UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

A Modified Echo State Network Model Using Non-Random Topology

Diana Carolina Roca Arroyo

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP SÃO CARLOS

Diana Carolina Roca Arroyo

A Modified Echo State Network Model Using Non-Random Topology

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Zhao Liang
Co-advisor: Prof. Dr. Israel Tojal da Silva

USP – São Carlos
August 2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

R669m

Roca Arroyo, Diana Carolina
A Modified Echo State Network Model Using Non-Random Topology / Diana Carolina Roca Arroyo; orientador Liang Zhao; coorientador Israel Tojal Da Silva. -- São Carlos, 2023.
83 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.

1. Echo State Networks. 2. Non Random Topologies. 3. Clustered Networks. 4. Complex Networks. 5. Reservoir Computing. I. Zhao, Liang, orient. II. Tojal Da Silva, Israel, coorient. III. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Diana Carolina Roca Arroyo

Um Modelo de Redes de Estado de Eco Modificado Usando Topologías Não Aleatórias

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutora em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃČO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Zhao Liang
Coorientador: Prof. Dr. Israel Tojal da Silva

USP – São Carlos
Agosto de 2023

Este trabalho é dedicado as pessoas mais importantes para mim: minha familia, o amor da minha vida, a Meire, e a todas as pessoas que fazem parte da minha vida e tenho a fortuna de chamar amigos.

A Deus seja toda a glória e honra por sempre...

ACKNOWLEDGEMENTS

The author of this work would like to thank:

- The Center for Mathematical Sciences Applied to Industry (CeMEAI) funded by FAPESP (grant 2013/07375-0) for the computational resources to carry out this research.
- The Center for Artificial Intelligence (C4AI-USP) and the support from the São Paulo Research Foundation (FAPESP grant 2019/07665-4) and from the IBM Corporation. It was supported by FAPESP grant (2022/12991-0).
- Agência de Fomento CAPES-Brasil, for finnacial support. Grant (PROEX-10973721/D1)

In particular, I would like to thank my family, my parents, Evaristo Roca and Celia Arroyo, my siblings, Luz Roca and Andrés Roca, my great friend Meire, my dear friends, and all the people who believed in me and were part of this process that began as an uncertain dream but has become a reality. I also want to thank my tutors, Professor Liang Zhao and Professor Israel Tojal, for all their support, knowledge, patience and time invested in this work, without them it would not have been possible.

“Disse-lhe Jesus: Não te hei dito que, se creres, verás a glória de Deus?”
João 11:40

RESUMO

ROCA, D. C. **Um Modelo de Redes de Estado de Eco Modificado Usando Topologías Não Aleatórias**. 2023. 83 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

A computação de reservatório é um tipo de redes neurais recorrentes adequadas para o processamento de dados temporais/sequenciais. Um dos modelos representativos da computação de reservatório é a rede de estado de eco (no inglês Echo State Network, ESN), que mapeia os padrões de entrada-saída através de uma projeção não linear de alta dimensão, denominada reservatório. No modelo clássico de ESN, o reservatório é composto por um grande número de neurónios com ligações aleatórias, formando uma rede complexa aleatória do tipo Erdös-Rényi. Desta forma, o reservatório mapeia os dados de entrada num espaço de dimensão superior para resolver o problema da inseparabilidade linear, aplicando um algoritmo de regressão linear simples na fase de treino. Inspirados na estrutura modular de um cérebro humano, propomos novas ESNs utilizando topologia não aleatória como reservatório por modelos de redes complexas e modelos gerados via agrupamento de dados. A topologia de conetividade baseada em redes complexas no reservatório são: redes aleatórias, redes livre de escala e redes de mundo pequeno. Para gerar os reservatórios com clusters, propomos utilizar os algoritmos clássicos de agrupamento de dados: K-means, Partitioning Around Medoids, e algoritmo de Ward para simular estruturas de comunidades. Também geramos as redes livre de escala e de pequeno mundo agrupadas como reservatórios. A principal hipótese desta abordagem é que as estruturas de rede não aleatórias, especialmente as redes com clusters como reservatórios, podem captar melhor a informação de diferentes classes dos dados de treino, de tal forma que um grupo de comunidades da rede pode servir para codificar uma determinada classe de dados. Simulações numéricas em aplicações de previsão de sinais mostram que os modelos propostos apresentam uma melhoria no seu desempenho e um custo computacional mais baixo em comparação com as ESN clássicas. A utilidade dos nossos métodos baseados em ESNs foi demonstrada considerando a classificação de tecidos em imagens histopatológicas de lâminas inteiras (WSI) coradas com hematoxilina e eosina (H&E). Os modelos propostos para classificar componentes de tecido em WSI fornecem resultados que superam as técnicas clássicas e as do estado da arte.

Palavras-chave: Redes de Estado de Eco, Computação de Reservatório, Clusterização, Redes Neurais Recorrentes.

ABSTRACT

ROCA, D. C. **A Modified Echo State Network Model Using Non-Random Topology**. 2023. 83 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

Reservoir Computing is a type of recurrent neural networks suitable for temporal/sequential data processing. One of the representative Reservoir Computing models is the Echo State Network (ESN), which maps the input-output patterns through a high dimensional nonlinear projection, called reservoir. In the classic model of ESN, the reservoir is compound of a large number of neurons with random connections, forming a Erdös-Rényi random complex network. In this way, the reservoir maps the input data into a higher dimensional space to overcome the linear inseparable problem by applying a simple linear regression algorithm in the training phase. Inspired by the modular structure of a human brain, we propose new ESNs using non-random typology as reservoir by complex network models and clustering models. The connectivity topology based on complex networks in the reservoir are: random networks, scale-free networks, and small-world networks. To generate the clustered reservoirs, we propose to use the classic data clustering algorithms: K-means, Partitioning Around Medoids, and Ward algorithm to simulate community structures. We also generate the clustered scale-free and small-world networks as reservoirs. The main hypothesis of this approach is that the non-random network structures, especially, the clustered networks as reservoirs can better capture the information of different classes of the training data in such a way that each of a group of network communities may be served to encode a certain class of data. Numerical experiments in signal prediction applications show that the proposed models have an improvement in its performance and lower computational cost compared to the classical ESNs. The utility of our ESN-based methods has been shown by considering the tissue classification on hematoxylin and eosin (H&E) stained histopathological whole slide images (WSI). The proposed models to classify tissue components on WSI provide results that overcome classical and the state-of-the-art techniques.

Keywords: Echo State Network, Reservoir Computing, Clustering, Recurrent neural network.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER

1

INTRODUCTION

Currently, the development of scientific advances applied in all areas is constantly growing due to the increasing use of artificial intelligence as a novel technology to develop and perform vast different tasks. One of the main subareas of artificial is machine learning, such as Artificial Neural Networks (ANNs). ANNs are powerful computational techniques inspired by biological neural networks (HAYKIN, 2007). From the simplest models to more complex network structures have been designed and adapted to each type of situation or context of the problem being solved (HAYKIN, 2009; HAYKIN, 2007).

In their pioneer work, McCulloch and Pitts introduced an artificial neural network with simple threshold units called *Perceptron*, which can solve simple classification tasks in linearly separable datasets (HAYKIN, 2007). Later, multi-layer feed-forward neural networks have been proposed to face more complex tasks that the *Perceptron* network can not solve, like XOR function. Since then, more complex neural networks have been developed like, well-known, Recurrent Neural Network (RNN) which differs, with the already mentioned networks, on having at least one cyclic pathway of synaptic connections. Although RNNs work well for non-linear dynamical system approximation, they have some salient disadvantages, such as hyper-parameter setting. Moreover, all network connection weights need to be trained for each training sample, leading to high computational cost (SOLTANI; BENMOHAMED; LTIFI, 2023). In order to overcome these shortcomings of RNNs, Reservoir Computing (RC) framework has been proposed.

RC is a type of RNN, that is aroused as result of unification of several approaches in literature for time series prediction. According to (SOLTANI; BENMOHAMED; LTIFI, 2023), a reservoir computing system consists of two main components: 1) a reservoir, with a fixed number of artificial neurons, that maps inputs into a high-dimensional space and 2) a readout for pattern analysis that use the information from the high-dimensional states in the reservoir. The main features relies on the simplicity of its training phase because only the output weights are trained (CUCCHI *et al.*, 2022; SOLTANI; BENMOHAMED; LTIFI, 2023) with a simple method such as

linear regression and classification, making it computationally inexpensive and thus avoiding the well-known problem of the disappearance of the gradient (REHMER; KROLL, 2020; SOLTANI; BENMOHAMED; LTIFI, 2023) that other types of recurrent and feedforward networks present. Liquid State Machine (LSM), Backpropagation Decorrelation Neural Networks and Echo State Network model (ESN) are main types of RC. ESN model differs from LSM model in the type of neurons that is used; the difference between ESN and Backpropagation Decorrelation Neural Network is the feedback connections from the output layer into hidden layer, and output layer itself do not usually used (CUCCHI *et al.*, 2022).

