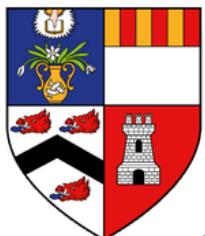


1495



UNIVERSITY OF
ABERDEEN
OILTHIGH OBAR DHEATHAIN

Is Chaos Good for Reservoir Computing?

Author:
Justinas Švykas

Supervisor:
Dr. Murilo Da Silva Baptista

2023-2024

This thesis is presented as a partial fulfillment of the requirements for the course PX4011, as part of an honors degree program in Applied Mathematics at the University of Aberdeen

This thesis was submitted on April 25, 2024

Abstract

The prediction of chaotic signals is vital task due to its frequency in the world, although complex due to sensibility to the initial conditions, which makes predictability decrease over time. Reservoir computing stands out as a cutting-edge machine learning model for predicting such signals when training data is limited. It is particularly suited to handling temporal data and employs a training methodology that sidesteps the vanishing gradient issue, a common challenge in neural network training. This thesis presents a method to categorise autonomous neural networks into irregular, nearly-irregular, and regular behaviors through the use of custom metrics derived from fast Fourier transform and Pearson correlation analyses. The main result is that neural networks with irregular or chaotic-like behaviors do not improve the prediction of chaotic signals, in contrast to regular and nearly-irregular reservoirs which offer superior predictions. Additionally, it is believed that configuring reservoir architecture in a specific way and tuning its hyper-parameters to induce Collective Almost Synchronous (CAS) behavior in the autonomous neural network could enhance the predictive capability of Reservoir Computing. Nevertheless, the expected CAS behavior was not observed in the studied networks, possibly due to the adoption of a random reservoir topology rather than a scale-free one.

Declaration of Originality

I hereby declare that this thesis is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university. My sources of information have been specifically acknowledged.

Signed: 

Acknowledgments

I want to thank my supervisor Murilo. In these past 8 months I have learned from Murilo a lot, while having a lot of fun and laughter (some lecturers told me they hear my laugh in Murilo's office). I have been always looking forward to our weekly meetings even if it was a bad day, as I knew after it I would come out with a smile. Also, I am thankful for Murilo's supervision, as without it this thesis would not be what it is today.

Also, I want to thank my lecturer Sandip for answering my pestering questions about phase space. Also, I want to thank my lecturer Roland for teaching me about the Lorenz system like its mixing dough of bread and telling me about researchers career.

I want to thank my family for their everlasting and loving support. Vaida thank you for sending me beer brewer pills, Martynai thank you for helping me with the Contract, Dad thank you for Sidhartha, and Mom for your garden flowers.

I feel very thankful for my friends. I want to thank everyone in the swing dance society for being such a loving bunch. Also, I want to thank Vlad for guided meditations. I want to thank Nathan for talking with me about girls and of course - meaning of life.

I want to thank all of the University of Aberdeen and its staff for giving me the opportunity to learn so many fascinating things. Also, I want to thank for giving me the opportunity to study in University of Hong Kong. I feel very thankful for that as there I could learn from the best, create life-lasting friendships, and see the vast and beautiful world.

Contents

1	Introduction	6
2	Machine Learning Methods	7
2.1	Neural Networks	8
2.2	Reservoir Computing	9
2.3	Initialize Reservoir Layer	9
2.4	Descriptions of Neural Networks	11
2.5	Training and prediction Phase	12
3	Non-linear analysis methods	15
3.1	Time-series analysis quantities and diagrams	15
3.2	Chaotic maps	16
3.3	Collective almost synchronous behavior	17
4	Analysis of hyper-parameters	19
4.1	Spectral radius and coupling	19
4.2	Complex Network Analysis	20
4.3	Finding Irregular Behaviour	24
5	Prediction	28
5.1	Network predictions	28
5.2	Error	32
5.3	Looped Network	34
6	Collective Almost Synchronization (CAS)	37
6.1	Bifurcations	38
6.2	Similar Neurons	42
7	Conclusions	46
8	Appendix A	51

1 Introduction

Artificial neural networks (ANNs) are machine learning techniques that are inspired by their biological counterparts. They are most commonly used to tackle computational and mathematical problems that require compressing enormous amount of data which for humans would be staggering to do. Recently, innovations and developments have been made possible by ANNs, in fields such as image and speech recognition, self-driving cars, chat-bots, image creation and predictions of chaotic signals. Despite this there are limits - huge amounts of data, electricity and computing are required to train a model.

This thesis explores reservoir computing (RC), a framework within artificial neural networks (ANNs), focusing on its potential for enhanced prediction of chaotic signals. Recent studies show that RC is particularly adept when historical training data is limited, outperforming many state-of-the-art models in such scenarios [1]. A key advantage of RC is its training method, which sidesteps the common vanishing gradient problem [2] seen in traditional ANNs which requires to suitably change the weights of the connections between pair of neurons (simulating how our plastic brain learns). This is achieved by fixing the network topology describing how neurons connect in the neural network (the reservoir) and training only the output weight matrix rather than every layer of the artificial neural network, making RC less computationally demanding as the training dataset grows [3]. However, it's noteworthy that with ample data, advanced architectures like LSTMs [4] and transformers [5] have successfully addressed the vanishing gradient issue and according to recent result it seems to surpass RC's predictive capabilities for chaotic signals [1]. Throughout this thesis, we employ the echo state network (ESN) model of RC, introduced by Jaeger [3].

This thesis investigates the enhancement of reservoir computing by leveraging chaos. The investigation is threefold: firstly, it involves classifying reservoirs into three categories—chaotic-like, nearly-chaotic-like, and periodic-like—during autonomous operation without input or output. This classification hinges on metrics derived from the fast Fourier transform [6], and the rate of decay in Pearson correlation [7] that can be used to infer the type of dynamical behavior [8] (chaotic, quasi-periodicity or periodic) of the neurons. Since this work does not use ergodic quantities such as Lyapunov exponents [9, 10] to prove chaos, it instead introduces custom definitions for network behaviors: 'irregular' corresponds to chaotic-like, 'nearly-irregular' to quasi-periodic-like, and 'regular' to periodic-like behaviors. The interest is to first investigate how these behaviors appears in terms of the hyperparameters such as probability of connections between neurons, the coupling strength, and the size of the reservoir

The subsequent phase of the research assesses the efficacy of various types of reservoirs, autonomously set to behave either chaotic-like, or periodic-like, in predicting chaotic signals. A significant observation from this study suggests that reservoirs exhibiting chaotic traits do not inherently enhance the precision of predictions for chaotic signals. Interestingly, reservoirs displaying near-chaotic and regular behaviors achieve comparable predictive accuracy. This supports T. Carroll research in showing RC predicts the best at "edge of chaos" [11]. The work further evaluates the impact of training the network with different variables of the Lorenz system [12], revealing that incorporating the z-variable may hinder rather than aid prediction. This is so as Lorenz has symmetry by the z variable, meaning that the same z value can have different x and y values.

The final segment of this thesis scrutinizes the proposition that integrating chaotic neurons within a reservoir, arranged to induce Collective Almost Synchronization (CAS) as outlined in [13], could potentially improve its performance. This proposition was motivated by findings in previous studies such as [14]. However, this thesis differs from previous research in CAS as in this thesis I

hypothesize that CAS can be found by coupling maps that are at the "edge of chaos" [11] instead of maps which are chaotic. My investigation showcases networks featuring both distinctly regular and irregular neuronal behaviors. However, the networks examined did not demonstrate simultaneous regular and irregular neuronal activities, a main characteristic of the CAS phenomenon. Consequently, this thesis concludes that CAS was not observed within the examined networks. The main reason for the absence of CAS is attributed to the coupling map which is not chaotic. However, this also can be caused by the random initialization of the reservoir's weight matrix rather than employing a scale-free network architecture. I believe that a scale-free topology [15], characterized by a mix of sparsely and highly connected neurons, would facilitate the coexistence of periodic and chaotic behaviors. Conversely, the random weight matrix used in this study results in neurons with intermediate levels of connectivity.

The thesis is organized as follows: Chapter 2 introduces essential machine learning concepts to provide a foundational understanding for the thesis. Chapter 3 discusses the nonlinear analysis methods employed in this research. Chapter 4 delves into the reservoir's intricacies, classifying them into irregular and regular categories. Chapter 5 presents the results from chaotic signal prediction experiments. Chapter 6 reflects on the search for CAS within the network and discusses the implications of the findings. Chapter 7 concludes the thesis by summarizing the key outcomes and suggesting avenues for future research.

I have used Python to create all of the results discussed in this thesis which can be found in Google Colab notebook [16]

2 Machine Learning Methods

Clarifying terminology is crucial in the context of Machine Learning (ML) algorithms, as certain terms are often mistakenly used interchangeably. In this chapter, I will establish clear definitions for several key concepts:

Artificial intelligence, or AI, is technology that enables computers and machines to simulate human intelligence and problem-solving capabilities [17]. Some applications of AI include: generating text, images, videos, or other data [18]. Examples of nowadays widely known include ChatGPT [19] and Midjourney [20].

Machine learning (ML) is a branch of artificial intelligence and computer science that focuses on using data and algorithms to enable AI to imitate the way that humans learn, gradually improving its accuracy [21]. Intuitively machine learning is the underlying algorithm under any AI application. Examples of ML include artificial neural networks, deep learning, recurrent neural networks, reservoir computing, linear regression, and transformers.

Artificial Neural Networks or ANNs are computational models inspired by the human brain's structure, consisting of interconnected units (neurons) that process information in a layered manner.

Deep Learning is a branch of ML that utilizes deep neural networks (By strict definition, a deep neural network is a neural network with three or more layers [22]) to analyze various forms of data with complex structures.

Recurrent Neural Networks is a type of deep learning which uses sequential data or time series data. They are distinguished by their "memory" as they take information from prior inputs to influence the current input and output. While traditional deep neural networks assume that inputs and outputs are independent of each other, the output of recurrent neural networks depend on the prior elements within the sequence [23].

Reservoir Computing is a type of recurrent neural network that is trained with a fixed, randomly generated reservoir and a trainable output layer. The main advantage of this framework is that only the connections to the output layer are trained, saving time and computational effort. [24]

Unlabeled data does not have explicit labels or annotations, often used in unsupervised learning tasks. Whereas **labelled data** is accompanied by labels or annotations, which provide a reference for supervised learning models.

Unsupervised Learning is type of ML that involves training models on data without explicit labels, allowing the model to identify patterns and structures on its own. Whereas **Supervised Learning** is another type of ML where the model is trained on a labeled dataset, learning to predict the output from the input data. When predicting a chaotic signal using RC, I will be dealing with supervised learning of labeled data.

Parameters: In the context of machine learning, a parameter is a configuration variable that is internal to a model, and whose value is deduced from historical training data. For example, the weights of a neural network can be considered a parameter.

Hyper - Parameter: Configurations set prior to the training process that guide the learning process. For example test-train ratio, learning rate, number of iterations etc. To achieve the best results one who implements the machine learning model has to optimise hyper-parameters.

2.1 Neural Networks

Before discussing the artificial neural networks architecture introduced by Jaeger in 2001 [3] it is important to notice that they are inspired by their biological counter parts. In this thesis I will be using various definitions that is coming from our brain. In Figure 1 a comparison is drawn between a biological neuron and an artificial neuron which is the most basic building block of ANN.

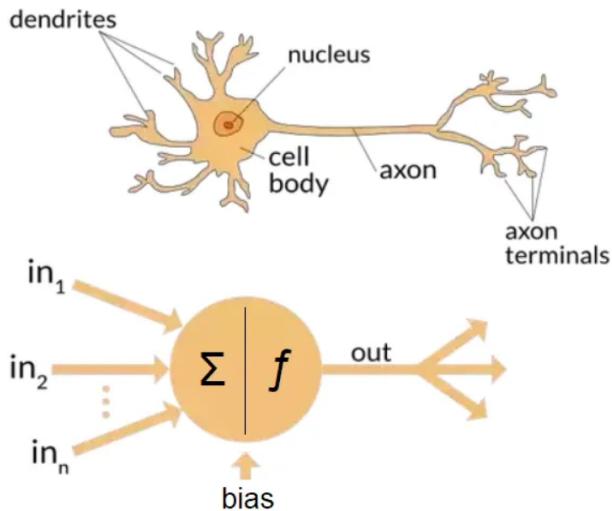


Figure 1: A comparison of a biological neuron from the brain and an artificial neuron [25].

The strengths of the connections between biological neurons will often change when exposed to external stimuli, this is how biological creatures like humans retain information and how learning takes place. Biological neurons are connected via axons and dendrites. They communicate in different ways, either by chemical or electrical synapses. The former is a unidirectional connection

mediated by ionic channels, and the latter a bidirectional connection mediated by the cells electric potential difference.

In this thesis, I will model both types in the artificial neuron networks. The connectivity of the neurons within the network modeling the synapses is represented by a matrix, either adjacency or a Laplacian matrix. Each element of this matrix has weights representing the strength of the synapsis connecting 2 neurons, so I will refer to it as the weight matrix. Notice that in the types of networks studied in this thesis, the elements of the connectivity matrix will not change their weights when exposed to external stimuli - training phase. In the following subsections, training phase will adjust only output layer weights showcasing the main advantage of reservoir computing over any other architecture. Thus the architecture as proposed by Jaeger [3] is not an exact replica of a plastic biological brain that learns by changing the strength of synapsis.

2.2 Reservoir Computing

Despite this widely acknowledged potential, and despite a number of successful academic and practical applications, the impact of RNNs in nonlinear modeling has remained limited for a long time. The main reason for this lies in the fact that RNNs are difficult to train by gradient-descent-based methods, which aim at iteratively reducing the training error. The RC paradigm saves time, and computational effort and avoids instability [26] of gradient-descent RNN training by setting up the fixed weight matrix W and training only the output layer W_{out} [27]. Because of these reasons RC is used in various ways among which are the prediction of chaotic signals [28], separations of chaotic signals [28], and prediction of EEG signals [14].

This thesis specifically focuses on improving reservoir computing Echo State Network (ESN) architecture. Let me introduce ESN as proposed in Jaeger [3] and furthered by Parlitz Zimmermann[26] as shown in Figure 2, which has typically considered neurons that behave periodically, or a network that behaves regularly without external stimuli. This type of behavior is assured by Echo State Property (ESP) [3] about which I will discuss more in the next subsection.

For clarity, the neural network evolves in discrete time. I will denote $T \in \mathbb{N}$ as number of iterations the network is run for and $n \in [0, T - 1]$ as the discrete time (first iteration denoted by $n = 0$ due to the convention in my programming language of choice). Also, h will denote the integration step of the measurements of a given time series (the trajectory of the Lorenz attractor which I will elaborate on in the next chapter. RC has different forms but ESN is the implementation of choice in this work, so in the context of the remainder of the thesis the terms ESN, RC and network will all be used interchangeably to describe the networks that are subsequently created.

2.3 Initialize Reservoir Layer

The reservoir layer W can be created following 3 steps:

1. Create weight matrix W :

The reservoir (hidden) layer is defined as connections between neurons. This can be described as weight matrix $W \in \mathbb{R}^{N \times N}$. The entries of W are assigned randomly from the interval $[-1, 1]$.

2. Make matrix W Sparse:

In reservoir computing not all of the neurons are connected. Thus the majority $(1-\epsilon)\%$ of W entries zero, where $\epsilon \in [0, 1]$ corresponds to the sparsity of the matrix. In this thesis

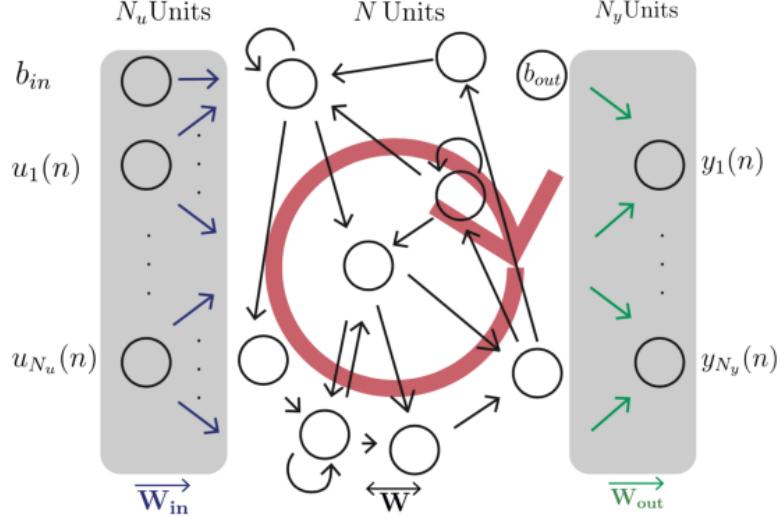


Figure 2: ESN general structure as described by Jaeger [3], then Parlitz and Zimmermann [26]. From left to right the input signal \vec{u}_n passes through input weight matrix W_{in} of N_u neurons, through a reservoir of N neurons with weights W , then output weight matrix W_{out} with N_y neurons where the output signal \vec{y}_n is calculated.

sparsity $\epsilon = 0.2$ corresponds to keeping 20% of W entries to be nonzero, unless explicitly said otherwise.

3. Normalize and rescale the reservoir layer W by spectral radius ρ :

To normalize W I divide each entry by the largest absolute eigenvalue, meaning that the spectral radius of the normalized matrix is 1. The normalized matrix is then rescaled to a new spectral radius by multiplying each entry by the desired value ρ . This can be written in the following equation:

$$W_{\text{new}} = \rho \left(\frac{W_{\text{old}}}{\max\{|\mu_1|, |\mu_2|, \dots, |\vec{u}_n|\}} \right) \quad (1)$$

Where \vec{u}_n ($n = 1, \dots, N$) are the eigenvalues of W and $\rho \in \mathbb{R}_{++}$ (In this thesis I will investigate $\rho \in [0, 3]$ only).

The spectral radius ρ is one of hyper-parameters which plays pivotal role in assuring ESP holds creating a network that behaves regularly. Echo state property characterizes valid ESN dynamics and essentially says that the reservoir state should asymptotically depend only on the driving input signal (the network state is an echo of the input), while the influence of initial conditions should progressively vanish with time. For most practical purposes, the echo state property is assured if the reservoir weight matrix W is scaled so that its largest absolute eigenvalue is less than 1 (This is the case if $\rho < 1$ in Equation (1)). Or, using another term, W is contracting. It is necessary to say that $\rho < 1$ does not amounts to a necessary and sufficient condition for the echo state property. ESP is violated if $\rho > 1$ in reservoirs using the \tanh function as neuron non-linearity, and for zero input. Contrary to widespread misconceptions, the ESP can be obtained even if $\rho > 1$ for non-zero input (including bias inputs to neurons), and it may be lost even if $\rho < 1$, although it is hard to construct systems where this occurs [27]. Notice that I will be working with reservoir that when autonomous behave non-periodically, so ESP should not be expected to hold.

Once the connections in the reservoir are initialised let me turn to the input and output matrices. The connections between input layer composed of N_u neurons and the reservoir layer composed of N neurons are stored in the matrix $W_{in} \in \mathbb{R}^{N \times N_u+1}$ whose entries are set randomly from the interval [-1,1]. The output connections are stored in W_{out} and have dimensionality $W_{out} \in \mathbb{R}^{N_y \times 1+N+N_u}$ whose entries are also set randomly in the interval [-1,1], but as mentioned before, these will change in response to training, unlike the entries of W and W_{in} which remain set once assigned. The output layer has N_y neurons.

2.4 Descriptions of Neural Networks

This chapter focuses on the neural networks descriptions for the reservoir and its input and output layers. As I have mentioned in before - neurons communicate both chemically and electrically. In this thesis, I will simulate both of these situations.

The dynamical state of neurons in the reservoir layer is described by the vector $\vec{s}_n \in \mathbb{R}^{N \times 1}$. Each component of \vec{s}_n represents the state of one of the neurons at time n . They are initially assigned values randomly from the interval [1, 1]. Neurons states which communicate chemically are adjusted via the autonomous update equation, a discrete map proposed by Parlitz and Zimermann [26]:

$$\vec{s}_n = (1 - \alpha) \vec{s}_{n-1} + \alpha \tanh(W \vec{s}_{n-1}), \quad (2)$$

with typically leaking rate $\alpha \in [0, 1]$. But notice that in this thesis I will also investigate $\alpha \in [1, 2]$.

Alternatively, neurons can interact electrically as proposed in [14], in which case the state s_n^i of a neuron $i \in [0, N-1]$ is adjusted via:

$$s_n^i = (1 - \alpha) s_{n-1}^i - \alpha \sum_{j=0}^{n-1} W_{ij} (s_{n-1}^j - s_{n-1}^i), \quad (3)$$

Notice that if $W_{ij} \neq 0$, then neuron i interacts with neuron j . Using Equation (3) in experimentation I encountered overflow - states quickly diverged to infinity. Compared to the chemical state Equation (2) activation function (\tanh) put the states in a closed set this way avoiding overflow. To have dynamics which is bounded and is similar to Equation (2) I decided to divide the second term by $2\hat{k}_i$ updating Equation (3) into Equation (4):

$$s_n^i = (1 - \alpha) s_{n-1}^i - \frac{\alpha}{2\hat{k}_i} \sum_{j=0}^n W_{ij} (s_{n-1}^j - s_{n-1}^i). \quad (4)$$

where $\hat{k}_i = \sum_{j=0}^{n-1} W_{ij}$ is the weighted degree of neuron i . This makes proposed network to have bounded dynamical behaviour.

During the experimentation phase, I noticed that electrical (or linear) connectivity between neurons computationally takes significantly longer than chemical. This happens due to element-wise multiplication as represented by Equation (4) and implemented computationally without any vectoral representation. Thus to reduce computational time I introduce vector-wise multiplication, and implement also vector computation in computer code, with the help of the weighted Laplace matrix changing electrical state Equation (4) into the following:

$$\vec{s}_n = (1 - \alpha) \vec{s}_{n-1} + \frac{\alpha}{2\hat{k}_i} \hat{L} \vec{s}_{n-1}, \quad (5)$$

where $\hat{L} = I\hat{k} - W$ and $\hat{k} \in \mathbb{R}^{N \times 1}$. Equation (4) equals Equation (5) as can be seen from the following equations:

$$\sum_{j=0}^{n-1} W_{ij}(s_{n-1}^j - s_{n-1}^i) = \sum_{j=0}^{n-1} W_{ij}s_{n-1}^j - \sum_{j=0}^{n-1} W_{ij}s_{n-1}^i = \sum_{j=0}^{n-1} W_{ij}s_{n-1}^j - \hat{k}_i s_{n-1}^i = W\vec{s}_{n-1} - I\hat{k}\vec{s}_{n-1} = -\hat{L}\vec{s}_{n-1}$$

Now using either chemical network Equation (2) or electrical network (5) I can run our reservoir autonomously without any input or output. That can be done by saving the evolution of the state vectors over time into only-state matrix $S \in \mathbb{R}^{N \times T}$ where T is the number of iterations the system is run for, about which I will elaborate in the next subsection.

In chapter 6 to find CAS I need to introduce a bounded chaotic map as in [14] [29] and [13]. For example, F can be a Hindmarsh-Rose map [13], Kuramoto's Oscillators [29], or Lorenz system. Thus, I updated Equation (3) into a more general case:

$$\vec{s}_n = (1 - \alpha)F(\vec{s}_{n-1}) + \frac{\alpha}{2\hat{k}_i}\hat{L}\vec{s}_{n-1}, \quad (6)$$

where F is any map i.e. periodic or chaotic. Similarly, Equation (2) can be updated into a more general Equation (7):

$$\vec{s}_n = (1 - \alpha)F(\vec{s}_{n-1}) + \alpha \tanh(W\vec{s}_{n-1}). \quad (7)$$

Notice that Equations (2) and (5) are special cases of these general case Equations (??) when F is an identity map i.e. $F(s_{n-1}) = s_{n-1}$.

Let me now introduce how external perturbations are added to the neural network description. To further analyse how the network predicts certain dynamics I need to input a vector $\vec{u}_n \in \mathbb{R}^{N_u}$ and $b_{in} = 1$ into the network through associated input weights W_{in} . I do that by updating state equations into

$$\vec{s}_n = (1 - \alpha)F(\vec{s}_{n-1}) + \alpha \tanh(W\vec{s}_{n-1} + W_{in}[b_{in}; \vec{u}_n]), \quad (8)$$

$$\vec{s}_n = (1 - \alpha)F(\vec{s}_{n-1}) + \alpha(\hat{L}\vec{s}_{n-1} + W_{in}[b_{in}; \vec{u}_n]). \quad (9)$$

where notation $[\vec{a}; \vec{b}]$ means concatenate a new vector from some vectors $\vec{a} \in \mathbb{R}^{N_a \times 1}$ and $\vec{b} \in \mathbb{R}^{N_b \times 1}$ into $[\vec{a}; \vec{b}] \in \mathbb{R}^{(N_a+N_b) \times 1}$. e.g.

$$[\vec{a}; \vec{b}] = \begin{bmatrix} \vec{a} \\ \vec{b} \end{bmatrix}. \quad (10)$$

In general, states of neurons in this thesis are described using either a chemically connected network as in Equation (8) or an electrically connected one as in Equation (9).

