



OPEN

Extrapolating tipping points and simulating non-stationary dynamics of complex systems using efficient machine learning

Daniel Köglmayr[✉] & Christoph Räth

Model-free and data-driven prediction of tipping point transitions in nonlinear dynamical systems is a challenging and outstanding task in complex systems science. We propose a novel, fully data-driven machine learning algorithm based on next-generation reservoir computing to extrapolate the bifurcation behavior of nonlinear dynamical systems using stationary training data samples. We show that this method can extrapolate tipping point transitions. Furthermore, it is demonstrated that the trained next-generation reservoir computing architecture can be used to predict non-stationary dynamics with time-varying bifurcation parameters. In doing so, post-tipping point dynamics of unseen parameter regions can be simulated.

Small perturbations in a complex system can dramatically change its evolution¹. A lack of precision in determining the exact state of the system can lead to an amplified lack of certainty about the future behavior of the system. This is the case even when we know its true governing equations and the exact boundary conditions. How can we then deal with complex systems where, in addition, we do not know the governing equations and must rely solely on observational data?

In recent years promising and remarkably efficient machine learning methods were proposed that use observational data as training data to autonomously generate a model that can explain the data^{2–4}. One prominent example is a recurrent neural network method called reservoir computing^{5,6} (RC). A reservoir computer creates a high-dimensional nonlinear representation of the observed dynamical system and synchronizes it with the corresponding input data. The synchronized representation is then trained on the desired output target so that the reservoir computer becomes an autonomous dynamical system whose output dynamics resemble that of the analyzed system. This way, it can achieve cutting-edge performances in predicting short- and long-term behavior of chaotic systems and outperforms other machine learning approaches like LSTMs or DNNs^{7,8}. In September 2021, Gauthier et al. published the next-generation reservoir computing architecture (NG-RC), highlighting its lack of randomness, the fewer hyperparameters, the smaller amount of required training data, and its performance gain in speed compared to the traditional approach⁹. In traditional reservoir computing, randomly initialized matrices are used to feed the input variables of the dynamical system into a high-dimensional state space that is nonlinearized by applying a nonlinear activation function. The NG-RC uses a library of unique polynomials of time-shifted input variables to achieve a nonlinear dimensionality expansion. In both cases, the resulting state space is consistently trained on the desired output target using ridge regression to become an autonomous dynamical system. Both methods can generally be deployed with small state spaces, which, combined with the computational cheap regression, lead to highly efficient algorithms.

So far, these algorithms have been used mainly for analyzing *stationary* dynamical systems, where the boundary conditions of the system are assumed to be fixed, i.e., time-independent. In this case, the qualitative behavior of the system, such as periodicity or chaoticity, remains the same over time. However, in most real-world systems, the boundary conditions can change over time, possibly leading to a qualitative change in the behavior of the system, e.g., from stable periodicity to chaos or from chaos to system collapse. These systems are called *non-stationary* dynamical systems, and the boundary condition under which the system undergoes such a critical transition is termed *tipping point*. Extrapolating tipping points is of great interest in many scientific fields. A prominent example is the evolution of the climate system influenced by atmospheric greenhouse gas concentrations, for which several tipping points are predicted¹⁰. Irrgang et al. surveyed the role of artificial intelligence

German Aerospace Center (DLR), Institute for AI Safety and Security, 89081 Ulm, Germany. ✉email: daniel.koeglmayr@dlr.de

for earth system modeling. They highlighted the concern that current classic earth system models might not be capable of predicting future abrupt climate changes¹¹. Hence, data-driven methods that capture the underlying physics seem suitable to augment classic models.

Reservoir computing-based methods for analyzing non-stationary dynamical systems and, thus, for the possible data-driven extrapolation of tipping points are still in their early stages. Two different approaches are emerging in the current literature, which mainly differs from each other in the form of their training data. One approach directly uses non-stationary time series to train the reservoir computer^{12–14}. The other approach uses several stationary data samples from different boundary conditions, i.e., bifurcation parameters, to train the reservoir computer^{15–17}. The latter uses the multifunctional capabilities of reservoir computing, which are complemented by an additional parameter channel to test for new an unseen parameter regions. It has been shown that a reservoir computer can be optimized to predict several different dynamical systems with a single trained architecture^{18–22}. Kim et al. showed that reservoirs can learn with this approach and the additional parameter channel to interpolate and extrapolate translations, linear transformations, and bifurcations of the Lorenz attractor¹⁵. In¹⁶, Kong et al. statistically evaluated the parametrization of a system collapse, or global bifurcations, of a chaotic food chain model and a generic power system model. Kong et al. were also able to reconstruct bifurcation diagrams of driven chaotic systems¹⁷.

In this work a framework for parameter-aware next-generation reservoir computing is developed. By means of the examples used in¹⁶, the functionality of the developed method is demonstrated and it is shown that the method is capable of accurately reconstructing bifurcation diagrams and simulating non-stationary dynamics, even in situations where the data is limited and the parameterization of the training data is far from the global bifurcation of interest.

Results

Recently, a new type of RC called next-generation reservoir computing (NG-RC) has been introduced for the analysis of dynamical systems. In its functional core, the algorithm first collects the time-shifted input variables of the time series data to be analyzed into a vector. In a second step, each unique polynomial combination of certain orders of the entries in the previously collected vector is determined and appended. In this way, the feature vector is created. During training, the linear mapping of the feature vector to the corresponding next time series data point is optimized using ridge regression. Due to this minimal architecture, the NG-RC features excellent speed and lacks any randomness. Besides these operational advantages, it has been shown in several publications that NG-RC requires significantly less training data than the already data sparing traditional reservoir computer^{9,23–25}, which makes NG-RC a highly efficient method for analyzing and predicting dynamical systems.

The algorithm proposed in this paper models an additional input channel for a bifurcation parameter into the NG-RC architecture by adding the parameter times a scaling parameter to each entry of the feature vector (see Methods). This allows the algorithm to learn a dynamical system also in terms of its bifurcation parameter. After training, the parameter can be varied so that the prediction of the algorithm can be tested for unseen parameter regions. The parameter-aware next-generation reservoir computing architecture is applied below to two systems of ordinary differential equations, a generic power system model²⁶ and a chaotic food chain model²⁷. Both systems were examples for statistical evaluation of tipping points using traditional reservoir computing¹⁶, moreover, their equations contain terms that cannot be directly represented by the polynomial structure of the feature vector, making them informative test systems for the parameter-aware NG-RC.

For the generic power system model, the NG-RC architecture is used to predict the bifurcation diagram and to extrapolate the tipping point. To evaluate its prediction quality, the largest Lyapunov exponents are compared with those of the model equations. The Lyapunov exponent is a measure of the long-term statistical behavior, or statistical climate, of a time series and indicates how chaotic or periodic a time series is (see Methods). It is also demonstrated that the correct choice of the scaling parameter is important and affects the quality of the prediction. For the chaotic food chain model, the influence of the scaling parameter on the extrapolation is further investigated. For that, 15 bifurcation diagrams predicted by the same NG-RC architecture are shown, which only differ in a slightly different scaling parameter. Furthermore, it is shown that the trained NG-RC can be used to simulate non-stationary dynamics in unseen parameter regions, capturing the main behavior of the dynamics even after passing through a tipping point.