ESN model is suitable for non-linear approximation problems such as: Identification Systems (SCHWEDERSKY; FLESCH; DANGUI, 2022), Time Series Prediction (ZHENG *et al.*, 2020), Pattern Recognition (JAMSHIDI; DANESHFAR, 2022), Modeling Neural Plasticity for Classification and Regression (SUN *et al.*, 2022), among others. The reservoir of ESN model is used as a processing layer and is not modified during its training phase. From the reservoir dynamic is obtained the reservoir state, which is determined by the history of its inputs and outputs. For a good performance, this reservoir must satisfy a condition about its dynamics state (echo property, see 2.1.1. This property states that the dynamic state of the reservoir is influenced by the spectral radius, the highest eigenvalue of a matrix, which is a parameter that has a high impact over the performance model and the capacity of good estimations (SUN *et al.*, 2022). Although the ESN model inherits the main benefit of RC techniques (simple training phase), they have been criticized because the configuration about connections between internal units are generated randomly, ie. those connections do not follow a pattern or a connection rule. According to (JAEGER, 2002), the more complex the task to be performed, the more complex the reservoir structure is required. In consequence, when the tasks turn more difficult, the stability of the trained dynamics becomes a critical issue for the ESN training phase (SUN *et al.*, 2022) and a random configurations into internal units of the reservoir becomes inadequate. Besides, an optimal configuration about the reservoir settings is not guaranteed because of its lack of information about how the reservoir works. This is the main reason that has motivated the development of this work.

As an alternative to treat the above mentioned issue, modifications of the reservoir topology have been proposed in literature. In (ARROYO *et al.*, 2020; KITAYAMA, 2022; JUNIOR; STELZER; ZHAO, 2020), the authors have proposed the substitution of classic random network by non-random complex as the reservoir topology in ESN models, like small-world and scale-free networks. In (JUNIOR; STELZER; ZHAO, 2020), the authors have proposed the incorporation of random clusters to the reservoir topology. These changes are inspired in order to imitate some forms and learning mechanisms from the human brain which have hierarchical and distributed structure (HILGETAG; GOULAS, 2020).

In this thesis, we focused on ESN model to analyze time series. Specifically, we are going to carry out this analysis facing two tasks: prediction and classification. Time series analysis

consists in extracting meaningful and valuable information and other features with the purpose of discovering patterns from the time series data. The objective is to build a adequate model to represent the dynamics of the data. For prediction tasks, a model that predicts future values based on previously observed values. Let's consider a three-dimensional dynamical system $x(t), y(t), z(t) \in \mathbb{R}^2$, the time series $x(t), y(t), z(t)$ are known during a fixed interval of time $t \in [0, T]$ for $T \in \mathbb{R}$, $x(t)$ is the input signal and $y(t), z(t)$ are the output signals. The predictor estimates the output values of $y(t)$ and $z(t)$ for a time $t > T$ from knowledge of input $x(t)$. Figure 1 illustrates this problem.

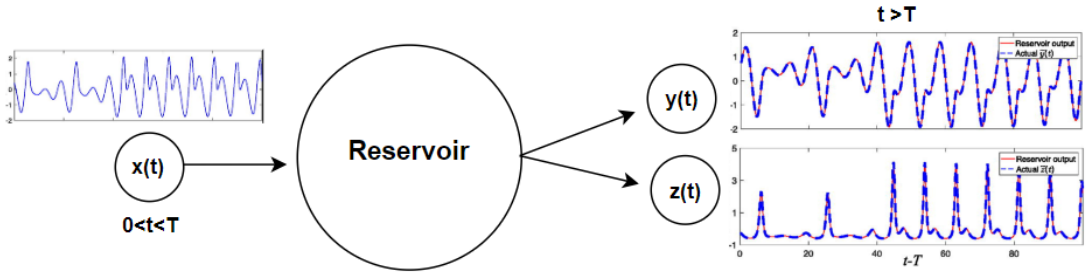


Figure 1 – An illustration about prediction time series task using Reservoir Computing framework. Let's consider a three-dimensional dynamical system $x(t), y(t), z(t) \in \mathbb{R}^2$, the time series $x(t), y(t), z(t)$ are known during a fixed interval of time $t \in [0, T]$ for $T \in \mathbb{R}$, $x(t)$ is the input signal and $y(t), z(t)$ are the output signals. The predictor estimates the output values of $y(t)$ and $z(t)$ for a time $t > T$ from knowledge of input $x(t)$. Elaborated by the author.

Classification tasks aim to train a classifier using a labeled dataset, in order to map from the input space to a probability distribution over the class variable values (labels). In formal terms, for a classification task, let's consider $D = (X_1, Y_1), (X_2, Y_2), \ldots, (X_M, Y_M)$ a collection of pairs (X_i, Y_i) where X_i is a univariate or multivariate time series and Y_i as its corresponding label vector. Without loss of generality, for a dataset with K classes, the label vector Y_i is a vector of length K, where each element $i \in [1, K]$ is equal to 1 if the class of X_i is i and 0 otherwise. The Figure 2 illustrates this problem.

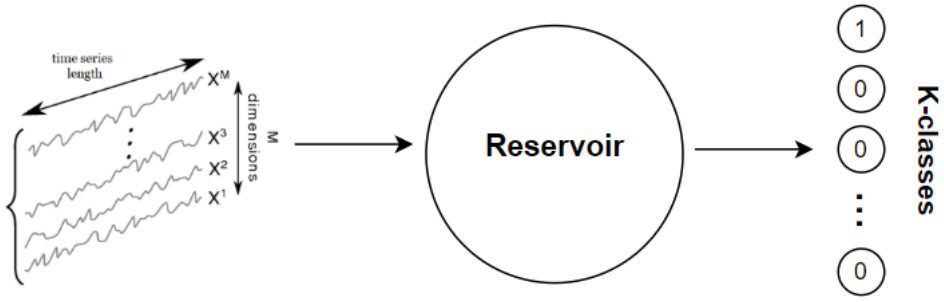


Figure 2 – An illustration about classification time series task using Reservoir Computing framework. Let's consider $D = (X_1, Y_1), (X_2, Y_2), \ldots, (X_M, Y_M)$ a collection of pairs (X_i, Y_i) where X_i is a univariate or multivariate time series and Y_i as its corresponding label vector. For a dataset with K classes, the classifier estimates the output, a label vector Y_i with length K, where each element $i \in [1, K]$ is equal to 1 if the class of X_i is i and 0 otherwise. Elaborated by the author.

1.1 Objectives and Contributions

The main objective of this work is to study ESN models with various reservoir topology. Specifically, we propose to construct the reservoirs using clustered non-random networks in contrast to the Erdös-Rényi random networks in the classic ESNs. Specifically, to generate the clustered reservoirs, we propose to use the classic data clustering algorithms: K-Means, Partitioning Around Medoids, and Ward algorithm to simulate community structures. We also generate the clustered scale-free and small-world networks as reservoirs.

It should be made clear here that scale-free networks, small-world networks and clustered Erdös-Rényi random networks have been applied to reservoir construction and good results have been obtained (ARROYO *et al.*, 2020; KITAYAMA, 2022; JUNIOR; STELZER; ZHAO, 2020). The novelty of this work is the use of non-random networks, such as scale-free networks, small-world networks, and clustered networks generated by applying data clustering algorithms, as reservoirs. Those changes are applied for two type of task: 1) In supervised learning tasks, such as the classification tasks, we need to distinguish various groups (classes) of data each other. We hope that the clusters structure inside of a reservoir can help capture certain features of the input sequences in the training phase to use them in testing phase to face classification task. Moreover, the neural networks of human (animal) brain present clustered structure. Therefore, we are interested in studying the clustered reservoirs in this work and our work is inspired by the biological neural networks. 2) Real-world networks are rarely random ones and the nodes or links in a network have different levels of importance. For this reason, we combine the cluster structure and non-random structure to construct new reservoirs in ESNs.

Still in this work, we use the proposed ESNs to real applications. As we mentioned before, we treat two types of tasks: prediction and classification. As the first application, we apply the proposed model to prediction task in two study cases: Rössler and Lorenz systems. As the second result, we apply the proposed model to human tissue medical image classification problem. The human tissue medical images went through a preprocessing in which they will be converted and treated as time series. For more details of this process see 3.2.2.

Due to the good performance of the ESN in prediction and classification tasks, in this work, we study ESN with new structures of reservoirs, specifically, we study the clustered scale-free and small-world reservoirs. We hope the clustered and non-random structures can help to distinguish training and testing data in an natural way. Consequently, the proposed ESNs can get better performance. This work presents the following contributions:

- We study the ESN with new structures of reservoirs. We show that the ESN with clustered non-random reservoirs can achieve better classification accuracy than the classic one.

