Using Reservoir Computing to Predict Chaotic Systems

ENAE 646: Advanced Dynamics of Aerospace Systems

Ahmed Ashry
5-11-2023

1. Introduction

In the realm of machine learning, the prediction of chaotic systems represents a formidable challenge due to their inherent complexity and unpredictability. Traditional architectures and methodologies often struggle to effectively capture the intricate dynamics of such systems. Against this backdrop, Reservoir Computing (RC), and more specifically, Echo State Networks (ESN), have emerged as a compelling alternative. The present report seeks to delve into the intricacies of these innovative strategies, shedding light on their potential in predicting chaotic systems.

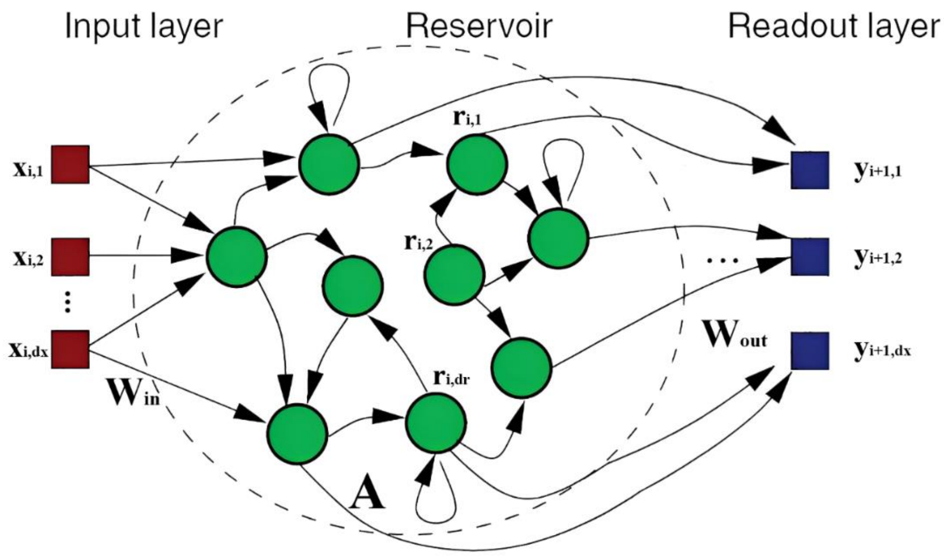
Reservoir Computing, a subset of recurrent neural networks, distinguishes itself through its unique architecture. It employs a fixed, large reservoir of neurons that process temporal data efficiently, while the only trainable component is the output layer. The Echo State Network, a notable variant of RC, utilizes an echo state property in the reservoir to capture and process complex dynamics inherent in chaotic systems.

The Echo State Property (ESP), an essential characteristic of the ESN, ensures that the reservoir's memory is influenced by its input signals, thereby "remembering" past inputs for producing meaningful output predictions. Several control parameters, including the size of the reservoir, sparsity, spectral radius (ρ), and leaking rate (α), significantly influence the ESN's performance. This report seeks to elaborate on using ESN to predict chaotic systems and tuning the parameters to optimally achieve that.

The relation between ESN and the Chaos Theory is a central point of our exploration, with specific examples such as the Lorenz attractor and Rossler attractor serving to illustrate the practical implications. The Lorenz Attractor, a paradigm of chaos, exhibits unpredictable behavior, thereby showcasing the proficiency of ESN in predicting complex systems.

The report draws on research findings, including findings from Jaeger (2002), to illuminate the finer nuances of ESN. The ultimate aim is to get a comprehensive understanding of Reservoir Computing, Echo State Networks, and their promising role in predicting chaotic systems. As I navigate this report, I will continually emphasize the importance of fine-tuning ESN parameters to achieve optimal performance, and the role of manual experimentation in this process.