Power system model

Dobson and Chiang formulated a set of generic equations to model the collapse of electrical power systems. This can be caused by the dynamic response of the system to disturbances, which may lead to a progressive drop in voltage, causing what is known as a “voltage collapse” or blackout²⁶. In the upper plot of Fig. 1, the bifurcation diagram of the generic equations of the power system model is scattered in red. The corresponding Lyapunov exponents are plotted to measure the dynamic behavior of the system. It evolves from a periodic dynamic to a chaotic one for increasing bifurcation parameters. In some areas, it shows periodic windows. The system collapses at the critical bifurcation parameter $Q_{1c} = 2.989820$, and the total voltage drops to zero. The presented NG-RC architecture aims to reconstruct the bifurcation diagram with matching Lyapunov exponents. In this example, seven training data samples are taken from different and widely separated regions of the bifurcation diagram to be analyzed. The bifurcation parameter of these are highlighted as vertical green dashed lines.

Results

The parameter-aware NG-RC architecture was applied and tested with different scaling parameters. The reconstructed bifurcation diagram of the best performing architecture is scattered in the middle plot of Fig. 1 in green, and its corresponding Lyapunov exponents are plotted in blue. It captures the main dynamical behaviors of the

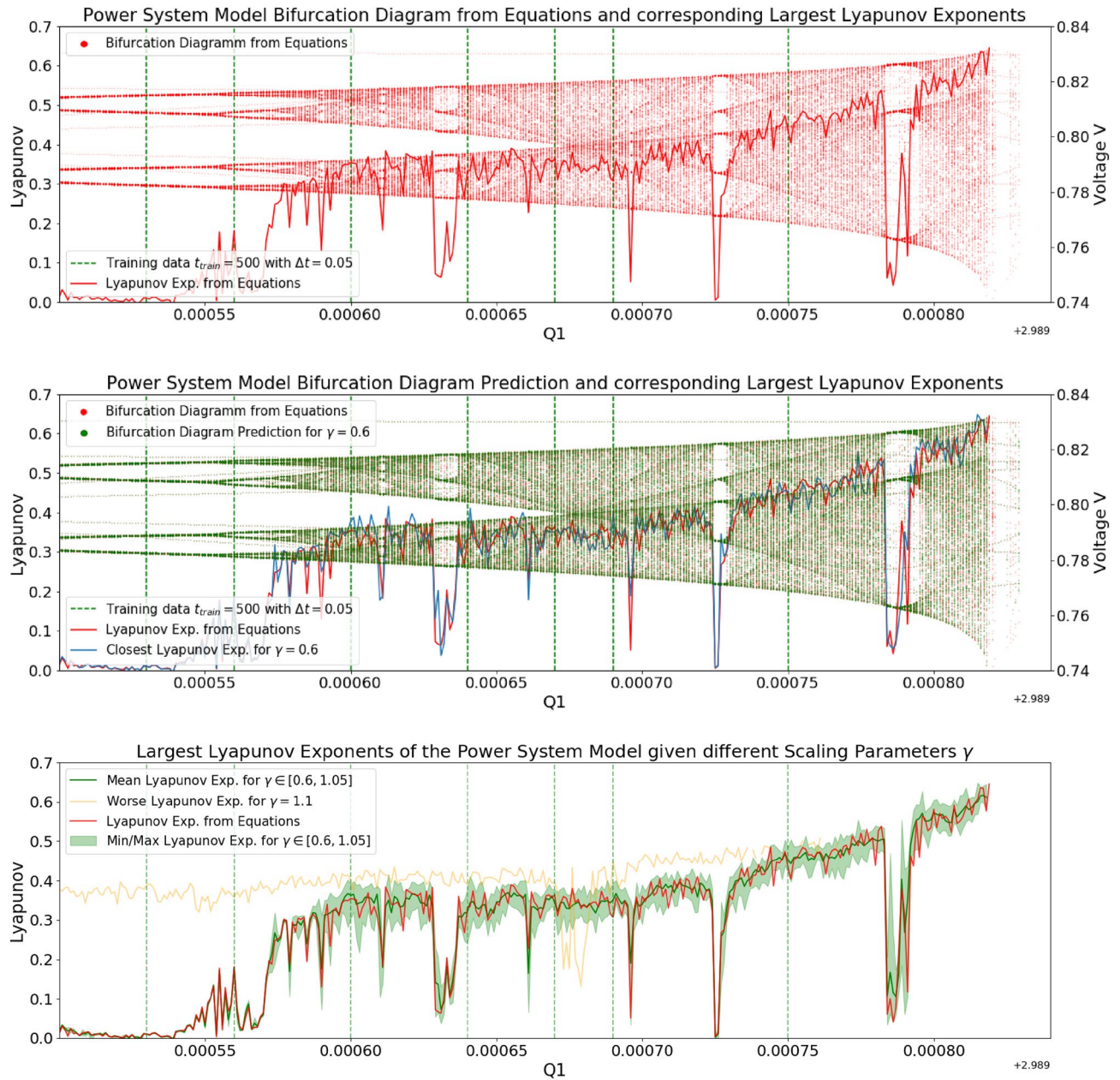


Figure 1. Upper Plot: The bifurcation diagram of the power system model with a voltage collapse at $Q_{1c} = 2.989 + 0.000819$. Middle Plot: The predicted bifurcation diagram of the power system model using the proposed NG-RC architecture with a scaling parameter of $\gamma = 0.6$ is scattered in green. The architecture was trained with seven stationary training data samples. The corresponding bifurcation parameters are highlighted as green dashed lines with 10000 training data points each. For visual comparison, the bifurcation diagram of the model equations is scattered in red and also shown in the upper plot. The respective largest Lyapunov exponents of the two systems are plotted for the quantitative comparison. Both show systematic similarities. Lower Plot: The NG-RC architecture is tested for different scaling parameters. The enclosed area of minimum and maximum predicted Lyapunov exponents for scaling parameters in $\gamma \in [0.6, 1.05]$ is highlighted in light green. All capture the systematic behaviors of the model. Above a scaling parameter of $\gamma = 1.1$, the NG-RC architecture loses its predictive power.

model. Between the area of training data samples, the architecture interpolates the periodic windows even though none of the samples were set in a similar region. In extrapolating the dynamics, the architecture captures the periodic window starting at $Q_1 = 2.989784$ and predicts the system collapse at $Q_{1c} = 2.989819$. The dynamical properties of the model were successfully captured for a set of scaling parameters in the range of $\gamma \in [0.6, 1.05]$. The corresponding minimum and maximum Lyapunov exponents of these parameters are plotted in the lower plot of Fig. 1. The Lyapunov exponents of the model equations lie well in between this region. The Lyapunov exponents for $\gamma = 1.1$ are plotted in yellow, showing that the NG-RC architecture could not predict the dynamical

properties of the model. For some parameters of the training data, the Lyapunov exponents of the predicted dynamics differ strongly from those of the training data, which allows for direct validation of prediction performance given the applied scaling parameter. This makes the introduced scaling parameter a functional new hyper-parameter, which is worthwhile tuning in this setup. Its functionality is further investigated in the next example.

