



Echo State Networks for the Prediction of Chaotic Systems

Daniel Estévez-Moya^{1,2}, Ernesto Estévez-Rams¹, and Hölger Kantz²

¹ University of Havana, Havana, Cuba

destevez@mpg.pks.de, estevez@fisica.uh.cu

² Max Planck Institute for the Physics of Complex Systems, Nöthnitzer Str. 38,
01187 Dresden, Germany

Abstract. Chaotic systems are complex and challenging to predict due to their sensitive dependence on initial conditions and the complexity of their dynamics. This paper improves the prediction accuracy of chaotic systems using Echo State Networks (ESNs) by considering a complex graph adjacency matrix as the recurrent layer. This helps ESNs better capture the nonlinear dynamics of chaotic systems. ESNs are a type of recurrent neural network with a fixed, randomly generated hidden layer, making them computationally efficient and able to handle large, high-dimensional datasets. This study applies ESNs to the Lorenz, Mackey-Glass, and Kuramoto-Sivashinsky models, which are well-studied examples of chaotic systems. Our results demonstrate that ESNs can effectively capture the dynamics of these systems and outperform existing prediction methods made with ESNs.

Keywords: Chaotic systems · Reservoir Computing · Echo State Networks · ESN · Recurrent neural networks · Nonlinear dynamics · Prediction · Lorenz model · Mackey-Glass model · Kuramoto-Sivashinsky model · Graph adjacency matrix · High-dimensional datasets

1 Introduction

Chaotic systems are ubiquitous in various fields, such as physics, biology, and economics. Understanding and predicting the behaviour of these systems is crucial for many applications, including weather forecasting, speech recognition and stock market prediction [4, 8, 17]. However, chaotic systems are notoriously difficult to predict due to their sensitive dependence on initial conditions and the complexity of their dynamics.

Recent advances in machine learning have shown promise in predicting chaotic systems, with Echo State Networks (ESNs) being one of the most successful approaches. ESNs are a type of recurrent neural network with a fixed, randomly generated internal state, making them computationally efficient and

Supported by Max Planck Institute for the Physics of Complex Systems.

able to handle large, high-dimensional datasets. This study applies ESNs to the Lorenz, Mackey-Glass, and Kuramoto-Sivashinsky models, which are well-studied examples of chaotic systems [7, 10, 13].

In this paper, we present an improvement in the prediction accuracy of chaotic attractors on these models using ESNs, considering its internal hidden layer as the adjacency matrix of a Watts-Strogatz graph. Our results demonstrate that with this consideration ESNs can effectively capture the dynamics of these systems and outperform existing reported prediction accuracy [14, 15].

2 Echo State Networks

Reservoir computing is a neurocomputing paradigm that employs a reservoir of dynamical nodes to compute temporal features internally, requiring only a linear readout layer to map the reservoir state to the desired output. There are two main paradigms: liquid state machines (LSM) [12] and echo state networks (ESN) [6].

Echo State Networks (ESNs) are recurrent neural networks successfully applied to various applications, including time-series prediction, speech recognition, and control problems [16]. Herbert Jaeger introduced them in 2001 to simplify the training of recurrent neural networks [6].

Mathematically, an ESN can be defined as follows: Let $\mathbf{h}(t)$ denote the state vector of the network at time t , $\mathbf{x}(t)$ be the input vector, and $\mathbf{y}(t)$ be the output vector. The dynamics of the ESN can be written as follows:

$$\mathbf{h}(t+1) = f(\mathbf{W}_{in}\mathbf{x}(t) + \mathbf{W}_r\mathbf{h}(t) + \mathbf{W}_{fback}\mathbf{y}(t)) \quad (1)$$

where \mathbf{W}_{in} is the input weight matrix, \mathbf{W}_r is the randomly initialized recurrent weight matrix, \mathbf{W}_{fback} is a feedback weight matrix from the outputs $\mathbf{y}(t)$ to the internal state $\mathbf{h}(t)$, and $f(\cdot)$ is a non-linear activation function, typically tanh or some other sigmoid function. The output of the ESN is then given by:

$$\mathbf{y}(t+1) = \mathbf{W}_{out}\mathbf{h}(t+1) \quad (2)$$

where \mathbf{W}_{out} is the output weight matrix learned during training.