2. Reservoir Computing – Mathematical Description



For data driven systems with a dataset containing N samples, denoted as $\boldsymbol{x}_i$, which are arranged in chronological order. Here, $\boldsymbol{x}_i$ is synonymous with $\boldsymbol{x}_t$, where each $\boldsymbol{x}_i$ belongs to a real-valued time-series vector with a dimension of $\boldsymbol{d}_x$. This forms the basis for the Reservoir Computing Recurrent Neural Network's discrete time update function, given as follows:

$$\boldsymbol{r}_{i+1} = (1-\alpha)\boldsymbol{r}_i + \alpha q(\mathbf{A}\boldsymbol{r}_i + \boldsymbol{u}_i) \quad (1)$$

The output layer of the Reservoir Computing network translates the output $\boldsymbol{y}_{i+1}$ as a linear conversion of a feature vector, which is constructed from the reservoir state $\boldsymbol{r}_{i+1}$. This relationship is expressed as:

$$\boldsymbol{y}_{i+1} = \mathbf{W}_{\text{out}}\boldsymbol{r}_{i+1} \quad (2)$$

In this context, the hidden variable $\boldsymbol{r}_i$, which belongs to a higher dimension (dr > dx), is coupled with $\boldsymbol{u}_i$, a linear transformation expressed as:

$$\boldsymbol{u}_i = \mathbf{W}_{in}\,\boldsymbol{x}_i$$

Here, $\mathbf{W}_{in}$ is a randomly selected matrix of weights with dimensions dr×dx. The matrix A also plays a significant role as a randomly chosen square matrix of weights (dr×dr). It is designed to exhibit particular properties, such as spectral radius for convergence or sparsity, in alignment with the "echo-state" property, which will be elaborated upon in the subsequent sections.

The read-out matrix, on the other hand, is formed through a linear transformation using a dx×dr matrix of weights, $\mathbf{W}_{\text{out}}$. This matrix, $\mathbf{W}_{\text{out}}$, is the only component that is trained to the data, enabling forecasts $\boldsymbol{y}_i$ given data $\boldsymbol{x}_i$. This selective training is a major advantage of Reservoir

Computing as it allows for simplified and cost-effective least squares computation. The function q is an "activation" function. Common choices for q include q(s) = tanh(s). Other activation functions commonly found in general neural network theory include sigmoidal functions and the ReLu function. The parameter α, with a value between 0 and 1, functions to decelerate the Reservoir Computing network and moderate the stability of the fitting.

Rather than meticulously training the matrices $\mathbf{W}_{\text{in}}$ and A to adapt to the data, these matrices are randomly determined. the elements of A are uniformly chosen, $\mathbf{A}_{\text{i,j}} \sim$ U(-β,β), where β functions as a scaling factor for the spectral radius. Furthermore, the read-in matrix is determined through a random selection, with $\mathbf{W}_{\text{in i,j}} \sim$ U(0,γ) and γ > 0 serving to scale the internal variables r.

In the application of Reservoir Computing, a critical step involves the utilization of the update equation for the reservoir states. By continually feeding data into the system and updating the reservoir states, denoted as 'r', we can amass these states in a matrix form, referred to as matrix R.

$$\boldsymbol{R} = [\boldsymbol{r}_{\boldsymbol{i+1}}|\boldsymbol{r}_{\boldsymbol{i+2}}| \dots |\boldsymbol{r}_{\boldsymbol{N}}]$$

The ultimate objective of this process is to align the reservoir states with the true variables, which are embodied in matrix X. This alignment is achieved through the application of a mapping matrix, denoted as V. It's important to note that V is equivalent to $\mathbf{W}_{\text{out}}$ but it's representative of the system before the fitting process has taken place.

$$\boldsymbol{X} = [\boldsymbol{x}_{\boldsymbol{i+1}}|\boldsymbol{x}_{\boldsymbol{i+2}}| \dots |\boldsymbol{x}_{\boldsymbol{N}}] = [\boldsymbol{V}\boldsymbol{r}_{\boldsymbol{i+1}}|\boldsymbol{V}\boldsymbol{r}_{\boldsymbol{i+2}}| \dots |\boldsymbol{V}\boldsymbol{r}_{\boldsymbol{N}}] = \boldsymbol{V}\boldsymbol{R}$$

Given this configuration, the task at hand essentially becomes a minimization problem. The aim is to minimize the discrepancy between the reservoir states and the true variables. To accomplish this, the least squares problem is solved, providing an optimal solution that minimizes the sum of the squares of the differences between the observed and predicted values.

$$\mathbf{W}_{\text{out}} = \underset{\mathbf{V}\in\mathbb{R}^{d_x \times d_r}}{\arg min} \parallel \mathbf{X} - \mathbf{VR} \parallel_F = \underset{\mathbf{V}\in\mathbb{R}^{d_x \times d_r}}{\arg min} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{V}\mathbf{r}_i\|_2$$