Chaotic food chain model

McCann and Yodzis showed that ecosystem behavior, when it transitions to chaotic transient dynamics, can cause sudden and unpredictable disappearance of populations. Due to the realistic and nonlinear functional response properties of productive environments, sudden and unexpected jumps to other dynamical population density attractors may occur, potentially causing the disappearance of a population²⁷. To model this, they used a three-species food chain model with a resource density R , a consumer density C , and a predator density P . The resource-carrying capacity K of the environment is taken as the bifurcation parameter. The bifurcation diagram of the model equations is scattered in Fig. 2 in red and with a larger K space in Fig. 3. This system of equations shows rich bifurcation structures. When the resource-carrying capacity K reaches a critical value of $K_{c_1} = 1.00050$, the chaotic oscillating predator density P suddenly drops to 0, and the predator population disappears. For $K_{c_2} = 1.04$, this density reappears in a reverse manner. Both system behaviors are global bifurcations. Notably, there is another one at $K_{c_3} = 0.96075$, where the predator density performs a sudden jump. This time the

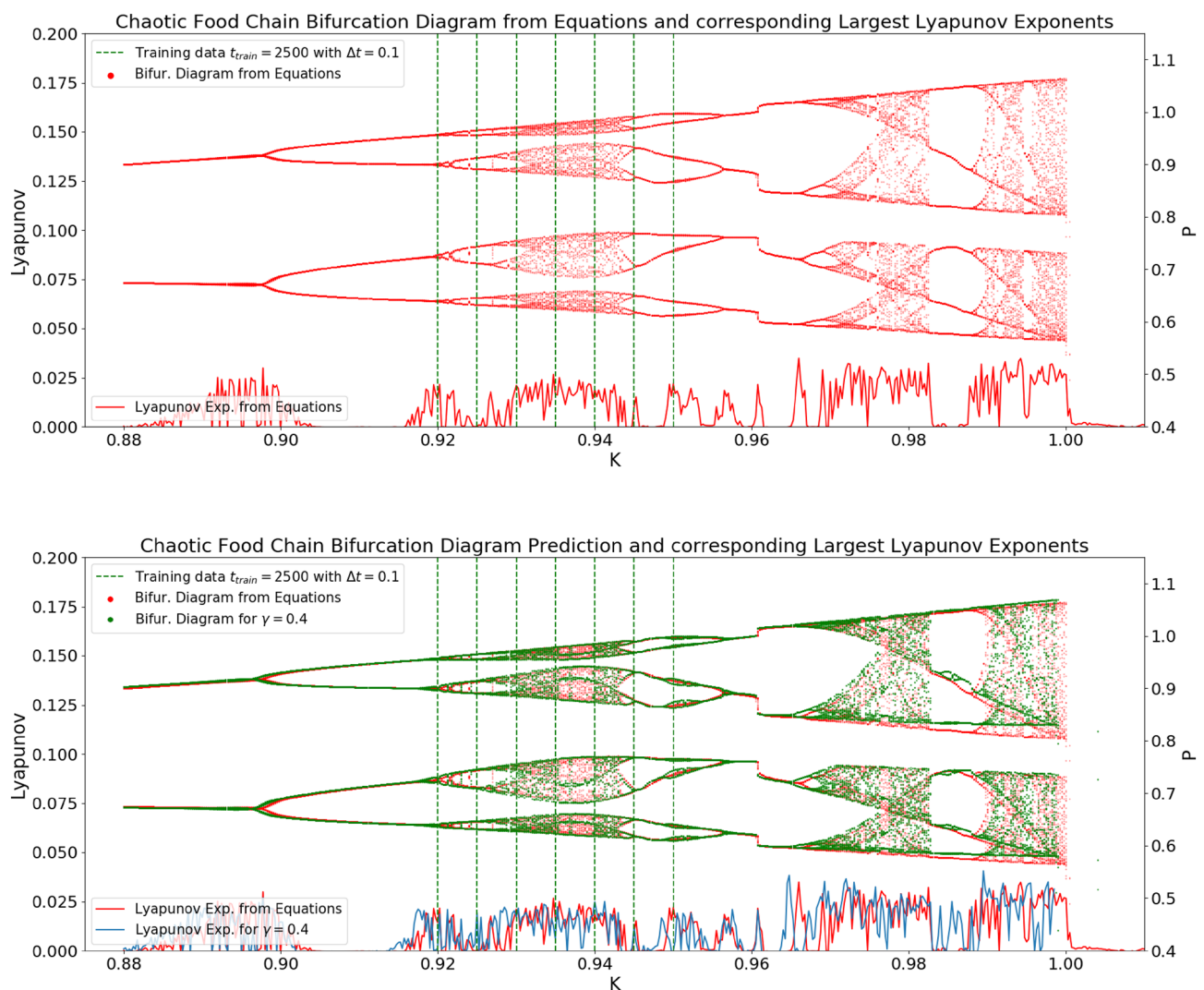


Figure 2. Upper Plot: The bifurcation diagram of the chaotic food chain model. Lower Plot: Testing the extrapolation capabilities through a reduced parameterization range of training data samples. The predicted bifurcation diagram of the chaotic food chain model using the proposed NG-RC architecture with a scaling parameter of $\gamma = 0.4$ is scattered in green. The architecture was trained with seven stationary training data samples and with 25000 training data points each. The bifurcation diagram of the model equations is scattered in red. The respective largest Lyapunov exponents of the two systems are plotted for the quantitative comparison. The predicted diagram shows systematic similarities that are shifted on the K -axis the further away they are from the parameterization of the training data.