The critical feature of ESNs is the echo state property (ESP) [5], which allows the network to store and process information in its recurrent connections without complex training algorithms. The ESP refers to the ability of a recurrent neural network with a dynamic recurrent layer to process input information effectively and stably. ESP is achieved by randomly initializing the recurrent connections of the network and fixing them during training. The recurrent connections then act as a “reservoir” of information that can be used to process input data. Jaeger argues that a neural network complies with ESP if its recurrent layer can gradually forget irrelevant information and retain only the crucial features from the input signal.

To ensure the echo state property in an ESN, it is important for the spectral radius of the recurrent weight matrix \mathbf{W}_r to be less than 1. The spectral radius

can be controlled by scaling the weights of the recurrent matrix. This guarantees that the dynamics of the network converge over time and that the network does not amplify small perturbations in the input. Although this condition is very restrictive, other criteria that may also be applied to ensure the achievement of the ESP [19] have been developed.

Owing to the dynamics of a reservoir exhibiting the Echo State Property, the network can operate as a filter solely for the input signal, irrespective of the internal state's original conditions. Accordingly, the denomination "Echo State Network" alludes to the reservoir's echoing of the input history.

Another way to achieve the echo state property is to use a sparsely connected recurrent weight matrix, where only a tiny fraction of the connections are non-zero. It is then natural to consider this sparse matrix to be the adjacency matrix of a complex graph, as this allows the network to exploit the complex information propagation dynamics of the graph in its recurrent activity. The connectivity pattern encoded in the adjacency matrix captures the higher-order structure of the complex graph, which can influence the emergent computational properties of the reservoir. For this reason we then used the adjacency matrix of a Watts-Strogatz model as the recurrent layer of the ESN. This choice of Watts-Strogatz topology for the reservoir is motivated by the capacity of its small-world networks to facilitate complex information flow dynamics that we seek to leverage for computational purposes within the ESN framework.

Another advantage of ESNs is that they can be trained using simple linear regression techniques, which makes them computationally efficient and easy to implement. During training, the output weight matrix is learned by minimizing the mean squared error between the network and target output using ridge regression [3]. The input and feedback weight matrices can also be optimized during training to improve further the network's performance [11].

2.1 Training an ESN

The training process for an ESN has two main stages: internal state harvesting stage and regression stage.

After initializing the ESN with random weights and achieving ESP, input data is fed into the network during the *harvesting stage*. The network processes this input data, generating an internal state $h(t)$ for each time step t . These sequence of internal states at each time step are then collected and used to train the output layer.

During the output *regression stage*, the collected internal states are used to train the output matrix W_{out} . This is done by solving a linear regression problem of the form $W_{out}h(t) = y(t+1)$ for W_{out} , where W_{out} represents the weights that map the internal states to the desired output, and $y(t)$ is the target output at time step t , in our case the next time step of the time series, i. e. $x(t+1)$. The weights W_{out} are typically learned using a regularized least-squares regression method such as Ridge regression. Ridge regression is a version of linear regression that adds a penalty term to the least squares objective function to

prevent overfitting. The objective function for this regression is given by:

$$\mathbf{y}_{target} - \mathbf{W}_{out}\mathbf{h} + \beta|\mathbf{W}_{out}|^2 \quad (3)$$

where \mathbf{y}_{target} is the target output, \mathbf{h} is the state vector of the ESN, and β is a regularization parameter that controls the strength of the penalty term. The output weight matrix is learned by minimizing this objective function. Other methods, such as gradient descent or the Moore-Penrose pseudoinverse [1], can also achieve similar results.

One important consideration in training ESNs is the choice of hyperparameters, such as the spectral radius, input scaling factor, the regularization parameter for ridge regression, and the reservoir size (i.e., the number of neurons in the ESN). These hyperparameters can significantly impact the performance of the ESN and are often chosen through trial and error or using more sophisticated methods such as grid search or Bayesian optimization [2, 9, 18].