2.5 Training and prediction Phase

Before discussing how to predict let me introduce some definitions. Let our target vector be $y_n \in \mathbb{R}^{N_y \times 1}$ and store it into columns of $Y \in \mathbb{R}^{N_y \times T}$.

$$Y = [y_0, y_1, y_2, \dots, y_{T-1}] \in \mathbb{R}^{(1+N_y) \times T}. \quad (11)$$

The target vector represents what we would like our trained reservoir to reproduce or predict. For a network with input let me store \vec{u}_n , $b_{out} = 1$ and \vec{s}_n in extended states defined as follows

$$\vec{x}_n = [b_{out}; \vec{s}_n; \vec{u}_n] \in \mathbb{R}^{(1+N+N_u) \times 1}. \quad (12)$$

Let me also define the extended state matrix $X \in \mathbb{R}^{(1+N+N_u) \times T}$ whose columns are the extended states \vec{x}_n where $n \in [0, T - 1]$

$$X = [x_0, x_1, x_2, \dots, x_{T-1}] \in \mathbb{R}^{(1+N+N_u) \times T}. \quad (13)$$

The network is run for a total of T iterations. Total run time is divided into 3 phases: Transient, Training and Prediction phases such that $T = T_{tran} + T_t + T_p$. To begin with, the network is run for T_{tran} increments to allow network to settle down. Then the network is shown part of target data $Y_t \in \mathbb{R}^{N_y \times T_t}$ in order to "learn" its relationship (represented by weight matrix W_{out}) to associated extended states $X_t \in \mathbb{R}^{(1+N+N_u) \times T_t}$ where X_t represents the extended matrix in Equation(13) for the time window of the training phase. To learn W_{out} we first have to turn our attention to what we need to predict. For now assume we have learned relationship W_{out} and use it and extended states \vec{x}_n to predict \vec{y}_n for $n \in [T_{tran} + T_t, T - 1]$. This can be written in a simple linear relationship

$$\vec{y}_n = W_{out}\vec{x}_n = W_{out}[b_{out}; \vec{s}_n; \vec{u}_n]. \quad (14)$$

where $\hat{y}_n \in \mathbb{R}^{N_y \times 1}$ is prediction vector. In matrix notation, this can be given as

$$Y_p = W_{out}X_p \quad (15)$$

Where $Y_p \in \mathbb{R}^{N_y \times T_p}$ and $X_p \in \mathbb{R}^{1+N+N_u \times T_p}$, representing Y (in Equation (11)) - where Y_p should predict the target signal - for the time windows Training and X (in Equation (13)) - representing the extended state of the reservoir - for the time windows for Training.

In order to calculate W_{out} , or in other words, to train the reservoir, I first have to define a metric with which I measure the error of the predictions of the reservoir and the target signal. In this thesis I will be using Root Mean Squared Error defined as follows

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2}, \quad (16)$$

where n is number of observations, y_i is the target value for the i -th observation and \hat{y}_i is the predicted value for the i -th observation. To calculate W_{out} , in classic RNN approach that is done iteratively adjusting W_{out} by minimising the cost function

$$C(W_{out}) = \sqrt{\frac{1}{n} \sum_{i=0}^{T_t-1} (y_i - W_{out}\vec{x}_i)^2 + \lambda \text{Tr}(W_{out}W_{out}^T)} \quad (17)$$

The first term measures the RMSE error of the prediction and the second is a regularisation term with strength $\lambda > 0$ which is used to avoid overfitting of W_{out} . An illustrative example of overfitting is given in the Appendix A (Figure 96). In a typical RNN, training is usually done iteratively using the gradient descent method as in Equation (17), an explanation with a picture of which is given in Appendix A (Figure 97). As I have discussed before this comes with its shortcomings.

The beauty of reservoir computing lies in that it can avoid gradient descent by directly calculating the Moore–Penrose pseudoinverse X_t^+ of X_t and W_{out} as

$$W_{out} = Y_t X_t^{++}. \quad (18)$$

Direct pseudoinverse calculations exhibit high numerical stability, but are expensive memory-wise for large state-collecting matrices $X_t \in \mathbb{R}^{(1+N+N_u) \times T_t}$ thereby limiting the size of the reservoir N and/or the number of training samples T_t [27].

This issue is resolved in by manipulating Equation (15), multiplying both sides for X^T , where transpose of X is denoted X^T :

$$W_{out} X_t X_t^T = Y_t X_t^T. \quad (19)$$

A naive solution of it would be

$$W_{out} = Y_t X_t^T (X_t X_t^T)^{-1} \quad (20)$$

Note that in this case $Y_t X_t^T \in \mathbb{R}^{N_y \times N}$ and $X_t X_t^T \in \mathbb{R}^{N \times N}$ do not depend on the length T_t of the training sequence, and can be calculated incrementally while the training data are passed through the reservoir. Thus, having these two matrices collected, the solution complexity of Equation (20) does not depend on T either in time or in space. Also, intermediate values of W_{out} can be calculated in the middle of running through the training data, e.g., for an early assessment of the performance, making this a “semi-online” training method. The method in Equation (20) has more numerical stability, compared to Equation (18). By using the pseudoinverse $(X_t X_t^T)^+$ instead of real inverse $(X_t X_t^T)^{-1}$ in Equation (18) we gain more stability [27]. In addition, this method enables one to introduce ridge, or Tikhonov, regularization elegantly:

$$W_{out} = Y_t X_t^T (X_t X_t^T + \lambda I)^+ \quad (21)$$

where $I \in \mathbb{R}^{N \times N}$ is the identity matrix and $\lambda > 0$ is regularization factor.

Now the training phase used in this thesis can be summed up in 3 steps:

- Harvest extend states x_n for $n \in [T_{tran}, T_{tran} + T_t - 1]$ during training which requires T_t number of column stacking operations to create $X_t \in \mathbb{R}^{1+N+N_u \times T_t}$;
- Create training data $Y \in \mathbb{R}^{N_y \times T}$ about which I will discuss in the next chapter;
- Calculate W_{out} using Equation (21) at the end of $T_{tran} + T_t$ iteration.

After training phase, network is ready to move into prediction phase and calculate predictions \vec{y}_n for $n \in [T_{tran} + T_t, T - 1]$ using Equation (14). Notice that in this approach Equation (21) is completely left out during the prediction phase, otherwise the network will be retrained according to its own predictions. Predicted outputs are already slightly inaccurate, so allowing W_{out} to change in this way will cause errors to accumulate until the model no longer reflects the situation it was intended for. Notice the network can be adjusted to predict $\tau \in \mathbb{N}$ iterations in the future by training the network with adjusted desired output vector $\vec{y}_{n+\tau}$ for $n \in [T_{tran} - \tau, T_{tran} + T_t - 1]$.

From Equations (14) and (15), it is clear that at each n the network is predicting y_n , depending only on a given \vec{u}_n stored in x_n , and the trained W_{out} . It is important to note that the network is fed input \vec{u}_n for $n \in [0, T_{tran} + T_t - 1]$ which can be equal to \vec{y}_{n-1} e.g. input x coordinate of chaotic signal and predict x coordinate. In this case, after training the reservoir can be looped to use its own predictions as input vector. To be precise, at the point of first prediction $n_{p1} = T_{tran} + T_t - 1$ I make $\vec{y}_{n_{p1}} = \vec{u}_{n_{p1}+1}$ and from that point onward $\vec{y}_{n-1} = \vec{u}_n$ for $n \in [T_{tran} + T_t, T - 1]$. This approach of feeding the output back in as an input will be demonstrated in Chapter 5, which is an example of a feed-back loop introduced in the work by Jaeger in 2001 [3]. Whereas, in [28] the input is a mixture of signals, and the prediction is a component of this mixture. This means that the output at each step cannot be used as the input for the proceeding step. Overall the presence of a feed-back loop depends on the task each network is trained for.

3 Non-linear analysis methods

3.1 Time-series analysis quantities and diagrams

In this chapter, I define the various quantities and diagrams utilized in this thesis to characterize the dynamical behavior of the reservoir. Let me denote $n \in [0, T - 1]$, where T represents the number of increments desired for plotting.

States diagram plots the states s_n of N neurons in different colors, illustrating the dynamic activity of each neuron over time.

Mean field or average of states is the mean \bar{s}_n of all neurons N at each iteration n , spanning all T iterations. The mean is calculated as follows:

$$\bar{s}_n = \frac{1}{N} \sum_{i=1}^N s_n^i. \quad (22)$$

Variance σ^2 of states across the desired T iterations, computed by the equation:

$$\sigma_n^2 = \frac{1}{N} \sum_{i=1}^N (s_n^i - \bar{s}_n)^2 \quad (23)$$

Fourier spectra in terms of magnitude is calculated using fast Fourier transform of an average of states as in Equation (22), offering insights into the frequency components of the network.

Pearson correlation coefficient is:

$$r(\Delta t) = \frac{\left| \sum_{n=1}^{T-\Delta t} (\bar{s}_n - \bar{s})(\bar{s}_{n+\Delta t} - \bar{s}) \right|}{\sum_{n=1}^{T-\Delta t} (\bar{s}_n - \bar{s})^2} \quad (24)$$

with $\bar{s} = \frac{1}{T} \sum_{n=1}^T \bar{s}_n$ and $\Delta t = 10$.

Bifurcation diagrams shows transitions in network dynamics by plotting local maxima and minima of the state vector s_n^i for the i-th neuron as certain hyper-parameters of the network are varied. These diagrams help pinpoint critical values of hyper-parameters at which phenomena such as saddle-node, transcritical, pitchfork, and Hopf bifurcations occur, thereby fundamentally altering the dynamics of the reservoir. Occasionally, bifurcation diagrams will include multiple neurons to examine whether they exhibit similar behavior.

Delay diagrams visualise the dynamics of two aligned neurons in the reconstructed phase space, employing delay embedding techniques with specified delay (τ_d). Where by aligned neurons I mean finding the time lag (τ_l) of neuron j by maximizing their cross-correlation (cross-correlation is calculated via Python function [30]) to align their trajectories effectively for comparison. Essentially x axis visualise two aligned neurons (s_n^i and $s_{n+\tau_l}^j$) in the x axis against the same 2 neurons with specified delay ($s_{n-\tau_d}^i$ and $s_{n+\tau_l-\tau_d}^j$) on the y axis. This method is useful for uncovering dynamic structures such as limit cycles or chaotic attractors.

Heat Maps provide a visual representation of how a network responds to changes in two hyper-parameters, using color to indicate the impact of each metric.

Running average is a method to smooth out the noise in time-series which allows for clearer observation of the underlying trends. In this thesis running average (a_n) at iteration n calculated as the mean of the previous $w = 10$ observations

$$a_n = \frac{1}{w} \sum_{i=n-w+1}^n a_i. \quad (25)$$

Definitions used to categorize network behaviors in this thesis are as follows:

- **Regular behavior:** characterized by a Pearson Correlation that oscillates with Δt and spectra (FFT) with well-defined peaks, indicating periodic oscillatory behavior, or a roughly constant behavior suggesting an equilibrium point.
- **Nearly-irregular:** features an oscillating Pearson correlation with respect to Δt with slowly decaying amplitude as Δt increases and spectra (FFT) that shows well-defined peaks alongside finite but small amplitude broadband, suggesting quasi-periodic or weakly chaotic behaviors.
- **Irregular:** defined by a decaying Pearson correlation with respect to Δt and broadband spectra, indicative of chaotic behavior.

The choice to detect behavior based on the correlation decay, the Fourier spectra, and the visual inspection of the mean field was based on the fact that measuring the Lyapunov Exponents [8, 10] is a much more complex task, risking taking crucial time for the overall honors thesis work. It is however worth to comment that much of work done to demonstrate chaos has been based on the study of the correlation decay as well as on the Fourier spectra [8, 10].

3.2 Chaotic maps

In this thesis I will be predicting the Lorenz system as proposed by Lorenz [12]. This system can be described by the following equations:

$$\frac{dx}{dt} = \sigma(y - x), \frac{dy}{dt} = x(\rho - z) - y, \frac{dz}{dt} = xy - \beta z, \quad (26)$$

Where $\sigma = 10, \rho = 28, \beta = \frac{8}{3}$. This equation is not analytically solved, and to solve it I rely on numerical techniques providing us with input vector \vec{u}_n and desired output vector \vec{y}_n .

4th order Runge-Kutta or RK4 is the numerical method of choice used to approximate the solution of ordinary differential equations $\frac{dy}{dt} = f(t, y)$. With an initial condition $y(t_0) = y_0$, the RK4 method provides the following iterative formula for each step:

$$\begin{aligned} k_1 &= h \cdot f(t_n, y_n) \\ k_2 &= h \cdot f\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) \\ k_3 &= h \cdot f\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right) \\ k_4 &= h \cdot f(t_n + h, y_n + k_3) \\ y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \quad (27)$$

where h is the chosen step size.

It is essential to give an example of how an input vector can be defined in discrete time to prevent any confusion. Say the input vector is 3 dimensional as follows

$$\vec{u}_n = (x(nh), y(nh), z(nh)) \quad \text{for } n \in [0, T - 1] \quad (28)$$

where input vector \vec{u}_n consists of the Lorenz system's state variables x , y , and z at Lorenz time nh , aligning the discrete time indexing with physical time measured in multiples of h . Similarly, the output vector \vec{y}_n can be defined. Note that \vec{u}_n and \vec{y}_n might differ as discussed before.

Notice, for maps F that have a constant Jacobian, or a constant derivative, the Lyapunov exponent (LE) can be calculated simply by

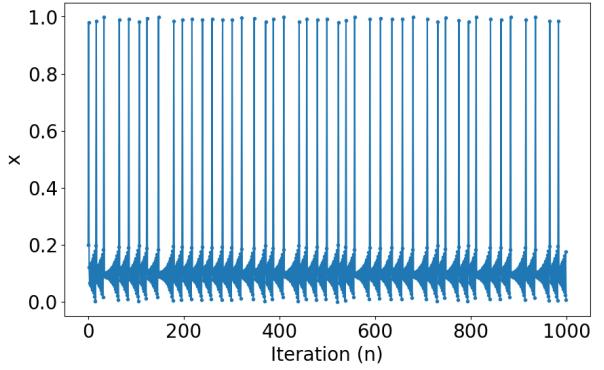
$$LE = \ln |DF| \quad (29)$$

where DF is the derivative of the map with respect to the variable, i.e., $\frac{dF}{dx}$ [10].

For the rest of the thesis for function F in Equations (8) and (9), I will use a simple map as follows

$$F(s_{n+1}) = (a + bs_n) \mod 1, \quad (30)$$

where $a \in [0.15, 0.25]$ and $b \in [-1.15, -1]$, as proposed in [31]. The map's behavior is illustrated in the Figure to the right. Notice this map has $DF = b$, consequently $LE = \log |b|$ by Equation (29). Thus for all parameters in the domain of b , the map has $LE > 1$ and thus is chaotic, except for $b = -1$ for $LE = 0$ ([10]). I will use this map to look for Collective Almost Synchronization (CAS). In this thesis I will use $b = -1$ and $a = 0.15$ to make the map behave to what I believe to be the "edge of chaos".



Map as in Equation (30) iterated for 1000 steps with initial condition $s_0 = 0.2$

3.3 Collective almost synchronous behavior

In this subsection I will define collective almost synchronous behavior (CAS) as proposed in [13]. It is important to study the emergence of collective behavior as it is common in nature [32, 33, 34]. If s_n^i is the variable describing the phase of a node i in the network at the iteration n , and \bar{s}_n represents the mean field defined as in Equation (22), collective behavior may appear in the network mean field. Behavior in a network is characterized by not only $\bar{s}_n \neq s_n^i$ but also nodes by nodes evolving in a way that cannot be described by the evolution of only one individual node when isolated from the network.

In contrast to collective behavior, another widely studied behavior of a network is when all nodes behave equally, and their evolution can be described by an individual node when isolated from the network. This state is known as complete synchronisation [35] and appears when $s_n^i = s_n^j = \bar{s}_n$ for all time, where i and j are some nodes of network. In networks whose nodes are coupled by non-linear functions, such as those that depend on time-delays [36] or those that describe how neurons chemically connect [37], the evolution of the synchronous nodes might be different from the evolution of an individual node when isolated from the network.

Collective almost synchronous behavior is highly dependent on the number of connections between neurons. Degrees defined as $d_i = \sum_{j=1}^N A_{ij}$, where d_i represents the number of connections for i -th neuron and $A \in \mathbb{N}^{N \times N}$ is matrix which entries are 0 or 1. To describe connections I will be using weighted degrees defined as $k_i = \sum_{j=1}^N W_{ij}$ where W is our reservoir weight matrix, and W_{ij} is its row- i and column- j component. Loosely, when CAS is in place neurons with similar and strong connections behave regularly while those with low connections behave irregularly, chaotically. Let

me define this behavior more rigorously starting with local mean field of i -th neuron as follows

$$\hat{s}^i(t) = \frac{1}{k^i} \sum_{j=1}^N W_{ij} s_n^j. \quad (31)$$

CAS is a phenomenon that appears in a complex network that produces a weaker form of synchronization [13]. In this phenomenon, nodes are in weak interaction (weak coupling strength) and if only sparsely connected will typically behave independently and chaotically. However, for neurons strongly connected forming a local cluster, they will experience a roughly constant local mean fields. Every other neuron in the network with similar local connection will experience a similar local mean field and all these neurons will behave regularly ("periodically") in a similar way. The trajectory of these neurons is what is called the CAS pattern. The CAS pattern is a solution of a simplified Equation (33) describing a network that has neurons i for which the local mean field is roughly constant, or $\hat{s}^i = C^i$. The expected value of the local mean field is defined as

$$C_i = \lim_{t \rightarrow \infty} \int \hat{s}_i(t) dt. \quad (32)$$

Then the Equation (4) of the electrical network in discrete time can be described in terms of the local mean field by

$$s_n^i = (1 - \alpha) F(s_{n-1}^i) + \frac{\alpha}{2} s_{n-1}^i - \frac{\alpha}{2} C^i + \delta. \quad (33)$$

where δ is residual term ($\delta = \bar{s}^i - C^i$). Equation (4) becoming (33) can be seen from $\frac{1}{k^i} \sum_{j=0}^{N-1} W_{ij} s_{n-1}^j \approx C^i$ and following operations:

$$(1 - \alpha) F(s_{n-1}^i) + \frac{\alpha}{2k_i} s_{n-1}^i \sum_{j=0}^{N-1} W_{ij} - \frac{\alpha}{2k_i} \sum_{j=0}^{N-1} W_{ij} s_{n-1}^j = (1 - \alpha) F(s_{n-1}^i) + \frac{\alpha}{2} s_{n-1}^i - \frac{\alpha}{2k_i} \sum_{j=0}^{N-1} W_{ij} s_{n-1}^j. \quad (34)$$

The following are the two criteria for node i to present the CAS phenomenon:

- Criterion 1. The central limit theorem can be applied. Therefore, the larger the degree of a node, the smaller the variation in its local mean field.
- Criterion 2. The CAS pattern describes a stable periodic orbit denoted by $B_{p_i}(t)$, where $p_i = \alpha \hat{k}_i$, a quantity that measures the influence on the node i of the nodes that are connected to it.

If a network has the CAS phenomenon then there has to exist nodes i such that

$$|s^i(t) - B_{p_i}(t - \tau_i)| < \varepsilon_i, \quad (35)$$

for all time t (or all timesteps n in discrete setting), where ε_i is a small quantity, not arbitrarily small, but reasonably smaller than the envelop of states $s^i(t)$ (s_n^i in discrete setting) and τ_i denotes time lag between $s^i(t)$ and $B_{p_i}(t)$.

4 Analysis of hyper-parameters

In this chapter, I will be investigating how hyper-parameters influence the autonomous network's dynamical behavior. In the first subsection, I will discuss the influence of spectral radius and coupling on the network's dynamical behavior. In the second subsection, I will summarize the results of Sec. 4.1 by explaining the range of hyper-parameters that produce different dynamical regimes. While in the third subsection I will argue for this method of categorizing networks using customised metrics. In this chapter I will be investigating only chemically connected reservoir without input based on Equation (2).

4.1 Spectral radius and coupling

In this subsection, I will look into spectral radius (ρ) and coupling (α) when the network has 10 neurons ($N = 10$). I believe they are one of the most influential hyper-parameters regarding the behavior of the networks. This is because ρ represents the global expanding ($\rho > 1$) -potentially producing irregular behavior - or contracting ($\rho < 1$) behavior - potentially inducing regular behavior - of the network induced by its connection topology for the absence of inputs as discussed about ESP in chapter 2, and α represents the contribution of the individual neuron to the network.

To begin with, consider a weakly coupled network with $\alpha = 0.05$ and $\rho < 1$. The neurons tend to asymptotically converge towards a fixed point as can be seen in Figure 3. As α is increased from 0.05 to 0.5 the neurons converge to fix point 0 faster as in Figure 4.

Now let me take a look at reservoirs when $\rho > 1$. If α is increased from 0.05 to 0.5 notice again that states reach equilibrium faster in just 15 increments in Figure 7 compared to 150 increments in Figure 5. Thus overall I can conclude that the bigger coupling α is, the faster neurons reach an equilibrium.

As ESP suggests network contracts when $\rho < 1$ seen in Figures 3 and 4 where neurons asymptotically converge to a fixed point at 0. Notice if $\rho > 1$, individual neurons also reach equilibrium, but each at a different value as seen in Figure 5. As a whole network reaches a fixed point at around -0.12 from Figure 6. Furthermore, notice that in Figure 8 neuron states evolve more irregularly than those in Figure 3 even though coupling is the same. Thus it seems that the bigger the spectral radius ρ is, the more irregular the network is. As irregularity is an indication of chaos, this suggest that chaos might be achieved by tuning ρ to values close to or larger than 1.

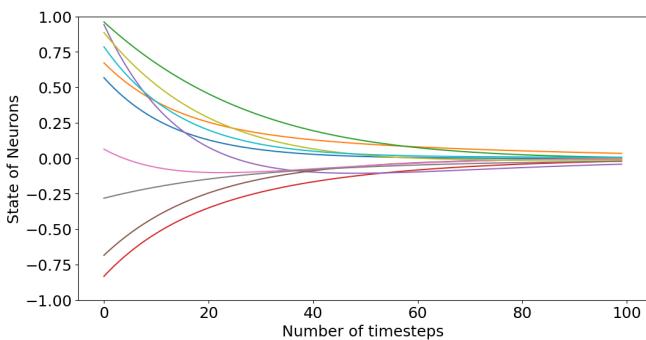


Figure 3: States of neurons over $T=100$ in reservoir with $\rho = 0.5$ and $\alpha = 0.05$

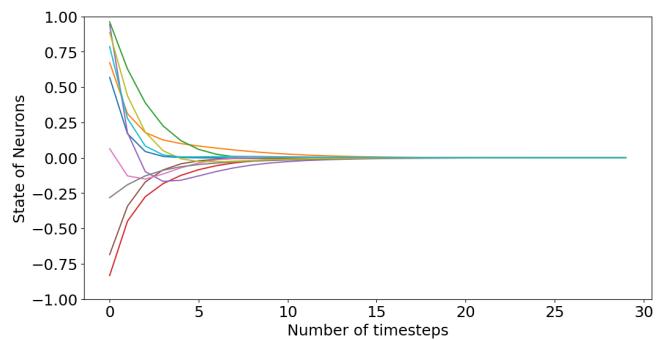


Figure 4: States of neurons over $T=30$ in reservoir with $\rho = 0.5$ and $\alpha = 0.5$

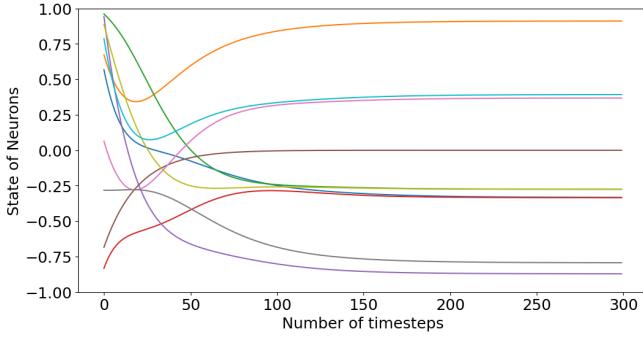


Figure 5: States of neurons over $T=300$ in reservoir with $\rho = 1.5$ and $\alpha = 0.05$

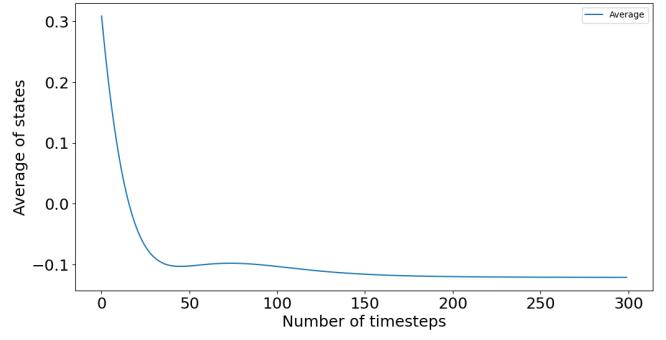


Figure 6: Average of neurons states over in reservoir with $\rho = 1.5$ and $\alpha = 0.05$

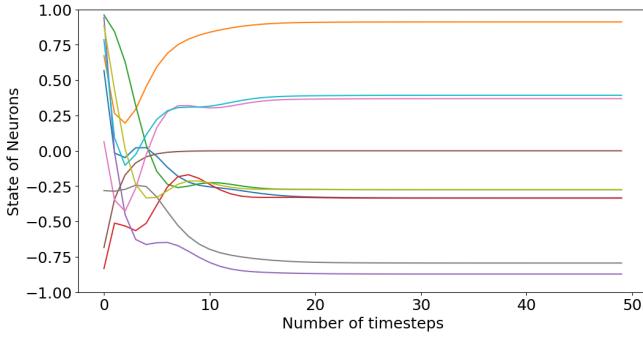


Figure 7: States of neurons over $T=50$ in reservoir with $\rho = 1.5$ and $\alpha = 0.5$

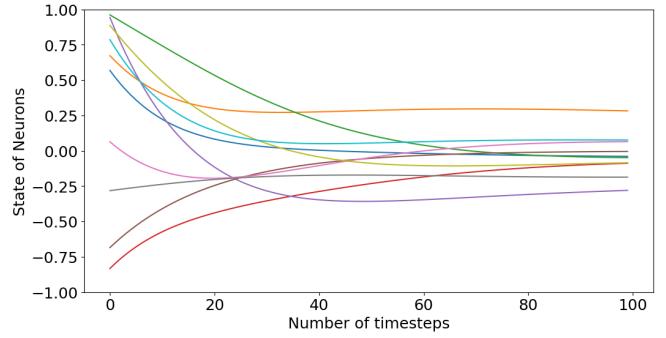


Figure 8: States of neurons over $T=100$ in reservoir with $\rho = 0.95$ and $\alpha = 0.05$

4.2 Complex Network Analysis

In this subsection, I will look into more complex Reservoirs increasing the number of neurons to ($N = 100$). To better understand and characterize these complex networks dynamical behaviors I will measure the quantities: average states \bar{s}_n , fast-Fourier transform, variance σ_n^2 and Pearson correlation $r(\Delta t)$.

