}_j$ is a matrix. Combining the matrices $\tilde{\mathbf{O}}_j$ for all time series

$$\tilde{\mathbf{O}} = [\tilde{\mathbf{O}}_1,\ \tilde{\mathbf{O}}_2, \dots,\ \tilde{\mathbf{O}}_{n_p}], \tag{A10}$$

we finally write the regression problem in the form of

$$\tilde{\mathbf{Y}} = \mathbf{W}_{\text{out}}\tilde{\mathbf{O}}. \tag{A11}$$

Formally, this equation has the same form as Eq. (15), and given the new definitions for $\tilde{\mathbf{Y}}$ [Eq. (A8)] and $\tilde{\mathbf{O}}$ [Eq. (A10)], its solution can be expressed as Eq. (16).

[1] B. Ramadevi and K. Bingi, Chaotic time series forecasting approaches using machine learning techniques: A review, Symmetry **14**, 955 (2022).
[2] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks, Technical Report GMD Report 148, German National Research Center for Information Technology, 2001.
[3] 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).
[4] M. Lukoševičius and H. Jaeger, Reservoir computing approaches to recurrent neural network training, Comput. Sci. Rev. **3**, 127 (2009).
[5] M. Lukoševičius, A practical guide to applying echo state networks, in *Neural Networks: Tricks of the Trade*, 2nd ed., edited by G. Montavon, G. B. Orr, and K.-R. Müller (Springer, Berlin, 2012), pp. 659–686.
[6] K. Nakajima and I. Fischer, eds., *Reservoir Computing: Theory, Physical Implementations, and Applications* (Springer, Singapore, 2021).
[7] H. Zhang and D. V. Vargas, A survey on reservoir computing and its interdisciplinary applications beyond traditional machine learning, IEEE Access **11**, 81033 (2023).
[8] D. J. Gauthier, E. Bollt, A. Griffith, and W. A. S. Barbosa, Next generation reservoir computing, Nat. Commun. **12**, 5564 (2021).
[9] D. J. Gauthier, I. Fischer, and A. Röhm, Learning unseen coexisting attractors, Chaos **32**, 113107 (2022).
[10] S. Shahi, F. H. Fenton, and E. M. Cherry, Prediction of chaotic time series using recurrent neural networks and reservoir computing techniques: A comparative study, Mach. Learn. Appl. **8**, 100300 (2022).
[11] S. Liu, J. Xiao, Z. Yan, and J. Gao, Noise resistance of next-generation reservoir computing: a comparative study with high-order correlation computation, Nonlinear Dyn. **111**, 14295 (2023).
[12] D. Köglmayr and C. Räth, Extrapolating tipping points and simulating non-stationary dynamics of complex systems using efficient machine learning, Sci. Rep. **14**, 507 (2024).
[13] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science **304**, 78 (2004).
[14] 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).
[15] Z. Lu, B. R. Hunt, and E. Ott, Attractor reconstruction by machine learning, Chaos **28**, 061104 (2018).
[16] 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).
[17] A. Griffith, A. Pomerance, and D. J. Gauthier, Forecasting chaotic systems with very low connectivity reservoir computers, Chaos **29**, 123108 (2019).
[18] P. Vlachas, J. Pathak, B. Hunt, T. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos, Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics, Neural Networks **126**, 191 (2020).
[19] V. Pyragas and K. Pyragas, Using reservoir computer to predict and prevent extreme events, Phys. Lett. A **384**, 126591 (2020).
[20] A. Röhm, D. J. Gauthier, and I. Fischer, Model-free inference of unseen attractors: Reconstructing phase space features from a single noisy trajectory using reservoir computing, Chaos **31**, 103127 (2021).
[21] J. A. Platt, S. G. Penny, T. A. Smith, T.-C. Chen, and H. D. Abarbanel, A systematic exploration of reservoir computing for forecasting complex spatiotemporal dynamics, Neural Networks **153**, 530 (2022).
[22] Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, Reservoir observers: Model-free inference of unmeasured variables in chaotic systems, Chaos **27**, 041102 (2017).
[23] 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).
[24] J. Z. Kim, Z. Lu, E. Nozari, G. J. Pappas, and D. S. Bassett, Teaching recurrent neural networks to infer global temporal structure from local examples, Nat. Mach. Intell. **3**, 316 (2021).
[25] 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).
[26] L.-W. Kong, Y. Weng, B. Glaz, M. Haile, and Y.-C. Lai, Reservoir computing as digital twins for nonlinear dynamical systems, Chaos **33**, 033111 (2023).
[27] H. Luo, Y. Du, H. Fan, X. Wang, J. Guo, and X. Wang, Reconstructing bifurcation diagrams of chaotic circuits with reservoir computing, Phys. Rev. E **109**, 024210 (2024).
[28] L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification, Phys. Rev. X **7**, 011015 (2017).
[29] D. Canaday, A. Griffith, and D. J. Gauthier, Rapid time series prediction with a hardware-based reservoir computer, Chaos **28**, 123119 (2018).
[30] 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 **115**, 100 (2019).
[31] M. Nakajima, K. Tanaka, and T. Hashimoto, Scalable reservoir computing on coherent linear photonic processor, Commun. Phys. **4**, 20 (2021).
[32] S. Kan, K. Nakajima, Y. Takeshima, T. Asai, Y. Kuwahara, and M. Akai-Kasaya, Simple reservoir computing capitalizing on the nonlinear response of materials: Theory and physical implementations, Phys. Rev. Appl. **15**, 024030 (2021).
[33] M. Cucchi, S. Abreu, G. Ciccone, D. Brunner, and H. Kleemann, Hands-on reservoir computing: A tutorial for practical implementation, Neuromorph. Comput. Eng. **2**, 032002 (2022).
[34] T. Lymburn, A. Khor, T. Stemler, D. C. Corrêa, M. Small, and T. Jüngling, Consistency in echo-state networks, Chaos **29**, 023118 (2019).
[35] J. A. Platt, A. Wong, R. Clark, S. G. Penny, and H. D. I. Abarbanel, Robust forecasting using predictive generalized synchronization in reservoir computing, Chaos **31**, 123118 (2021).
[36] A. Shirin, I. S. Klickstein, and F. Sorrentino, Stability analysis of reservoir computers dynamics via Lyapunov functions, Chaos **29**, 103147 (2019).