In this work, we will be using a free-running ESN with no input vector $x(t)$; only the feedback from the network output will be combined with the internal state, i. e.:

$$\mathbf{h}(t+1) = f(\mathbf{W}_r\mathbf{h}(t) + \mathbf{W}_{fback}\mathbf{y}(t)) \quad (4)$$

The training will consist of feeding the network with the actual signal data to make the internal state harvesting. Afterwards, \mathbf{W}_{out} is computed by a Ridge regression and the network loop is closed to make the predictions, Fig. 1.

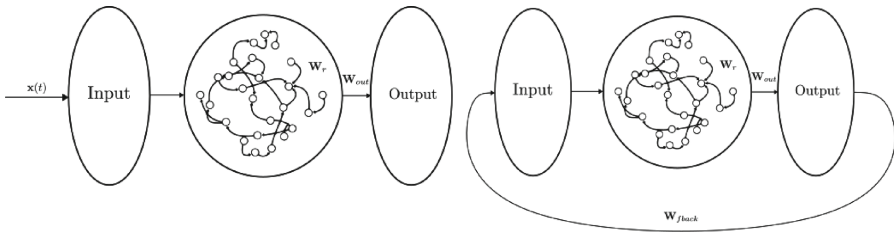


Fig. 1. Diagram of an Echo State Network. Left: the open loop phase with correct signal input. Right: prediction phase with network feedback.

In the training phase (open loop), the network will be fed with the actual data from the time series as if it were the network's output. Then in a prediction phase (closed loop), once the output matrix \mathbf{W}_{out} is obtained, the feedback will be the one provided by the network, Fig. 1.

3 Results

We applied our ESN model to predict the trajectories of the Lorenz, Mackey-Glass, and Kuramoto-Sivashinsky systems. We used an adjacency matrix generated from a Watts Strogatz small-world graph with a rewiring probability of

0.5 and an average degree of 6 for the recurrent weight matrix. Ridge regression was used to train the output weights. The reservoir size and spectral radius of the recurrent matrix for the Lorenz system and the Kuramoto-Sivashinsky, were the same as reported in [14, 15].

3.1 Lorenz System

The Lorenz system describes the dynamics of convective fluid motion [10]. This system is described by the equations:

$$\dot{x} = \sigma(y - x) \quad (5)$$

$$\dot{y} = x(\rho - z) - y \quad (6)$$

$$\dot{z} = xy - \beta z \quad (7)$$

For the simulations, we will use the classic parameter choice of $\sigma = 10$, $\rho = 28$, $\beta = \frac{8}{3}$, where the system exhibits chaotic behaviour; hence it makes it a good benchmark for the network to compare.

Figure 2 shows a plot of our prediction horizon compared to previous work. Our model achieved significantly more prolonged and more accurate predictions of the Lorenz attractor, clearly capturing the folding and stretching dynamics of the system. We could predict the Lorenz trajectory up to 10 time units in the future before diverging from the correct trajectory, compared to 5 time units reported by Ott et al. in [15].

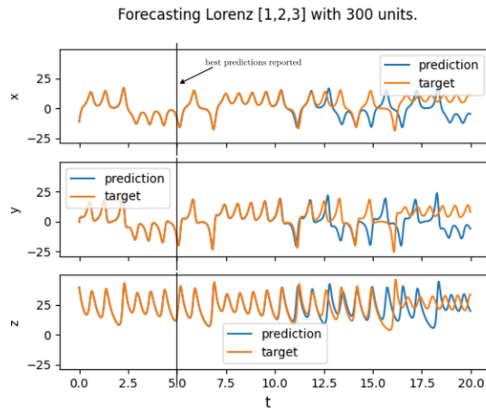


Fig. 2. Prediction horizon of the Lorenz system

Although the predictions eventually diverged from the actual trajectory after some time steps due to the chaotic nature of the system, it is worth noting that the ESN was still able to capture the complex internal dynamics of the system. This is evident from Fig. 3, which shows the Lorenz attractor obtained from the

ESN predictions. The fact that the ESN produced a similar attractor to the actual Lorenz attractor indicates that it captured the essential features of the Lorenz system's dynamics.