Reservoir 1 will be defined as network with $\rho = 0.5$ and $\alpha = 0.05$, investigations will be carried out for $T=100$. Figures from 11 to 12

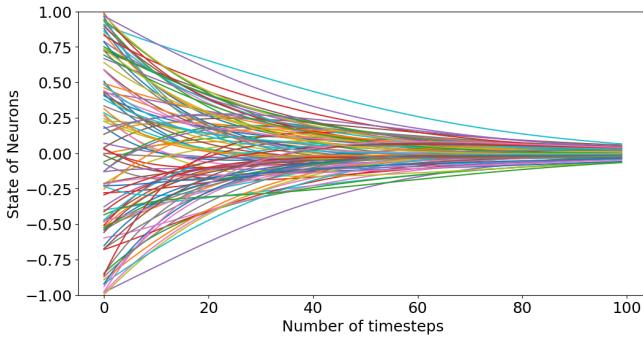


Figure 9: States in Reservoir 1

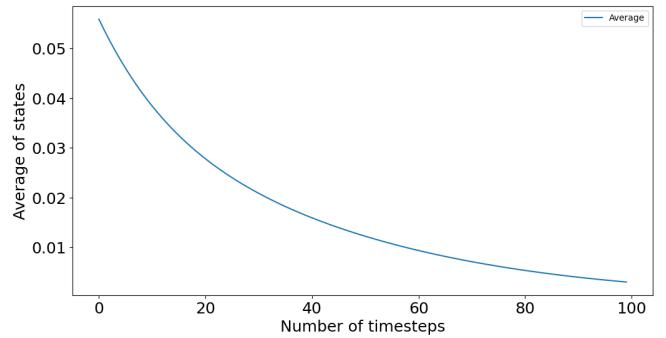


Figure 10: Mean field in Reservoir 1

show that the neurons in this reservoir are asymptotically approaching an equilibrium state indicating regular behavior. Figure 9, 10 and 11 for the time-series of neurons, the mean field of the

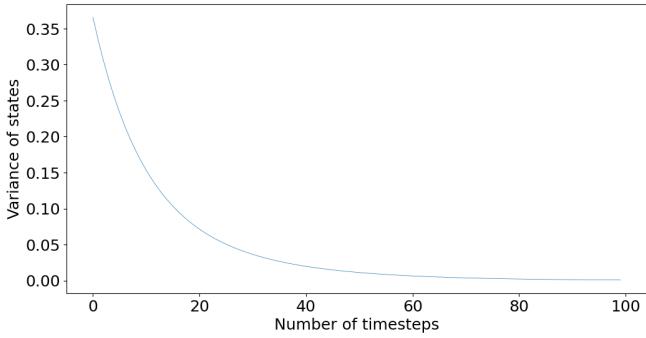


Figure 11: Variance in Reservoir 1

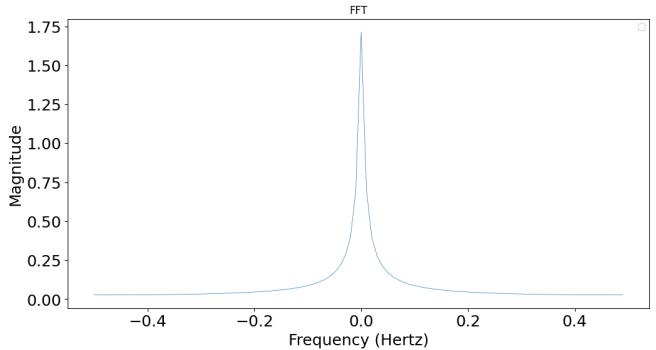


Figure 12: FFT of Reservoir 1

network and the variance, respectively converge to 0. While the FFT in Figure 12 shows a well defined peak at zero, indicating approaching to the equilibrium point, a orbit with infinite period and 0 frequency.

Reservoir 2 is defined with $\rho = 1.5$ and $\alpha = 0.05$ investigated over $T = 1000$ iterations. Notice that I investigated this network with 10 neurons in previous chapter (Figure 5). But after increasing N to 100 in reservoir 2 neurons start to behave irregularly without a well defined period (Figure 13). But in terms of mean field (Figure 14) reservoir 2 seems to behave regularly. This is further demonstrated by oscillating Pearson correlation in Figure 15 and well defined peak in Figure 16.

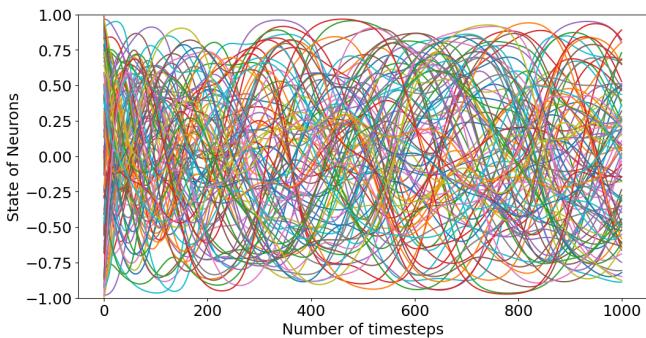


Figure 13: States in Reservoir 2

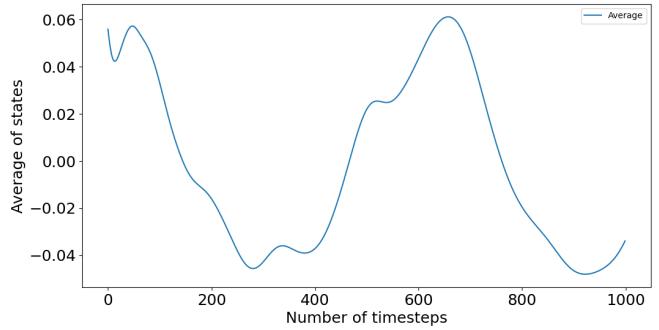


Figure 14: Mean field in Reservoir 2

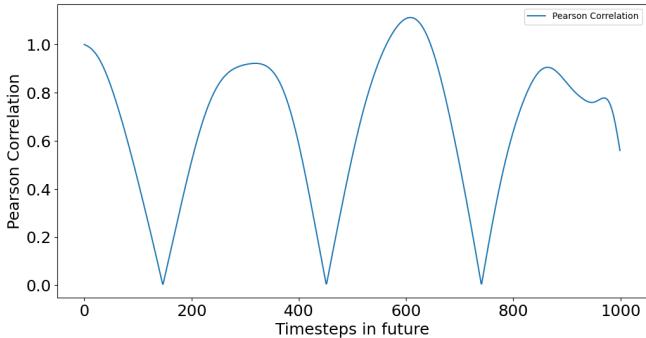


Figure 15: Pearson Correlation in Reservoir 2

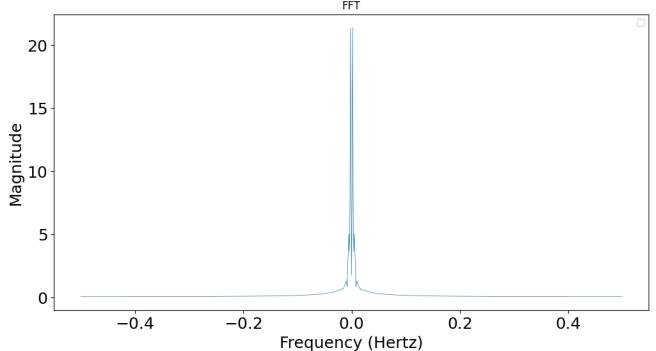


Figure 16: FFT of Reservoir 2

Reservoir 3 is defined with $\rho = 1.5$ and $\alpha = 0.6$. The mean field of the network in Figures 17 shows that overall network oscillates. In Figure 18 Pearson correlation oscillates and decays slowly,

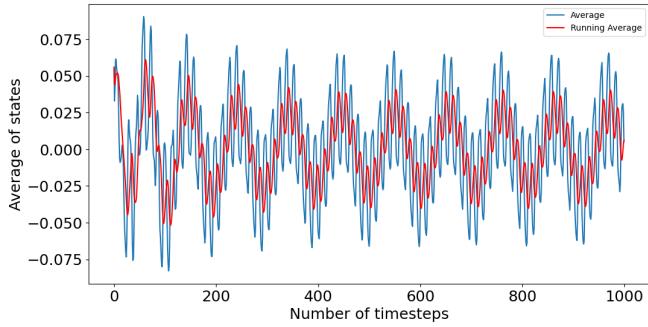


Figure 17: Mean field (blue) and its running average (red) in Reservoir 3

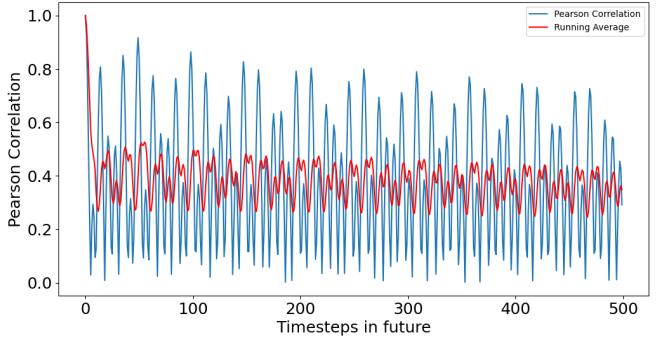


Figure 18: Pearson correlation (blue) and its running average (red) in Reservoir 3

while spectra and it plotted logarithmically in Figures 19 and 20 shows several well-defined peaks alongside finite but small amplitude broadband, both indicators of as nearly-irregular behavior.

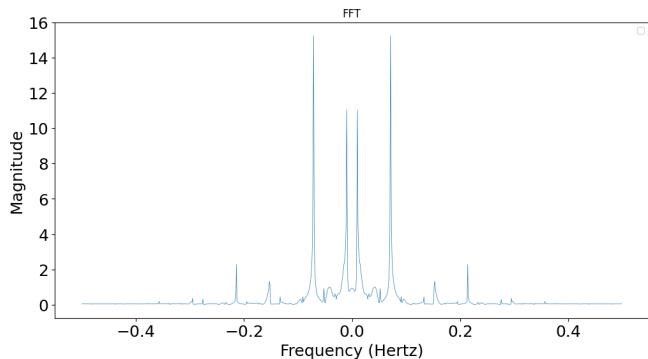


Figure 19: FFT of Reservoir 3

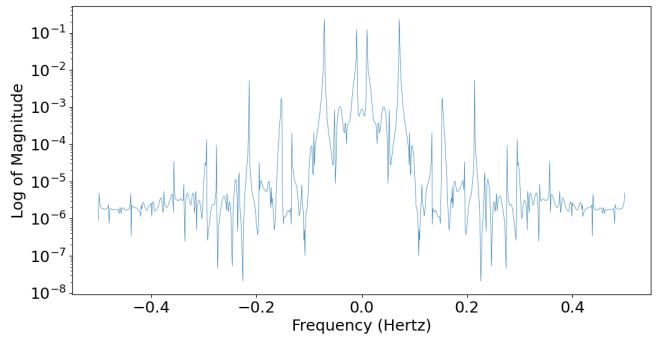


Figure 20: FFT of Reservoir 4 plotted logarithmically

Reservoir 4 is defined with $\rho = 1.5$ and $\alpha = 1.1$. In this network, neurons are behaving irregularly (Figure 21) in fact so strongly that they jump out of $[-1,1]$ (range of attractor). After

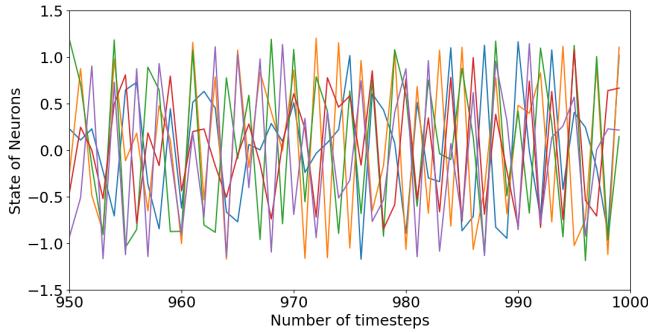


Figure 21: 10 neurons in different colors in Reservoir 4

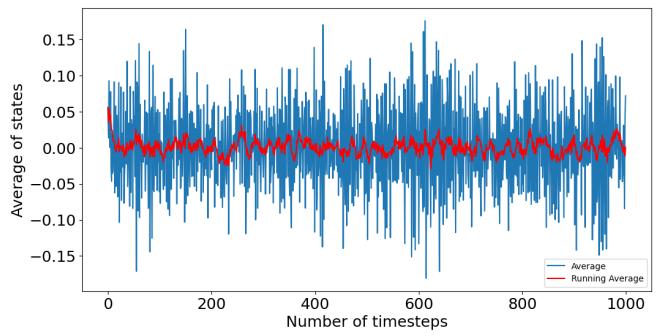


Figure 22: Mean field (blue) with its running average (red) in Reservoir 4

further investigation in the next subsection, I notice that increasing α leads to increases in the

size of the attractor. The mean field oscillates similarly to what I would obtain for a signal that is chaotic (Figure 22), Pearson correlation quickly decay (Figure 23) and FFT is broadband (Figure 24), all features of irregular behavior.

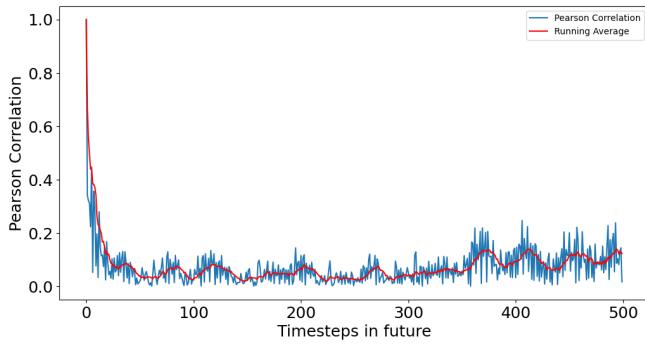


Figure 23: Pearson Correlation of Reservoir 4

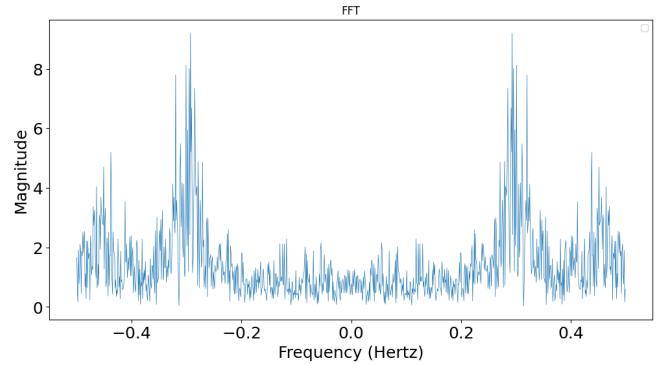


Figure 24: FFT of Reservoir 4

Reservoir 5 is defined with $\rho = 2$ and $\alpha = 2$. From Figure 25 I can see that neurons are oscillating rapidly. Furthermore, mean field (Figure 26) shows that reservoir 5 diverges.

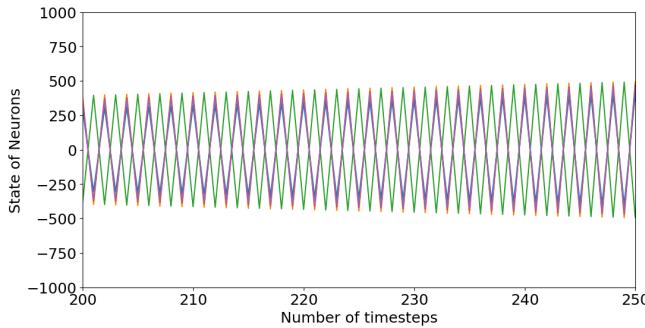


Figure 25: 5 neurons in different colors in Reservoir 5

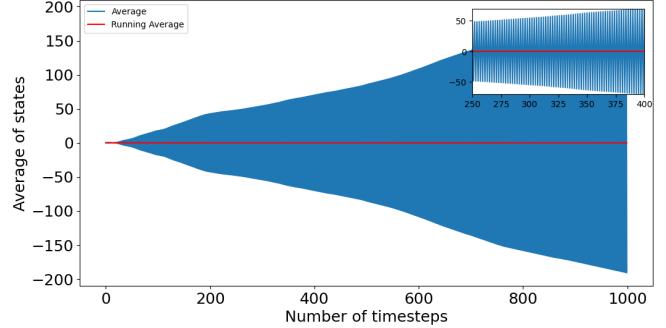


Figure 26: Mean field (blue) with running average (red) in Reservoir 5

Conclusions: After investigating various reservoirs I can make the following preliminary considerations regarding the relationship between the hyper-parameters ρ and α and the behavior of the reservoir. Notice however that more rigorous analysis will be further made to refine, demonstrate or amend these following statements:

- If $\alpha \Rightarrow 2$ and $\rho > 1$ then reservoir diverges to infinity.
- If $1 < \alpha < 2$ and $\rho > 1$ then reservoir exhibits Irregular behaviour.
- If $\alpha < 1$ and $\rho > 1$ then reservoir exhibits Nearly-Irregular behaviour.
- If $\alpha < 1$ and $\rho < 1$ then reservoir converges to 0 exhibiting Regular behavior.
- If $1 < \alpha < 2$ and $\rho < 1$ then reservoir exhibits all behaviors - Regular, Nearly-Irregular, Irregular - except for diverging to infinity. The behavior depends on the values of ρ and α .

4.3 Finding Irregular Behaviour

In this subsection, I will refine the considerations done in the previous subsection, but in terms of more rigorous statements. My further analysis to characterise behavior in a reservoir will be based on 3 custom metrics made from Pearson correlation and FFT spectra:

- Number of spikes above a certain threshold in FFT spectrum
- Integral of FFT spectrum above a certain threshold
- Integral of Pearson correlation curve

Spike number and Integral in FFT spectrum: I have decided to extract a number of spikes above a threshold in the magnitude as this allows to count the number of well defined peaks in FFT spectrum. If the number of peaks is small (for example 0 or 1) this can indicate regular behavior, while moderately large numbers (for example numbers from 2 to 10) can indicate nearly-irregular behavior. Much larger number of spikes (for example more than 10) indicating irregular behavior. To detect a broadband spectrum the integral of FFT above threshold is used. While the FFT integral does not differentiate well between regular and nearly-irregular behavior, it can well characterize a broadband spectrum, with large values indicating irregular behavior. Figure 27 illustrates a broadband FFT spectrum where a large number of peaks can be identified above the threshold line (in red).

Integral of Pearson Correlation curve: Integral of Pearson correlation under running average can characterize regular behavior by small numbers, nearly-irregular behavior by large values as oscillates in time and does not decay in time, and irregular by an integral with moderate values as the correlation rapidly decays, but oscillates near 0. Figure 28 shows the correlation curve for a broadband mean field. It quickly decays with a further oscillatory behavior for large iterations. This suggests that there is always a kind of regular component in the network.

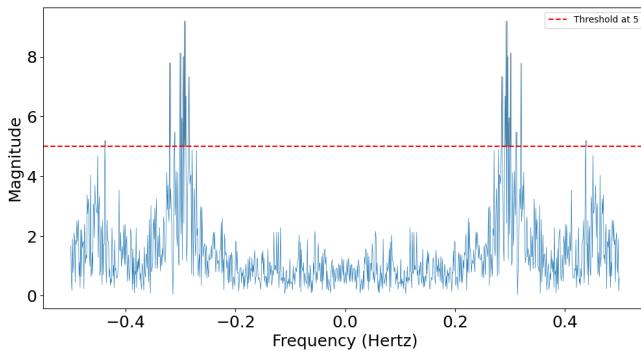


Figure 27: FFT spectrum number of peaks is 36, integral is ≈ 0.039 when the threshold is of magnitude 5 indicated by red line in the reservoir with $\rho = 1.5$, $\alpha = 1.1$ and $N = 100$.

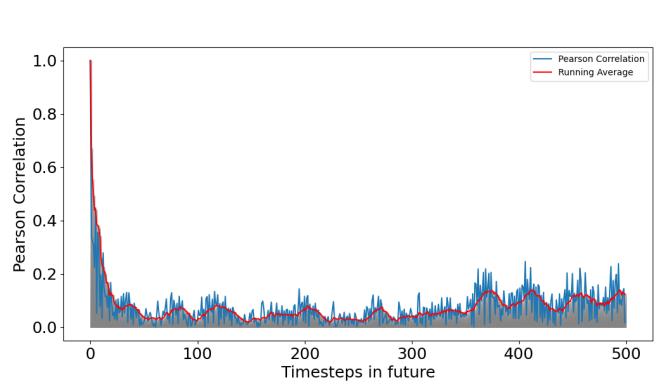


Figure 28: Integral of Pearson correlation curve under running average is ≈ 35 in Reservoir with $\rho = 1.5$, $\alpha = 1.1$ and $N = 100$

Heatmap of spikes in Figure 29 indicates that networks set with hyperparameters which are colored in deep purple behaves regularly and light blue behaves irregularly. This allows to refine the statements in subsection 4.2:

- If $\alpha \Rightarrow 2$ and $\rho > 1$ then reservoir diverges to infinity is indeed the case. Furthermore, light blue color when $\alpha \Rightarrow 2$, shows that the reservoir diverges to infinity no matter what its spectral radius ρ .
- If $1 < \alpha < 2$ and $\rho > 1$ then reservoir exhibits irregular behavior, which indeed seems the case looking at light blue.
- If $\alpha < 1$ and $\rho > 1$ then reservoir exhibits nearly-irregular behavior might be incorrect, due to the reason that the number of spikes significantly increases shown in navy color when $\rho \geq 2$ which indicate irregular behavior.
- If $\alpha < 1$ and $\rho < 1$ then reservoir exhibit regular behavior shown by purple color.
- Statement if $1 < \alpha < 2$ and $\rho < 1$ then reservoir exhibits all behaviors must be correct as that does seem the case as all colors can be seen in that area.

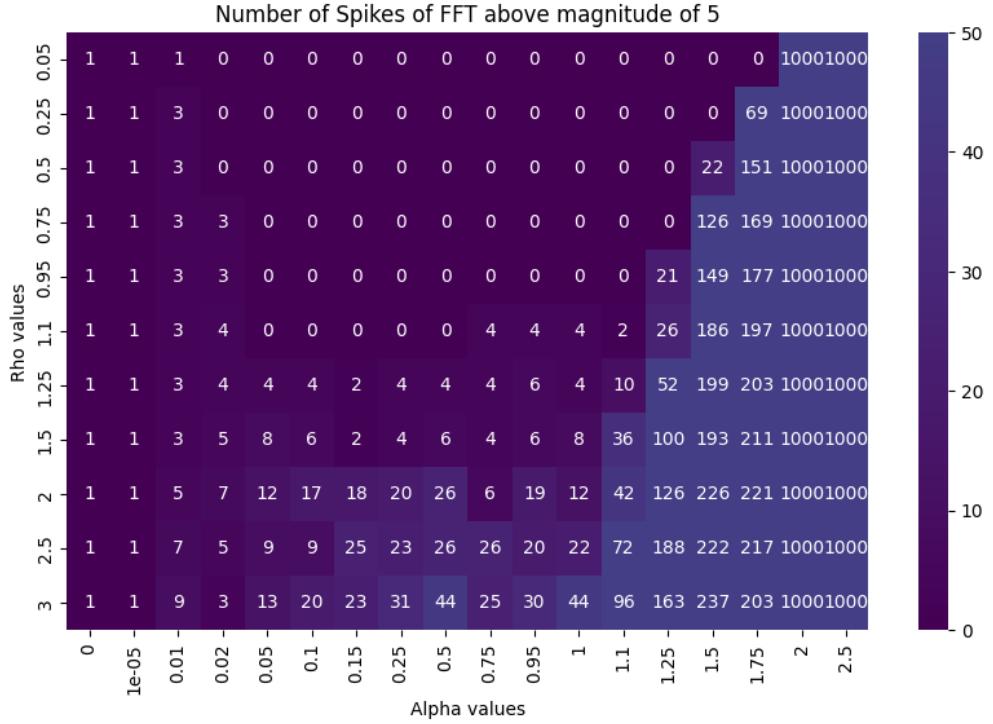


Figure 29: Heatmap of spikes above magnitude 5 in reservoirs constructed with different spectral radius and coupling values. Deep purple indicates small while light blue indicates big number of spikes.

