IISER PUNE

# Semester Project Report

Aug-Nov 2021

**Echo State Networks**

*Author:*
Reetish Padhi

*Submitted to:*
Prof. Joy Merwin Monteiro

December 5, 2021

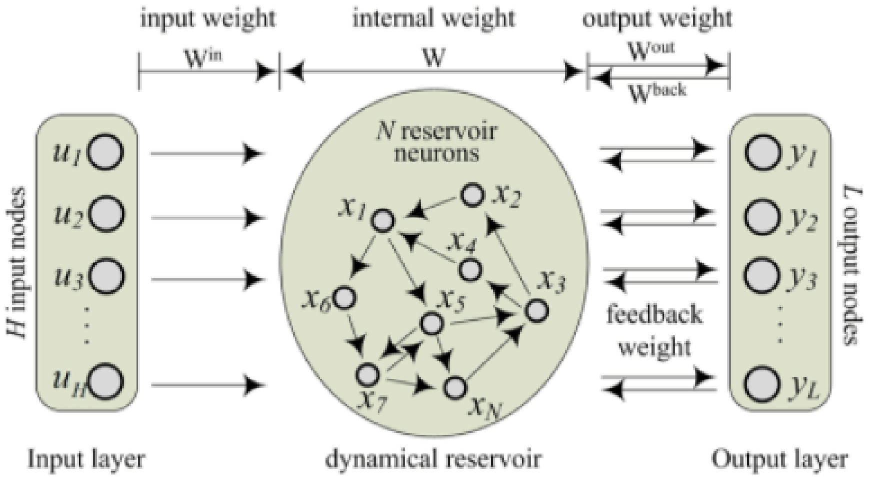
# Table of Contents

# 1 Neural Networks

A neural network is a series of algorithms that aims to replicate underlying relationships in a set of data. A neural network contains layers of interconnected nodes. Each node receives a weighted signal from the previous layer which is then passed through the activation function to yield an output. The weights in the network are then trained to predict the desired output. Their name and structure are inspired by the biological neurons in the brain. The main advantage of Neural Networks is that once the network is trained, the prediction takes very little time. This is particularly useful in climate modelling because it allows one to construct data driven surrogate models. Such data driven models can be used to accelerate and/or improve the predictions and simulations of complex dynamical systems and even provide a means to model processes which are not very well understood [1].

# 2 Echo State Network

Echo state network is a type of Recurrent Neural Network, part of the reservoir computing framework. The main idea is to create a large, random, fixed RNN and to project an input impulse on the network so as to induce a non linear signal response in the neurons. The network is then trained to take the right linear combination of signals to obtain the desired output signal. Herein lies the biggest advantage of using ESNs. It is computationally faster than BPTT algorithms used to train RNNs and doesn't suffer from vanishing/exploding gradients since the training procedure is merely a linear regression task [1] [2].



### 2.1 Echo State Property

A network is said to have the echo state property if the effect of initial conditions vanish with time (The effect due to the random choice of initial reservoir state dies out after enough iterations). This condition is met when the spectral radius (maximum eigenvalue of the Adjacency Matrix) is less than 1. In such cases with more iterations the network tends to converge to a state rather than diverge. The properties of reservoir states being state contracting, state forgetting, and input forgetting [2] are all equivalent to the network having echo states. This ensures that errors don't accumulate and blow up with time.

### 2.2 Matrix Equations

We consider a reservoir with N neurons where x(t) refers to the state of the reservoir at time t. $W^{in}$ refers to the input matrix whose purpose is to project the input u(t) onto the reservoir. $W$ refers to the adjacency matrix of the reservoir. y(t) is the output vector and $W^{out}$ is trained to map the reservoir state x(t) to the desired y(t).

$x(n+1) = f(W^{in}u(n+1) + Wx(n) + W^{back}y(n))$

The reservoir is updated based on the above equation. The activation function can be a variety of different functions like tanh, sigmoid, relu.