In practice, however, this minimization problem is often approached using Tikhonov regularization, which is also commonly referred to as ridge regression. This method provides a closed-form solution that is designed to ensure the stability of the solution and prevent overfitting, a common pitfall in machine learning where the model becomes too tailored to the training data and performs poorly on unseen data. The stability and generalizability of the solution are maintained by incorporating a regularization coefficient, denoted as \lambda. This coefficient effectively controls the balance between the complexity of the model and its ability to fit the training data, helping to create a model that is well-suited for predicting future data. The close form solution for $\mathbf{W}_{\text{out}}$ is as follows,

$$\mathbf{W_{out}} = \mathbf{XR^T}(\mathbf{RR^T} + \lambda\mathbf{I})^{-\mathbf{1}}$$

Once the training phase has been completed, the Reservoir Computing system is primed to commence making predictions. In this phase, the computed output from the training stage is redirected and fed back into the system as the original data. This feedback mechanism effectively represents the input, xi, by the computed output of the preceding step, where $\boldsymbol{x}_i = \boldsymbol{y}_i = \mathbf{W}_{out}\,\boldsymbol{r_i}$

Upon the completion of this step, the reservoir system transitions into an autonomous operational mode. This is a significant shift as the system is no longer driven by the input data, but rather operates independently to generate output predictions according to the following equation

$$\boldsymbol{r}_{i+1} = (1-\alpha)\boldsymbol{r}_i + \alpha q(\mathbf{A}\boldsymbol{r}_i + \mathbf{W}_{in}\,\mathbf{W}_{out}\,\boldsymbol{r_i})$$

And the output $\boldsymbol{y}_{i+1} = \mathbf{W}_{\text{out}}\boldsymbol{r}_{i+1}$ is the predicted value.

Depending on the needs of the application, these predictions can be extended indefinitely into the future. The final stage of the process involves comparing the original data with the predicted data, providing a tangible measure of the system's performance. This cycle of training, feedback, autonomous prediction, and comparison forms the cornerstone of the Echo State Network (ESN) methodology.

A key feature of ESN, and indeed its main advantage over traditional Recurrent Neural Networks (RNNs), is its ability to streamline the training process. Unlike traditional RNNs, which require the training of all weights, ESN only necessitates the training of the read-out matrix, denoted as $\mathbf{W}_{\text{out}}$. The read-in matrix ($\mathbf{W}_{\text{in}}$) and the inner layer recurrence matrix (A) are randomly initialized at the outset. Their properties, however, are not arbitrary, but are judiciously chosen to ensure the manifestation of a critical feature known as the "echo state" property. The echo state property is a crucial concept in understanding the functionality of ESNs, which we will delve into in the following sections.

2.1 Echo State Network

A deeper exploration of Echo State Networks (ESNs) reveals that their functionality and performance are intricately tied to a handful of key parameters. These parameters, when precisely adjusted, allow the ESN to effectively process complex temporal data and make meaningful predictions.

At the heart of ESNs lies the concept of the Echo State Property (ESP). This property ensures stability within the reservoir and allows it to retain a "memory" of past inputs. This ability to remember and build upon past inputs is what allows the ESN to generate meaningful and contextually accurate predictions.

The reservoir size is another critical parameter that requires careful calibration. The size of the reservoir plays a significant role in influencing the performance of the ESN. A larger reservoir can improve the network's ability to approximate the temporal data. However, it also adds to the complexity of the model, making it more challenging to manage and process. Therefore, striking a balance between performance and complexity is crucial when determining the appropriate reservoir size.

Sparsity within the reservoir is another consideration. The Erdős–Rényi model is often used as a benchmark for the distribution of non-zero elements within the reservoir. A sparse reservoir, where many of the connections are zero, can often improve the performance of the ESN.

The spectral radius (ρ) is another parameter that has a significant influence on the echo state property. This parameter needs to be fine-tuned according to the specific requirements of the problem at hand, usually starting from unity and adjusting as needed.

The leaking rate (α) represents the update rate of the nodes within the reservoir. This parameter can be interpreted as the time between successive time steps, or as a measure of the sampling rate. The leaking rate needs to be carefully adjusted to ensure the reservoir updates at an appropriate pace.

Importantly, ESNs are known to be highly sensitive to changes in these parameters. These parameters need to be adjusted in conjunction to achieve optimal performance in predicting chaotic systems. Current standard practice involves manual experimentation and practical tricks collected over the years. A valuable resource in this regard is Lukoševičius (2012), which compiles successful observations and methods from practical implementations, offering a useful guide for parameter tuning in ESNs.