Heatmap integral of FFT spectrum above the threshold in Figure 30 indicates that networks set with hyperparameters which are colored in deep purple indicates regular behavior, light blue produces nearly-irregular behavior and yellow indicates irregular behavior. This allows to adjust statements in subsection 4.2 with conclusions from the results shown in Figure 29:

- Statement that if $\alpha \Rightarrow 2$ then reservoir diverges to infinity is supported by this heatmap.
- Statement that if $1 < \alpha < 2$ and $\rho > 1$ then reservoir exhibits irregular behavior, is not entirely correct. If I look at the reservoir with $\rho = 1.1$, $\alpha = 1.1$ I can see that integral is zero. Notice that the value of the integral gradually increases (from blue to yellow) as α is increased from 1 to 2 and ρ is decreased. Thus indicating that when $1 < \alpha < 2$, reservoirs exhibit all behaviors indicated by negative dependence of ρ and α .
- Statement that if $\alpha < 1$ and $2 > \rho > 1$ then reservoir exhibits nearly-irregular behavior supported by the heatmap.
- Statement that if $\alpha < 1$ and $\rho > 2$ then reservoir exhibits irregular behavior is not entirely correct. The change between nearly-irregular and irregular seems to happen in the boundary between the blue and yellow colors, or when the network behavior changes from nearly-irregular to irregular behavior with most of irregular behavior concentrated around the reservoir with $\rho = 3$ and $\alpha = 0.25$.
- Statement that if $\alpha < 1$ and $\rho < 1$ then the reservoir exhibits regular behavior is supported by this heatmap.

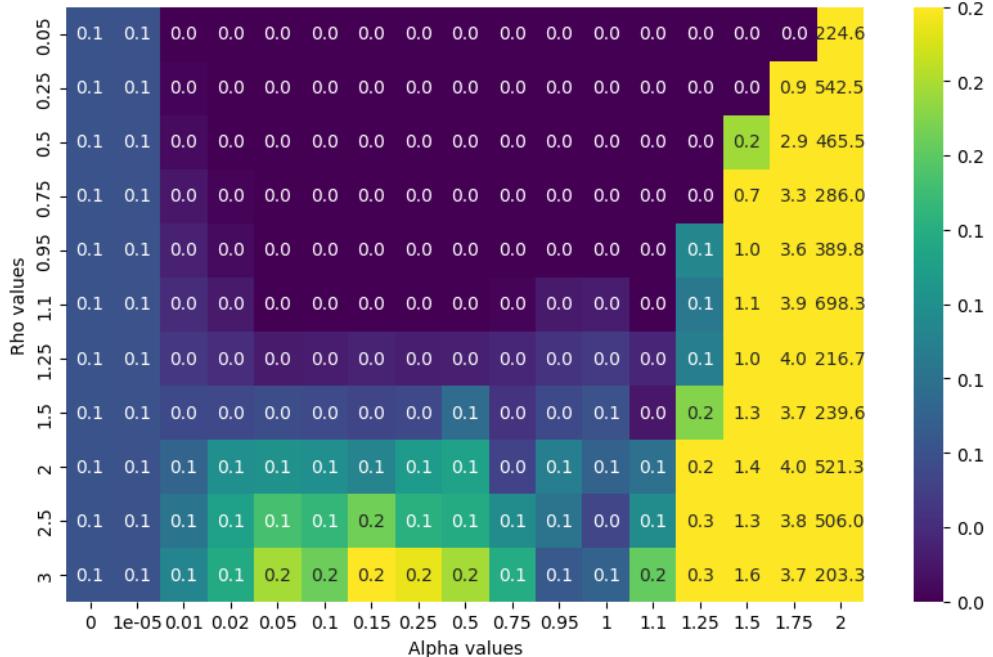


Figure 30: Heatmap integral of FFT power spectrum above magnitude 5 color-coded in reservoirs constructed with different spectral radius and coupling values. Deep purple indicates small while yellow indicates big integral of the FFT spectrum.

Heatmap of integral of Pearson correlation curve in Figure 31 indicates that network set with hyperparameters which are colored in yellow and green produce nearly-irregular behavior while blueish green indicates either irregular or regular behaviors (For this it is important to look at numbers). More conclusions can be drawn from the results of this figure, further refining the statements in subsection 4.2.

- Statement that if $\alpha \Rightarrow 2$ then reservoir diverges to infinity is unsupported by this heatmap.
- Statement that when $1 < \alpha < 2$, reservoirs exhibit all behaviors indicated by negative dependence of ρ and α is supported by this heatmap
- Statement that if $\alpha < 1$ and $2 > \rho > 1$ then reservoir exhibits nearly-irregular behavior is strongly supported by this heatmap by yellow and yellow-greenish colors.
- Statement that if $\alpha < 1$ and $\rho > 2$ then network behavior change between nearly-irregular and irregular seems to happen in the boundary between the blue and yellow colors, or when the network behavior changes from nearly-irregular to irregular behavior with most of irregular behavior is supported by the heatmap
- Statement that if $\alpha < 1$ and $\rho < 1$ then reservoir exhibits regular behavior is supported by this diagram.

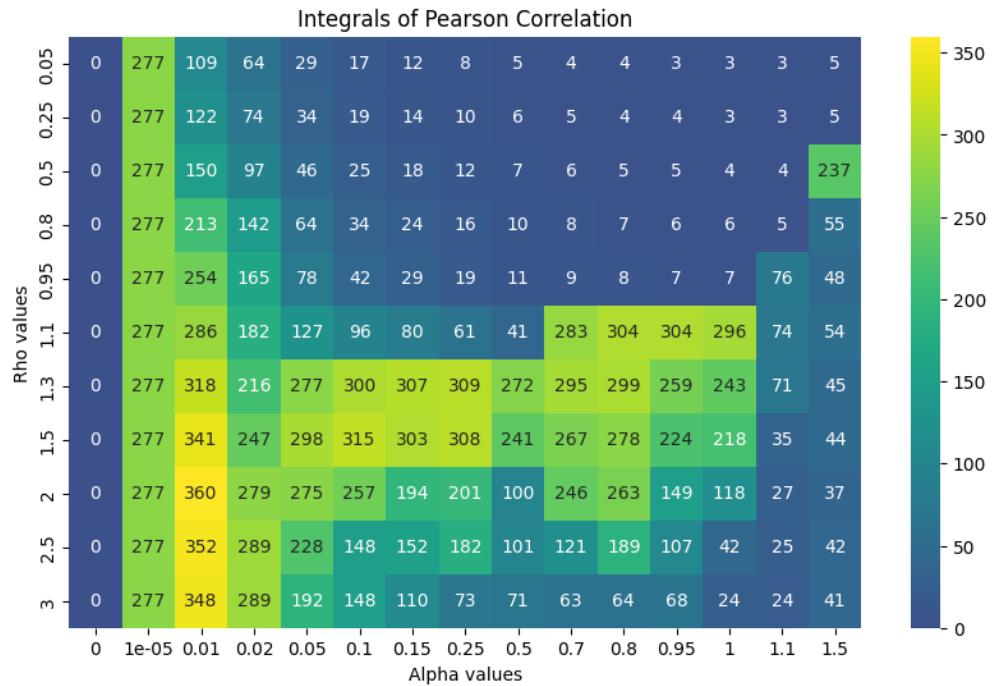


Figure 31: Heatmap of the integral of Pearson correlation curve color-coded in reservoirs constructed with different spectral radius ρ and coupling values α . Yellow indicates large and blue indicates small values for the integral of the Pearson correlation curve

The effect of sparsity in the network and the number of neurons in the network behavior can be seen in Figures from 98 to 101 with same metrics and can be found in Appendix A.

5 Prediction

Throughout this chapter, the network is constructed as described in Chapter 2. During this chapter, I will be using reservoirs connected chemically as described by Equation (8) where F is an identity map with $\rho = 0.8$, $\alpha = 0.8$, and $N = 100$, unless otherwise specified. I call thus constructed network **Reservoir 6**. Using the previous statement constructed in chapter 4 I know this network behaves regularly. Input, training and test data for this chapter will be provided by the Lorenz system described by Equation (26) and integrated by RK4 (27) with integration step $h = 0.01$. In the following figures, trajectories being generated by the reservoir to predict the Lorenz system will be shown in blue, and the original trajectories of the Lorenz system, will be shown in green.

This chapter will start with an investigation of the input vector on predictions and analysis of performance to predict future for a time interval τ in the future. Followed by an investigation on how reservoirs set with different hyperparameters perform. The chapter will conclude with a looped network investigation.

5.1 Network predictions

Prediction $\tau = 1$ and $\vec{u}_n = [x; y; z]$: First let me predict Lorenz system for the next step i.e. $\tau = 1$ using reservoir 6 with input vector $\vec{u}_n = [x; y; z]$ where x, y, z are Lorenz variables. Prediction by reservoir 6 for the next step ($\tau = 1$) has only neglectable error as can be seen in Figures 32, 33, 34 and 35.

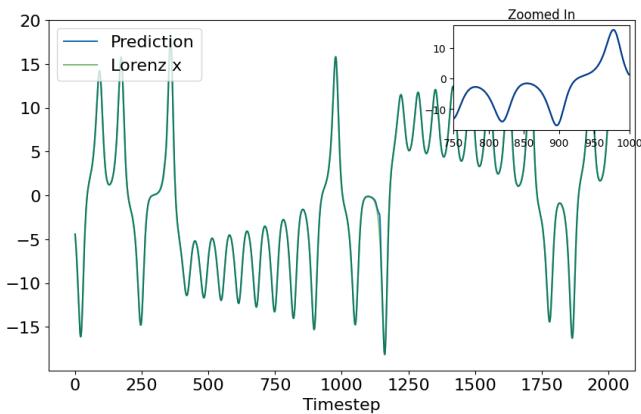


Figure 32: Prediction of x by reservoir 6 ($\tau = 1$, $\vec{u}_n = [x; y; z]$) in blue and integrated by RK4 in green.

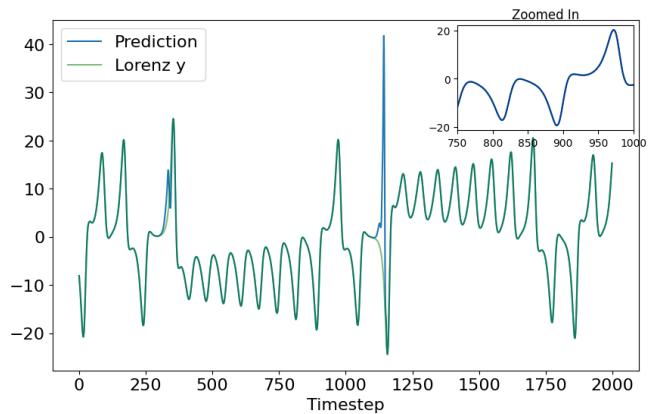


Figure 33: Prediction of y by reservoir 6 ($\tau = 1$, $\vec{u}_n = [x; y; z]$) in blue and integrated by RK4 in green.

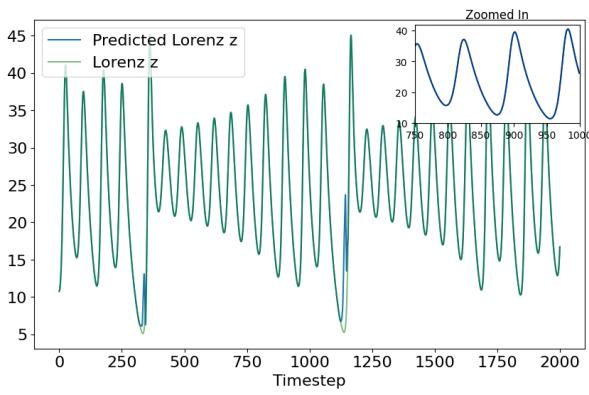


Figure 34: Prediction of z by reservoir 6 ($\tau = 1$, $\vec{u}_n = [x; y; z]$) in blue and integrated by RK4 in green.

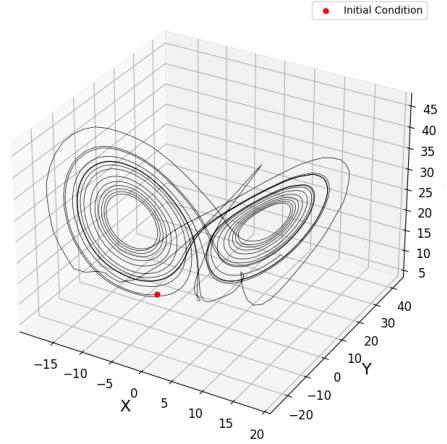


Figure 35: Prediction of Lorenz system by reservoir 6 ($\tau = 1$, $\vec{u}_n = [x; y; z]$).

Divergent jumps: However, notice there is a noticeable inaccuracy happening at around 1150 timestep in Figure 33.

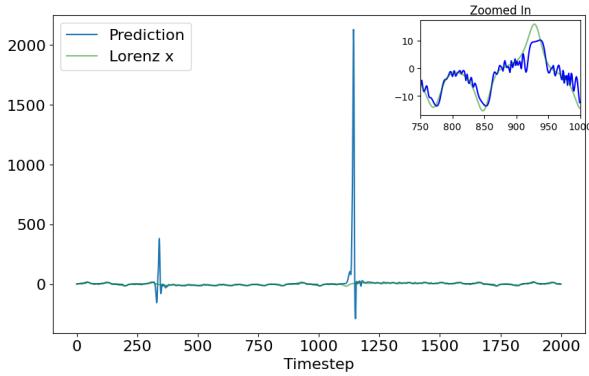


Figure 36: Prediction of x by reservoir 6 ($\tau = 50$, $\vec{u}_n = [x; y; z]$) in blue and integrated by RK4 in green.

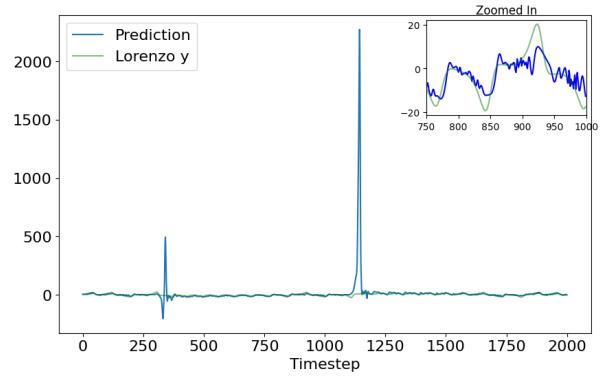


Figure 37: Prediction of y by reservoir 6 ($\tau = 50$, $\vec{u}_n = [x; y; z]$) in blue and integrated by RK4 in green.

The network jumps out of the attractor but rapidly comes back. This can also be seen in Figures 36, 37, 38 and 39. Nevertheless, overall the reservoir computing approach predicts reasonably well as can be seen from zoomed-in areas of noted Figures.

The jumps happen close to $x=y=0$, at the low values for z . I suppose that this inaccuracy in prediction at this point happens because this is a critical point in the Lorenz system. This system has a symmetry over the inversion through the z axis, that is $x = -x$ and $y = -y$. The $x = y = z = 0$ is an equilibrium point, and it is a critical one, in the sense that from it 2 homoclinic orbits depart, one going to positive side of x and another to the negative side. This is the most nonlinear region of the Lorenz system, and it is not unexpected that the reservoir is not able to accurately predict the future steps for trajectories near this critical point. Basically, arbitrarily small perturbation around this point can take the trajectory in totally different directions [38].

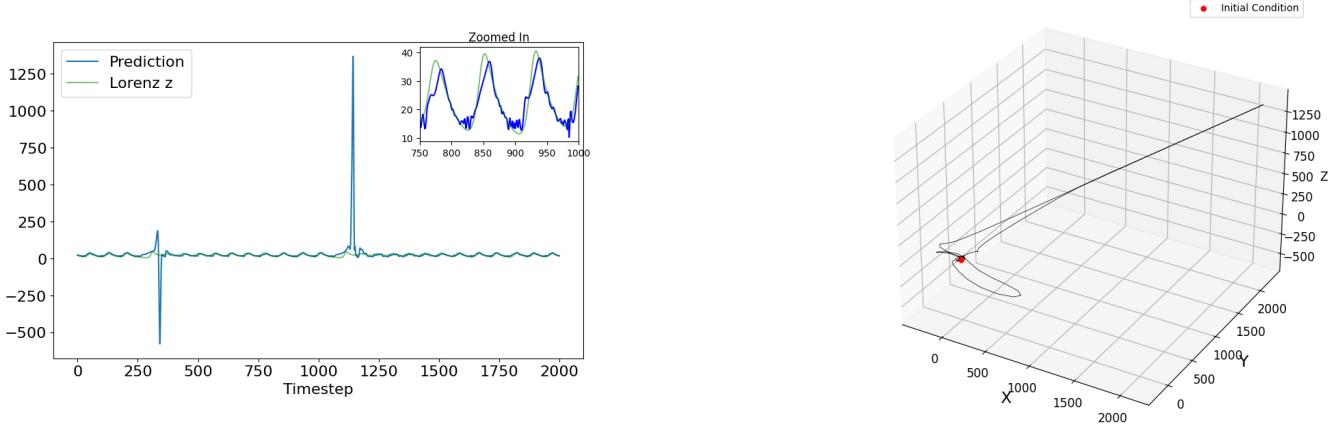


Figure 38: Prediction of z by reservoir 6 ($\tau = 50$, $\vec{u}_n = [x; y; z]$) in blue and integrated by RK4 in green.

Figure 39: Prediction of Lorenz system by reservoir 6 ($\tau = 50$, $\vec{u}_n = [x; y; z]$).

Influence of input vector \vec{u}_n : Divergent jumps pose a problem in getting accurate results. After further experimentation, I noticed that this error can be prevented by avoiding using the input z into the network. Using input vector \vec{u}_n composed of only x and y I managed to get rid of the divergent jump error as this can be seen in Figures 40, 41, 42 and 43.

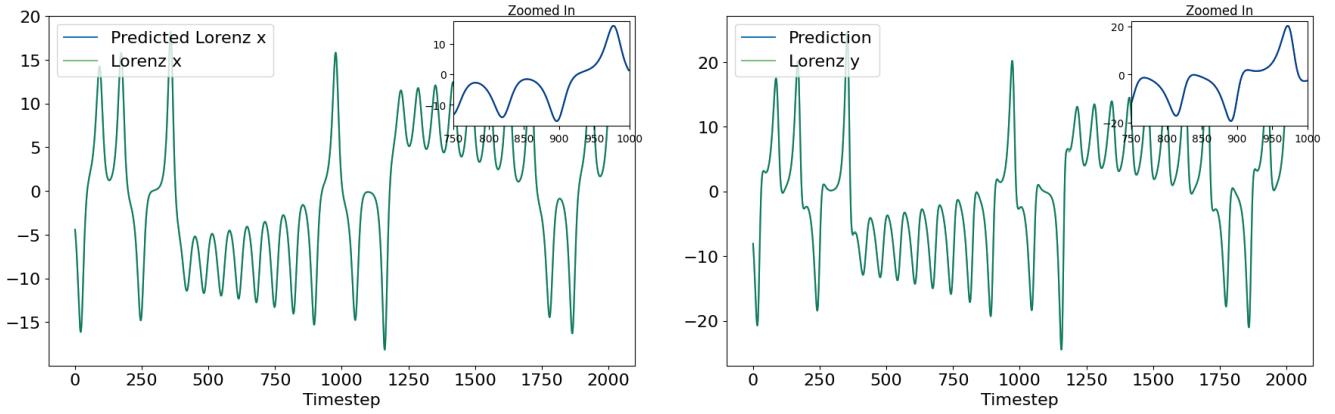


Figure 40: Prediction of x by Reservoir 6 ($\tau = 1$, $\vec{u}_n = [x; y]$) in blue and integrated by RK4 in green.

Figure 41: Prediction of y by Reservoir 6 ($\tau = 1$, $\vec{u}_n = [x; y]$) in blue and integrated by RK4 in green

Notice that, the network now predicts without divergent jump error, but it comes with a cost of accuracy in predicting z as can be seen in comparing Figures: 42 with 34 and 43 with 35. Nevertheless, if I increase the number of steps I look into the future (τ) and investigate predictions then I can deduce that on the whole, the network predicts better with less input. This can be seen by comparing predictions to previous results.

The Lorenz is least observable by the z variable. Which means that reconstructions of the Lorenz system by only the z variable do not reproduce the topology of the attractor [39]. Whereas discussions about observability are out of the scope of this thesis, the z variable is not a good variable to be used to predict the state of the other variables of the Lorenz system, and as such, eliminating it from the input vector has had the effect of improving predictions.

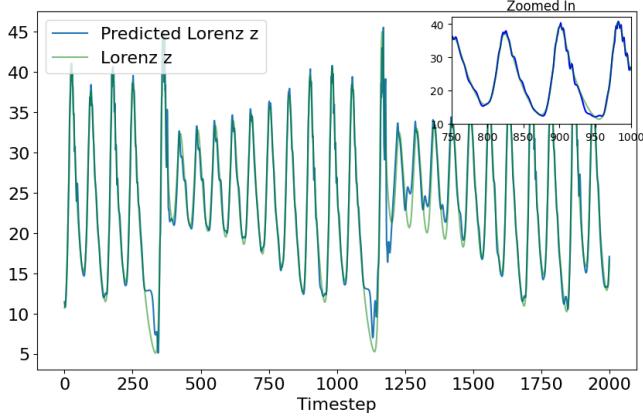


Figure 42: Prediction of z by Reservoir 6 ($\tau = 1$, $\vec{u}_n = [x; y]$) in blue and integrated by RK4 in green.

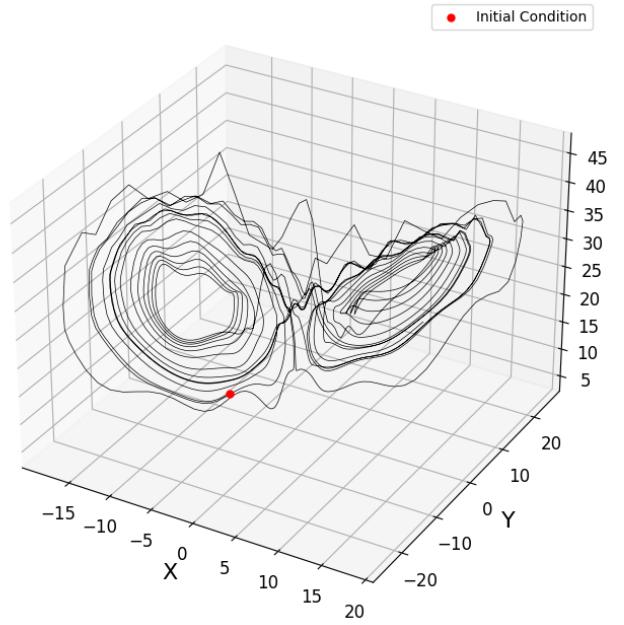


Figure 43: Prediction of Lorenz system by Reservoir 6 ($\tau = 1$, $\vec{u}_n = [x; y]$).

For example, reservoir 6 with 2-dimensional input predicting y for $\tau = 50$ steps in the future as in Figure 45 has less error than the same network with 3-dimensional input as in Figure 37. On another note, the reservoir predicts $\tau = 1$ (Figure 40) and $\tau = 50$ (Figure 44) reasonably well compared to $\tau = 100$ (Figure 47).