- We study how the parameters of the ESN can impact the models' performance. Through our study, a guidance on parameter selection is provided.

- We apply the proposed ESN to predict unmeasured variables in Rössler and Lorenz dynamic systems.
- We apply the proposed ESN to human tissue image classification problem. The good classification accuracy on the large image datasets indicates that the proposed approach is useful for tissue cancer diagnosis.

1.2 Organization of the Document

The remaining sections are organized as it follows. In Section 2, we discuss the relevant concepts, models and techniques. Specifically, descriptions of the classic ESN and complex networks are presented in this section. A detailed explanation about the proposed model is presented in Section 3. Then, in Section 4, the experimental results with the modify architecture of the classic ESN model to prediction task and a comprehensive parameter study is performed and presented. Likewise, the experimental results with the modify architecture of the classic ESN model to classification task and a comprehensive parameter study is performed and presented in Section 5. Finally, in Section 6, the conclusions are presented.

CHAPTER

2

LITERATURE REVIEW OF RELEVANT CONCEPTS, MODELS, AND TECHNIQUES

In this chapter, we present a literature review on the fundamental concepts, models, and techniques, which are relevant to this work. Firstly, we review the classic ESN models and their properties. Then, we present the complex network models, which were applied to extend the classic ESNs in this thesis. After that, we review some classic data clustering algorithms, which be used for reservoir construction in this work. Finally, we give an overview on the human tissue image classification, which is an important application of the extended ESNs.

2.1 Echo State Networks

The Echo State Network (ESN) model was proposed by Jaeger in 2001 (JAEGER, 2002). It is a new and robust type of recurrent neural network that consist of a network of three layers. The hidden layer, called *reservoir*, has a fixed number of neurons known as *reservoir units*. One characteristic of this model is to work with a huge quantity of neurons, providing dynamic into the reservoir. This model works like a coupled system of non-linear functional elements. Each neuron, at each iteration, calculates an output by applying an activation function (generally *tanh*) and the output is feed to other neurons in the reservoir or in the output layer, forming a recurrent loop. Thus, the reservoir processes the data that is passed directly through of the input nodes and model the dynamical of its entries, while the output at each time step is determined by a linear readout. The great advantage of this model is that just the weights of the output layer are trained avoiding the problem of vanishing of the gradient (REHMER; KROLL, 2020).

2.1.1 The Classic ESN and Its Properties

The ESN model is very attractive not only for the inherited benefits of Reservoir Computing approach but because is more simple to implement than others approaches (CUCCHI

et al., 2022). Generally, the previous statement is because ESN model has absence the cyclic connections between neurons in the output layer, transforming the learning phase in a simple regression linear task. The Figure 3 shows a example of a classical ESN architecture.

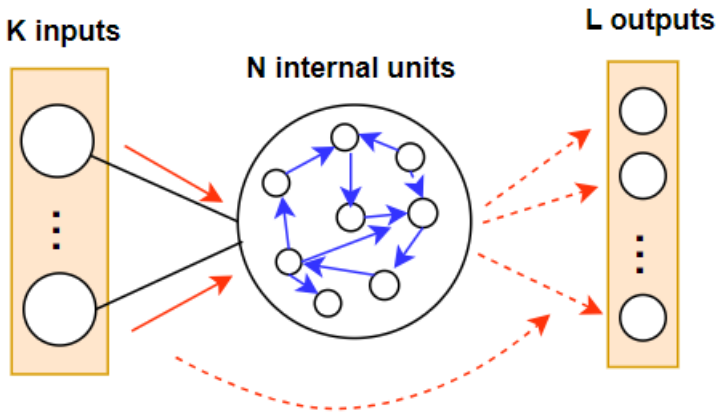


Figure 3 – The basic network architecture of a ESN. Dotted arrows mark connections which are trained. Elaborated by the author.

We are going to consider an ESN constituted by K inputs units $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_K)$, one hidden layer that consists of a reservoir with N internal units, $\mathbf{r} = (r_1, r_2, r_3, \ldots, r_N)$, where $\mathbf{r}$ represents the system state vector at time (iteration) t, and one output layer with L units, $\mathbf{y} = (y_1, y_2, y_3, \ldots, y_L)$. The weight matrix of the model is defined as

- W^{in}: weight matrix from inputs units to internal units: $W^{in} = (w_{ij}^{in}) \in Mat(\mathbb{R}^{NxK})$
- W : weight matrix between internal units or reservoir matrix: $W = (w_{ij}) \in Mat(\mathbb{R}^{NxN})$
- W^{out}: weight matrix from internal units to output units: $W^{out} = (w_{ij}^{out}) \in Mat(\mathbb{R}^{LxN})$

If there exists feedback into the network, we denote W^{back} as weight matrix of these connections. In order to make the ESN model to work, the dynamics of the reservoir must be damped. This condition is possible if the reservoir matrix satisfies the echo state property (JAEGER, 2002). According to (JAEGER, 2002), we define the previous assumption in the next definition.

A network has the echo state property if the current state of the network is only determined by the values of the past inputs and outputs. In other words, for each internal unit of the reservoir, r_i, there is a e_i function, called an echo function, which maps the input/output pair (x_i, y_i) of its history to the current state. Thus the current state for one reservoir's unit $r_i(t)$ at the time instance t is given by the equation (2.1)

$$r_i(t) = e_i((x(t), y(t)), (x(t-1), y(t-1)), \ldots) \tag{2.1}$$

The echo state property is connected to algebraic properties of the reservoir matrix. In practice, it is difficult to guarantee the echo state property in a neural network, instead, it is useful to know when this property is not satisfied. In (JAEGER, 2002), Jaeger proposes a sufficient condition for the non-existence of echo states. This condition is based on the fact that if the corresponding reservoir matrix of an ESN model has a spectral radius greater than unity, the neural network does not have the echo states. The spectral radius of a matrix A, described in equation 2.2, is defined as the highest eigenvalue of a matrix.

$$\rho(A) := \max_i(|\lambda_i|), \quad \lambda_i \text{ is eigenvalue of } A \tag{2.2}$$

In consequence, the choice of the spectral radius ρ is very important for the success of ESN training. Small values of ρ means a fast dynamic reservoir and large values of ρ (close to the unity) mean slow dynamic reservoir.

The activation function for the internal units of the reservoir is given by:

$$\mathbf{r}(t+1) = (1-\alpha)\mathbf{r}(t) + \alpha\mathbf{f}(W\mathbf{r}(t) + W_{in}\mathbf{x}(t) + \zeta) \tag{2.3}$$

where $0 < \alpha \leq 1$ is the “leakage rate", which causes the reservoir to evolve more slowly as $\alpha \longrightarrow 0$, ζ represents the bias and the function $f = tanh$ is the activation function of neurons. The output of the network is given by:

$$\mathbf{y}(t) = W_{out}\mathbf{r}(t) + c \tag{2.4}$$

The model presented here is inspired by the the network designed in (ARROYO *et al.*, 2020). As a solution to the randomness problem in the classical ESN model, several studies have been conducted: unsupervised initialization of the input connections using the K-means algorithm on the training data (STEINER; JALALVAND; BIRKHOLZ, 2021) and modification of the reservoir’s structure such as Multiple Reservoir structure model (LI; LIU; TANAKA, 2023). In the last one, authors created a model called an HP-MRESN, this model consist on a combination between a multi-reservoir, MRESN, with the Hodrick–Prescott (HP) filter for nonlinear time series prediction. The proposed model comprises three basic components: a time series decomposer, a reservoir state extractor, and an ensemble decoder.

According to (JAEGER, 2002), the effectiveness of the ESN is strongly influenced by the size of the reservoir generating a disadvantage as the high computational cost. Some references show a better performance of the ESN model than classical when modified the reservoir topology (ARROYO *et al.*, 2020; KITAYAMA, 2022; JUNIOR; STELZER; ZHAO, 2020). In (ARROYO *et al.*, 2020), complex networks like scale-free and small-world are used as reservoir topology instead of random networks. In (JUNIOR; STELZER; ZHAO, 2020), clustered networks as reservoirs have been proposed using two complex networks configurations as reservoir: erdös renyi network and barabási network.

2.1.2 ESNs for Signal Prediction

For prediction task, many approaches have been develop, such as the dynamic ensemble deep Echo state networks (ESN), proposed by (GAO *et al.*, 2023), in order to learn the dynamic characteristics of a wave height from input signal to the network trough a representation created by the dynamic ensemble ESN. In this work, an architecture for many reservoirs is built in a hierarchical order. For training, an independent readout module is used for each reservoir, integrating all readout layers using a dynamic ensemble block. Another approach, develop in (VIEHWEG; WORTHMANN; MäDER, 2023), researchers focused in demostrate design guidelines for the hyper-parameter optimization ESNs and its impact in the performance. For the experiments, the work was applied for the prediction of chaotic time series. This work provides a guideline about the hyper-parameter configurations helping researchers when training new models.