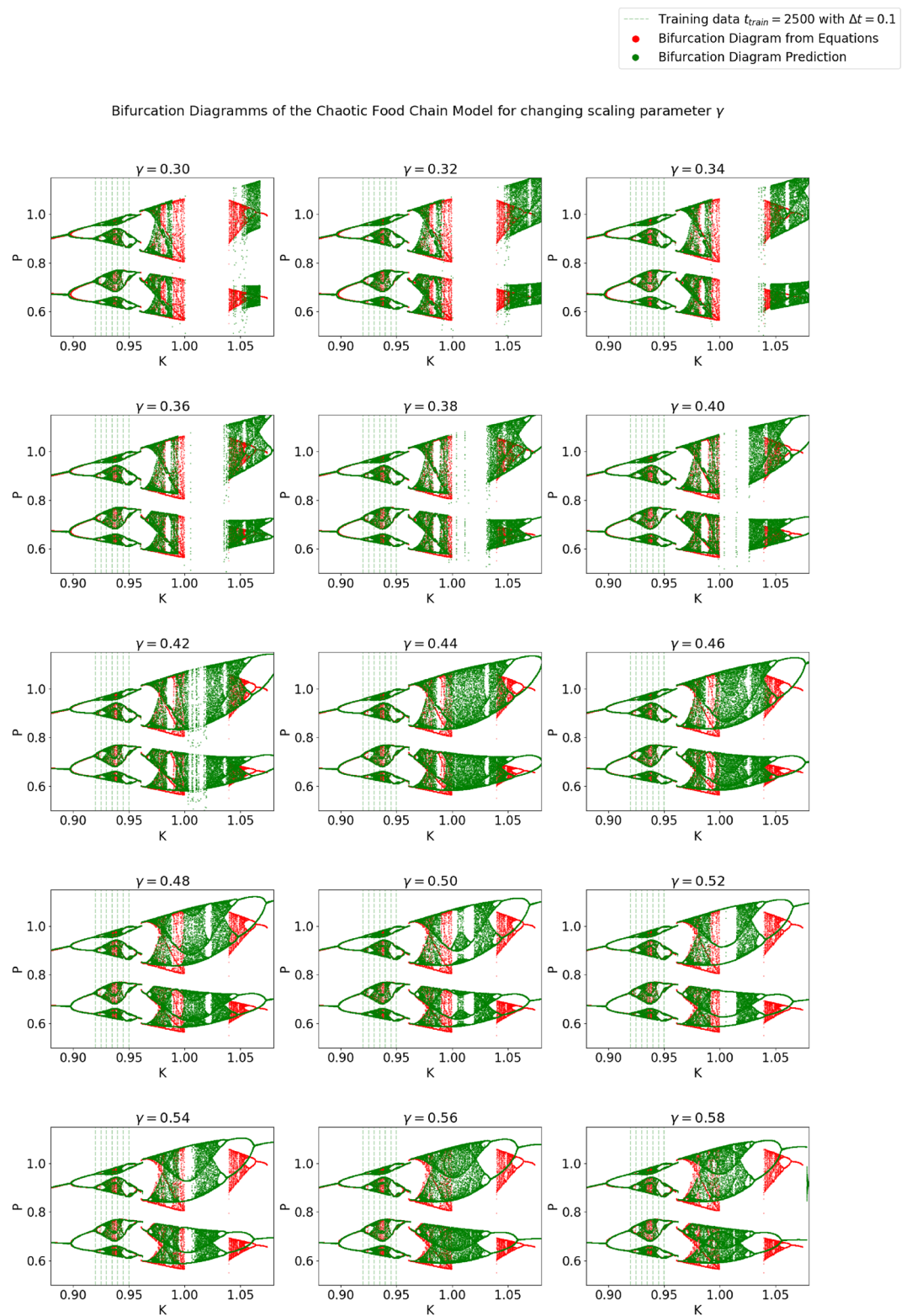


Figure 3. The NG-RC architecture was tested for different scaling parameters on the chaotic food chain model. This illustrates the effect of the newly introduced hyperparameter on the system behavior of the learned NG-RC dynamic and the extrapolation capabilities of tipping points.

NG-RC architecture aims to reconstruct the bifurcation diagram given stationary training data samples, which are narrowed down to a more minor part of the bifurcation diagram compared to those taken in the previous example. The performance of the extrapolation of tipping points is evaluated regarding the introduced scaling parameter. Non-stationary dynamics are simulated, passing through the tipping point K_{c_3} .

Results

The prediction performance of the NG-RC architecture was investigated for different scaling parameters. The best performing architecture with $\gamma = 0.4$ is scattered in Fig. 2 in green. The nearest tipping point from the parameterization of the training data at $K_{c_3} = 0.96075$ and the subsequent transition from chaoticity to periodicity at $K = 0.983$ are accurately predicted. The tipping point at $K_{c_1} = 1.00050$ was predicted with $K = 0.99875$. The bifurcation that is farthest away, $K_{c_2} = 1.04$, was extrapolated with $K = 1.027$ (see Fig. 3). A qualitative difference between the prediction and the model is that the predicted trajectory after the tipping point at $K = 0.99875$ goes to minus infinity, whereas the real one goes to zero. Clear topological differences can be seen between the training data samples at $K = 0.935$ and $K = 0.94$. This area incorrectly shows the properties of a periodic window. This also applies to the interpolation for different scaling parameters. Thus, the effect of the scaling parameter on the interpolation capabilities between the parameter space of training data samples is limited. As expected, accurate extrapolation of possible tipping points becomes more difficult the further they are from the parameterization of the training data. Looking at the evolution of the bifurcation diagrams for different scaling parameters in Fig. 3 was instructive to see the influence of the scaling parameter on the extrapolation capability. From $\gamma = 0.3$ on to $\gamma = 0.4$, the increasing scaling parameter stretches the bifurcation topology in the range of $K \in [0.95, 1]$. Interestingly, there is a qualitative change in the bifurcation diagram when the scaling parameter stretches it over the actual tipping point at $K_{c_1} = 1.00050$. The tipping point prediction is lost, and the NG-RC transforms the two parts of the bifurcation diagram into a continuous one. This behavior generally allows practical parameter tuning of the scaling parameter by introducing validation data to determine the necessary degree of stretching. Moreover, the trained NG-RC architecture for $\gamma = 0.4$ is used to simulate non-stationary dynamics using Eq. (15). In Fig. 4, the bifurcation parameter switches from $K = 0.955$ to $K = 0.965$ over the tipping point at $K_{c_3} = 0.96075$. The predicted trajectory captures this transition. Using the identical trained NG-RC architecture, a sinusoidal and linearly increasing function of K is taken as another example. The result is shown in Fig. 5. The dominant dynamical behaviors concerning the bifurcation parameter regions are captured in the prediction. These examples illustrate the applicability of this architecture, which enables the simulations of non-stationary dynamics as a function of a time-varying bifurcation parameter.