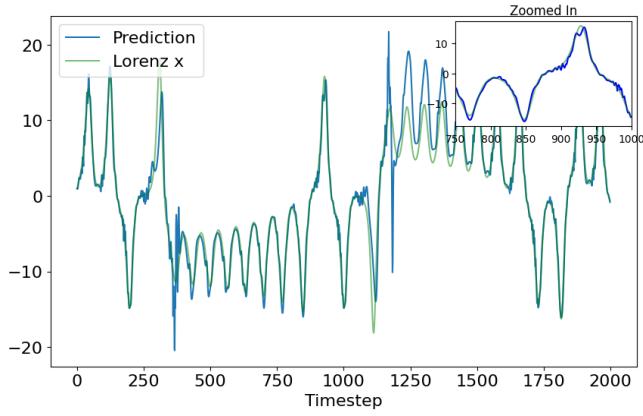


Figure 44: Prediction of x by reservoir 6 ($\tau = 50$, $\vec{u}_n = [x; y]$) in blue and integrated by RK4 in green

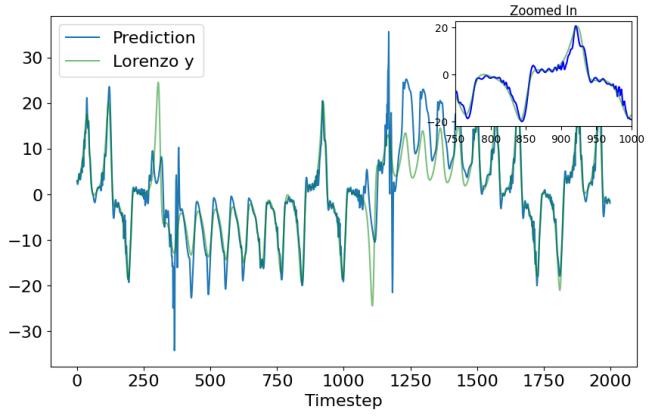


Figure 45: Prediction of y by reservoir 6 ($\tau = 50$, $\vec{u}_n = [x; y]$) in blue and integrated by RK4 in green

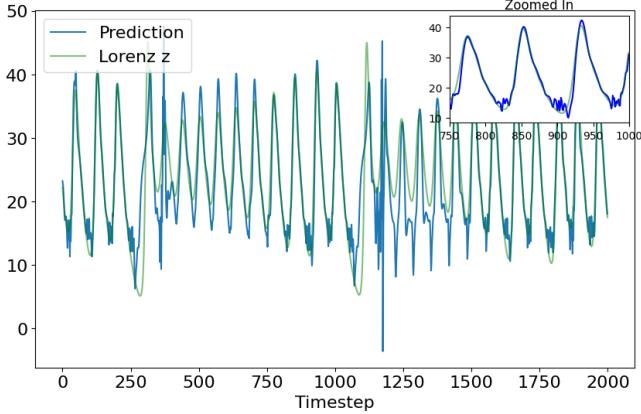


Figure 46: Prediction of z by reservoir 6 ($\tau = 50$, $\vec{u}_n = [x; y]$) in blue and integrated by RK4 in green

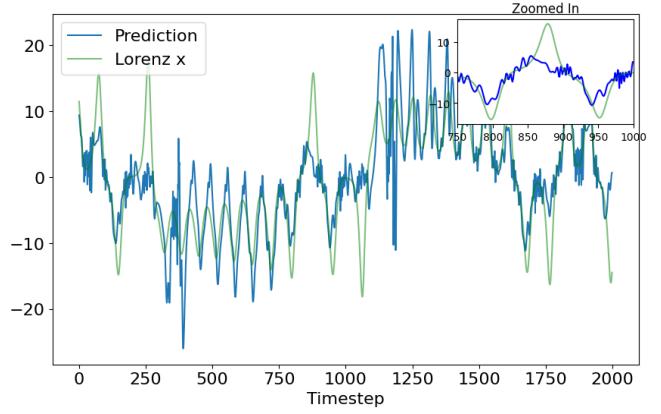


Figure 47: Prediction of x by reservoir 6 ($\tau = 100$, $\vec{u}_n = [x; y]$) in blue and integrated by RK4 in green

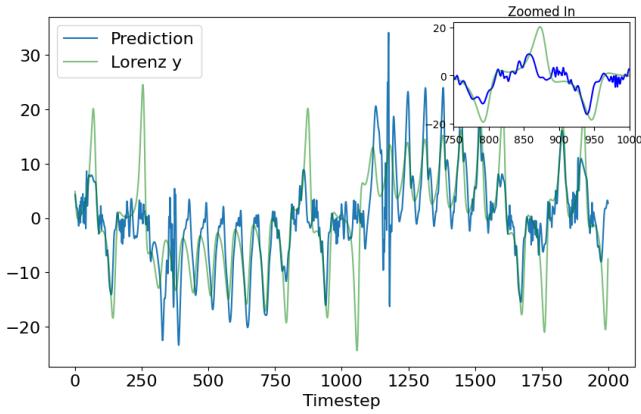


Figure 48: Prediction of y by reservoir 6 ($\tau = 100$, $\vec{u}_n = [x; y]$) in blue and integrated by RK4 in green

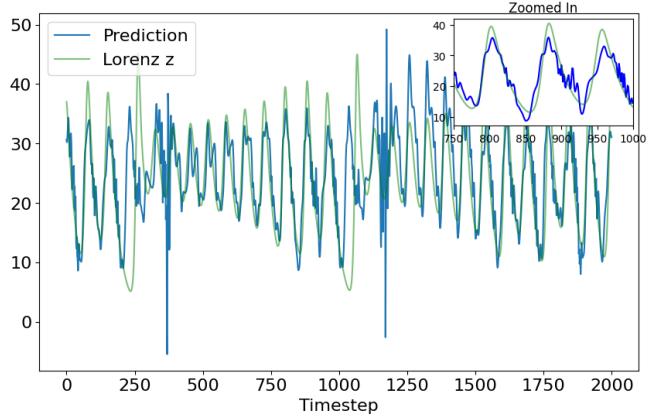


Figure 49: Prediction of z by reservoir 6 ($\tau = 100$, $\vec{u}_n = [x; y]$) in blue and integrated by RK4 in green

5.2 Error

In the last subsection, I investigated predictions loosely using visual inspection. In this subsection, I will investigate results analytically using root mean squared error (RMSE) Equation (16):

- Reservoir 6 predicts Lorenz system better with 2-dimensional input vector $\vec{u}_n = [x; y]$ instead of 3-dimensional one $\vec{u}_n = [x; y; z]$. This is evident in Figure 51 error reaches at most 6 while in Figure 50 error reaches 20 at $\tau = 20$.
- Predicting over $\tau = 50$ steps in the feature of Lorenz system with Reservoir 7 is not accurate (Figure 51).

Notice that the diagrams presented here are based on predictions of Lorenz's x-axis only. Nevertheless, conclusions are the same after further investigation into other predicted axes, which I will ignore here for the sake of continuity.

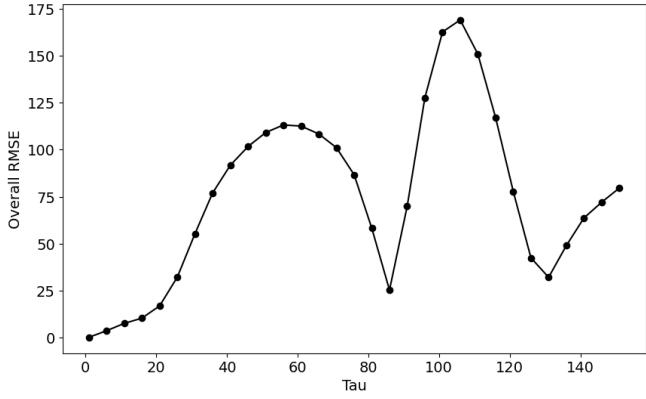


Figure 50: RMSE for prediction of Lorenz x-axis by reservoir 6 ($\vec{u}_n = [x; y; z]$) with respect to τ

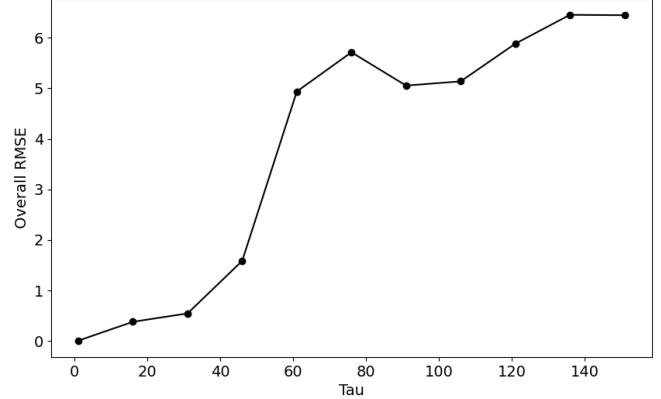


Figure 51: RMSE for prediction of Lorenz x-axis by reservoir 6 ($\vec{u}_n = [x; y]$) with respect to τ

RMSE Heat map rho-alpha Here I investigate heatmap for reservoirs constructed by different coupling α and spectral radius ρ and used to predict Lorenz x-axis and evaluated by RMSE in Figure 52. In this diagram yellow means high error and dark navy low error. Dark navy region in the heatmap corresponds to reservoirs constructed with roughly $\alpha \in [0.05, 0.95]$ and $\rho \in [0, 2]$. These reservoirs are all either regular behavior or nearly-irregular behavior by the statements from previous chapter and confirmed via heatmaps in Figures 29, 30 and 31. A similar trend can be seen when τ is increased or decreased (Appendix A Figures 103 and 104). Thus to conclude the reservoirs predict the best when they behave regularly or nearly-irregularly.

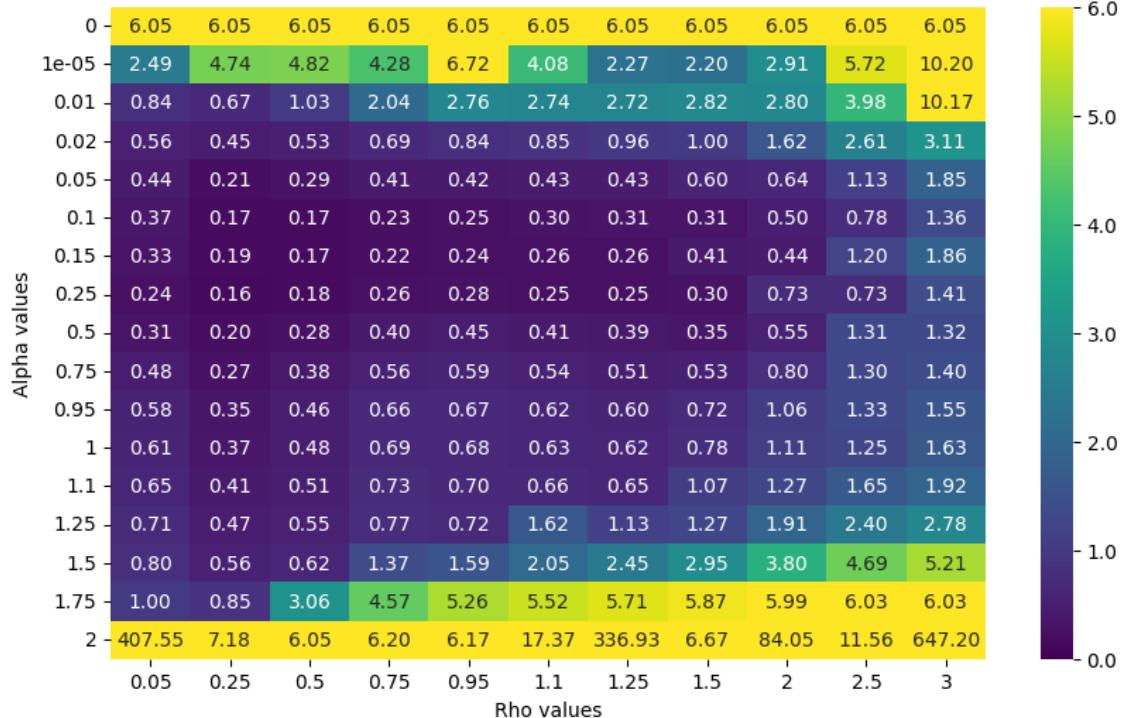


Figure 52: Heatmap over networks constructed by different coupling α and spectral radius ρ and used to predict Lorenz x-axis $\tau = 25$ timesteps into future and colored by RMSE.

RMSE Heat map rho-N: Here I investigate heatmap for reservoirs constructed by different spectral radius ρ and number of neurons in reservoir N . The main trend is that increasing the number of neurons allows networks to predict better which can be seen from dark navy color regions corresponding to low RMSE in the heatmap Figure 53. Also, note that the best prediction is around $\rho = 1.1$ and $N = 250$ which supports T. Caroll statement [11] that the reservoir computing approach predicts the best at "edge of chaos".

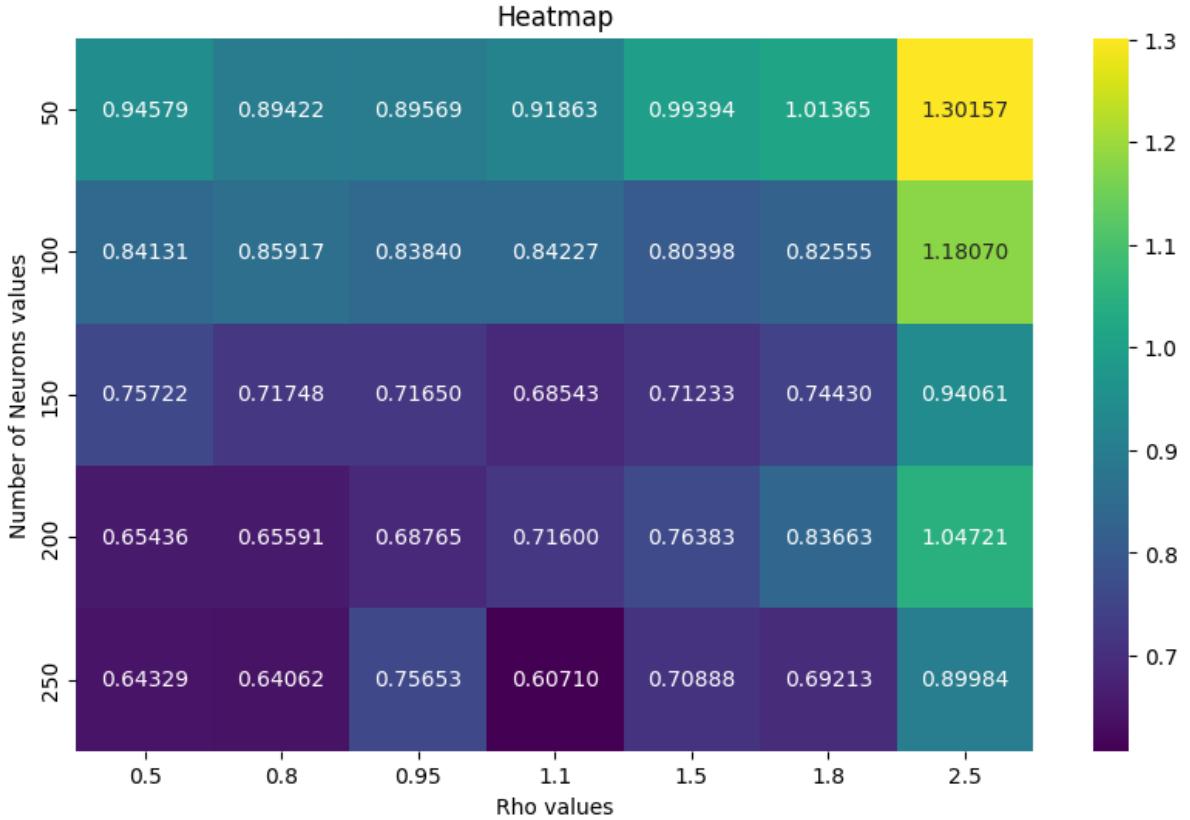


Figure 53: Heatmap over networks constructed by different number of neurons N and spectral radius ρ and used to predict Lorenz x-axis $\tau = 25$ timesteps into future and colored by RMSE.

5.3 Looped Network

How come I am predicting Lorenz system so well? The sensitivity to initial conditions means that with chaotic systems, it is impossible to make firm long-term predictions, because you can never know exactly, to the infinite decimal point the state of the system [40]. Why then my predictions seems to predict remarkably well? As mentioned in Chapter 2, I input Lorenz system into the network at each timestep and then based on that predict τ steps into the future (this depends on what the model was trained for). Predictions are not fed back as input signals into the reservoir, as the trajectory of the Lorenz integrated is the one serving as the input to predict the next step based on that again. In this subsection, I will address this by looping the reservoir to use its own predictions to predict the next step. Notice that errors will accumulate over time because the reservoir will not receiving a correct input signal to predict a target signal, but an onput signal generated by a previous prediction, thus the error will reach the size of the attractor quite fast.

Predictions in Figures 54 and 55 show reservoir 6 with $T_t = 5000$ iterations of training predicts reasonably well for around 200 timesteps. At timestep 250 the network reaches an error with the size of the attractor and around timestep 300 gives up on predicting and oscillates through the whole network forming a stable orbit. (In real time this can be calculated to be 2, 2.5 and 3). I hypothesise that after increasing training time (T_t) the reservoir will perform better. From Figures 56 and 57 I notice similar behavior from reservoir 6 with more training time. The only noticeable difference is that network predictions resemble Lorenz x variable and do not settle down on periodic orbit as time increases. If I further increase training time, the reservoir reaches the error of the whole attractor at only 125 timestep (This can be seen from Figures 58 and 59) hinting that this will not help in predicting.

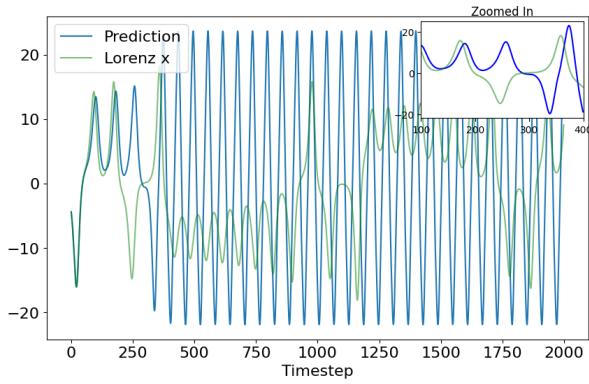


Figure 54: Prediction of Lorenz x-axis by looped reservoir 6 ($\vec{u}_n = [x; y]$, $T_t = 5000$) in blue and integrated by RK4 in green.

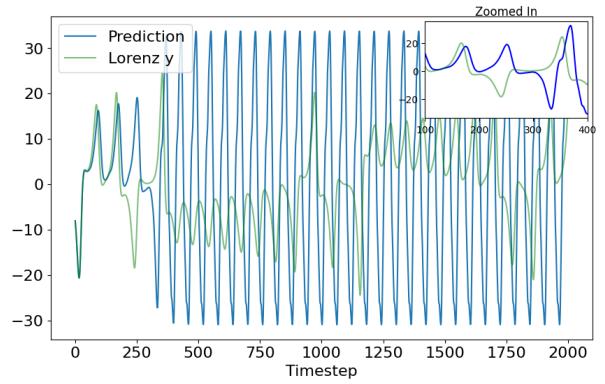


Figure 55: Prediction of Lorenz y-axis by looped reservoir 6 ($\vec{u}_n = [x; y]$, $T_t = 5000$) in blue and integrated by RK4 in green.

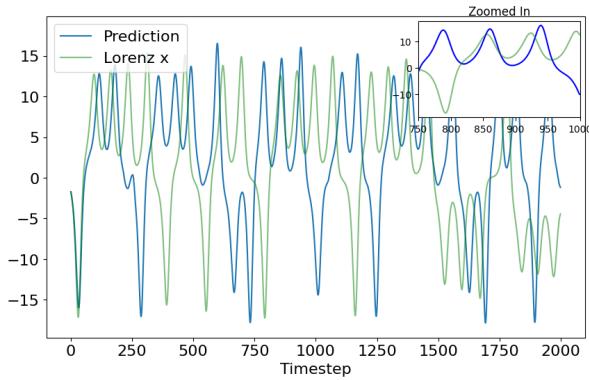


Figure 56: Prediction of Lorenz x-axis by looped reservoir 6 ($\vec{u}_n = [x; y]$, $T_t = 15000$).

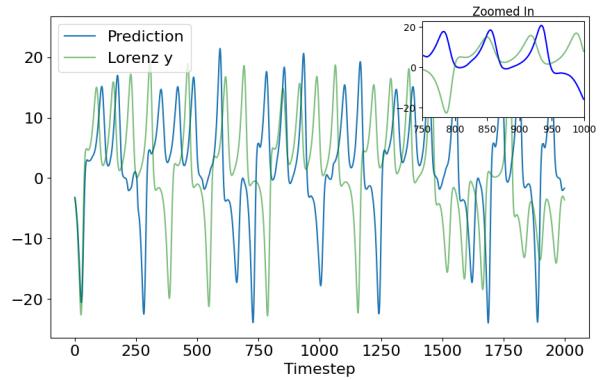


Figure 57: Prediction of Lorenz y-axis by looped reservoir 6 ($\vec{u}_n = [x; y]$, $T_t = 15000$).

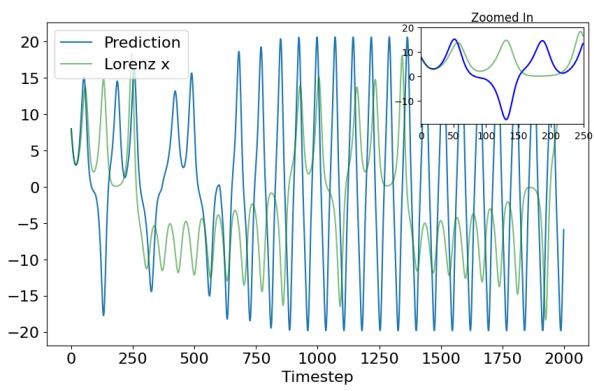


Figure 58: Prediction of Lorenz x-axis by looped reservoir 6 ($\vec{u}_n = [x; y]$, $T_t = 45000$).

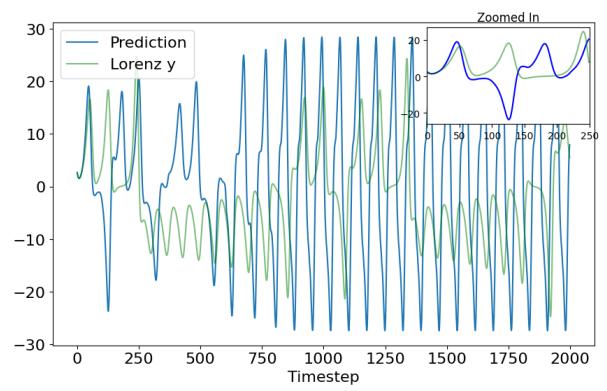


Figure 59: Prediction of Lorenz y-axis by looped reservoir 6 ($\vec{u}_n = [x; y]$, $T_t = 45000$).

6 Collective Almost Synchronization (CAS)

In this Chapter, I try to find CAS by varying coupling strength (α) and spectral radius (ρ). As I discussed in Chapter 3 recall that neurons with small connectivity behaving irregularly and neurons with high connectivity behaving regularly is an indication of CAS phenomena. Also, neurons with similar connectivity behaving similarly is an indication of CAS as well. In subsection 6.1 I investigate bifurcation diagrams of different neurons to find specifications of said hyper-parameters that might indicate CAS. While in subsection 6.2 I investigate these potential areas by looking at similar neurons and their orbits.

Before plunging into search for CAS I believe it is important to investigate connectivity of neurons. Figures 60 and 61 show degrees and weighted degrees respectively, both of whom are

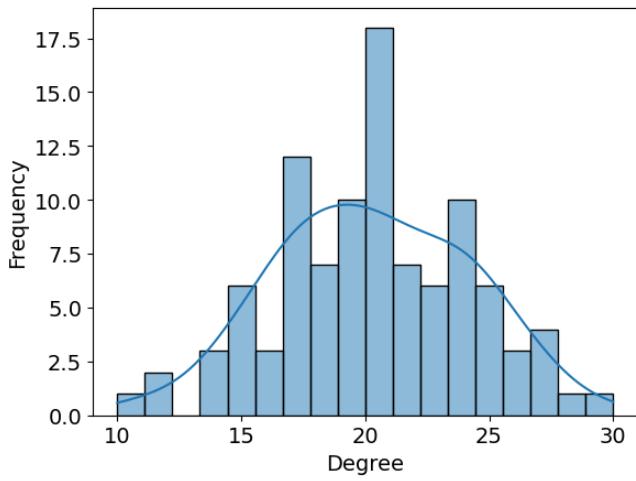


Figure 60: Degree distribution of Reservoir 8 with $\rho = 0.8$ and $\alpha = 0.8$

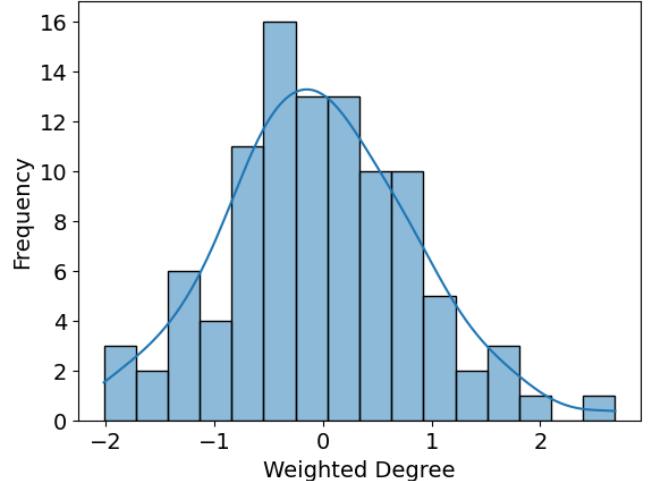


Figure 61: Weighted degree distribution of Reservoir 8 with $\rho = 0.8$

centered around 0. In this thesis, I will refer to neuron index as shown in Figure 62. Notice

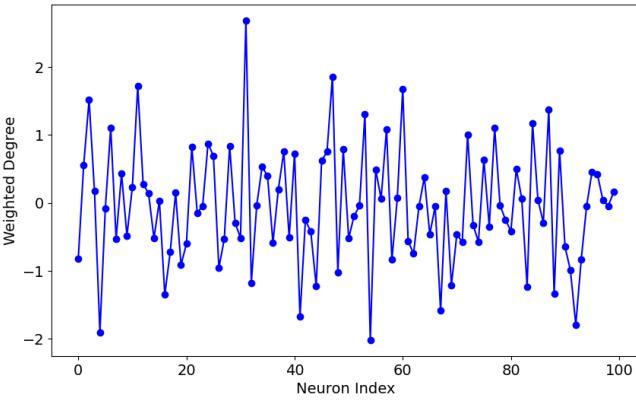


Figure 62: Neurons index and associated weighted degrees in Reservoir 8 with $\rho = 0.8$

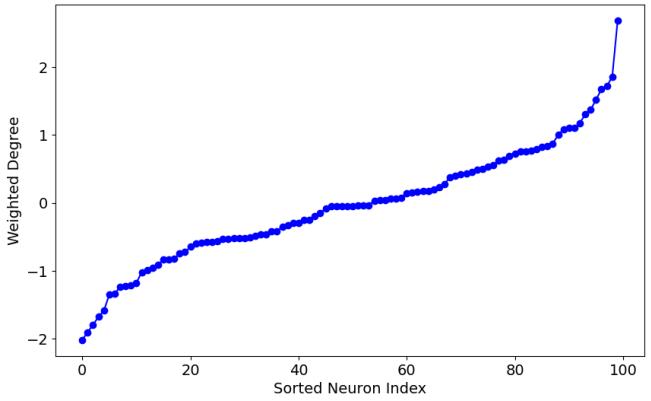


Figure 63: Sorted Neurons index and associated weighted degrees in Reservoir 8 with $\rho = 0.8$

that these indexes can be sorted according to associated weighted degree as in Figure 63 which highlights similar weighted degrees. Notice that depending on ρ value the k of the associated neuron will change which can be seen from Equation (1).