On the other hand, in (GHIMIRE *et al.*, 2023), the authors proposed a hybrid combination technique, which consist on mixture between two types of netwroks: Convolutional Neural Networks, CNN, and Echo State Network, ESN, called CESN. For the experiments, the parallel architecture was applied in the Daily electricity demand data from four sites (Roderick, Rocklea, Hemmant and Carpendale), located in Southeast Queensland, Australia. The proposed nodel was compared with other state-of-art methods. In (DING *et al.*, 2023), a serial-parallel dynamic echo state network, called (SP-ESN), with dual dynamic characteristics is proposed. This structure can capture the short-term memory information from the input sequence with the auto-correction ability of the prediction error. For the experiments, five wind speed datasets were used from Ningxia Hui Autonomous Region and Qinghai Province of China. For more approaches and applications of ESNs for signal prediction, see (SOLTANI; BENMOHAMED; LTIFI, 2023)

2.2 Complex Networks

The concept of complex network comes from the extension of the graph (Newman, 2010). A graph $G(V,E)$ is defined as a non empty set with V vertices which are connected to each other, according to a configuration, through a set of elements E called edges (Newman, 2010). A complex network is a large scale graph, where the topology that governs the connections configuration is not trivial (Newman, 2010). It is said that a complex network is directed if the links have selected directions, otherwise it is undirected. Figure 4 shows an example of a directed network. Complex network is a interesting area which has been originated in the middle of years 1930 in the branch of sociology and it can be used to represent many real-world systems, some examples are Internet, biological neural networks, chains food among others (Barabási; Pósfai, 2016). As follows, we describe briefly the most common models of complex networks.

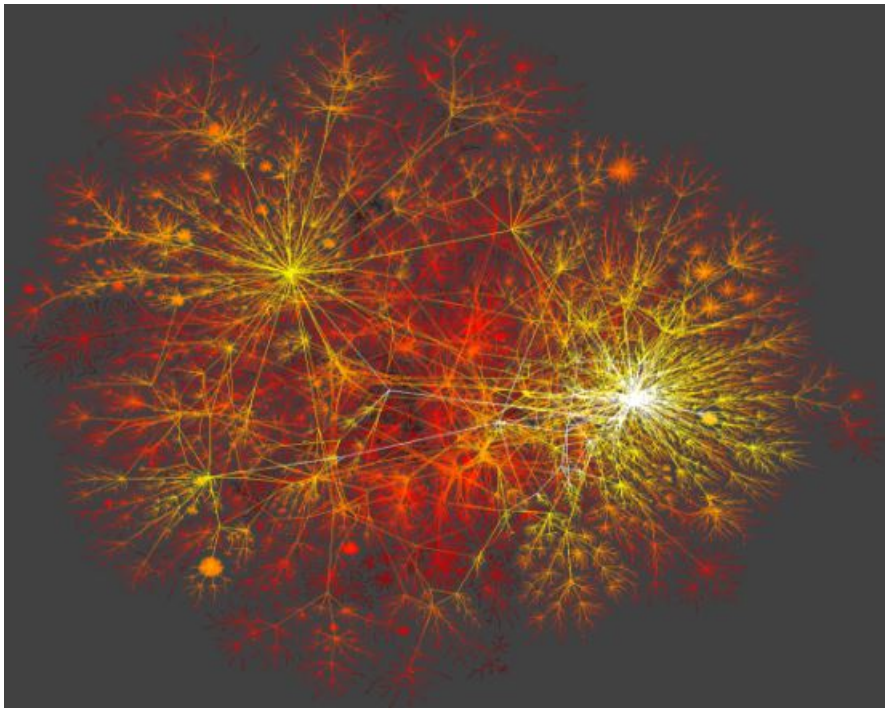


Figure 4 – A macroscopic snapshot of internet connectivity (Barabási; Pósfai, 2016).

2.2.1 Random Networks

Random networks are introduced for the first time by Anatol Rapoport, a Russian immigrant to the United States (Barabási; Pósfai, 2016). However, the study of random networks reached prominence thanks to the fundamental work of the hungarian mathematicians, Pál Erdös and Alfréd Rényi, who have played an important role in understanding the properties of these networks (Barabási; Pósfai, 2016). In their honor, commonly, the random networks are called the Erdös-Rényi networks. A random network is a set with N nodes and connections between the nodes are randomly setup, i.e. each pair of nodes is connected with probability p. According to (Barabási; Pósfai, 2016), there are two definitions of a random network:

- $G(N,L)$ Model: N labeled nodes are connected with L randomly placed links. Erdös and Rényi used this definition in their string of papers on random networks.

- $G(N,p)$ Model: Each pair of N labeled nodes is connected with probability p.

Hence, the $G(N,p)$ model fixes the probability p that two nodes are connected and $G(N,L)$ model fixes the total number of links L. While in $G(N,L)$ model the average degree of a node is simply $< k >= 2L/N$, others network characteristics are easier to calculate in the $G(N,p)$ model (Barabási; Pósfai, 2016). The number of links in a random network can vary between realizations because this value is determined by N and p. As p increases, a random network becomes denser (Barabási; Pósfai, 2016).

The degree distribution, p_k, is the probability that any randomly chosen node has degree k and it is given by the following expression (Barabási; Pósfai, 2016)

$$p_k = \binom{N-1}{k} p^k (1-p)^{N-1-k} \tag{2.5}$$

For a particular realization of a random network with a maximum number of links $\frac{N(N-1)}{2}$, the probability of a random network has a L links, denoted p_L, follows a Binomial distribution[1]:

$$p_L = \binom{\frac{N(N-1)}{2}}{L} p^L (1-p)^{\frac{N(N-1)}{2} - L} \tag{2.6}$$

then, the expected number of links, $< L >$, according to 2.6 is given by

$$< L >= p\frac{N(N-1)}{2} \tag{2.7}$$

Despite the fact that some random networks are generated with the same parameters N and p, the realizations can be different as we can see in Figure 5.

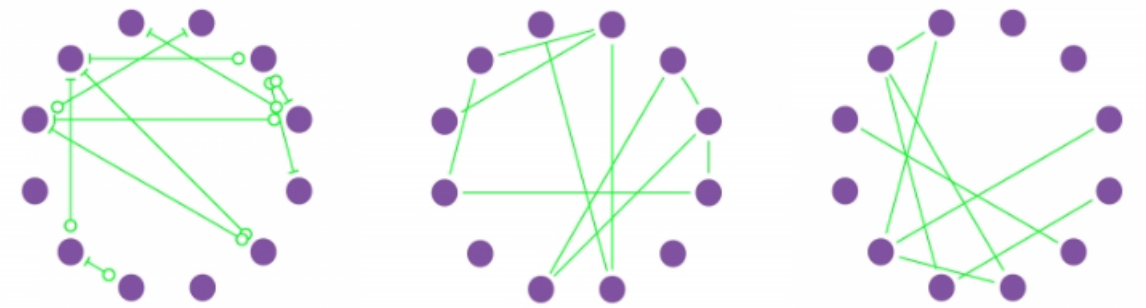


Figure 5 – Three realizations of a random network generated with the same parameters $p = \frac{1}{6}$ and $N = 12$. Despite the identical parameters, the networks not only look different, but they have a different number of links as well $(L = 10, 10, 8)$ (Barabási; Pósfai, 2016).

2.2.2 Scale-Free Networks

Generally speaking, a scale-free network consists of a small number of high degree nodes and a large number of low degree nodes. Scale-free networks are also called Barabási-Albert (BA) networks in honor of the creators (Barabási; Pósfai, 2016). Scale-free networks can be generated using the growth and preferential attachment mechanism, i.e., these networks are in continued growth and tend to link to the more connected nodes, in contrast to the random

[1] <https://www.sciencedirect.com/topics/engineering/binomial-distribution>

network that assumes that the number of nodes is fixed and the interaction between nodes is randomly setup (Barabási; Pósfai, 2016). The degree distribution of a scale-free network follows the *Power Law* (Barabási; Pósfai, 2016). This property says that the fraction p_k of nodes in the network with k connections is according to the expressions below

$$p_k \sim k^{-\lambda}$$

where λ is a parameter that represents power law exponent. Generally $2 < \lambda < 3$. Figure 6 shows how looks the degree distribution of the scale free networks. One common feature of scale-free networks is the presence of *"hubs"*, which are nodes with have high degree to link to another nodes, i.e. nodes have a huge quantity of connections. This last fact gives rise to *"rich-gets-richer phenomenon"* (Barabási; Pósfai, 2016). In the consequence to the preferential attachment, new nodes in the network are more likely to connect to the more connected nodes, hubs, than to the low degree nodes, or nodes with few connections. The generation of a scale-free network starts with m_0 nodes, which are initially randomly linked and each node has at least one link. As mentioned before, the network develops following the concept of growth and preferential attachment, so, at each timestep, a new node is added with m, $(m \leq m_0)$, links that connect it to nodes already in the network. This process is illustrated in Figure 7.