$y(n+1) = f(W^{out}(u(n+1), x(n+1), y(n)))$

The above equation denotes that the output vector can depend on u(t) , x(t), y(t-1).

## 3 Important Parameters in training

In order to get a better understanding of the parameters and their effect on the predictions, we take a simple periodic function. The Echoes package [3] has been used to set up the ESN and generate the predictions

$$sinx + sin2x$$

## 3.1 Size of the reservoir

The size of the reservoir is nothing but the number of neurons in the reservoir. Larger Reservoir allows for more diversity. Naturally smaller reservoirs aren't able to capture the complete essence of the data. However VERY large reservoirs can cause stability issues because a lot of the nodes remain 'unused'. Moreover the reservoir size places an upper bound on the maximum 'memory' of the network [4]. In this case, the 500 neuron reservoir does a reasonable job initially but is unstable, the 1000 neuron reservoir does a reasonably good job with stable solutions. A higher reservoir size however doesn't always ensure that the predictions are better as is evident from figure 1.

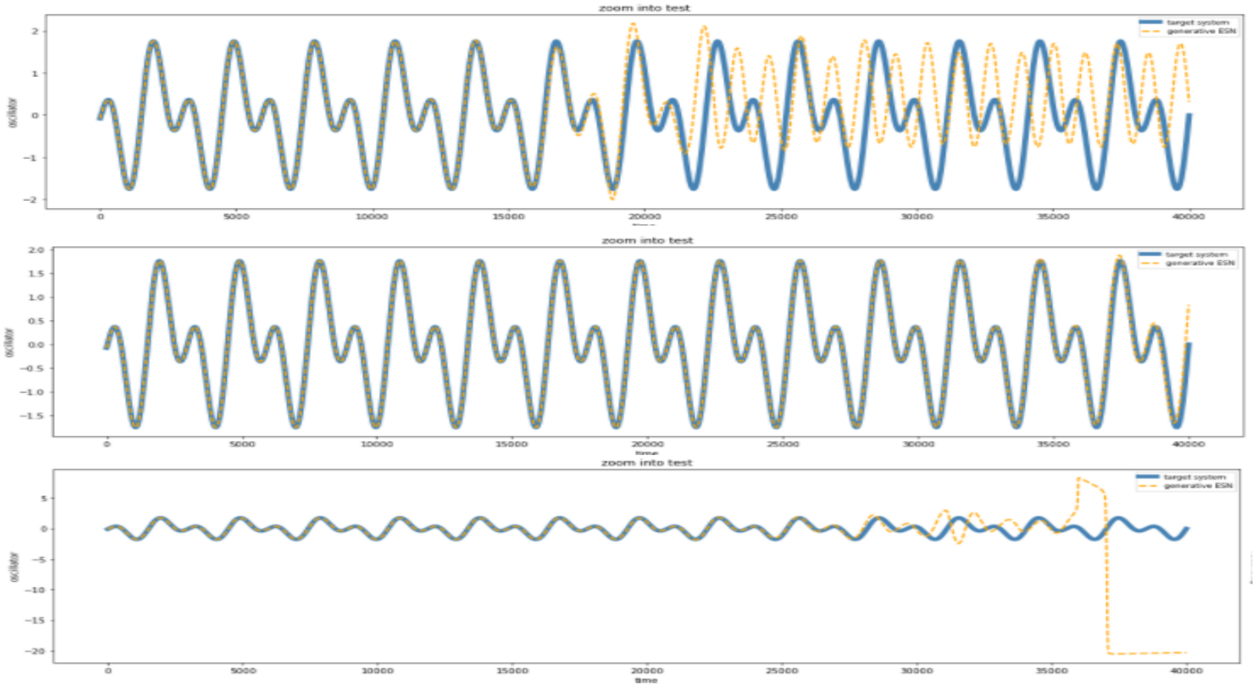


Figure 1: Effect of Reservoir Size: N=2500(top), N=1000(middle), N=500(bottom)

## 3.2 Spectral Radius

Measure of the decay rate of an impulse. This parameter is also important in determining whether a network has the echo state property or not. It is defined as the largest Eigenvalue of the reservoir matrix $W$. For, example SR = 0.999 would 'remember' more than SR = 0.1 and hence be more suitable for slower dynamics.Typically a higher SR means that the stability of the network is a little harder to achieve because impulses don't die out as easily. One can use a larger reservoir with a higher SR.