6.1 Bifurcations

Notice that in the following bifurcation diagrams, regular behavior is shown by local max and min staying around some specific areas, while irregular behavior is indicated by them jumping around through the whole attractor.

Reservoir 7 is defined based on chemical connection as in Equation 8 where F is an identity map and $N = 100$. In Figure 64 notice that the neuron stays in equilibrium at 0 when $\rho = 0.8$ and $\alpha < 2$ while when $\alpha = 2$ neuron diverges towards infinity. This does not indicate CAS. Next,

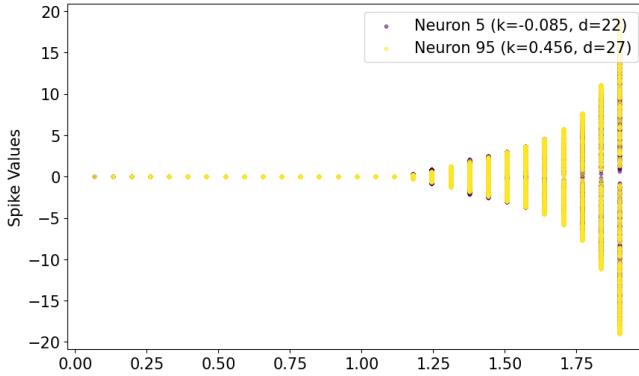


Figure 64: Bifurcations of neurons of index 95 (yellow) and 5 (purple) in reservoir 7 with $\rho = 0.8$, and coupling α as a variable on horizontal axis.

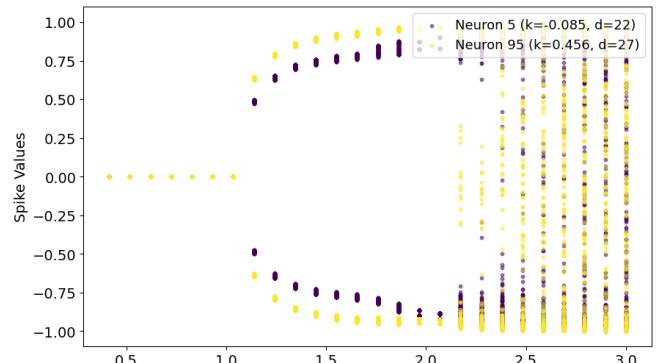


Figure 65: Bifurcations of neurons of index 95 (yellow) and 5 (purple) in reservoir 7 with $\alpha = 0.8$, and spectral radius as a variable on horizontal axis.

neurons 5 and 95 in reservoir 7 with ρ as a variable and $\alpha = 0.8$ as in Figure 65. Notice that when $\rho < 1$ neurons are at equilibrium at 0. Whereas if $1 < \rho < 2$, neurons become less stable and neuron 5 is no more similar to neuron 95. This is an interesting region, but not an indication of CAS as both neurons behave regularly. In this region local minima and maxima remain the same in specific ρ , indicating periodic orbits. Continuing, when $\rho > 2$ neurons become a lot more irregular which can be seen from their states jumping around the attractor.

Turning attention to Figures 66 and 67, notice when $\alpha = 1.3$ the neuron diverges to irregularity at around $\rho = 0.7$ while when $\alpha = 1.8$ the neuron diverges to irregularity around $\rho = 0.2$.

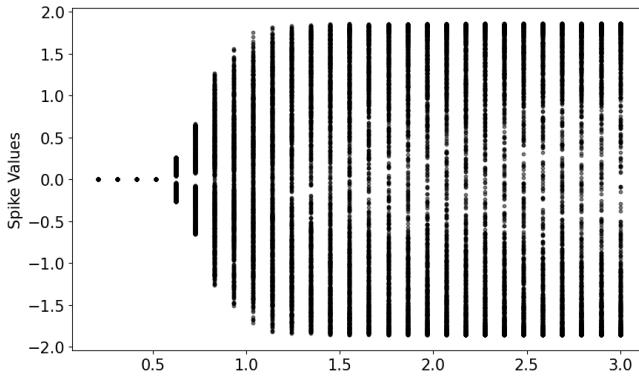


Figure 66: Bifurcations of neuron of index 95 in reservoir 7 with $\alpha = 1.3$, and spectral radius ρ as a variable on horizontal axis.

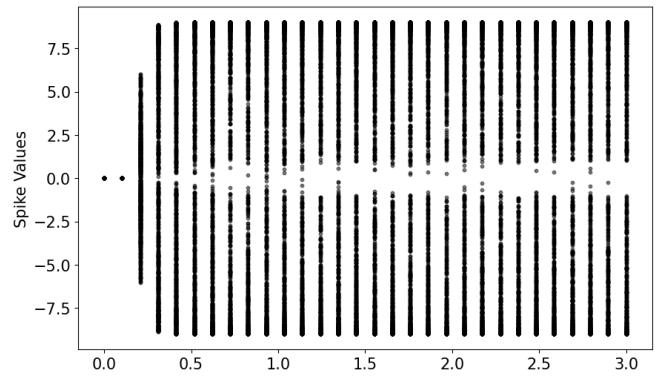


Figure 67: Bifurcations of neuron of index 95 in reservoir 7 with $\alpha = 1.8$, and spectral radius as a variable on horizontal axis.

I can argue that when $1 < \alpha < 2$ there is linear relationship between ρ and α i.e. neurons reach irregularity when $\alpha + \rho = 2$. Notice that all of this coincides with our conclusions in Chapter 4 which further amplifies their credibility.

Reservoir 8 is based on electrical connection as in Equation (9) where F is an identity map and $N = 100$. In this reservoir, neurons are stationary and are at the equilibrium point 0, see Figure 68. The difference between reservoir 8 and 7 is most notable when spectral radius is varied which seems to have no effect as neurons are stationary in the reservoir with electrical connection (Figure 69) whereas in chemical one (reservoir 7) neurons behavior change from stationary to regular and to irregular behavior.

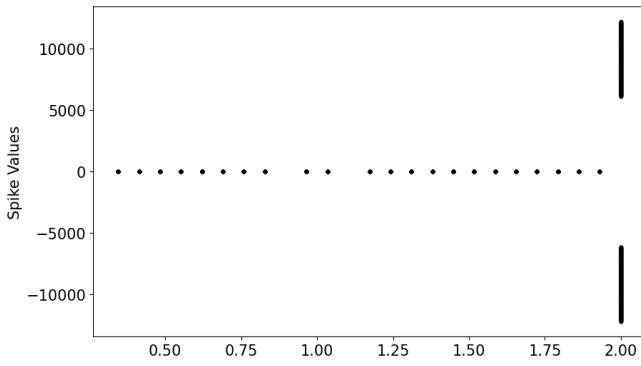


Figure 68: Bifurcations of neuron of index 95 in reservoir 8 with $\rho = 0.8$, and coupling as a variable on horizontal axis.

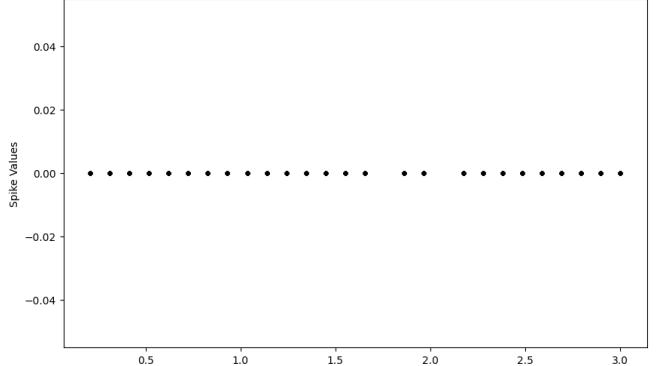


Figure 69: Bifurcations of neurons of index 95 in reservoir 8 with $\alpha = 0.8$, and spectral radius as a variable on horizontal axis.

Reservoir 9 is based on chemical connection as in Equation 8 where F is an chaotic map as in Chapter 3 with $N = 100$. From Figures (70) and 71 one can see that there is significantly more irregularity and neurons behave differently from each other.

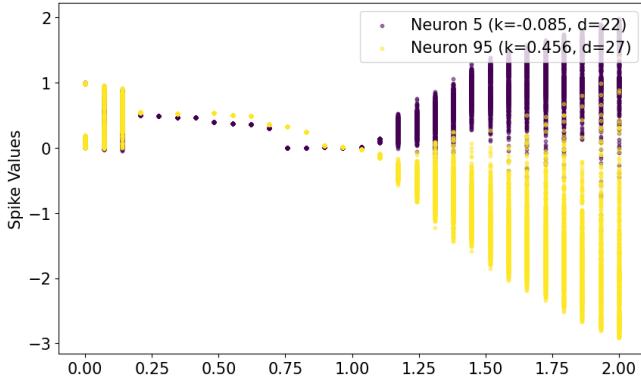


Figure 70: Bifurcations of neurons 5 and 95 ($k_5 \approx -0.085$ and $k_{95} \approx 0.456$) in reservoir 9 with $\rho = 0.8$, and coupling as a variable on horizontal axis.

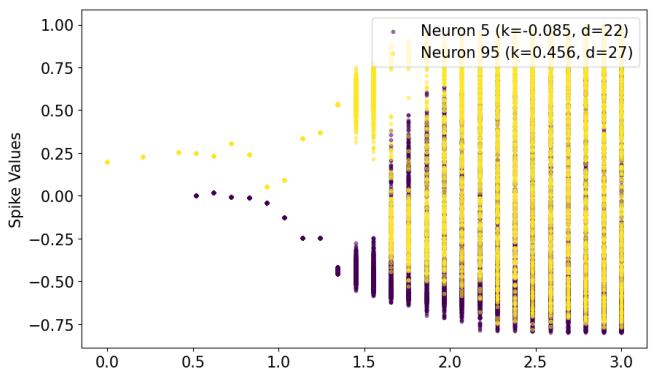


Figure 71: Bifurcations of neurons of 5 and 95 ($k_5 \approx -0.085$ and $k_{95} \approx 0.456$) in reservoir 9 with $\alpha = 0.8$, and spectral radius as a variable on horizontal axis.

This is to be expected as reservoir 9 is composed of units that are chaotic. In Figure 72 notice neurons behave differently when the coupling is around 0.2 or larger. In the same area of varied parameter α , neurons with similar weighted degrees behave similarly, see Figure 73. Similar

degrees neurons acting similarly while different degrees neurons act differently is indication of CAS. With aid of Figures 74 and 75 I further my suspicions that this range of coupling in reservoir 7 might allow CAS behavior. But notice that all neurons in for the reservoir set to operate in this parameter ranges behave regularly which does not indicate CAS.

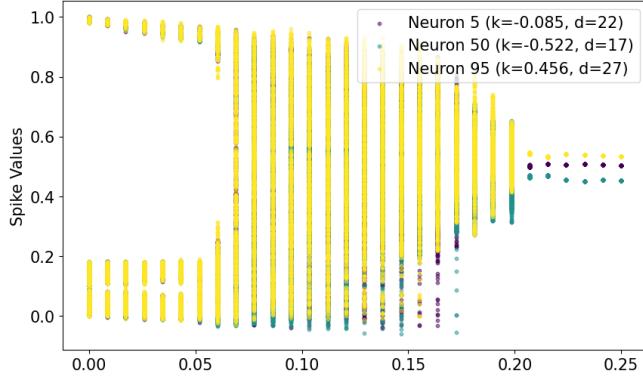


Figure 72: Bifurcations of neurons 5, 50 and 95 ($k_5 \approx -0.085$, $k_{50} \approx -0.522$ and $k_{95} \approx 0.456$) in reservoir 9 with $\rho = 0.8$, and coupling as a variable from 0 to 0.25.

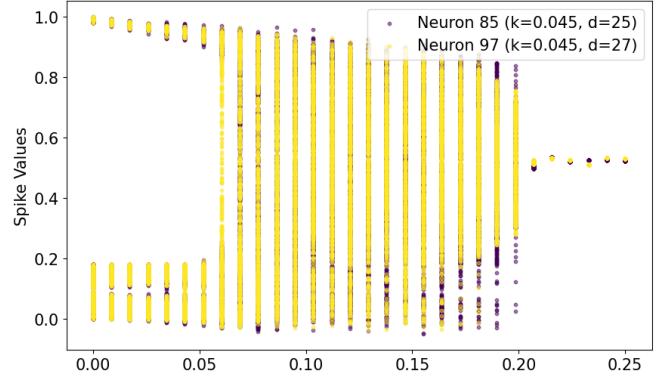


Figure 73: Bifurcations of neurons of 85 and 97 ($k_{85} \approx 0.045$ and $k_{97} \approx 0.045$) in reservoir 9 with $\rho = 0.8$, and coupling as a variable from 0 to 0.25.

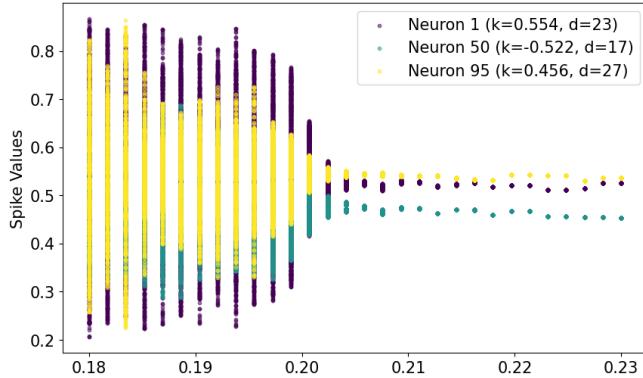


Figure 74: Bifurcations of neurons 5, 50 and 95 ($k_5 \approx -0.085$, $k_{50} \approx -0.522$ and $k_{95} \approx 0.456$) in reservoir 9 with $\rho = 0.8$, and coupling as a variable from 0.18 to 0.23.

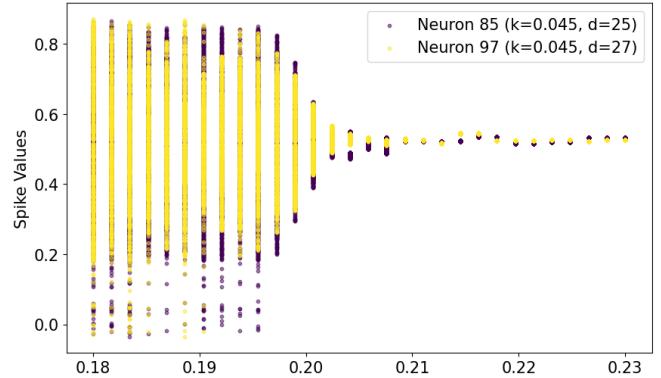


Figure 75: Bifurcations of neurons of 85 and 97 ($k_{85} \approx 0.045$ and $k_{97} \approx 0.045$) in reservoir 9 with $\rho = 0.8$, and coupling as a variable from 0.18 to 0.23.

Reservoir 10 is based on electrical connection as in Equation 6 where F is a chaotic map discussed in Chapter 3 with $N = 100$. Notice Figure 76 looks very similar to Figure 68 with only minor fluctuation at small coupling. After further investigation, I deduced that it is in fact irregular behavior (Figure 77). But what surprised me looking at Figure 77 was neurons behavior at $\alpha \approx 0.19$. Neuron 5 and neuron 95 are both behaving with a periodicity 4 but in different orbits. The neurons with different degrees behave differently which is indication of CAS, but both behave regularly which is not. I decided to investigate this coupling region looking at the dynamics of other neurons (neurons 10 and 90) and found similar behaviour in different coupling values (Figures 78 and 79). As before reservoir connected electrically stays in equilibrium when I vary spectral radius (Figure 80). Interesting behaviour shows up when I pick coupling which induces different neurons behaviour and their periodic orbits remain as I vary spectral radius (Figure 80). Furthermore, in Figure 81 when coupling is small in reservoir there are some regions where similar neurons behave the same.

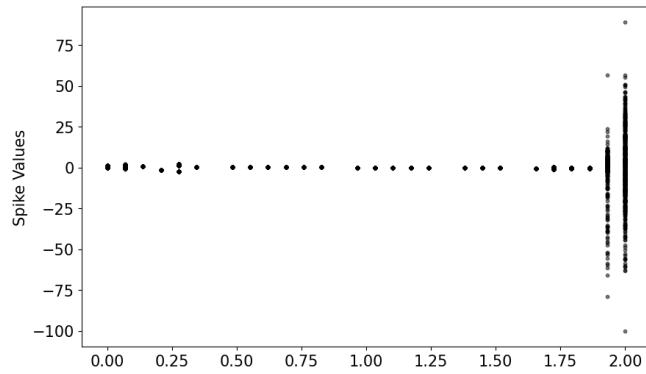


Figure 76: Bifurcations of neuron 95 in reservoir 10 with $\rho = 0.8$, and coupling as a variable on x axis

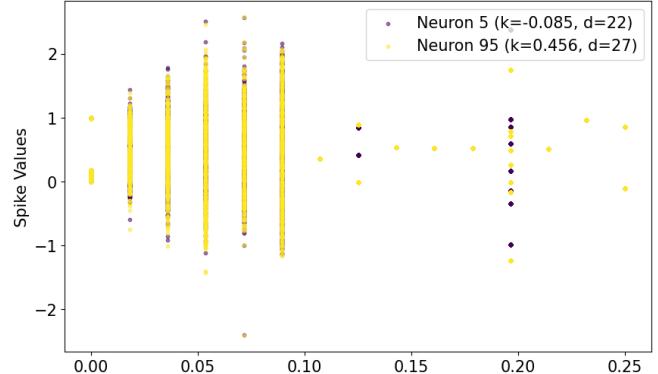


Figure 77: Bifurcations of neurons 5 and 95 in reservoir 10 with $\rho = 0.8$, and coupling as a variable on x axis

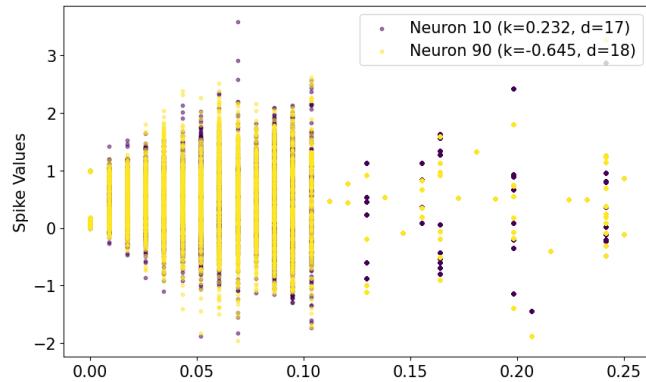


Figure 78: Bifurcations of neurons 10 and 90 in reservoir 10 with $\rho = 0.8$, and coupling as a variable from 0 to 0.25

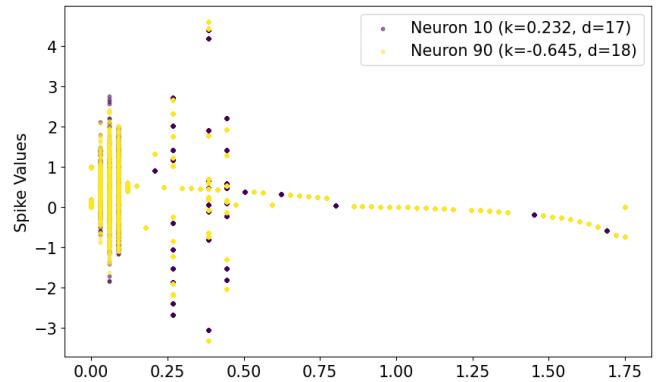


Figure 79: Bifurcations of neurons 10 and 90 in reservoir 10 with $\rho = 0.8$, and coupling as a variable from 0 to 1.75

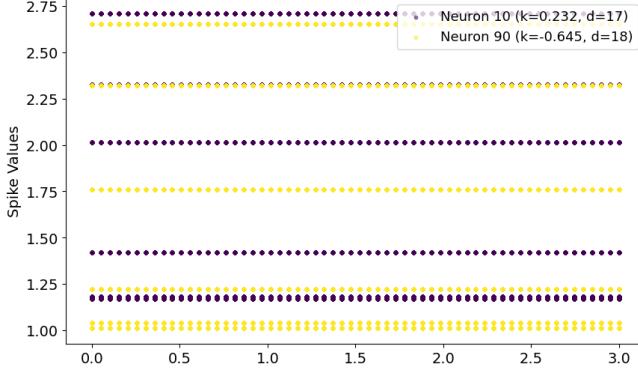


Figure 80: Bifurcations of neurons 10 and 90 in reservoir 10 with $\alpha \approx 0.2669$, and spectral radius as a variable from 0 to 1.75

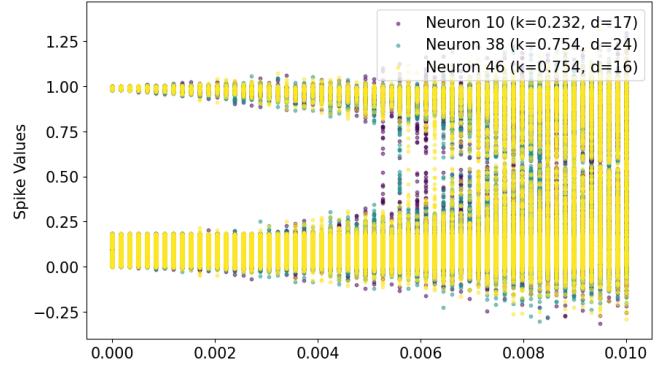


Figure 81: Bifurcations of neuron 10,38 and 46 in reservoir 10 with $\rho = 0.8$, and α as a variable from 0 to 0.01

6.2 Similar Neurons

I have not found any hyperparameter ranges that induces CAS in the reservoir. Nevertheless, I found some interesting regions worth looking into in case I missed something. In this subsection all reservoirs are constructed with $\rho = 0.8$ unless otherwise specified.

Reservoir 7: Now I will look at a set of individual neurons of interest from previous subsection. In Figure 65 notice that neurons behave different from each other when $\rho > 1$. But clearly irregularity ensues when $\rho > 2$. From Figure 82 notice that states of neurons 5 and 95, when $\rho \approx 1.5517$, move in periodic orbits. This does not say anything about CAS as their degree is different ($\hat{k}_5 \approx -0.165$ and $\hat{k}_{95} \approx 0.885$). Whereas in Figure 83 neurons 85 and 97 have similar degrees with only 0.02 difference, nevertheless they behave different from each other. Thus I can not find CAS in this network.

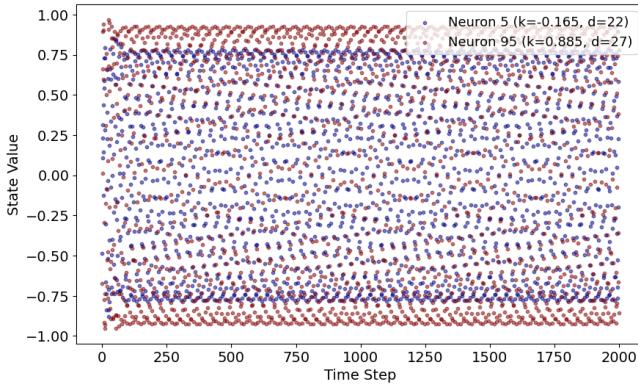


Figure 82: States of neuron 5 and 95 in reservoir 7 with $\alpha = 0.8$ and $\rho \approx 1.5517$ over 2000 time steps

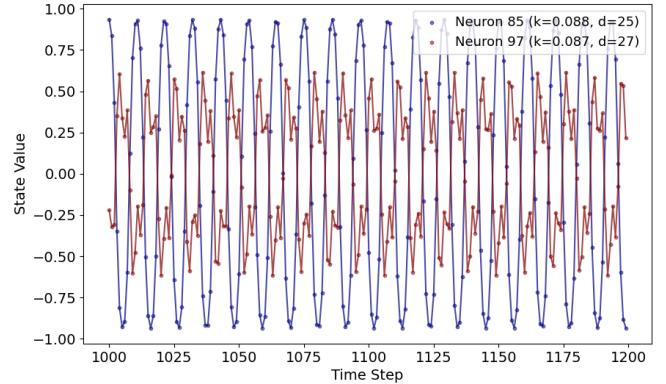


Figure 83: States of neuron 85 and 97 with $\hat{k}_{85} \approx 0.088$ and $\hat{k}_{97} \approx 0.087$ in reservoir 7 with $\alpha = 0.8$ and $\rho \approx 1.5517$ over 200 time steps

Reservoir 8 does not seem to have any CAS-related behavior. The neurons stays in equilibrium for most of the coupling range and all range considered for the spectral radius due to nature of its state equation.

Reservoir 9, as mentioned in previous subsection has some interesting behaviors thus I will investigate it further here. In Figure 84 notice two neurons with similar weighted degrees near similar location which rises my hopes of finding CAS. But, from Figure 85 my hopes are shattered as neurons 38 and 46 with similar weighted degrees are in different parts of the attractor.