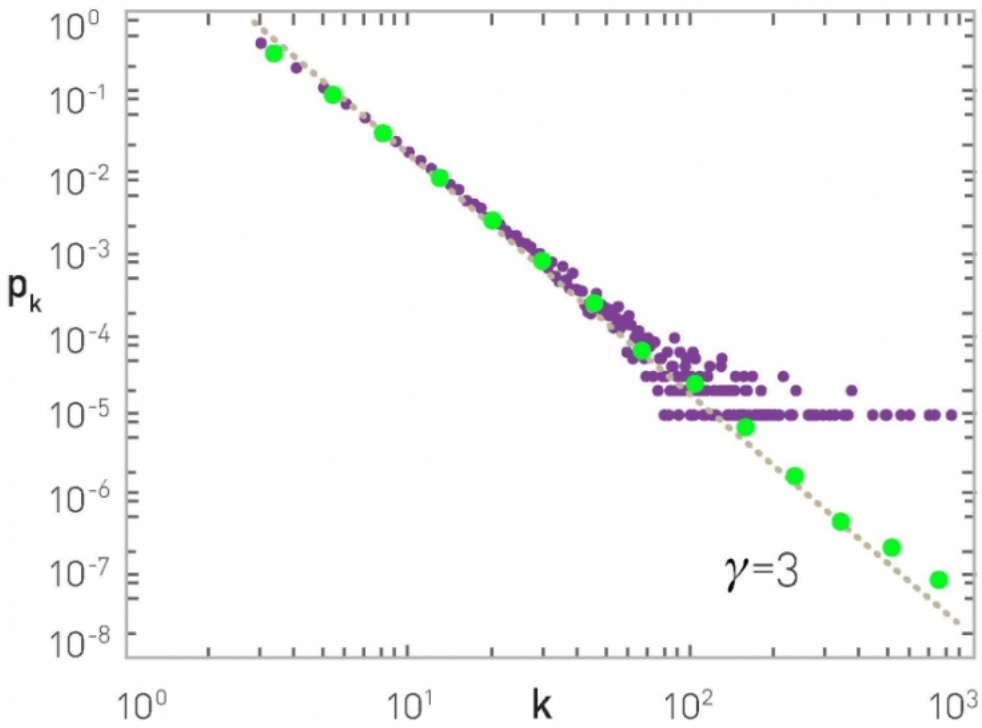


Figure 6 – The degree distribution of a network generated by the Barabási-Albert model. The figure shows p_k for a single network of size $N = 100000$ and $m = 3$ (Barabási; Pósfai, 2016).

The probability of creation of each new link between the newly added node and an particular existing node i depends on the degree k_i and it is given by the following expression

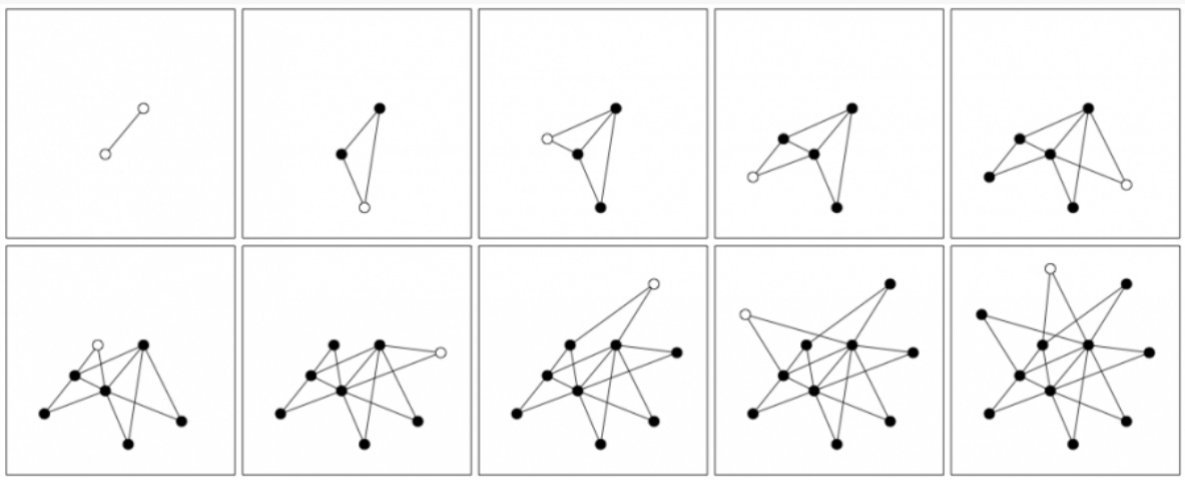


Figure 7 – The sequence of images shows nine subsequent steps of the Barabási-Albert model. Empty circles mark the newly added node to the network. At each stage, two links ($m = 2$) are inserted in the network using the preferential attachment rule.

(Barabási; Pósfai, 2016):

$$\prod(k_i) = \frac{k_i}{\sum_j k_j} \tag{2.8}$$

A particular fact in this type of networks is that it does not specify the precise initial configuration of the first m_0 nodes. Also, it does not specify whether the m links assigned to a new node are added one by one, or simultaneously. For last, this fact result in another mathematical detail because if the links are truly independent, then they could connect to the same node i, creating multi-links (Barabási; Pósfai, 2016).

2.2.3 Small World Networks

Maybe the most famous small-world networks are a social networks. Such networks received this name due to the so called small-world phenomenon. The small-world phenomenon, also known as six degrees of separation, states that if you choose any two individuals anywhere on earth, you will find a path of at most six acquaintances between them or said in the language of network science *the distance between two randomly chosen nodes in a network is short* (Newman, 2010). This phenomenon is illustrated in the Figure 8.

The small-world experiment was conducted by Stanley Milgram (Barabási; Pósfai, 2016; Newman, 2010). The objective of this experiment was examine the average path length for social networks of people in the United States. The fact that individuals who live in the same city are separated by a few handshakes from each other is no means surprising. The really fascinating thing in this network is that even individuals who are on the opposite side of the country can be connected to us by only a few acquaintances.

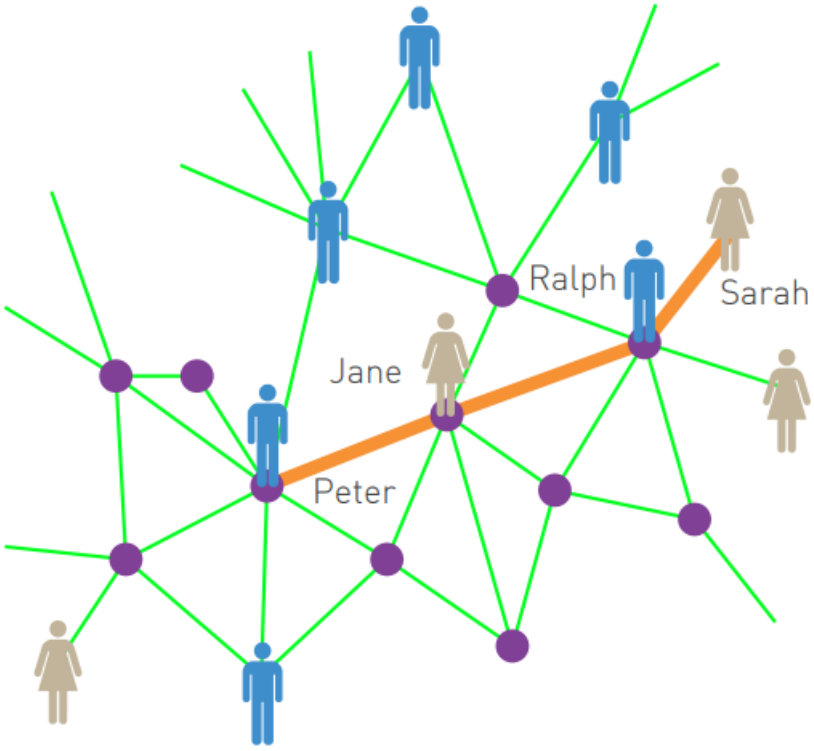


Figure 8 – According to six degrees of separation two individuals, anywhere in the world, can be connected through a chain of six or fewer acquaintances. This means that while Sarah does not know Peter, she knows Ralph, who knows Jane and who in turn knows Peter. Hence Sarah is three handshakes, or three degrees from Peter. In the language of network science six degrees, also called the small world property, means that the distance between any two nodes in a network is unexpectedly small (Barabási; Pósfai, 2016).