Discussion

A machine learning method based on parameter-aware next-generation reservoir computing was presented to investigate the bifurcation behavior of dynamical systems. It was shown that tipping points can be extrapolated. Moreover, the trained architecture can be used to simulate non-stationary dynamics and, with that, also, post-tipping point dynamics. The success of reservoir computing and next-generation reservoir computing relies on optimizing them into dynamical systems whose dynamics resemble that of the analyzed system. The presented implementation of the bifurcation parameter provided a functional input channel that allowed the investigation of the system dynamics in unseen parameter regions. It is noteworthy that, on the one hand, this method can capture high sensitivities of the analyzed dynamical system to the bifurcation parameter. This was shown in the generic power system model, where the sixth decimal place of the bifurcation parameter partly determines the dynamic behavior. On the other hand, this method integrated the bifurcation parameter, implemented by adding it times a scaling parameter to the feature space of NG-RC, so that the optimized NG-RC architecture was able to simulate and predict dynamics where the bifurcation parameter appears as an inverse parameter in the governing equation. This generally extends the applicability of this approach and was shown in the chaotic food

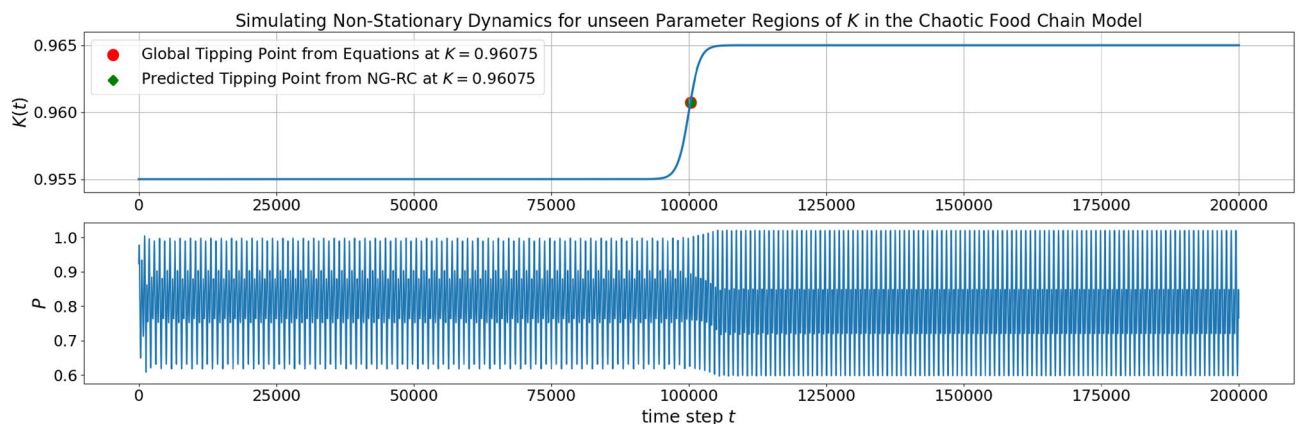


Figure 4. Simulating non-stationary dynamics of the chaotic food chain model with the trained NG-RC architecture of Eq. (27) for unseen parameter regions. The switch of the bifurcation parameter K across the tipping point at $K_{c_3} = 0.96075$ shows that the simulated trajectory captures the expected change in behavior (see Fig. 2).

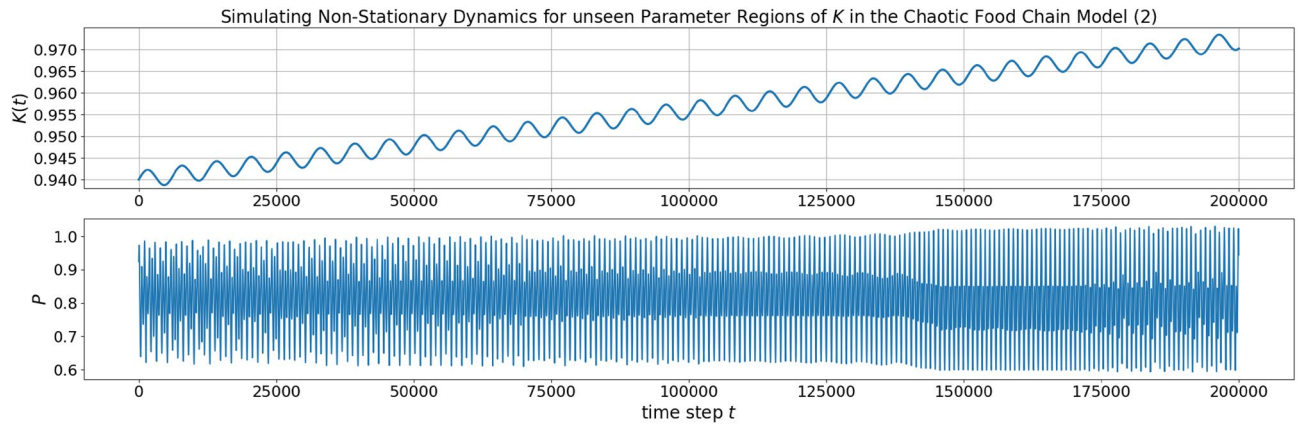


Figure 5. As another example, the trained NG-RC architecture of Eq. (27) is used to simulate non-stationary dynamics of the chaotic food chain model for a more complicated sinusoidal and linearly increasing bifurcation parameter K . The simulated trajectory captures the qualitative behaviors of the respective parameter regions (see Fig. 2).

chain example. Although both system equations contain terms that cannot be directly represented by the polynomial structure of the feature vector, the proposed NG-RC architecture was able to interpolate and extrapolate the system behaviors, which is another plus for its applicability.

So far, there are few publications in which reservoir computing methods are used to determine bifurcation diagrams of dynamical systems and their tipping points. In Kim et al.¹⁵, parameter-aware reservoir computing was used to accurately extrapolate the period doubling bifurcations of the Lorenz system around $\rho \approx 100$. For this, 4 training samples with 250000 training steps and 50000 synchronization steps were used, resulting in 1200000 data points. In Kong et al.¹⁷, the bifurcation diagram of a driven Lorenz-96 system was predicted with parameter-aware RC. This was done using 4 training samples with 140000 training steps and 800 synchronization steps each, resulting in 563000 data points. In the results presented here, the parameter-aware NG-RC required 70014 data points to train the architecture on the power system model and 175112 data points for the chaotic food chain model. A general statement that NG-RC requires significantly less training data than traditional RC, even in the case of parameter-aware extrapolation, would be overstated due to the lack of a direct comparison of the two methods. However, the results presented here provide a first tendency that the required training data can significantly be reduced. In terms of setting up a working architecture, the parameter-aware RC approach has eight tunable hyperparameters¹⁷, while the proposed NG-RC architecture has six, most of which are far less comprehensive to optimize. In addition, the NG-RC works completely without randomness. Instead of random matrices, the polynomial architecture generally ensures higher interpretability and, together with the previously mentioned points, a more direct setup to deploy a working architecture without stochastic realizations of the reservoir system. In the context of this work, the NG-RC architectures were not extensively optimized nor comprehensively investigated concerning the minimal required training data. However, if the tendency holds, further applications emerge. Since most real-world dynamical systems are of non-stationary nature, the less training data needed, the better non-stationary data samples can be approximated as stationary data. Which can improve the prediction of tipping points based on non-stationary time series data. Consequently, the here proposed parameter-aware NG-RC is an efficient, model-free, and data-driven method for extrapolating the behavior of dynamical systems and simulating non-stationary dynamics.