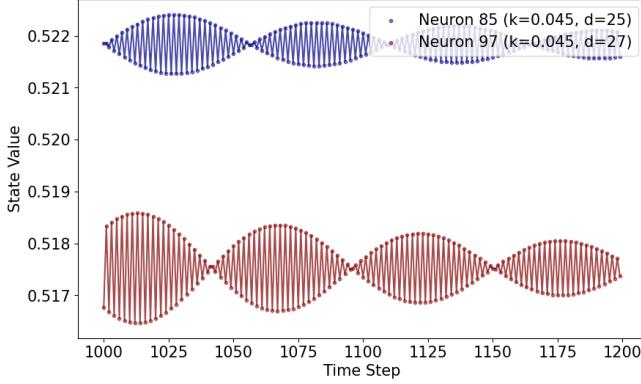


Figure 84: States of neuron 85 and 97 in reservoir 9 with $\alpha \approx 0.2127$ and $\rho = 0.8$ over 200 time steps.

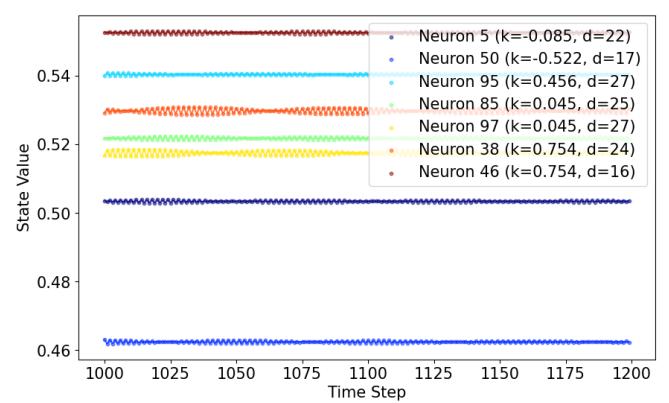


Figure 85: States of 7 neurons in reservoir 9 with $\alpha \approx 0.2127$ and $\rho = 0.8$ over 200 time steps.

Reservoir 10 shows some interesting behavior around $\alpha \approx 0.1964$ which I investigated in Figure 77. From Figure 86 notice that all neurons end up in one of 2 stable orbits 10-cycles. Furthermore, similar neurons seem to behave the same way (Figure 87). Nevertheless, this does not look like CAS as all neurons behave regularly. Notice that neurons are demonstrating synchronous-like behavior i.e. 1 neuron can describe half of the network. Similar behaviour can be found when $\alpha = 0.25$, but in that case all neurons end up in one of 2 stable orbit 3-cycles (Appendix A Figure 102). Investigation of more diagrams which I will leave out here for conciseness led us to believe that neurons form similar periodic orbits when coupling is from around 0.1 to around 0.6.

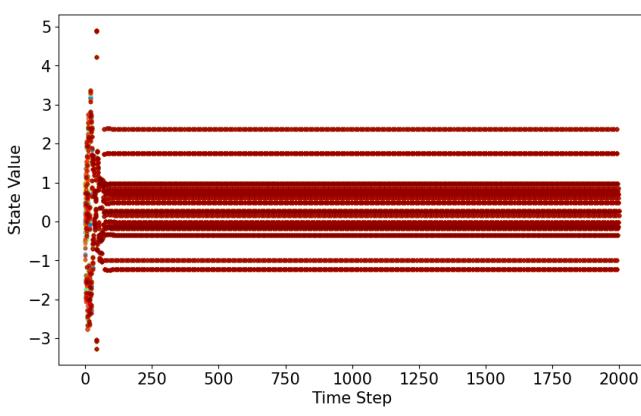


Figure 86: States of all neurons in reservoir 10 with $\alpha \approx 0.1964$ and $\rho = 0.8$ over 2000 time steps.

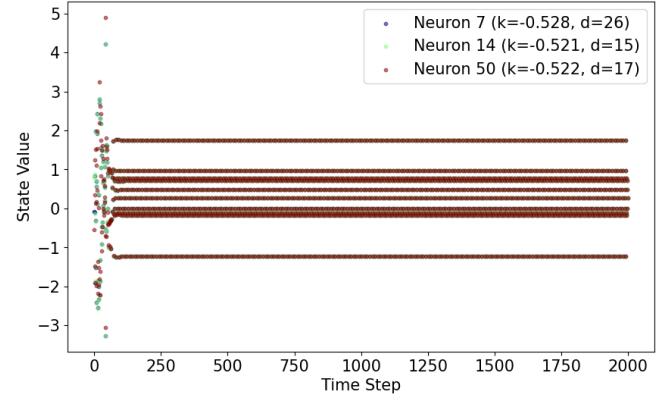


Figure 87: States of neuron 7, 14 and 50 with $k_7 \approx -0.528$, $k_{14} \approx -0.521$ and $k_{50} \approx -0.522$ in reservoir 10 with $\alpha \approx 0.1964$ and $\rho = 0.8$ over 2000 time steps.

Reservoir 11 is based on electrical connection as in Equation (6) where F is an chaotic map discussed in Chapter 3 with $N = 100$ and $\alpha \approx 0.00542$. My hypothesis is that in other

already studied reservoirs, I have not managed to find CAS mostly due to the reason that this specific behavior appears with small coupling. Reservoirs 11 and 12 are created to investigate small coupling region as in Figure 81.

Figures 88 and 89 shows time-series of all neurons in reservoir 11. Further investigating similar neurons I found similar behavior for short periods as in Figure 90 where timesteps are between 1000 and 1110. I wondered whether the neurons could have the same orbit but lagged in time as in Equation (35). To verify whether the orbits were similar but were lagged in time, as it typically happens during the CAS phenomenon, Figure 91. (This is the Delay diagram discussed in Chapter 3.1). The results allows me to see that reservoir does not exhibit CAS despite having adjusted a potential time delay between the orbits, they are still different. Two orbits that have the same CAS pattern are two very similar regular orbits, only shift in time by a delay.

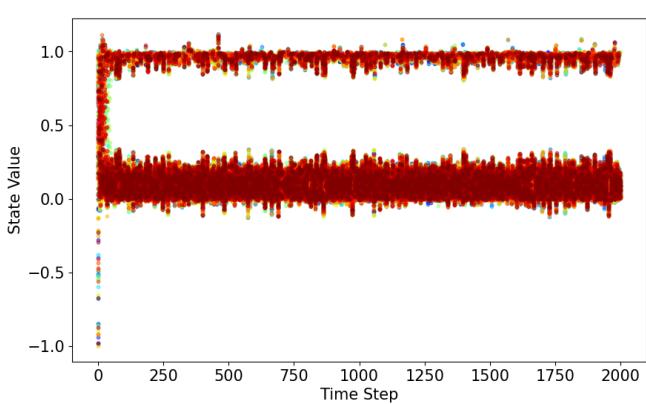


Figure 88: States of all neurons in reservoir 11.

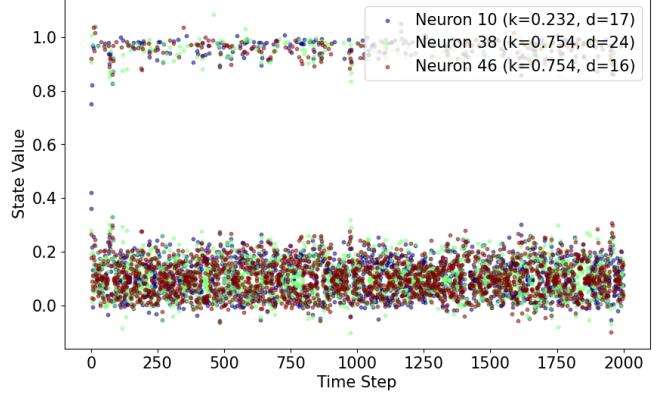


Figure 89: States of neuron 10, 38 and 46 in reservoir 11.

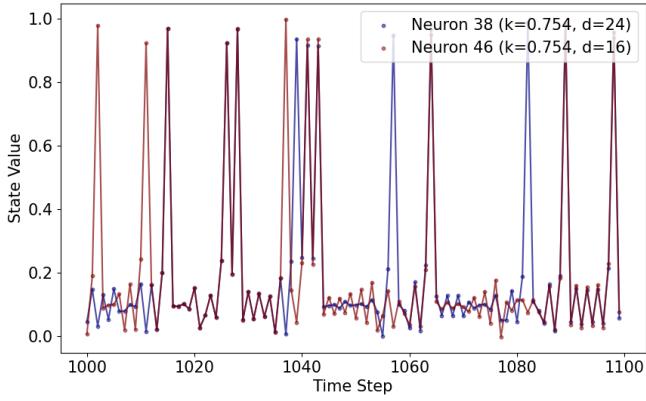


Figure 90: States of neuron 38 and 46 in reservoir 11 over 100 timesteps.

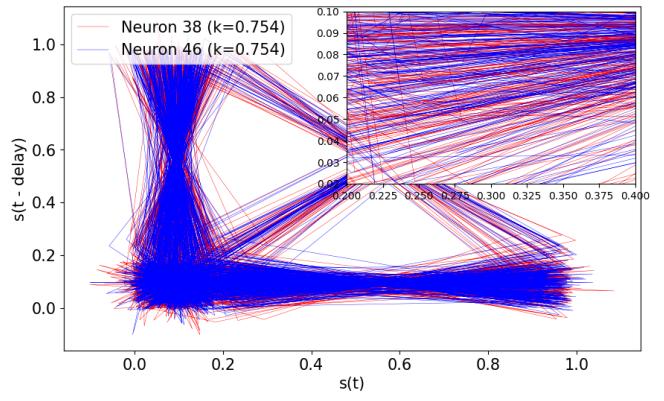


Figure 91: Phase space of neuron 38 and 46 in reservoir 11 with time lag 0 and delay 15.

Reservoir 12 is based on electrical connection as in Equation (6) where F is an chaotic map discussed in Chapter 3 with $N = 100$ and $\alpha \approx 0.00338$. Notice overall the reservoir has neurons with trajectories occupying smaller portions of the state space, so they are more regular (see Figure 92) than with higher coupling in reservoir 11 (Figure 88). Also, notice that all 3 neurons behave very similarly to each other still behaving in a non-periodic fashion characterised by 2 lines distinct

diverging from state value around 0.1 (Figure 93) a different behavior compared to reservoir 11 where neurons are all behaving more irregularly (Figure 89).

In Figure 94 notice the orbits in the reservoir are not similarl but only out of synchrony. See that the time between spikes for the blue trajectory can be quite longer than the time between spikes of the red trajectory. They are different orbits, more regular, but still behaving in a irregular way, chaotic fashion, not periodic. The conclusion is that these orbits are similar but different as shown by the time between spiking. Even with an adjustment of the time-delay (as in 95), they are still looking different. Thus no CAS observed here.

As coupling is decreased further to $\alpha=0.00169$ and 0.0054 I notice irregularity ensuing again (Appendix A Figures 107 and 108). In this thesis reservoir 12 is the closest potentially CAS network. I hypothesise that if CAS exists in reservoir 10 like connected network it is somewhere where $\alpha \in (0.00169, 0.0054)$.

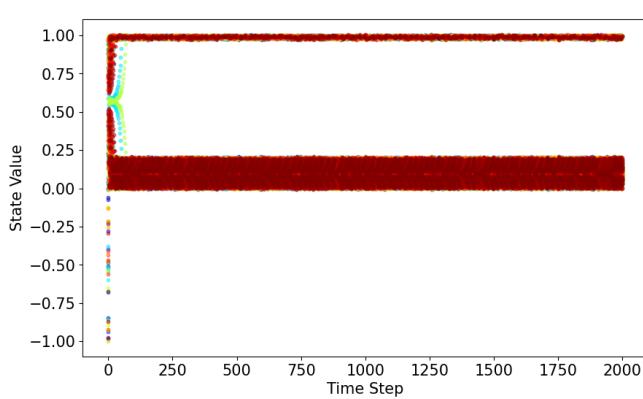


Figure 92: States of all neurons in reservoir 12.

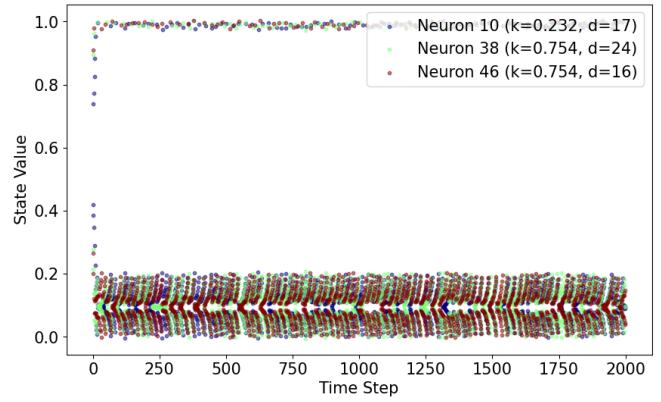


Figure 93: States of neuron 10, 38 and 46 in reservoir 12.

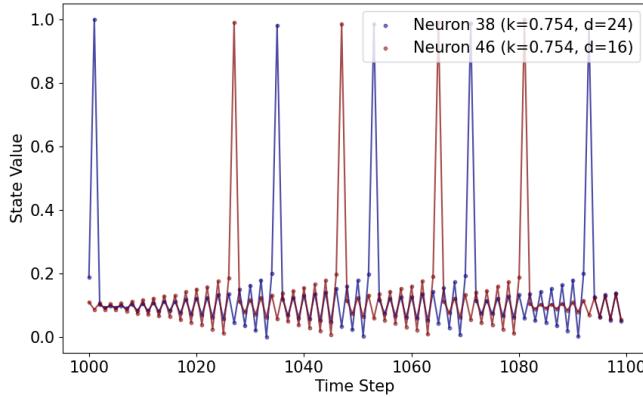


Figure 94: States of neuron 38 (red) and 46 ($k_{38} = 0.754, k_{46} = 0.754$) in reservoir 12 over 100 timesteps.

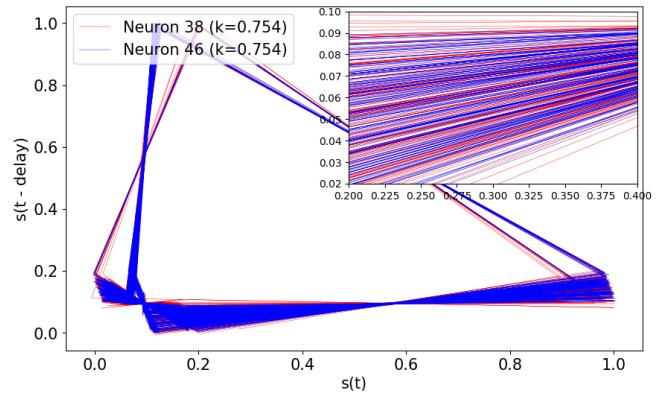


Figure 95: States of neuron 38 (red) and 46 (blue) ($k_{38} = 0.754, k_{46} = 0.754$) in reservoir 12 with time lag 28 and delay 15.

7 Conclusions

This thesis has navigated the complex interplay between chaos and reservoir computing, offering insights into the predictive capabilities of various reservoir types when predicting chaotic signals. Chapter 4 introduced a novel categorization method for reservoirs based on hyper-parameters, leading to the discernment of reservoirs into irregular, nearly-irregular and regular types. The pivotal discovery presented in Chapter 5 is that irregular reservoirs do not enhance the prediction of chaotic signals. On the contrary, reservoirs exhibiting regular and nearly-irregular behavior delivers superior predictions. Furthermore, the incorporation of the z variable from the Lorenz system data did not yield improvements in the predictions within the studied network configurations.

The quest for Collective Almost Synchronization (CAS) within the networks remained elusive. Chapter 6 highlighted a potential interval to CAS appearance that invites further exploration. However, it is crucial to acknowledge that the absence of CAS could be inherently tied 3 things:

- F is not chaotic ($LE = 0$ when $b = -1$) as required for CAS phenomenon to appear.
- Topology of the network is random instead of scale-free.
- This thesis uses weighted degrees instead of degrees as in research papers where the CAS phenomenon appears [13, 14, 29].

Future research can be built on this thesis with ease using the provided Python code notebook [16]. I believe research in the following areas can be beneficial to the research community:

- To further understand irregular reservoirs ability to predict it would be a good idea to analyse looped network using Lyapunov time which directly tells when the error will reach the whole attractor [41].
- Conduct an in-depth exploration within the identified interval in reservoir 10 ($\alpha \in (0.00169, 0.0054)$) where I believe CAS might be found.
- Reconfiguring the network addressing reasons for the absence of CAS and checking whether CAS emerges
- Upon identifying a reservoir with CAS characteristics, compare its predictive performance against other reservoirs and models.

I believe by advancing these suggested lines of inquiry, the research community stands to gain a deeper comprehension of reservoir computing mechanisms and their optimization for chaotic systems.

Special Acknowledgments

I have used ChatGPT from OpenAI [19] to help me to detect and resolve errors in my code and draw figures. Also, I have used it as a search engine to find information. Furthermore, as I am not a native English speaker, I used it to enhance my thought's readability. By no means does my use of ChatGPT imply that I have used it to create this thesis by generating pages. All of the pages are written by me with supervision from Murilo. Thoughts that are not mine are cited accordingly.

References

- [1] Gilpin, W., 2023. Model scale versus domain knowledge in statistical forecasting of chaotic systems. *Physical Review Research*, 5(4), p.043252.
- [2] : Vanishing gradient problem, Wikipedia, The Free Encyclopedia, Wikimedia Foundation, 28 January 2024, https://en.wikipedia.org/wiki/Vanishing_gradient_problem
- [3] 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.
- [4] : Long short-term memory, Sepp Hochreiter and Jürgen Schmidhuber December 1997 9(8):1735-80
- [5] : Attention is all you need, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, 12 Jun 2017, arXiv:1706.03762
- [6] : Fast Fourier Transform, Wikipedia, The Free Encyclopedia, Wikimedia Foundation, 21 April 2024, https://en.wikipedia.org/wiki/Fast_Fourier_transform
- [7] : Pearson Correlation, Wikipedia, The Free Encyclopedia, Wikimedia Foundation, 3 April 2024, https://en.wikipedia.org/wiki/Pearson_correlation_coefficient
- [8] : Julia Slipantschuk, Oscar F. Bandtlow, Wolfram Just, On the relation between Lyapunov exponents and exponential decay of correlations, 12 Sep 2012, arXiv:1209.2640
- [9] : Lyapunov Exponent, Wikipedia, The Free Encyclopedia, Wikimedia Foundation, 2 April 2024, https://en.wikipedia.org/wiki/Lyapunov_exponent
- [10] : Maria Angélica Araujo, Lyapunov exponents and extensivity in multiplex networks, Aberdeen University, 18th November 2019.
- [11] : Do Reservoir Computers Work Best at the Edge of Chaos? Thomas L. Carroll, 2 Dec 2020, arXiv:2012.01409
- [12] Lorenz, E. N., 1963: Deterministic Nonperiodic Flow. *J. Atmos. Sci.*, 20, 130–141
- [13] Baptista MS, Ren HP, Swarts JC, Carareto R, Nijmeijer H, Grebogi C. Collective almost synchronisation in complex networks. *PLoS One*. 2012;7(11):e48118. doi: 10.1371/journal.pone.0048118. Epub 2012 Nov 8. PMID: 23144851; PMCID: PMC3493579.
- [14] Ren, H.P., Bai, C., Baptista, M.S. and Grebogi, C., 2017. weak connections form an infinite number of patterns in the brain. *Scientific Reports*, 7(1), p.46472.
- [15] : Scale-free network, Wikipedia, The Free Encyclopedia, Wikimedia Foundation, 26 March 2024, https://en.wikipedia.org/wiki/Scale-free_network
- [16] Code for experimentations done in this thesis can be found in Google Colab Notebook accessible through the link here: <https://colab.research.google.com/drive/1k39xPxKLMTMk3ziCnx9E-TacVVbXsP?usp=sharing>

- [17] What is AI? An Article written by IBM. accessed April 24, 2024, available: <https://www.ibm.com/topics/artificial-intelligence>
- [18] Generative artificial intelligence, Wikipedia, The Free Encyclopedia, Wikimedia Foundation, 10 April 2024, https://en.wikipedia.org/wiki/Generative_artificial_intelligence
- [19] OpenAI (2023) ChatGPT-4 [Large language model].
- [20] Oppenlaender, Jonas, The Creativity of Text-to-Image Generation, ACM, 2022
- [21] What is ML? An Article written by IBM, accessed April 24, 2024, available: <https://www.ibm.com/topics/machine-learning>
- [22] What is deep learning? An article written by IBM company, accessed April 24, 2024 available: [https://www.ibm.com/topics/deep-learning#:~:text=Deep%20learning%20is%20a%20subset, AI\)%20in%20our%20lives%20today.](https://www.ibm.com/topics/deep-learning#:~:text=Deep%20learning%20is%20a%20subset, AI)%20in%20our%20lives%20today.)
- [23] What is Recurrent Neural Network? An article written by IBM company, accessed April 24, available: <https://www.ibm.com/topics/recurrent-neural-networks>
- [24] Schrauwen, D. Verstraeten, and J. Van Campenhout, “An overview of reservoir computing: theory, applications and implementations,” in Proceedings of the 15th european symposium on artificial neural networks. p. 471-482 2007, 2007, pp. 471–482.
- [25] N. Richard, “The differences between artificial and biological neural networks,” Towards Data Science, 2018.
- [26] R. S. Zimmermann and U. Parlitz, “Observing spatiotemporal dynamics of excitable media using reservoir computing,” Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 28, no. 4, p. 043118, 2018
- [27] Lukosevičius, M, Jaeger H. Reservoir computing approaches to recurrent neural network training. Comput Sci Rev. 2009;3(3):127–149.
- [28] Yahya Robert Scerbo, Decomposition of Chaotic Signals by Reservoir Computing 2021
- [29] Nguyen, P.T.M., Hayashi, Y., Baptista, M.D.S. et al. Collective almost synchronization-based model to extract and predict features of EEG signals. Sci Rep 10, 16342 (2020).
- [30] : SciPy 1.13.0, free and open-source Python library, function signal.correlate. Accessed: 13th April 2024. [Online]. Available: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.correlate.html>
- [31] : Bahramian A, Ramadoss J, Nazarimehr F, Rajagopal K, Jafari S, Hussain I. A simple one-dimensional map-based model of spiking neurons with wide ranges of firing rates and complexities. J Theor Biol. 2022 Apr 21;539:111062. doi: 10.1016/j.jtbi.2022.111062. Epub 2022 Feb 12. PMID: 35167840.
- [32] Cont R, Bouchaud JP (2000) Herd behavior and aggregate fluctuations in financial markets. Macroecon Dyn 4: 170–196.

- [33] Couzin ID, Krause J (2003) Self-organization and collective behavior in vertebrates. *Adv Study Behavior* 32: 1–75.
- [34] Helbing D, Farkas I, Vicsek T (2000) Simulating dynamical features of escape panic. *Nature* 407: 487–490.
- [35] Fujisaka H, Yamada T (1983) Stability theory of synchronized motion in coupled-oscillator systems. *Progress of Theoretical Physics* 69: 32–47.
- [36] Steur E, Nijmeijer H (2011) Synchronization in networks of diffusively coupled time-delay coupled (semi-)passive systems. *IEEE Trans Circuits Syst I* 58: 1358–1371.
- [37] Baptista MS, Kakmeni FMM, Grebogi C (2010) Combined effect of chemical and electrical synapses in hindmarsh-rose neural networks on synchronization and the rate of information. *Phys Rev E* 82: 036203-1–036203-12.
- [38] : The Lorenz system, by James Hately, accessible: <https://web.math.ucsb.edu/~jhately/paper/lorenz.pdf>
- [39] E. J. Bianco-Martinez, M. S. Baptista, C. Letellier, Symbolic computations of non-linear observability, *Physical Review E*, 062912 (2015)
- [40] Chaos Theory Explained: A deep dive into an unpredictable universe by Paul Sutter published in Space on March 18 2022. Available: <https://www.space.com/chaos-theory-explainer-unpredictable-systems.html>
- [41] : Lyapunov time, Wikipedia, The Free Encyclopedia, Wikimedia Foundation, 4 Agust 2023, https://en.wikipedia.org/wiki/Lyapunov_time
- [42] : ”Overfitting: Datarobot artificial intelligence wiki” accessed: 12th April 2024. [Online]. Available: <https://www.datarobot.com/wiki/overtting/>
- [43] “Social network for programmers and developers,” accessed: 15th Mar 2021. [Online]. Available: <https://morioh.com/p/15c995420be6>

8 Appendix A

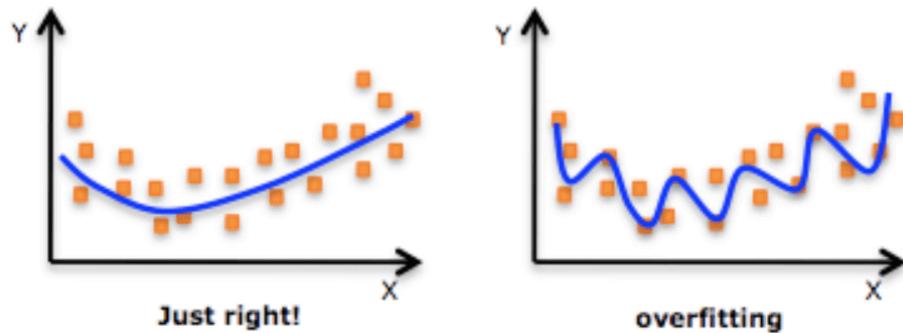


Figure 96: The image on the left shows an example of a model that matches the data suitably. The image on the right shows an example of a model that has been "overfitted", this means it has attuned to train data, but will perform significantly worse on test data due to its inability to capture general trend [42].