A method for constructing artificial small-world networks has been proposed in 1998 by Watts e Strogatz (WATTS; STROGATZ, 1998). In the construction process, the nodes in a network are initially only connected to their respective neighbors. Then, a small portion of linked are randomly changed, resulting in a small-world network. Mathematically, the small-world property states that, in a network of N nodes, the average distance between each pair nodes, $< d >$, is proportional to $\ln N$, rather than N or some power of N (WATTS; STROGATZ, 1998). In other words, it means that the average path length or the diameter of the network depends logarithmically on the network size. This fact is denoted through the expression 2.9 (WATTS; STROGATZ, 1998).

$$< d > \approx \frac{\ln N}{\ln < k >} \tag{2.9}$$

The expression 2.9 evidences the dependence of the average distance in a network on N and $< k >$. In the Table 1 shows the average distance for some well-known networks.

Network	N	L	$< k >$	$< d >$	d_{max}	$\frac{lnN}{ln < k >}$
Internet	192,244	609,066	6.34	6.98	26	6.58
WWW	325,729	1,497,134	4.60	11.27	93	8.31
Power Grid	4,941	6,594	2.67	18.99	46	8.66
Mobile Phone Calls	36,595	91,826	2.51	11.72	39	11.42
Email	57,194	103,731	1.81	5.88	18	18.4
Sciencie Collaboration	23,133	93,439	8.08	5.35	15	4.81
Actor Network	702,388	29,397,908	83.71	3.91	14	3.04
Citation Network	449,673	4,707,958	10.43	11.21	42	5.55
E. Coli Metabolism	1,039	5,802	5.58	2.98	8	4.04
Protein Interactions	2,018	2,930	2.90	5.61	14	7.14

Table 1 – Average distance for most commonly networks (Barabási; Pósfai, 2016).

2.2.4 Clustered Complex Networks

Usually, random clustered networks can be generated in the following way: Given N and M are number of nodes and number of clusters, respectively. The network is generated by the following rule: A pair of nodes is connected with probability p_{in} if they are in the same community, whereas a pair of nodes belonging to different groups is connected with probability p_{out}. The connections are made one by one until the network reaches a pre-defined average degree $< k >$ (Barabási; Pósfai, 2016).

One of the salient features of complex networks is the presence of communities. A community is a densely connected sub-network, while the the connections between different communities are relatively sparse. According to (Newman, 2010), there exist basically two general techniques to divide a network into groups or clusters: *graph partitioning* and *community detection* algorithms. Both of these techniques focus on the way how vertices in a network are associated in order to generate clusters. Graph partitioning is a classic problem that consists in dividing the vertices in a network into a given number of non-overlapping clusters with equal size with the number of inter-cluster edges being minimized (Newman, 2010). The main characteristic of this technique is the fact that the number and sizes of the clusters are fixed. On the other hand, the objective of community detection is to find a natural way to divide a network. The sizes of the clusters are not necessary fixed or predetermined and might, in principle, vary widely from one cluster to another (Newman, 2010). A given network might be divided into a few large cluster, many small ones, or a mixture of all different sizes. In other words, graph partition aims to find out an optimal division of the nodes in the underlying network, while community detection techniques usually provide efficient and sub-optimal solutions.

Community detection problems differ from graph partitioning in one simple thing: the number and size of the groups into which the network is divided are not specified, instead it is determined by the network itself. There are many algorithms to community detection clustering in complex networks: modularity based community detection such as *simple modularity maxi-*

mization, spectral modularity maximization, between based methods and hierarchical clustering. For a comprehensive review, see (ROSTAMI *et al.*, 2023).

As mentioned in the section 1, this work aims to replace the classical reservoir by clustered complex networks, analyzing their performance in relation to the number of clusters with the purpose to understand the role that clustered complex network plays as reservoirs and their relationship with the functioning of the ESN models. Community detection clustering makes sense to understand the natural way in which neurons group together to perform different tasks. However, we did not generate clustered networks as reservoirs using community detection algorithms. We generated a set of data points, apply data clustering algorithms, and construct the clustered networks taking into account the detected clusters of the data points. For this purpose, we applied several techniques to cluster the set of data points but we focused on those with better results in the model performance: three different data clustering algorithms, K-means, K-medoids and Ward algorithm, which will be reviewed in the next subsection.

2.3 Data Clustering Algorithms

In this section, we present the data clustering algorithms, which were used for clustered reservoir construction.

2.3.1 K-means Algorithm

K-means is one of a simple and popular unsupervised machine learning algorithms for data clustering and pattern reconigtion. Given a dataset, the purpose of K-means is to group similar data samples together and discover data patterns. The technique needs to initially setup a hyperparameter: the number of clusters or groups, K. Every data point is allocated to each of the clusters through reducing the in-cluster sum of square, in other words, to the nearest cluster. The “means" in the K-means refers to averaging of the data; that is, finding the centroid. A centroid is the representative of the center of the cluster. The Figure 9 illustrates the kmeans algorithm process.

For the construction of the reservoirs of the proposed model based on clustered networks which clusters were generated using kmeans algorithm, we have used within the implementation the available and updated modules of scikit learn[2]. As this algorithm usually depends on the initial points that are chosen as centroids, as well as on the difference in the choice of the initial samples, this leads to different results, for this reason, for the implementation (see 4.3), the method to the initialization was setup as “kmeans++", this choice selects initial cluster centroids using sampling based on an empirical probability distribution of the points’ contribution to the overall inertia. This technique speeds up convergence[2].

[2] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

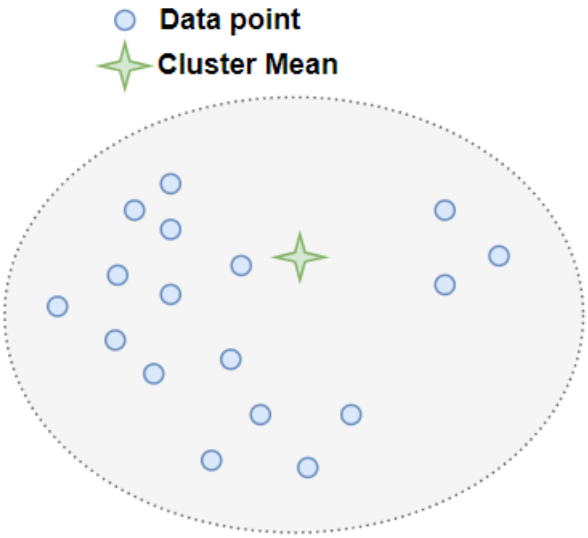


Figure 9 – An illustration about K-means clustering process. The centroid (green point) is the representative point of each cluster. The K-means algorithm assigns each incoming data point to one of the clusters by minimizing the sum in the cluster of squares.

2.3.2 K-medoids Algorithm

This algorithm is also called Partitioning Around Medoids (PAM). K-medoids algorithm is a clustering algorithm. K-medoids is more robust compared to K-means, as in K-medoids we find k as representative object to minimize the sum of dissimilarities of data objects whereas, K-Means used sum of squared Euclidean distances for data objects. The process to update the centroids is based in a computation of a loss function. The K-medoids algorithm is used to find medoids in a cluster which is centre located point of a cluster. The idea of K-medoids differs from K-means because the centroids are actual data-points. The Figure 10 illustrates how the kmedoids algorithm process is developed.

For the construction of the reservoirs of the proposed model based on clustered networks which clusters were generated using kmedoids algorithm, we have used within the implementation the available and updated modules of scikit learn extra[3]. For the implementation, the specific method to the initialization was setup as “heuristic++", this choice picks the n clusters points with the smallest sum distance to every other point and the metric used is the default metric used by the algorithm in the module.

2.3.3 Ward Algorithm

Ward algorithm is a hierarchical agglomerative clustering technique. This algorithm takes pairs of clusters and recursively merges them minimally increases within-cluster variance. Initially each data point is considered as an individual cluster. At each iteration, the similar

[3] https://scikit-learn-extra.readthedocs.io/en/stable/generated/sklearn_extra.cluster.KMedoids.html

Figure 10 – An illustration about K-medoids clustering process. Instead of calculating the mean of each group to determine its centroid, the medoid (blue point inside red triangle) is calculated. The medoid is the data of the group whose sum of the distances to the elements of the group is the smallest. The medoids are not new points in the space, but are chosen from the data itself. In contrast to K-Means, K-Medoids chooses data points as centroids.

clusters merge with other clusters until one cluster or K clusters are formed. Ward's minimum variance criterion minimizes the total within-cluster variance and this criterion determine if divide or cluster two previous groups. To implement this method, at each step, it finds the pair of clusters that leads to minimum increase in total within-cluster variance after merging. This increase is a weighted squared distance between cluster centers. To apply a recursive algorithm under a loss function, the initial distance between individual objects must be (proportional to) the squared Euclidean distance. The Figure 11 illustrates how the ward algorithm process is developed.

For the construction of the reservoirs of the proposed model based on clustered networks which clusters were generated using Ward algorithm, we have used within the implementation the available and updated modules of scikit learn extra[4].