2.2. Chaotic Systems – Applications

As we delve deeper into the application of Echo State Networks (ESNs), it becomes evident that their unique capabilities are particularly suited to tackling challenges posed by chaotic systems. As we've discussed in class, chaos theory pertains to systems that exhibit complex, unpredictable, and highly sensitive behavior. Their inherent nonlinearity and high sensitivity to initial conditions render the task of predicting their behavior a formidable one. Yet, ESNs stand out as a promising tool in this endeavor. To substantiate this claim, I'll illustrate the application of ESNs on two distinct chaotic systems: the Lorenz Attractor and the Rossler Attractor.

1- Lorenz Attractor

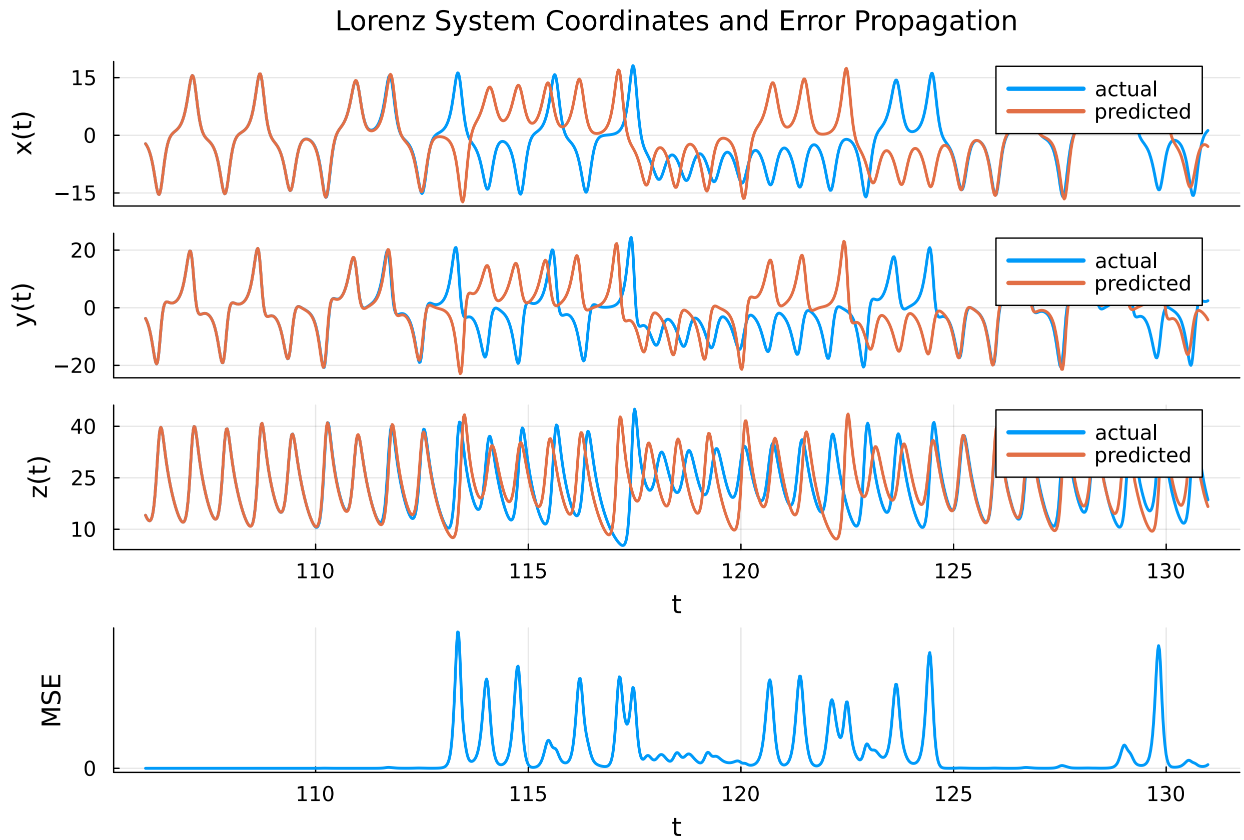
Starting with the Lorenz system, a paradigmatic example of a chaotic system. It comprises three coupled, nonlinear differential equations that encapsulate the chaotic behavior of certain physical systems, such as atmospheric convection. The challenge is to predict the future states of the Lorenz system based on a sequential arrangement of states.