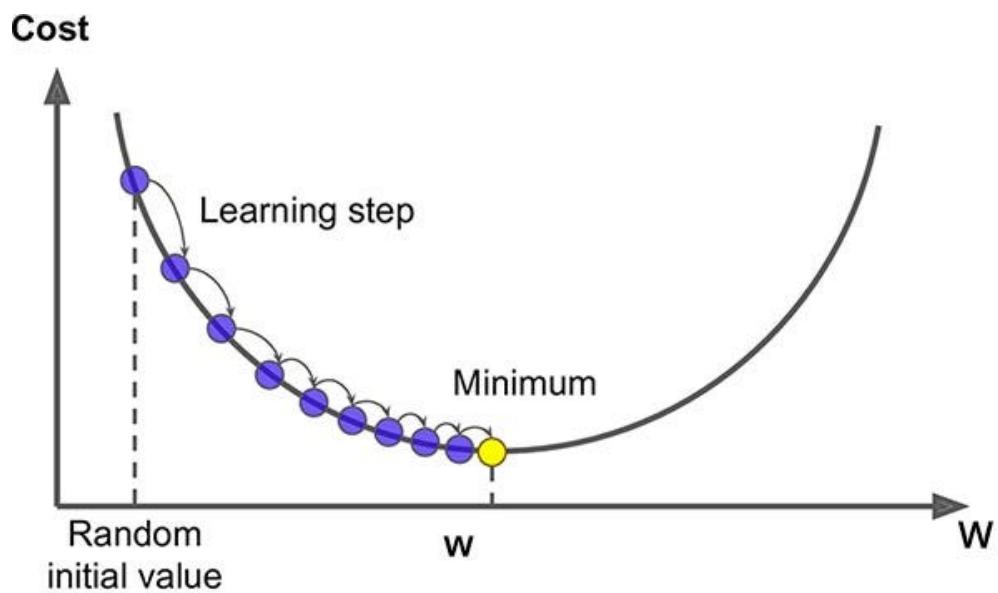


Figure 97: Gradient descent is often used during the training of a typical RNN. Over each iteration I move in the direction of steepest descent [43].

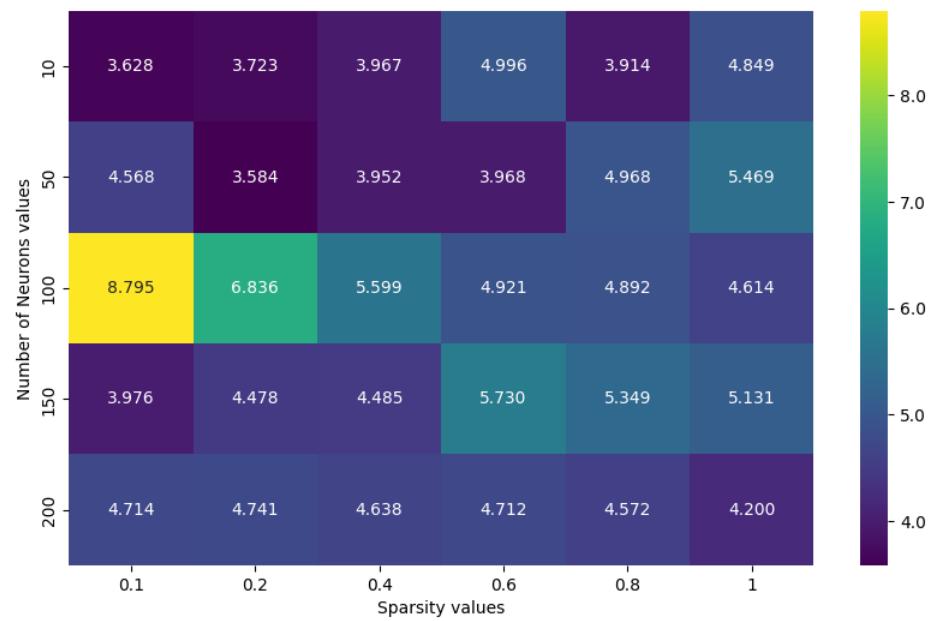


Figure 98: Heatmap of the integral of Pearson Correlation in Reservoirs ($\alpha = 0.8, \rho = 0.8$) constructed with different sparsity in weight matrix and number of neurons

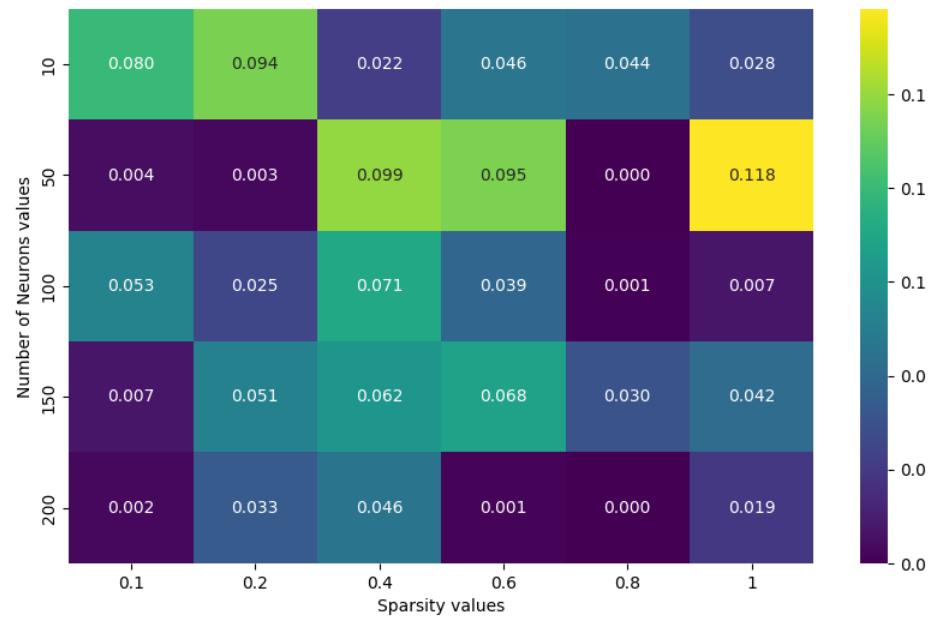


Figure 99: Heatmap of the integral of spectra above magnitude 5 in Reservoirs ($\alpha = 0.8, \rho = 1.3$) constructed with different sparsity in weight matrix and number of neurons

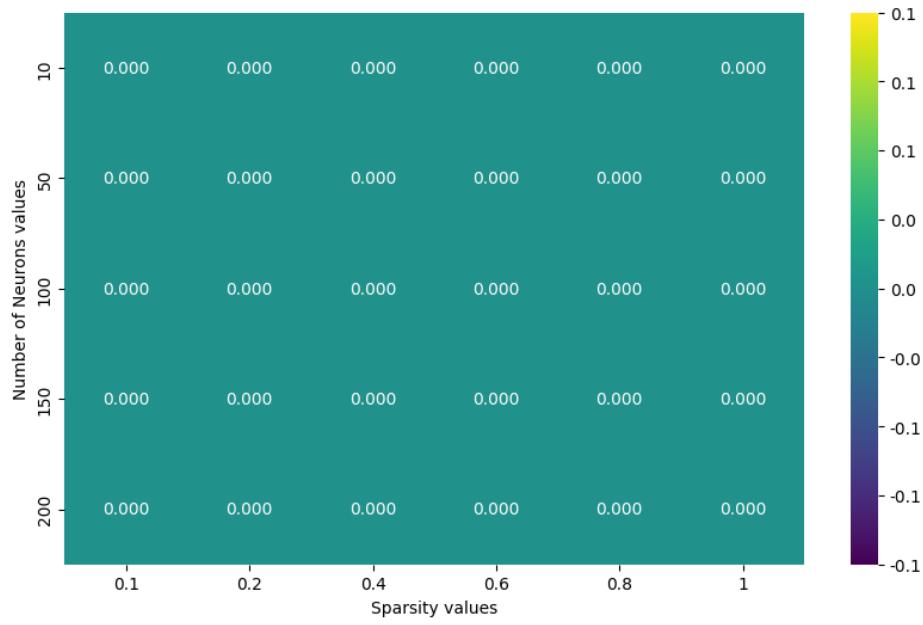


Figure 100: Heatmap of the integral of spectra above magnitude 5 in Reservoirs ($\alpha = 0.8, \rho = 0.8$) constructed with different sparsity in weight matrix and number of neurons

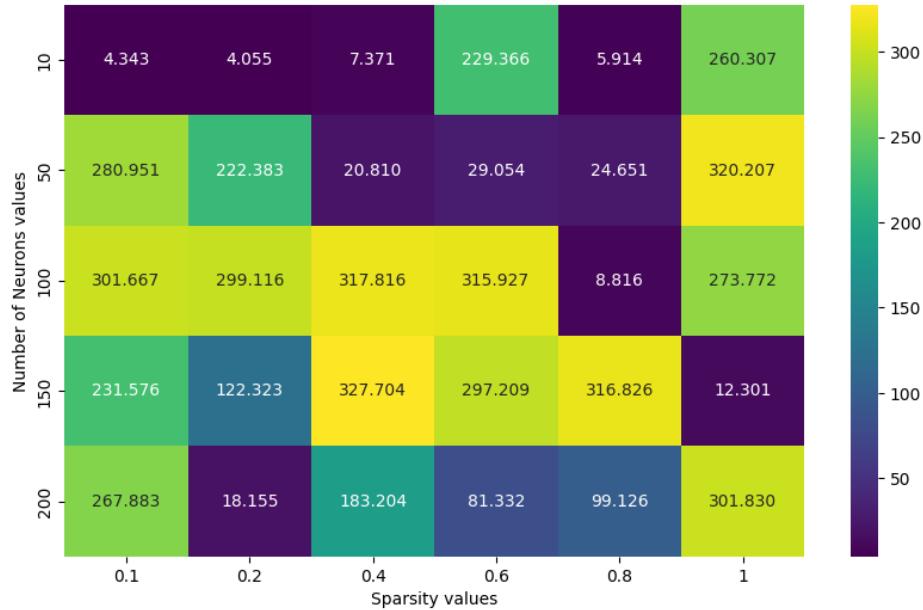


Figure 101: Heatmap of the integral of Pearson Correlation in Reservoirs ($\alpha = 0.8, \rho = 1.3$) constructed with different sparsity in weight matrix and number of neurons

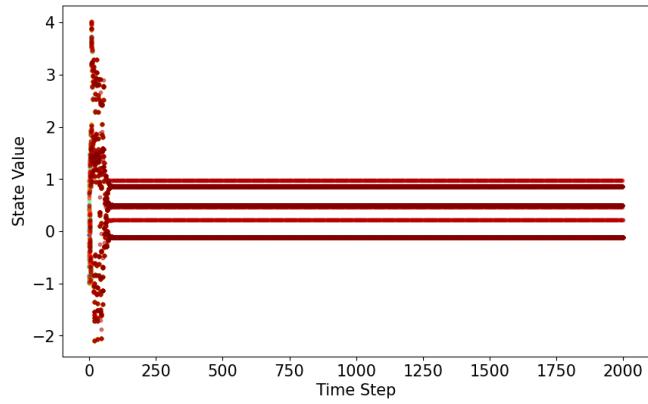


Figure 102: States of all neurons in reservoir 10 with $\alpha = 0.25$ and $\rho = 0.8$ over 2000 time steps

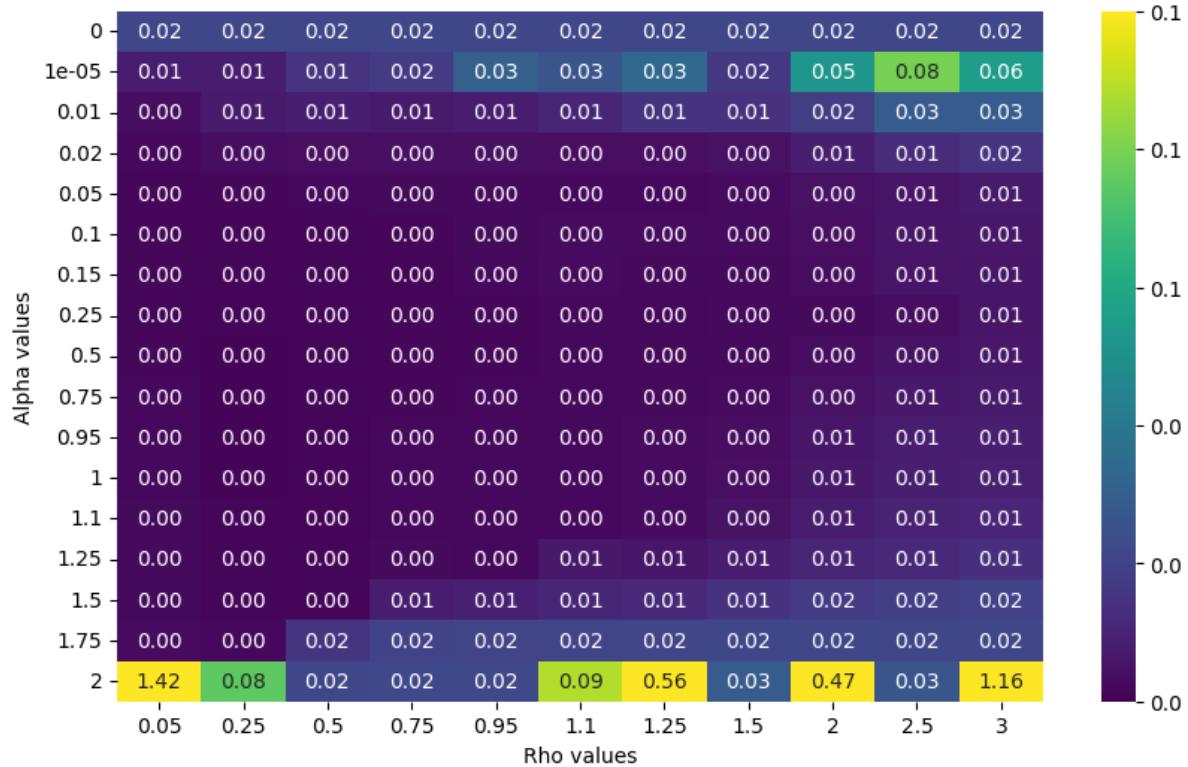


Figure 103: heat map over networks constructed by different alpha's and rho's and used to predict $\tau = 1$ timesteps in future of Lorenz system, evaluated by RMSE.

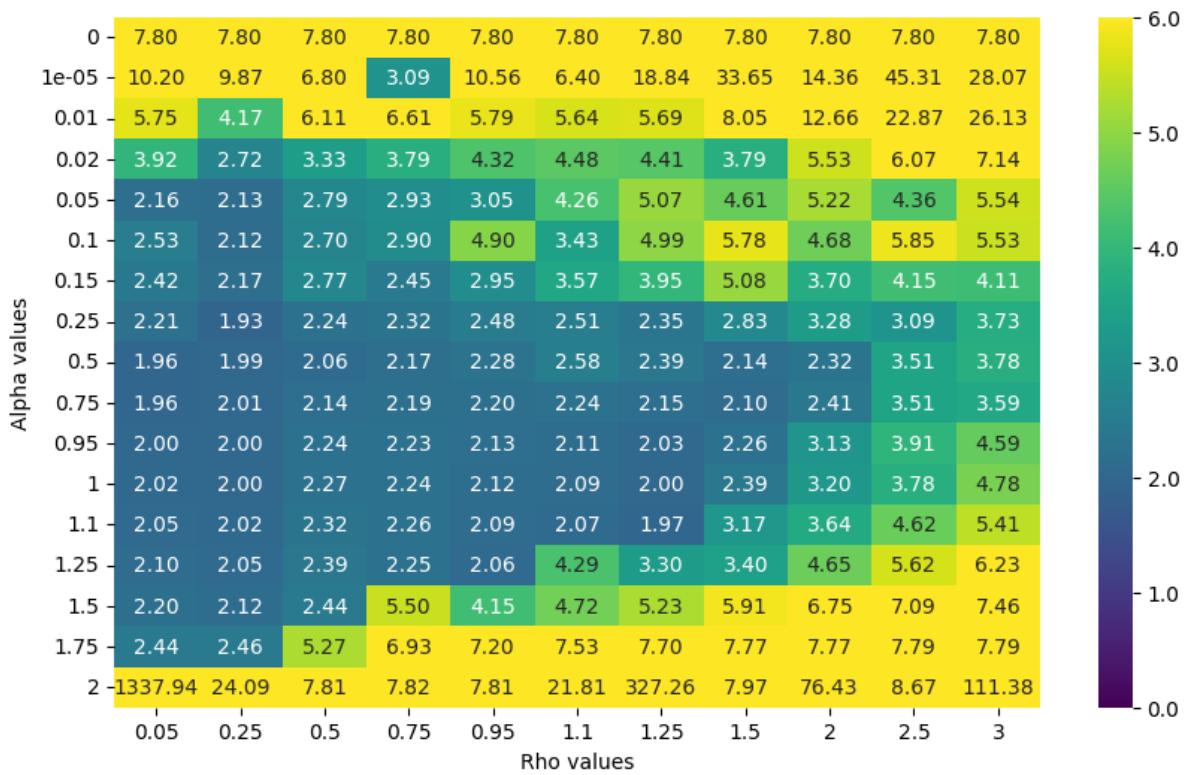


Figure 104: heat map over networks constructed by different alpha's and rho's and used to predict $\tau = 50$ timesteps in future of Lorenz system, evaluated by RMSE.

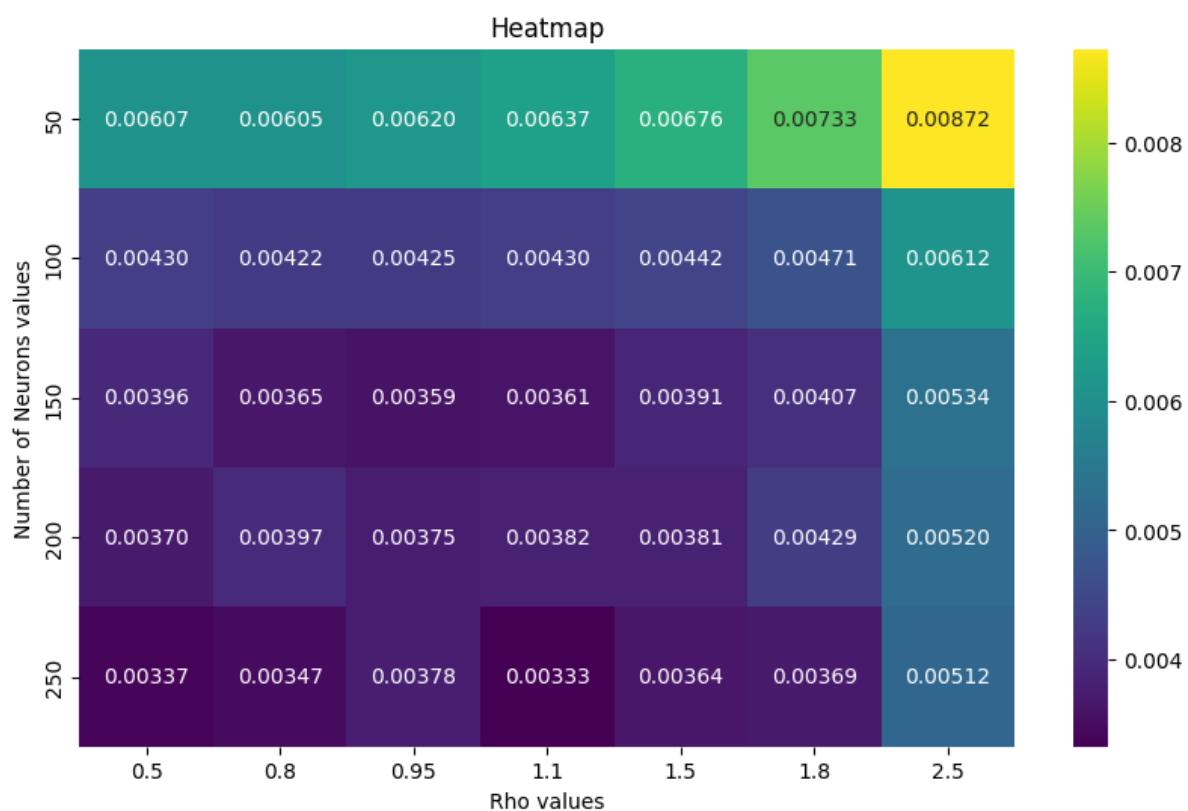


Figure 105: heat map over networks constructed by different alpha's and rho's and used to predict $\tau = 1$ timesteps in future of Lorenz system, evaluated by RMSE.

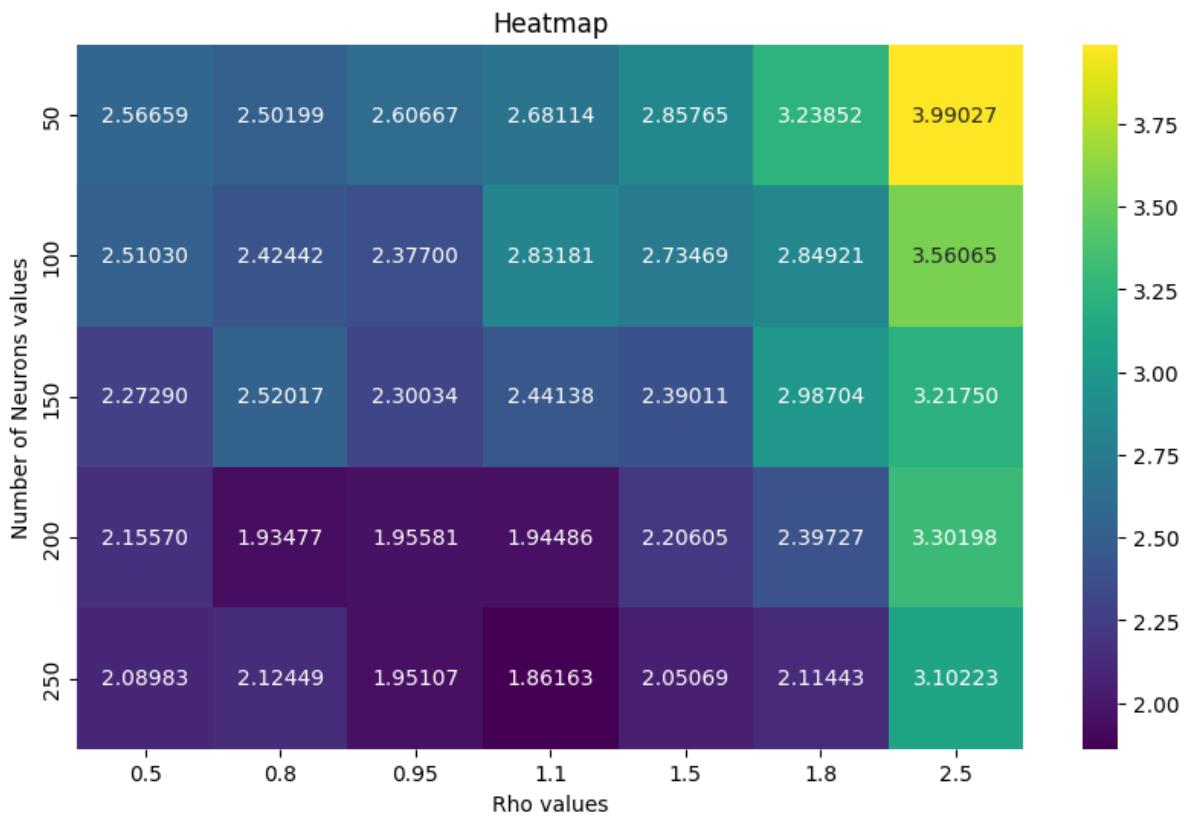


Figure 106: heat map over networks constructed by different alpha's and rho's and used to predict $\tau = 50$ timesteps in future of Lorenz system, evaluated by RMSE.

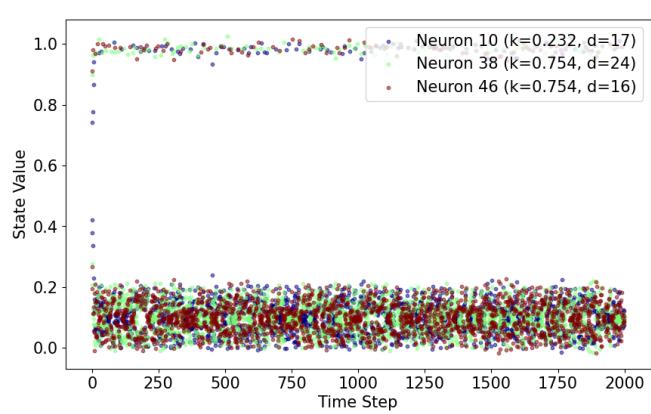


Figure 107: States of neuron 10, 38 and 46 In Reservoir 10 $\alpha \approx 0.00169$

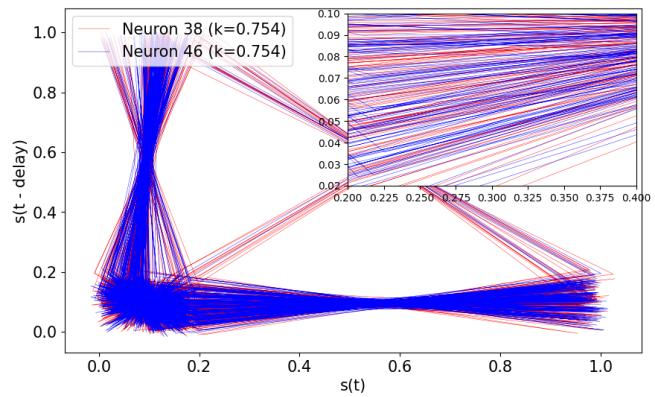


Figure 108: Delay diagram of neurons 38 and 46 in Reservoir 10 with $\alpha \approx 0.00169$ with time lag -6 and delay 15