[37] R. Martinenghi, S. Rybalko, M. Jacquot, Y. K. Chembo, and L. Larger, Photonic nonlinear transient computing with multiple-delay wavelength dynamics, Phys. Rev. Lett. **108**, 244101 (2012).

[38] T. L. Carroll and L. M. Pecora, Network structure effects in reservoir computers, Chaos **29**, 083130 (2019).

[39] M. Dale, S. O'Keefe, A. Sebald, S. Stepney, and M. A. Trefzer, Reservoir computing quality: Connectivity and topology, Nat. Comput. **20**, 205 (2021).

[40] L. Grigoryeva and J.-P. Ortega, Dimension reduction in recurrent networks by canonicalization, J. Geom. Mech. **13**, 647 (2021).

[41] X.-Y. Duan, X. Ying, S.-Y. Leng, J. Kurths, W. Lin, and H.-F. Ma, Embedding theory of reservoir computing and reducing reservoir network using time delays, Phys. Rev. Res. **5**, L022041 (2023).

[42] E. Del Frate, A. Shirin, and F. Sorrentino, Reservoir computing with random and optimized time-shifts, Chaos **31**, 121103 (2021).

[43] L. Gonon and J.-P. Ortega, Reservoir computing universality with stochastic inputs, IEEE Trans. Neural Netw. Learn. Syst. **31**, 100 (2020).

[44] A. G. Hart, J. L. Hook, and J. H. Dawes, Echo State Networks trained by Tikhonov least squares are L2(μ) approximators of ergodic dynamical systems, Physica D **421**, 132882 (2021).

[45] E. Bollt, On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD, Chaos **31**, 013108 (2021).

[46] W. A. S. Barbosa and D. J. Gauthier, Learning spatiotemporal chaos using next-generation reservoir computing, Chaos **32**, 093137 (2022).

[47] O. Rössler, An equation for continuous chaos, Phys. Lett. A **57**, 397 (1976).

[48] E. N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. **20**, 130 (1963).

[49] V. P. Vera-Ávila, R. Sevilla-Escoboza, A. A. Lozano-Sánchez, R. R. Rivera-Durón, and J. M. Buldú, Experimental datasets of networks of nonlinear oscillators: Structure and dynamics during the path to synchronization, Data Brief **28**, 105012 (2020).

[50] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, 2nd ed. (Cambridge University Press, Cambridge, 2004).

[51] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, Determining embedding dimension for phase-space reconstruction using a geometrical construction, Phys. Rev. A **45**, 3403 (1992).

[52] A. M. Fraser and H. L. Swinney, Independent coordinates for strange attractors from mutual information, Phys. Rev. A **33**, 1134 (1986).

[53] Y. Zhang and S. P. Cornelius, Catch-22s of reservoir computing, Phys. Rev. Res. **5**, 033213 (2023).

[54] M. Rosenstein, J. Collins, and C. Luca, A practical method for calculating largest Lyapunov exponents from small data sets, Physica D **65**, 117 (1993).

[55] L. N. Trefethen, *Approximation Theory and Approximation Practice, Extended Edition* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2019).