### 3.3 Sparsity

Proportion of the reservoir matrix ($W$) weights forced to be zero. A sparse network has multiple loosely coupled systems and hence this allows the inputs to 'echo' or bounce around in the reservoir. Higher sparsity also has the advantage that errors aren't able to propagate very quickly because the number of connections are less. Figure 2 compares the predictions using a low sparsity(0.5) and a high sparsity(0.95).

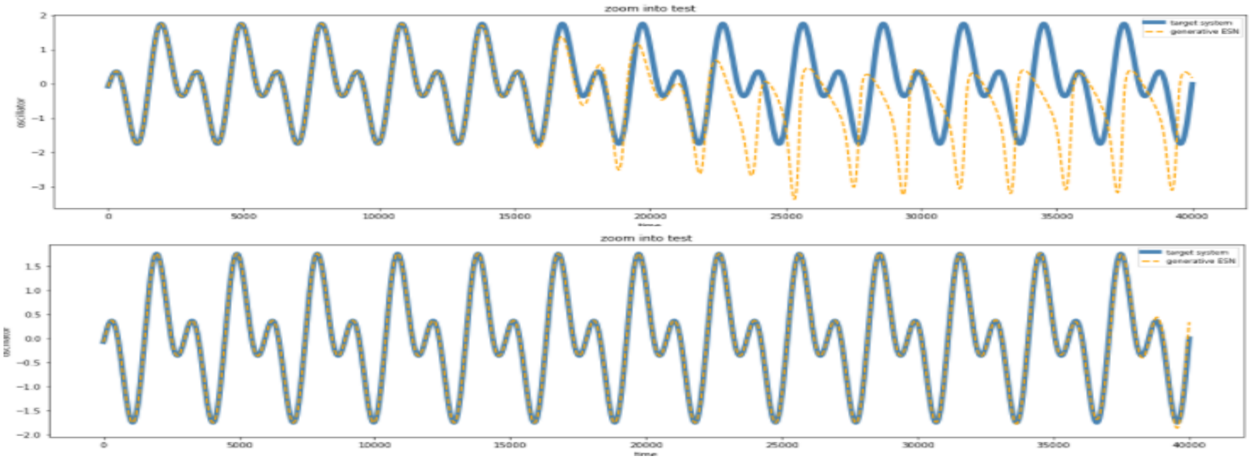


Figure 2: Effect of Sparsity: Sparsity=0.5(top), Sparsity=0.95(bottom)

### 3.4 Effect of Noise

Noise plays an important role in maintaining the stability of solutions. When learning to predict periodic paths, the network only needs to 'learn' z number of arguments where z is the time period of the periodic oscillations. This means that there are multiple possible solutions to the problem. However all such solutions might not be stable. Here is where the advantage of having a large reservoir comes into play. Essentially the trick is to insert some noise into the data so the weights can be trained to yield only the stable solutions. The magnitude of the noise is also important - noise larger than the input impulse will give wrong solutions. Figure 3 compares the stability of solutions with and without noise.