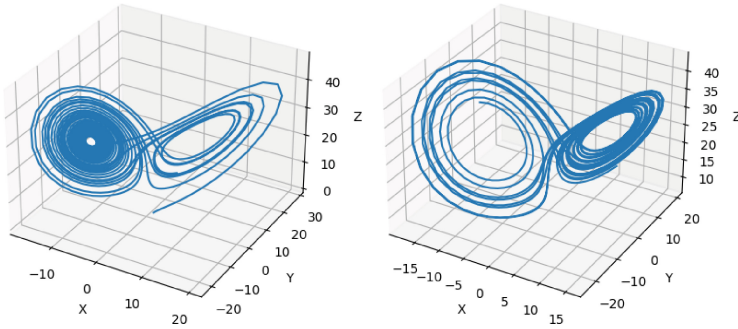


Fig. 3. Lorenz 3d plot. Left: the attractor of the original Lorenz model. Right: the ESN is running free after training. Both exhibit the same nature of internal dynamics.

Another way to confirm the system dynamics are being captured by the ESN is by examining the return plot of the system compared to that of the network, Fig. 4. This shows that the ESN can reproduce the characteristic folding structure and spread of points in the Lorenz attractor, indicating it has successfully learned the system's nonlinear dynamics. Any deviations in the return plot would suggest that the ESN is not fully capturing the dynamics of the chaotic system. Return plots are a useful visualization tool to complement other metrics like prediction error in evaluating the performance of a model on a chaotic system.

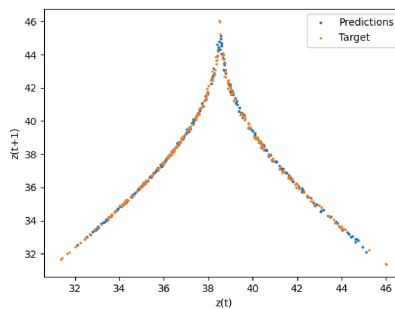


Fig. 4. Lorenz return map for the Z dimension, a comparison between the system and the ESN predictions.

3.2 Mackey-Glass System

The Mackey-Glass equation models physiological control systems with long delay times [13]. Physiological systems like blood cell production control involve feedback loops where the measurement and regulation happen at different time scales. This time delay can introduce instability and chaotic behaviour in the system dynamics. The Mackey-Glass equation captures this behaviour through a time-delayed differential equation. The system equations are:

$$\dot{x} = \frac{\alpha x(t - \tau)}{1 + x(t - \tau)^\beta} - \gamma x(t) \quad (8)$$

Here, $x(t)$ represents the state variable, like blood cell concentration. The first term on the right-hand side represents the production rate, where $x(t - \tau)$ is the delayed state. The second term represents the decay or loss rate. The parameters α , β , τ , and γ determine the behavior of the system. For certain parameter ranges, the delayed feedback through $x(t - \tau)$ can give rise to chaotic oscillations in $x(t)$. This simple equation exhibits phenomena like period doubling and route to chaos, making it useful for studying chaos in dynamic systems. It is typically used to benchmark the performance of recurrent neural networks due to the time-delay nature of its dynamics.

The parameter τ represents the **delay time**, which governs the “memory” of the system. When τ is small, the system behaves linearly. As τ increases, the system becomes increasingly chaotic due to the long delay.

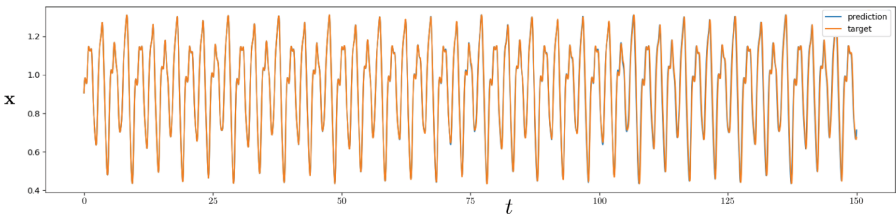


Fig. 5. Prediction horizon of the Mackey-Glass system. Comparison between the numerically calculated simulation and the predictions of the ESN.