2.4 Machine Learning for Human Tissue Image Classification

The classification of human tissues has become a major step in the field of medicine and biomedical research, as it enables the accurate diagnosis of diseases, the identification of molecular markers, and the selection of appropriate therapies (FITZGERALD *et al.*, 2021). In recent years, the application of artificial intelligence (AI) in human tissue classification has gained significant interest due to its potential to improve the accuracy and efficiency of this process. Such an approach can potentially save many life by performing early and automatic

[4] https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.ward.html

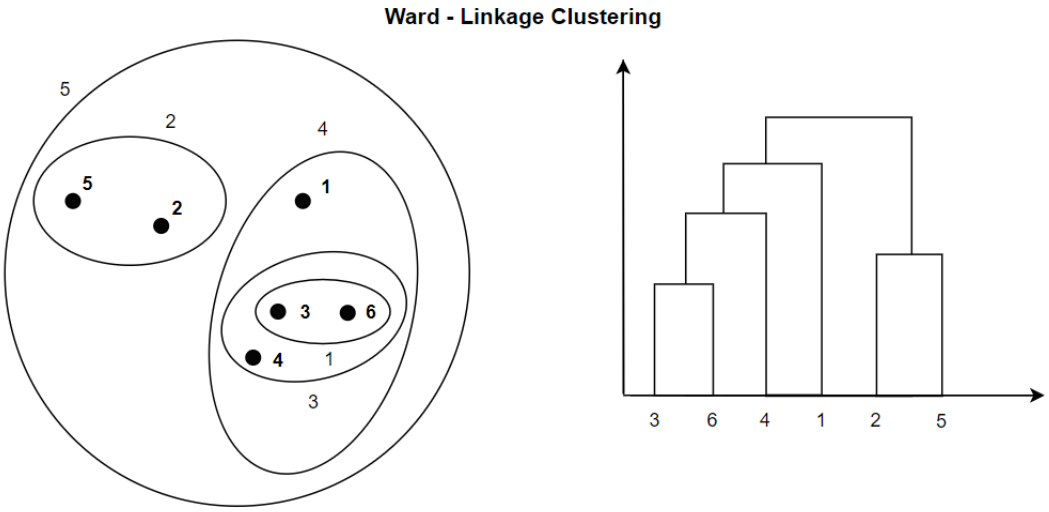


Figure 11 – An illustration about Ward clustering process. Initially each data point is considered as an individual cluster. At each iteration, the similar clusters merge with other clusters until one cluster or K clusters are formed. Ward's minimum variance criterion minimizes the total within-cluster variance and this criterion determine if divide or cluster two previous groups.

diagnosis. It can also facilitate and support the work of the people in charge of this process. In recent years, machine learning techniques have been widely used to develop tasks such as classification, detection, segmentation, registration and feature enhancement in diverse images, such as brain, breast, chest, eye and skin (JIANG *et al.*, 2023).

A wide variety of frameworks have been proposed in order to face tasks as the digital image classification, as the design of ESN structure for features extraction and segmentation of tomography images. The work developed in (JYOTHI; SINGH, 2023) focused on studies done in brain MRI. A survey provided an overview about recent works developed in brain tumor segmentation. The systematic literature review made present three approaches: classical scheme, machine learning strategy, and deep learning methodology meant for tumor segmentation. The overview raises a discussion about the basic convolutional neural network architecture, datasets, and the existing deep learning techniques for tissue segmentation coupled with classification. Another researchers were collected in the survey about Transformers in medical imaging were developed in (SHAMSHAD *et al.*, 2023). This manuscript documents an approach for gray scale images segmentation using a ESN. The proposed model is based on multiple features extraction from a single feature per image pixel according to intensity value from the reservoir. In this work, the reservoir reveals hidden image characteristics from the extracted features. For the experiments, the research was applied to segmentate 3D tomography image of a bone revealing details of the bone internal structure.

On the other hand, many techniques with deep neural networks for biomedical imaging have been developed in (NAZIR; DICKSON; AKRAM, 2023). The overview provides a detailed

literature review about applications of Transformers in medical imaging. The research was conducted covering various aspects, ranging from recent works to unsolved issues. The application areas covered using Transformers in medical image were segmentation, detection, classification, restoration, synthesis, registration, clinical report generation, among others. Also, CNNs were applied in Medical Image Analysis, as a described in (KSHATRI; SINGH, 2023). The document provides a description about Explainable IA techniques for biomedical imaging diagnostics. The researches made a categorization of these techniques, discuss open challenges and future works. The main intention of the authors is provided an deep explanation about how the artificial neural networks works and show what's behind the so-called black boxes. For brain tumor segmentation, the authors in (LIU *et al.*, 2023) using deep learning models for image analysis. The study areas of this manuscript covers brain magnetic resonance imaging (MRI) analysis, pre-processing, segmentation, data preparation, and post-processing. For more approaches of medical image classification, see (CHATURVEDI *et al.*, 2022; JIANG *et al.*, 2023).

CHAPTER

3

THE EXTENDED ECHO STATE NETWORKS WITH VARIOUS RESERVOIR TOPOLOGY

The development of this work consists of a design, construction and execution of a novel artificial neural network to solve different tasks: time series prediction and human tissue medical image classification. For each result, the development of the task is performed taking case studies. For the prediction task, the Rössler and Lorenz attractors are used. For the classification task, we were used 3 datasets consisting of one group with approximately $7,000$ images and two groups with $100,000$ images each one, all of them are medical images of human tissues. For both tasks, the extended Echo State Network models are applied as the main tool. In the proposed models, various topological reservoirs are considered with the special attention to the clustered scale-free and small-world networks. In particular, in the case of the medical image classification task, as the classical model is originally designed to work with time series but not images, a pre-processing method is designed to convert the original images to multivariate time series.

For both tasks, the proposed model retains essentially the same strategy: changes in the topology as an alternative to optimize some difficulties presented by the classical ESNs, such as the condition on the spectral radius and the complexity of the task required when working with a large amount of data, as well as the computational costs with large connection matrices in ESNs.

1. To carry out an experimental study on the new topology inserted into the reservoir: non-random topology inspired by complex and clustered networks.

2. To study the influence of the main parameters of the classical model, according to the literature.

3. To apply the proposed model to the study cases comparing its performance with the classical model and in some cases with other architectures.

3.1 ESN Design for Time Series Prediction Task

In this section, we present the modified ESN for time series prediction. We develop two types of approaches: 1) using complex networks and 2) using clustered networks. Both approaches focus on modifying the natural state of the reservoir to generate an alternative topology to perform the same task. Figure 12 shows an scheme of the architecture of the proposed model.

1. **Input Layer:** In this task, we use only one neuron in the input layer corresponding to a $x-$signal of Rössler and Lorenz systems. The input neuron is connected to all the neurons in the next layer (hidden layer).

2. **Hidden Layer:** Also known as the *reservoir layer*. In this layer, we construct the reservoir weight matrix, W, which is inserted in the network to obtain the reservoir state matrix (see Figure 12), $\mathbf{r}$, by applying the equation 2.3. The weight matrix, W, represents the connectivity topology among N reservoir nodes and it will be used to training phase. In this work, we generate the reservoir using the following different configurations. On one hand, the reservoirs are based on complex network structure: random, scale-free and small-world networks. On the other hand, the reservoirs are based on clustered structures. For more details of these configurations see the section 3.1.1.

3. **Training Phase:** After T_1 times, we collect the states in $\mathbf{r}$ obtaining the reservoir state matrix, a $N \times T_1$ matrix, which is used in the training phase. To train the network, we take the target training set and solve the simple linear regression task formed using sklearn modules of pyhton[1].

4. **Output Layer:** This layer has two neurons, one neuron per signal predicted. After training phase, we take the testing set to predict the $y, z-$signals.

3.1.1 Design and Generation of the Reservoir

In this section we detail how the topologies used in this work are generated. The design and generation of each type reservoir used in this work depends on the structure in which it is inspired. For this work, it was considered two types of structure: inspired in complex and clustered networks. In general, to generate the topology of connections between neurons inside of the reservoir, we need to generate the adjacency matrix, which determine how the neurons are connected. Thus, once the adjacency matrix is generated, the reservoir matrix (weights matrix) is obtained by assigning weights (values) to the connections previously established. These weights are uniformly distributed in the interval of $[-1, 1]$, as described in section 4.3.