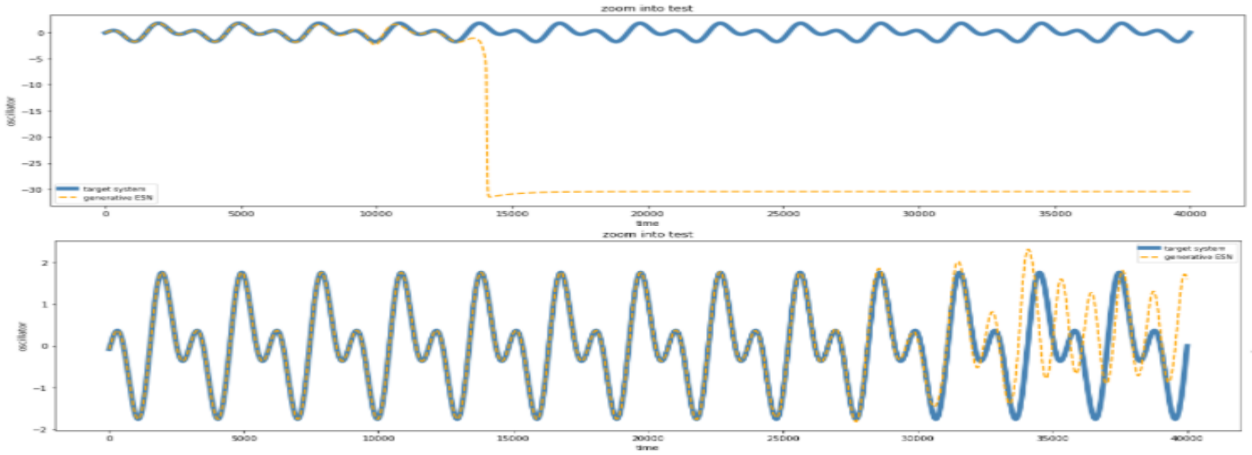


Figure 3: Effect of adding Noise to data: Without Noise(top), With Noise(bottom)

## 3.5 Leak Rate

Measure of the dependence of older states on current state. Higher leak rate means x(t+1) has a higher dependence on x(t).

## 3.6 Activation Functions - tanh and ReLU

$ReLU(x) = max(0, x)$
$tanh(x) = \frac{e^x - e^{-x}}{(e^x + e^{-x})}$

The choice of the activation function really depends on value of Spectral radius being used. for example tanh happens to be very unstable with larger values of SR. Thus one would require a large reservoir size to use tanh with a high spectral radius. However the ReLU happens to work really well with large SR even with smaller reservoirs. ReLU happens to yield much more stable solutions than tanh. ReLU's effectiveness depend on the leak-rate and a very low leak rate can lead to the states diverging in some cases.

In general ReLU seems to give better results (higher test scores and stable solutions) than tanh with smaller reservoir sizes. For example in figure 4 we see the comparison of tanh (SR=0.9, n=2000) with ReLU (SR=0.9, n=1000). Here a smaller reservoir with the ReLU activation function is able to outperform the larger reservoir with tanh as its activation function.