$$\frac{dx}{dt} = a(y - x)$$
$$\frac{dy}{dt} = x(b - z) - y$$
$$\frac{dz}{dt} = xy - cz$$

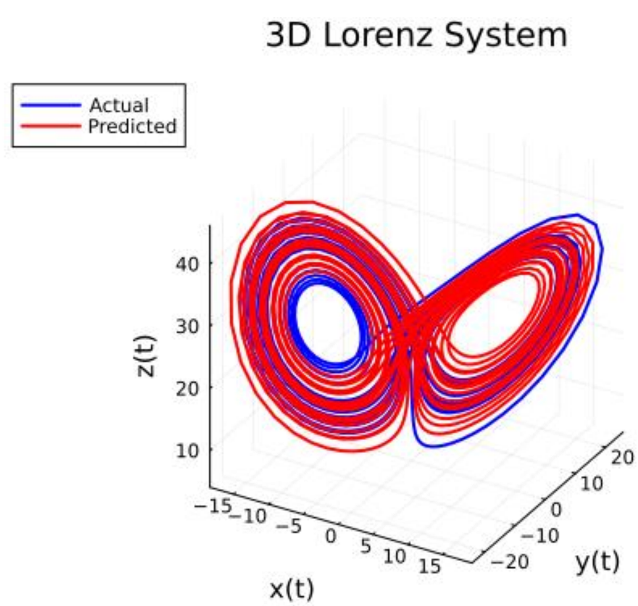
Upon applying the ESN with the parameters shown in the table, and observing the corresponding Lorenz prediction graph, we discern that the ESN is proficient in generating highly accurate short-term predictions. As we venture further into long-term predictions, though, certain errors start to surface. It's crucial to emphasize that these errors are within an acceptable margin as the overall dynamical behavior of the system remains intact. This is more evident in the three-dimensional plot; the predicted trajectory aligns with the actual one for a considerable span before starting to diverge.

ESN Parameters for Lorenz Table

Parameter	Value
Spectral Radius (ρ)	**1.2**
Number of reservoir states (N)	**300**
Leak rate (α)	**1**
Sparsity	**6/300**
Number of training points	**5000**
Number of prediction points	**1250**



This intriguing phenomenon is referred to as "reproducing the climate" of the chaotic system, underscoring the ESN's ability to grasp and reflect the system's core characteristics and dynamics. Even when the precise trajectory of the predictions deviates from the true values over time, the 'climate', or the general behavior, is maintained. This showcases the prowess of ESNs in dealing with intricate, chaotic systems like the Lorenz Attractor.