Methods

The method presented here is based on next-generation reservoir computing. Its architecture is extended by an input channel for a bifurcation parameter of a dynamical system. Therefore, the parameter is added to the NG-RC feature vector as a product with a scaling parameter to each element of the feature vector. The new feature vector is then extended with orders of itself. These steps are presented in detail below. A condensed mathematical description of the NG-RC is used, so that the applied architecture and its hyperparameters can be written as one equation. For the power system model results the architecture with its hyperparameters is described in Eq. (23) and for the chaotic food chain model in Eq. (27).

Next-generation reservoir computing

The d -dimensional data points $\mathbf{x} \in \mathbb{R}^d$ of the input data $\mathbf{X} = (\mathbf{x}_0, \dots, \mathbf{x}_n)$ are transformed with a polynomial multiplication dictionary \mathbf{P} into a higher dimensional state space. The unique polynomials of certain orders O , included in $\mathbf{P}^{[O]}$, are denoted by an index. For illustration purposes, we consider a two-dimensional input data point $\mathbf{x}_i = (x_{i,1}, x_{i,2})^T$ and transform it with the unique polynomials of order 1 and 2,

$$\mathbf{P}^{[1,2]}(\mathbf{x}_i) = \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,1}^2 \\ x_{i,2}^2 \\ x_{i,1}x_{i,2} \end{pmatrix}. \tag{1}$$

Further, Gauthier et al. introduced a time shift expansion \mathbf{L}_k^s of the input data. The k value indicates the number of past data points with which the current data point is concatenated. The s value indicates how far these points are separated in time. Following the previous example

$$\mathbf{P}^{[1,2]}(\mathbf{L}_{k=2}^{s=1}(\mathbf{x}_i)) = \mathbf{P}^{[1,2]} \left(\begin{pmatrix} x_{i,1} \\ x_{i,2} \\ x_{i-1,1} \\ x_{i-1,2} \end{pmatrix} \right) = \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ x_{i-1,1} \\ \vdots \\ x_{i,1}x_{i-1,2} \\ x_{i,2}x_{i-1,2} \end{pmatrix} = \mathbf{r}_{i+1}, \tag{2}$$

where $\mathbf{r}_{i+1} \in \mathbb{R}^N$ defines the feature vector with feature space dimension N . By concatenating this vector with powers of itself, higher-order features can be included in a computationally cheap way. For this purpose, an additional post-processing operator $\mathbf{q}_{[O_{states}]}$ (\mathbf{r}) is introduced, where O_{states} specifies which orders of the feature vector are to be concatenated. Defining \odot as the Hadamard product, \oplus as the vector concatenation operation, and specifying that for $0 \in O_{states}$ a bias term of dimension one is concatenated, the feature space can be extended, for example, for $O_{states} = [0, 1, 2]$, as shown below,

$$\mathbf{q}_{[0,1,2]}(\mathbf{r}_{i+1}) = 1 \oplus \mathbf{r}_{i+1} \oplus (\mathbf{r}_{i+1} \odot \mathbf{r}_{i+1}) = (1, r_{i+1,1}, \dots, r_{i+1,N}, r_{i+1,1}^2, \dots, r_{i+1,N}^2)^T = \tilde{\mathbf{r}}_{i+1} \in \mathbb{R}^{2N+1}$$

where $\tilde{\mathbf{r}}_{i+1} \in \mathbb{R}^{\tilde{N}}$ defines the expanded feature vector with dimension \tilde{N} . This vector is then mapped with a readout matrix \mathbf{W}_{out} onto the desired output target \mathbf{y}_{i+1} . During the training process, this mapping is optimized. In the training phase of the NG-RC, the input training data \mathbf{X} of length T is transformed into the feature matrix

$$\mathbf{R} = \mathbf{q}_{[O_{states}]}(\mathbf{P}^{[O]}(\mathbf{L}_k^s(\mathbf{X}))) \tag{3}$$

accordingly. Note that due to the k and s value, a warm-up time of $\delta t = ks$ is needed, where entries of the feature matrix at time $t < \delta t$ are not defined. Consequently, the output target matrix \mathbf{Y} needs to be adjusted. The output target matrix \mathbf{Y} is defined in the scope of this work as

$$\mathbf{Y} = (\Delta \mathbf{x}_{\delta t+1}, \dots, \Delta \mathbf{x}_T)^T \tag{4}$$

with $\Delta \mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_{i-1}$, such that the mapping is optimized to fulfill

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{W}_{out} \tilde{\mathbf{r}}_{i+1}. \tag{5}$$

The readout matrix \mathbf{W}_{out} is learned via ridge regression by optimizing

$$\mathbf{W}_{out} = \mathbf{Y} \mathbf{R}^T (\mathbf{R} \mathbf{R}^T + \beta \mathbf{I})^{-1}. \tag{6}$$

Matrix \mathbf{I} is an identity matrix, and β is the regression parameter. In this setup, the NG-RC is optimized to become a one-step-ahead integrator that drives the trajectory according to

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{W}_{out} \mathbf{q}_{[O_{states}]}(\mathbf{P}^{[O]}(\mathbf{L}_k^s(\mathbf{x}_i))). \tag{7}$$

Multifunctionality with input channel

Multifunctionality setup

To include the data of n trajectories into the training process of the NG-RC, the feature matrix of every trajectory \mathbf{X}_m for $m = 1, \dots, n$ is calculated with

$$\mathbf{R}_m = \mathbf{q}_{[O_{states}]}(\mathbf{P}^{[O]}(\mathbf{L}_k^s(\mathbf{X}_m))). \tag{8}$$

The resulting feature matrices are concatenated to

$$\mathbf{R}_M = \mathbf{R}_1 \oplus \mathbf{R}_2 \dots \oplus \mathbf{R}_n. \tag{9}$$

The output target matrix for each trajectory \mathbf{X}_m must also be concatenated to

$$\mathbf{Y}_M = \mathbf{Y}_1 \oplus \mathbf{Y}_2 \dots \oplus \mathbf{Y}_n. \tag{10}$$

This way, the identical training routine can be applied so that

$$\mathbf{W}_{out} = \mathbf{Y}_M \mathbf{R}_M^T (\mathbf{R}_M \mathbf{R}_M^T + \beta \mathbf{I})^{-1} \tag{11}$$

is optimized via ridge regression. Provided the training is successful, the \mathbf{W}_{out} can be used to predict the different trajectories

$$\mathbf{x}_{m,i+1} = \mathbf{x}_{m,i} + \mathbf{W}_{out} \mathbf{q}_{[O_{states}]}(\mathbf{P}^{[O]}(\mathbf{L}_k^s(\mathbf{x}_{m,i}))). \tag{12}$$

Multifunctionality setup with input channel

In the scope of this work, however, we use this architecture to modulate the bifurcation parameter for multiple stationary dynamics of a system into the feature vector so that the NG-RC can be tested on predicting the dynamics for new and unseen bifurcation parameters. Therefore, for every stationary dynamic \mathbf{X}_m in the training data, determined by its bifurcation parameter θ_m , we add to each element in the corresponding feature vector the bifurcation parameter θ_m multiplied by a scaling parameter γ ,

$$\mathbf{R}_m = \mathbf{q}_{[O_{states}]}(\mathbf{P}^{[O]}(\mathbf{L}_k^s(\mathbf{X}_m)) + \gamma\theta_m). \tag{13}$$