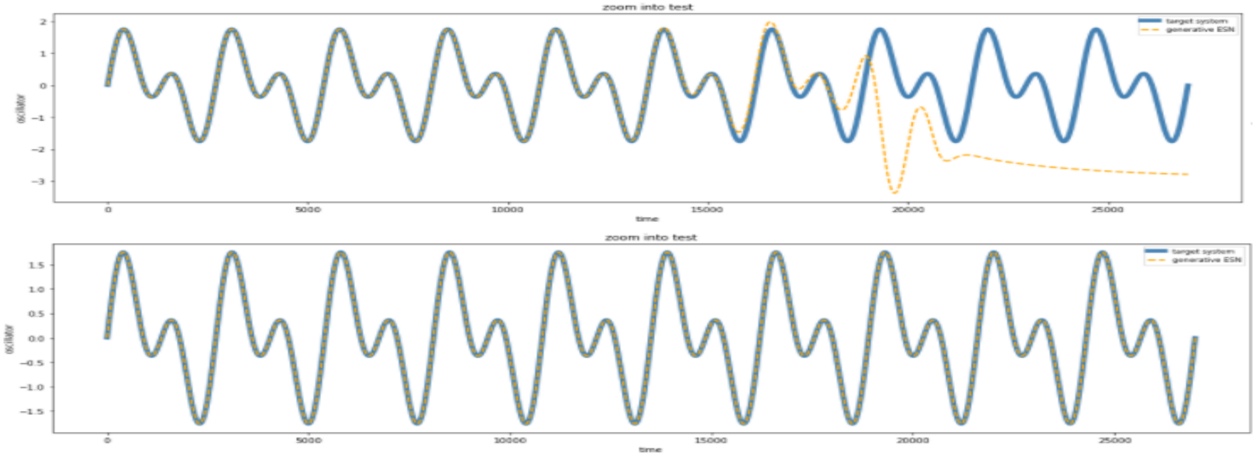


Figure 4: Comparison of tanh(top) and ReLU(bottom) with SR=0.9

# 4 Lorenz 63 (Chaotic system of equations)

$\frac{dx}{dt} = \sigma(y - x)$
$\frac{dy}{dt} = x(r - z) - y$
$\frac{dz}{dt} = xy - bz$

In 1963, Edward Lorenz developed a simplified three-variable model to investigate atmospheric convection. Here, $x$ is proportional to the rate of convection, $y$ is related to the horizontal temperature variation, and $z$ is the vertical temperature variation. There are three constant parameters - $\sigma = 10, r = 28, b = 8/3$. $\sigma, r, b$ relates to the Prandtl number, Rayleigh number, physical dimensions of the layer. In particular, Lorenz's model made it clear for the first time how an infinitesimally small change in the initial conditions of a system could end up having a dramatic effect on the subsequent behavior of the system [5].

## 4.1 ESN Prediction

The training data (950000 data points, 5000 data points for testing) is generated using the odeint function (Runge Kutta 4th order method) and is normalised by subtracting the mean and dividing by the standard deviation. The network consists of 3000 neurons with spectral radius of 0.6, sparsity of 0.9, leak rate of 0.3 with ReLU as the activation function. In Figure 5 the ESN is able to predict between 800-1700 time steps ($\Delta t = 0.01$) with reasonable accuracy however diverges beyond that.

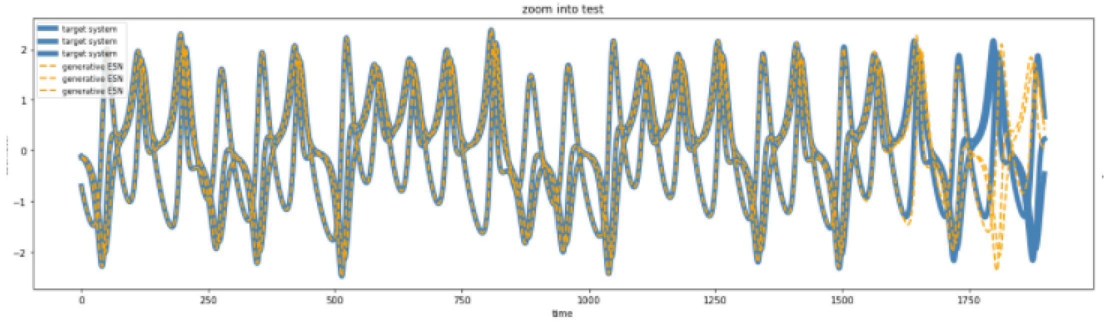


Figure 5: Predictions of ESN in 3d for the Lorenz 63

Figure 6 below shows the prediction of the ESN with different initial point and figure 7 denotes the evolution of the error on a logscale. The predictions seem to start diverging around t=500 and then again around t=775 and it is around the same time as when the error plot crosses 1 (on the log scale). A similar thing was observed in the error plots for the sin x + sin 2x case - the predictions weren't accurate after the error crossed 1 on the log scale.