For the Mackey-Glass system with $\alpha = 0.2$, $\beta = 10$, $\gamma = 0.1$, $\tau = 17$, where it exhibits chaotic behaviour, our ESN model was able to accurately predict the behaviour of the system entirely, an undoubted improvement compared with what has been done in previous ESN studies. Figure 5 shows the longer prediction horizon achieved by our model.

The performance likely reflects how well our model captured the Mackey-Glass system’s dynamics. The results demonstrate how important is the topology of the recurrent layer in ESN for modelling and predicting chaotic time series with long delay times.

3.3 Kuramoto Sivashinsky System

The Kuramoto-Sivashinsky model describes the propagation of a flame front in a premixed gas. The Laplacian and bi-Laplacian terms model heat diffusion and finite flame thickness, while the nonlinear term models convection due to gas expansion behind the flame front [7]. The system is described by the equation:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^4 u}{\partial x^4} = 0 \quad (9)$$

For this system, our ESN achieved predictions up to 12 Lyapunov times, as shown in Fig. 6, whereas, in previous studies, this was up to 6 Lyapunov times [14]. This indicates that our model can capture the complex spatiotemporal dynamics of the Kuramoto-Sivashinsky system for a more extended period.

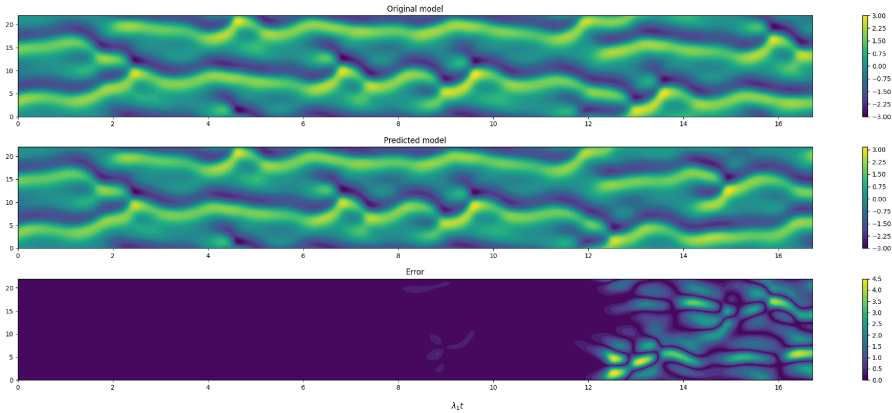


Fig. 6. Prediction horizon of the Kuramoto Sivashinsky system. Time is scaled according to the Lyapunov times of the system. The upper plot is the Kuramoto-Sivashinsky model numerical integration. The middle plot is the predictions made by the ESN. The lower plot exhibits the difference between the other two.

The fact that our model can better learn the system’s underlying patterns and long-term dependencies could be explained by a few possible reasons:

1. Complex dynamical systems like Kuramoto Sivashinsky exhibit complex information propagation, with interactions spanning both short and long timescales. Using a complex graph as the recurrent layer in our echo state network allows us to model this complex information flow better, as complex networks have been shown to capture real-world connectivity patterns more accurately.
2. The additional long-range connections in complex networks can help propagate information over longer timescales. This is crucial for predicting long-term behaviour in dynamic systems, which often depends on integrating information from both recent and distant history.

3. The non-locality and small-world properties of complex networks give them advantages in memory capacity and efficiency compared to an Erdos-Renyi network. Using the same computational resources, they can learn and store information from more inputs.

4 Discussion and Conclusions

Our ESN model achieved significantly more prolonged and accurate predictions for each chaotic system studied: the Lorenz, Mackey-Glass, and Kuramoto-Sivashinsky models.

For the Lorenz system, our model could predict the trajectory of up to 10 time units in the future compared to 5 time units reported in previous studies. The Lorenz attractor produced by the ESN closely resembled the actual Lorenz attractor, indicating that the ESN captured the underlying dynamics well.