This concept can be trained similarly by optimizing Eq. (11). The NG-RC then drives the trajectory according to

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{W}_{out} \mathbf{q}_{[O_{states}]}(\mathbf{P}^{[O]}(\mathbf{L}_k^s(\mathbf{x}_i)) + \gamma\theta). \tag{14}$$

In addition, this structure allows to change the bifurcation parameter per prediction step,

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{W}_{out} \mathbf{q}_{[O_{states}]}(\mathbf{P}^{[O]}(\mathbf{L}_k^s(\mathbf{x}_i)) + \gamma\theta_i). \tag{15}$$

Provided that the trained architecture can predict the dynamical behavior of the system for various unseen bifurcation parameters successfully, i.e., reconstruct its bifurcation diagram, a reasonable motivation for simulating non-stationary processes can be derived from Eq. (15).

Lyapunov exponent

Due to the definition of chaos and its sensitivity to initial conditions, evaluating dynamical predictions only on their deviation from the ground truth, i.e., with its short-time behavior, can not capture essential features of dynamical systems. To determine the systematic behavior of a dynamical system, it is necessary to determine the long-term properties of the trajectory. These are referred to as the statistical climate of the system, and its measurement can give rise to how chaotic or periodic the system is. Lyapunov exponents λ_i measure the temporal complexity of the dynamical system by measuring the average divergence rate of nearby points in phase space. This gives rise to its sensitivity to initial conditions for each dimension i and quantifies the time scale on which it becomes unpredictable^{28,29}. Suppose at least one Lyapunov exponent is positive. In that case, the system is considered chaotic. The magnitude of the largest Lyapunov exponent λ_{max} can then be taken to measure the degree of chaoticity the system exhibits. In the context of this work, the Rosenstein algorithm is used to calculate the largest Lyapunov exponent³⁰.

Power system model

Model equations

The model consists of four ordinary differential equations.

$$\dot{\delta}_m = \omega, \tag{16}$$

$$M\dot{\omega} = -d_m\omega + P_m - E_m Y_m \sin(\delta_m - \delta) V, \tag{17}$$

$$K_{qw}\dot{\delta} = -K_{qv2}V^2 - K_{qv}V + Q(\delta_m, \delta, V) - Q_0 - Q_1, \tag{18}$$

$$TK_{qw}K_{pv}\dot{V} = K_{pw}K_{qv2}V^2 + (K_{pw}K_{qv} - K_{qw}K_{pv})V + K_{qw}[P(\delta_m, \delta, V) - P_0 - P_1] - K_{pw}[Q(\delta_m, \delta, V) - Q_0 - Q_1] \tag{19}$$

where

$$P(\delta_m, \delta, V) = -E'_0 Y'_0 V \sin(\delta) + E_m Y_m V \sin(\delta_m - \delta),$$

$$Q(\delta_m, \delta, V) = -E'_0 Y'_0 V \cos(\delta) - (Y'_0 + Y_m)V^2 + E_m Y_m V \cos(\delta_m - \delta).$$

The real power demand P and the reactive power demand Q of the system appear in the differential equations of the load voltage V and the motor frequency δ . The variable δ_m describes the angle dynamics between two generators and ω the speed of a generator rotor. For a more technical description, we suggest the paper by Dobson et al.²⁶. Following the model parameterization used for the traditional reservoir computing approach¹⁶,

$$E'_0 = \frac{E_0}{(1 + C^2 Y_0^{-2} - 2CY_0^{-1} \cos(\theta_0))^{\frac{1}{2}}}, \tag{20}$$

$$Y'_0 = Y_0(1 + C^2 Y_0^{-2} - 2CY_0^{-1} \cos(\theta_0))^{\frac{1}{2}}, \tag{21}$$

$$\theta'_0 = \theta_0 + \tan^{-1} \left(\frac{CY_0^{-1} \sin(\theta_0)}{1 - CY_0^{-1} \cos(\theta_0)} \right) \quad (22)$$

are set as constants and $K_{pw} = 0.4$, $K_{pv} = 0.3$, $K_{qw} = -0.03$, $K_{qv} = -2.8$, $K_{qv2} = 2.1$, $T = 8.5$, $P_0 = 0.6$, $Q_0 = 0.3$, $P_1 = 0$, $Y_0 = 3.33$, $Y_m = 5$, $P_m = 1$, $d_m = 0.05$, $\theta_0 = 0$, $E_m = 1.05$, $M = 0.01464$, $C = 3.5$, $E_0 = 1$, $Q_0 = 1.3$.

Q_1 is taken as the bifurcation parameter. It determines the load reactive power demand of the system. The model bifurcation diagram was created using Runge-Kutta 4 (RK4), starting each time from $\mathbf{x}_0 = (\delta_{m,0}, \omega_0, \delta_0, V_0)^T = (0.17, 0.05, 0.05, 0.83)^T$ for $T = 10000$ time steps with time step size $\Delta t = 0.05$ and an bifurcation parameter step size of $\Delta Q_1 = 0.000001$.

NG-RC architecture

The training data is generated identically for each system parameter in

$$\mathbf{Q}_1^{train} = [2.98953, 2.98956, 2.98960, 2.98964, 2.98967, 2.98969, 2.98975].$$

The following NG-RC architecture

$$\mathbf{R}_{Q_1,m} = \mathbf{q}_{[0,1,2,3]}(\mathbf{P}^{[1,2,3]}(\mathbf{L}_2^2(\mathbf{X}_m))) + \gamma \mathbf{Q}_{1,m}^{train}, \quad (23)$$

is used and trained with a regression parameter of $\beta = 10^{-8}$. The expanded feature vectors have dimension $\tilde{N} = 493$. The warm-up times of length $\delta t = 4$ are simulated with RK4.

Chaotic food chain model

Model equations

The three-species food chain model with a resource density R , a consumer density C , and a predator density P is modeled as follows

$$\dot{R} = R \left(1 - \frac{R}{K} \right) - \frac{x_c y_c C R}{R + R_0}, \quad (24)$$

$$\dot{C} = x_c C \left(\frac{y_c R}{R + R_0} - 1 \right) - \frac{x_p y_p P C}{C + C_0}, \quad (25)$$

$$\dot{P} = x_p P \left(\frac{y_p C}{C + C_0} - 1 \right) \quad (26)$$

with $x_c = 0.4$, $y_c = 2.009$, $x_p = 0.08$, $y_p = 2.876$, $R_0 = 0.16129$, $C_0 = 0.5$ and the resource-carrying capacity K is taken as the bifurcation parameter¹⁶. The model bifurcation diagram was created using RK4, starting each time from $\mathbf{x}_0 = (R_0, C_0, P_0)^T = (0.6, 0.35, 0.9)^T$ for $T = 25000$ time steps with time step size $\Delta t = 0.1$ and an bifurcation parameter step size of $\Delta K = 0.00025$.