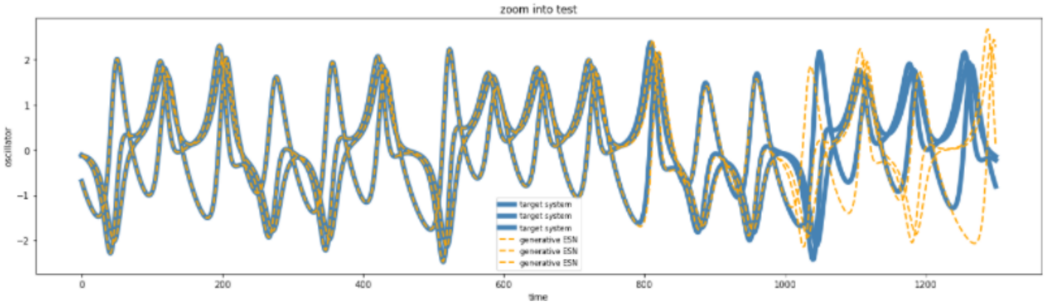


Figure 6: Predictions of ESN

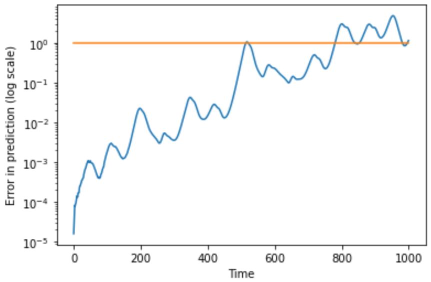


Figure 7: Difference between predicted and actual values(log scale); Yellow line denotes y=1

# 5 Reservoir State Activity

## 5.1 Periodic Signal

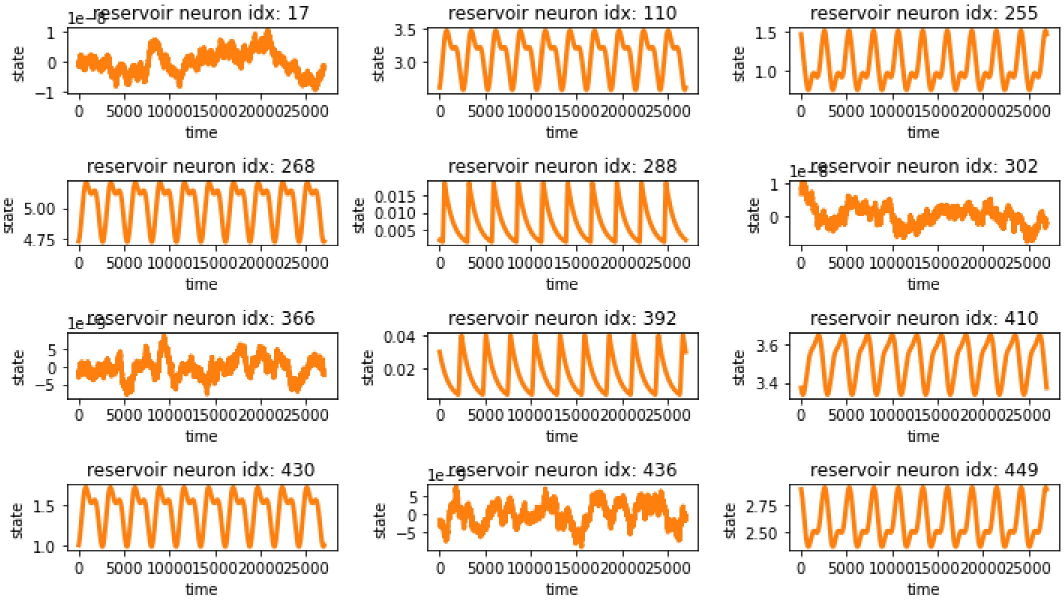


Figure 8: Reservoir State with sin x + sin 2x signal

Figure 8 shows the activity of 12 randomly chosen neurons (activation function - ReLU). Most of the neurons seem to have a periodic activation which resembles the desired the signal.One thing to note is that some of the neurons (17, 30, 366, 436) don't resemble the desired signal and have much lower activity than the other neurons. This might due to the high sparsity of the network. The signals also tend to resemble a Brownian motion so it might suggest that such neurons might 'model' the noise added to the data.