Our ESN achieved complete predictions for the Mackey-Glass system. This suggests that our model captured the Mackey-Glass system dynamics more accurately, likely due to using a small-world network as the recurrent layer.

For the Kuramoto-Sivashinsky system, our model achieved predictions up to 12 Lyapunov times compared to 6 Lyapunov times in previous studies. Again, this better performance can be attributed to using a small-world network in the ESN.

The longer prediction horizons achieved by our model can be attributed to the small-world network structure of the recurrent weights. The short path lengths and high clustering of the Watts-Strogatz graph allow information to propagate more efficiently through the reservoir, enabling it to extract and store relevant temporal patterns from the input data for longer. This shows that the topology of the recurrent connections can profoundly impact an ESN's computational abilities, with small-world graphs conferring significant advantages for chaotic systems prediction.

In summary, the improved performance of our model demonstrates the benefits of utilizing complex graph structures as recurrent layers for capturing complex spatiotemporal dynamics. The non-trivial topology of complex networks helps model the non-trivial information propagation in real-world dynamical systems. This highlights the importance of network structure in designing recurrent layers that can better learn long-term dependencies and predict long-horizon behaviours.

Acknowledgements. One of the author EER would like to acknowledge support from the Alexander von Humboldt Stiftung. This research was partially financed by the Cuban Ministry of Science under the Basic Science Program. Karel García is gratefully acknowledge for rewarding discussions.

References

1. Dresden, A.: The fourteenth western meeting of the American mathematical society. *Bull. Am. Math. Soc.* **26**(9), 385–396 (1920)

2. Feurer, M., Hutter, F.: Hyperparameter optimization. *Autom. Mach. Learn. Methods Syst. Challenges* 3–33 (2019). ISBN 978-3-030-05318-5
3. Hoerl, A.E., Kennard, R.W.: Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* **12**(1), 55–67 (1970)
4. Hsieh, D.A.: Chaos and nonlinear dynamics: application to financial markets. *J. Financ.* **46**(5), 1839–1877 (1991)
5. Jaeger, H.: The echo state: approach to analysing and training recurrent neural networks. GMD-Report, GMD - Forschungszentrum Informationstechnik (2001)
6. Jaeger, H., Haas, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* **304**(5667), 78–80 (2004)
7. Kuramoto, Y., Tsuzuki, T.: Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. *Progress Theor. Phys.* **55**(2), 356–369 (1976)
8. Langi, A., Kinsner, W.: Consonant characterization using correlation fractal dimension for speech recognition. In: *IEEE WESCANEX 95. Communications, Power, and Computing. Conference Proceedings*, vol. 1, pp. 208–213. IEEE (1995)
9. Li, D., Han, M., Wang, J.: Chaotic time series prediction based on a novel robust echo state network. *IEEE Trans. Neural Netw. Learn. Syst.* **23**(5), 787–799 (2012)
10. Lorenz, E.N.: Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**(2), 130–141 (1963)
11. Lukosevicius, M.: Echo state networks with trained feedbacks (2007)
12. Maass, W., Markram, H.: On the computational power of circuits of spiking neurons. *J. Comput. Syst. Sci.* **69**(4), 593–616 (2004)
13. Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control systems. *Science* **197**(4300), 287–289 (1977)
14. Pathak, J., Hunt, B., Girvan, M., Lu, Z., Ott, E.: Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. *Phys. Rev. Lett.* **120**, 3 (2018)
15. Pathak, J., Lu, Z., Hunt, B.R., Girvan, M., Ott, E.: Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data, October 2017
16. Salmen, M., Ploger, P.G.: Echo state networks used for motor control. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1953–1958. IEEE (2005)
17. Vaidyanathan, S., Volos, C.: *Advances and Applications in Chaotic Systems*, vol. 636. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-30279-9>
18. Yang, L., Shami, A.: On hyperparameter optimization of machine learning algorithms: theory and practice. *Neurocomputing* **415**, 295–316 (2020)
19. Yildiz, I.B., Jaeger, H., Kiebel, S.J.: Re-visiting the echo state property. *Neural Netw. Off. J. Int. Neural Netw. Soc.* **35**, 1–9 (2012)