NG-RC architecture

The training data is generated identically for each parameter in

$$\mathbf{K}^{train} = [0.92, 0.925, 0.93, 0.935, 0.94, 0.945, 0.95].$$

The following NG-RC architecture

$$\mathbf{R}_{K,m} = \mathbf{q}_{[0,1,2,3]}(\mathbf{P}^{[1,2]}(\mathbf{L}_4^4(\mathbf{X}_m))) + \gamma \mathbf{K}_m^{train} \quad (27)$$

is applied. The expanded feature vectors have dimension $\tilde{N} = 271$. A regression parameter of $\beta = 10^{-3}$ is used. The warm-up times of length $\delta t = 16$ are simulated with RK4.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 14 September 2023; Accepted: 23 December 2023

Published online: 04 January 2024

References

1. Lorenz, E. N. Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**, 130–141 (1963).
2. Brunton, S. L., Proctor, J. L. & Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci.* **113**, 3932–3937 (2016).
3. Ma, H., Haluszczynski, A., Prosperino, D. & R ath, C. Identifying causality drivers and deriving governing equations of nonlinear complex systems. *Chaos: Interdiscip. J. Nonlinear Sci.* **32**, 103128 (2022).
4. Huang, Y., Mabrouk, Y., Gompper, G. & Sabass, B. Sparse inference and active learning of stochastic differential equations from data. *Sci. Rep.* **12**, 21691 (2022).

5. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149 (2009).
6. Jaeger, H. & Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80 (2004).
7. Bompas, S., Geogot, B. & Guéry-Odelin, D. Accuracy of neural networks for the simulation of chaotic dynamics Precision: of training data vs precision of the algorithm. *Chaos: Interdiscip. J. Nonlinear Sci.* **30**, 113118 (2020).
8. Chattopadhyay, A., Hassanzadeh, P. & Subramanian, D. Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: Reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Process. Geophys.* **27**, 373–389 (2020).
9. Gauthier, D. J., Bollt, E., Griffith, A. & Barbosa, W. A. Next generation reservoir computing. *Nat. Commun.* **12**, 1–8 (2021).
10. Lenton, T. M. *et al.* Tipping elements in the Earth's climate system. *Proc. Natl. Acad. Sci.* **105**, 1786–1793 (2008).
11. Irrgang, C. *et al.* Towards neural earth system modelling by integrating artificial intelligence in earth system science. *Nat. Mach. Intell.* **3**, 667–674 (2021).
12. Lim, S. H., Theo Giorgini, L., Moon, W. & Wettlaufer, J. S. Predicting critical transitions in multiscale dynamical systems using reservoir computing. *Chaos: Interdiscip. J. Nonlinear Sci.* **30**, 123126 (2020).
13. Patel, D., Canaday, D., Girvan, M., Pomerance, A. & Ott, E. Using machine learning to predict statistical properties of non-stationary dynamical processes: System climate, regime transitions, and the effect of stochasticity. *Chaos: Interdiscip. J. Nonlinear Sci.* **31**, 033149 (2021).
14. Patel, D. & Ott, E. Using machine learning to anticipate tipping points and extrapolate to post-tipping dynamics of non-stationary dynamical systems. [arXiv:2207.00521](https://arxiv.org/abs/2207.00521) (2022).
15. Kim, J. Z., Lu, Z., Nozari, E., Pappas, G. J. & Bassett, D. S. Teaching recurrent neural networks to infer global temporal structure from local examples. *Nat. Mach. Intell.* **3**, 316–323 (2021).
16. Kong, L.-W., Fan, H.-W., Grebogi, C. & Lai, Y.-C. Machine learning prediction of critical transition and system collapse. *Phys. Rev. Res.* **3**, 013090 (2021).
17. Kong, L.-W., Weng, Y., Glaz, B., Haile, M. & Lai, Y.-C. Reservoir computing as digital twins for nonlinear dynamical systems. *Chaos: Interdiscip. J. Nonlinear Sci.* **33** (2023).
18. Flynn, A., Tsachouridis, V. A. & Amann, A. Multifunctionality in a reservoir computer. *Chaos: Interdiscip. J. Nonlinear Sci.* **31**, 013125 (2021).
19. Flynn, A., Herteux, J., Tsachouridis, V. A., R ath, C. & Amann, A. Symmetry kills the square in a multifunctional reservoir computer. *Chaos: Interdiscip. J. Nonlinear Sci.* **31**, 073122 (2021).
20. Flynn, A. *et al.* Exploring the limits of multifunctionality across different reservoir computers. In *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–8 (IEEE, 2022).
21. Herteux, J. & R ath, C. Breaking symmetries of the reservoir equations in echo state networks. *Chaos: Interdiscip. J. Nonlinear Sci.* **30**, 123142 (2020).
22. Lu, Z. & Bassett, D. S. Invertible generalized synchronization: A putative mechanism for implicit learning in neural systems. *Chaos: Interdiscip. J. Nonlinear Sci.* **30**, 063133 (2020).
23. Barbosa, W. A. & Gauthier, D. J. Learning spatiotemporal chaos using next-generation reservoir computing. *Chaos: Interdiscip. J. Nonlinear Sci.* **32** (2022).
24. Gauthier, D. J., Fischer, I. & R ohm, A. Learning unseen coexisting attractors. *Chaos: Interdiscip. J. Nonlinear Sci.* **32** (2022).
25. Haluszczyński, A., Koeglmayr, D. & R ath, C. Controlling dynamical systems to complex target states using machine learning: Next-generation vs. classical reservoir computing. In *2023 International Joint Conference on Neural Networks (IJCNN)*, 1–7 (IEEE, 2023).
26. Dobson, I. & Chiang, H.-D. Towards a theory of voltage collapse in electric power systems. *Syst. Control Lett.* **13**, 253–262 (1989).
27. McCann, K. & Yodzis, P. Nonlinear dynamics and population disappearances. *Am. Nat.* **144**, 873–879 (1994).
28. Wolf, A., Swift, J. B., Swinney, H. L. & Vastano, J. A. Determining Lyapunov exponents from a time series. *Physica D* **16**, 285–317 (1985).
29. Shaw, R. Strange attractors, chaotic behavior, and information flow. *Zeitschrift f ur Naturforschung A* **36**, 80–112 (1981).
30. Rosenstein, M. T., Collins, J. J. & De Luca, C. J. A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D* **65**, 117–134 (1993).

Acknowledgements

D.K. gratefully acknowledges the funding provided by Allianz Global Investors (AGI).

Author contributions

C.R. initiated the research. D.K. and C.R. designed the study. D.K. conducted the calculations. C.R. and D.K. interpreted and evaluated the findings. All authors reviewed the manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to D.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024