### 5.2 Chaotic Signal

Figure 9 shows the activity of 18 randomly chosen neurons from the reservoir when trained on data from the Lorenz 63. Again one can notice a lot of similarities in the activity of most of the neurons. The zero activity of Neuron 1986 seems a little bizarre though. It might be a consequence of the sparsity of the network.

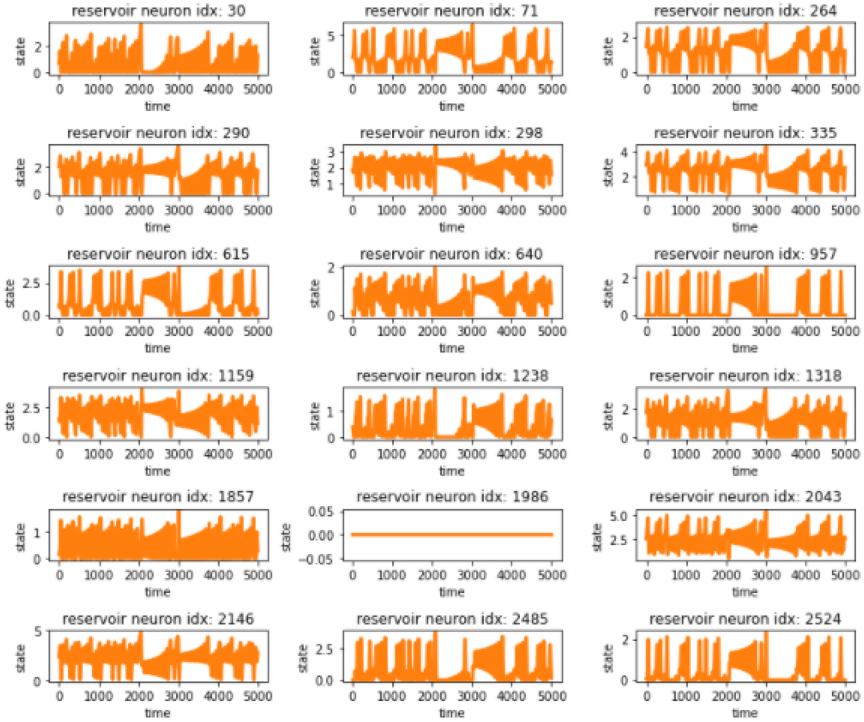

reservoir neuron idx: 30
reservoir neuron idx: 71
reservoir neuron idx: 264
reservoir neuron idx: 290
reservoir neuron idx: 298
reservoir neuron idx: 335
reservoir neuron idx: 615
reservoir neuron idx: 640
reservoir neuron idx: 957
reservoir neuron idx: 1159
reservoir neuron idx: 1238
reservoir neuron idx: 1318
reservoir neuron idx: 1857
reservoir neuron idx: 1986
reservoir neuron idx: 2043
reservoir neuron idx: 2146
reservoir neuron idx: 2485
reservoir neuron idx: 2524

Figure 9: Reservoir Activity for Lorenz 63 data

## 6 Conclusion

The network is able to make good short term predictions however the prediction horizon varies depending on the initial point. The error propagation in the network might be another interesting area to look into. This project helped me gain a better understanding of the parameters and functioning of echo state networks and I hope to be able to continue working on this in the future.

# References

[1] A. Chattopadhyay, P. Hassanzadeh, and D. Subramanian, “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 Processes in Geophysics*, vol. 27, no. 3, pp. 373–389, 2020.

[2] H. Jaeger, “The “echo state” approach to analysing and training recurrent neural networks-with an erratum note,” *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13, 2001.

[3] F. Damicelli, “echoes: Echo state networks with python.” `https://github.com/fabridamicelli/echoes`, 2019.

[4] H. Jaeger, “Echo state network,” *scholarpedia*, vol. 2, no. 9, p. 2330, 2007.

[5] 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: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 12, p. 121102, 2017.