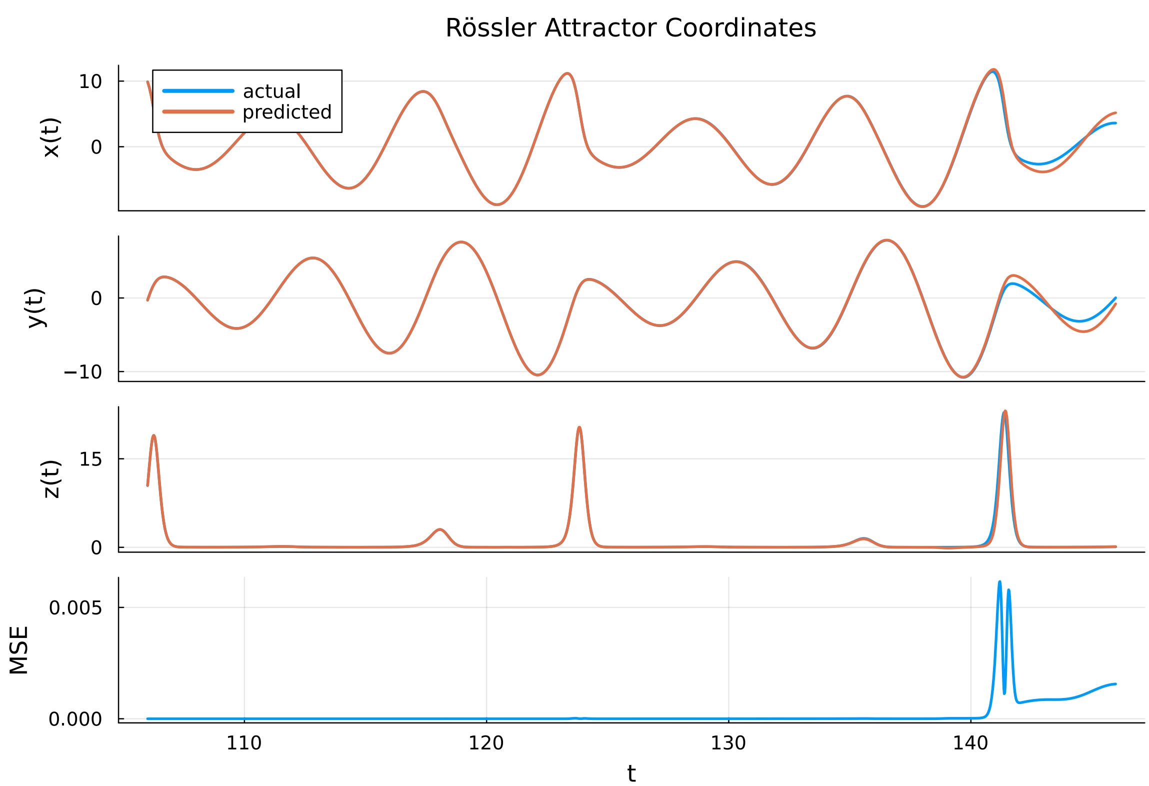


2- Rossler Attractor

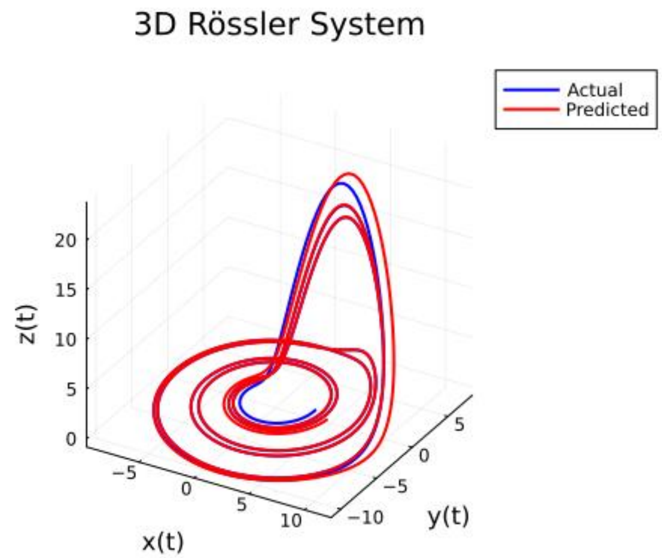
Shifting to the Rossler Attractor, another example of a chaotic system, we observe a similar pattern. Here too, the ESN's predictions align almost perfectly with the actual values for an initial period, before deviating in the long term, an expected characteristic of chaotic systems. The three-dimensional plots for the Rossler system further emphasize this point, revealing a congruence in the invariant measure between the predicted and actual system states.

ESN Parameters for Rossler Table

Parameter	Value
Spectral Radius (ρ)	**1.07**
Number of reservoir states (N)	**100**
Leak rate (α)	**1**
Sparsity	**6/300**
Number of training points	**5000**
Number of prediction points	**2000**



And in 3D, the plot is:



3. Conclusion:

In conclusion, Reservoir Computing (RC) and specifically Echo State Networks (ESNs) offer several advantages in predictive modeling and chaotic system analysis. ESNs provide faster training and reduced data requirements compared to traditional neural networks. They have been successfully applied in predicting chaotic systems such as the Lorenz attractor and Rossler system. The control parameters within ESNs play a crucial role in achieving optimal performance. Overall, RC and ESNs hold promise for advancing our understanding and prediction of complex dynamics.

References

Bollt, E. (2021) ‘Erratum: “on explaining the surprising success of Reservoir Computing Forecaster of Chaos? the universal machine learning dynamical system with contrasts to var and dmd” [chaos 31(1), 013108 (2021)]’, Chaos: An Interdisciplinary Journal of Nonlinear Science, 31(4), p. 049904. doi:10.1063/5.0050702.

Jaideep Pathak, Zhixin Lu, Brian R. Hunt, Michelle Girvan, Edward Ott; Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. Chaos 1 December 2017; 27 (12): 121102

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks. GMD Report 148, German National Research Center for Information Technology.

Lukoševičius, Mantas. (2012) “A practical guide to applying echo state networks.” Neural networks: Tricks of the trade. Springer, Berlin, Heidelberg.