[1] <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html>

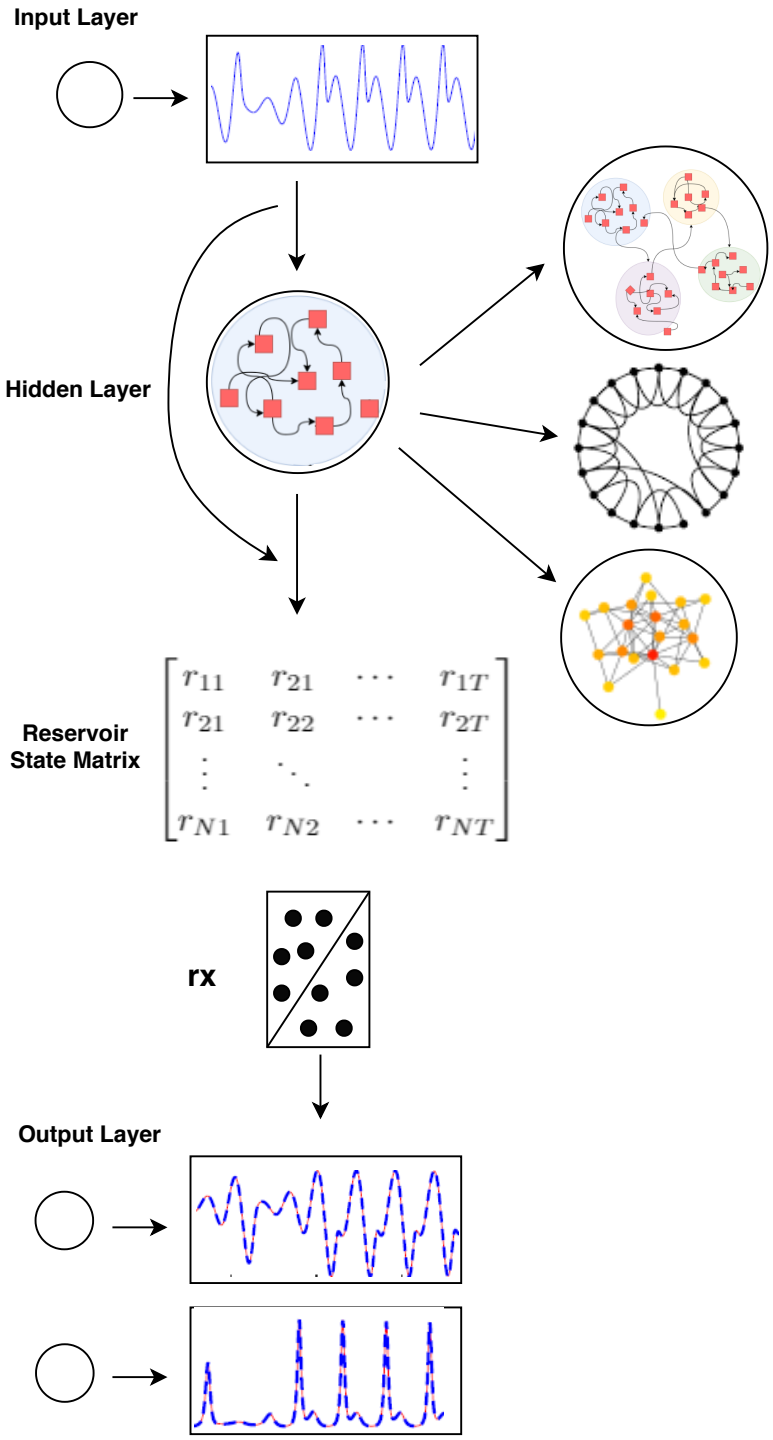


Figure 12 – The implemented neural network architecture. The arcitecture has three layers. The input layers has one neuron, the hidden layer is composed by the reservoir of neurons and the output layer has two neurons, one per signal y and z. In the reservoir layer, the reservoir states matrix is generated, which is used to training phase. Elaborated by the author.

3.1.1.1 Based on Complex Networks Structure

The following topology is inspired in complex networks such as scale free and small-world, chosen for its inherent non-random characteristics: preferential attachment and short path between any two pairs of nodes. For more details about the characteristics of these networks, see the table 2.

1. **Topology based on scale-free configuration:** The adjacency matrix required to obtain the reservoir matrix is originated by generating a scale-free graph. A graph of nodes is initialized with m_0 nodes, at first, they are randomly connected as long as each node has at least one link. Then, at each timestep, a new node is added which decides where to connect its m, $(m < m_0)$, links using preferential attachment, ie., nodes with high degree. For the implementation, we generate a Barabási-Albert graph using Networkx modules[2].

2. **Topology based on small-world configuration:** The adjacency matrix required to obtain the reservoir matrix is originated by generating a small-world graph. Initially, a ring with n nodes is created. Then each node in the ring is connected with its k nearest neighbors ($k-1$ neighbors if k is odd). To introduce the small world property, some edges are replacing by the following rule: for each edge $u-v$ in the underlying n-ring with k nearest neighbors with probability p replace it with a new edge $u-w$ with uniformly random choice of existing node w. For the implementation, we generate a Watts Strogatz small world graph using Networkx modules[3].

Topology/Char	Size	Av. Length	Cluster Coef.
Erdös	400	1.8012	0.1987
Barabási	400	1.9507	0.1760
Small World	400	2.5657	0.3700
Cluster 1	400	4.1878	0.53963
Cluster 2	400	2.8055	0.12705
Cluster 3	400	4.1117	0.45571

Table 2 – Characteristics of complex and clustered networks used in ESN models.

3.1.1.2 Based on Clustered Structure

In this work, to generate clustered topology, we use three clustering algorithm: K-means, K-medoids and Ward to simulate a community structure. The details about generation of the reservoir for each case are the same, only change the clustering technique by those mentioned

[2] <https://networkx.org/documentation/stable/reference/generated/networkx.generators.random_graphs.barabasi_albert_graph.html>

[3] <https://networkx.org/documentation/stable/reference/generated/networkx.generators.random_graphs.watts_strogatz_graph.html>

above respectively. For clustered topology, the generation of the reservoir matrix is described as follows:

1. The data points (from a chosen dataset) are mapped to the grid on the plane. Each data point is related with a node of the generated network.
2. A clustering algorithm (K-means, K-medoids and Ward) is used to cluster the data points.
3. For each cluster, it is established the backbone (centroid) as the main neuron (node) and local neuron for the remaining.
4. For the connections, all local neurons are connected to their respective backbone.
5. Creating interclusters connections: for connections between backbones, the limit is defined by the parameter *nc_inter*. Each backbone is connected to its *nc_inter* closer neighbors.
6. Creating source connections: These connections are between local neurons and its respective backbone. For each cluster, all local neurons are connected to its backbone resulting in a groups of hubs.
7. Creating intraclusters connections: for connections between local neurons, the limit is defined by the parameter *nc_intra* (don't include backbone connection). The attachment rule for local neurons is propose as follows: if *nc_intra* is less than number of neighbors, the local neuron is connected to *nc_intra* closer neighbors, otherwise, the local neuron is fully connected to all neighbors.
8. The neighborhood of a local neuron is defined by the open ball whose radius is the distance between the local neuron to its backbone. This process is illustrated by Figure 13. The red point refers to backbone of a current cluster, black point refers to current node to link and gray points refer to the remaining nodes in a cluster.
9. The adjacency matrix of the graph obtained is extracted and the reservoir matrix (weights matrix) is obtained by assigning weights (values) to the connections previously established. These weights are uniformly distributed in the interval of $[-1,1]$.
10. The reservoir matrix is re-scaled in order to the spectral radius be a value in the interval $(0,1]$.

In the simulations of the prediction task, the clustered reservoirs, generated using the K-means, K-medoids and Ward algorithms, are named as Cluster 1, Cluster 2, and Cluster 3, respectively. Figure 14 shows an example how internal units look like after clustering, the black points refer to backbones. In the Table 2, the characteristics of clustered and complex networks used in the ESN models are presented.

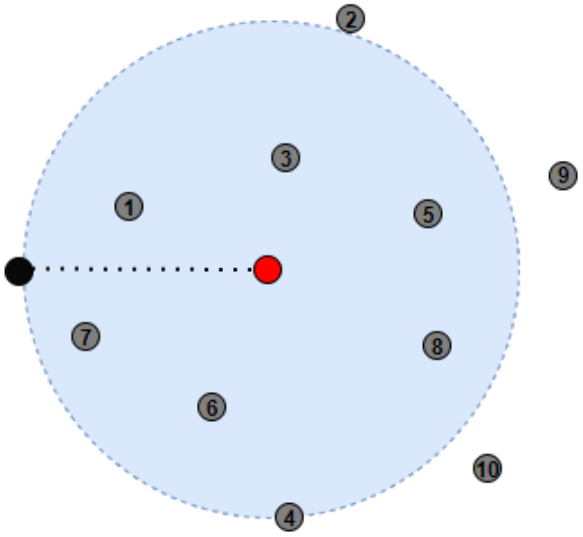


Figure 13 – Illustration of a neighborhood in a clustering process. The neighborhood (blue region) of a local neuron inside a cluster is generated from diameter between the actual local neuron (red circle) and the backbone (black circle). Gray points represents the neighbors of the actual local neuron. The number of connections is setup by the parameter *nc_intra*, then the actual local neuron will be connected with its *nc_intra* closer neighbors. Elaborated by the author.

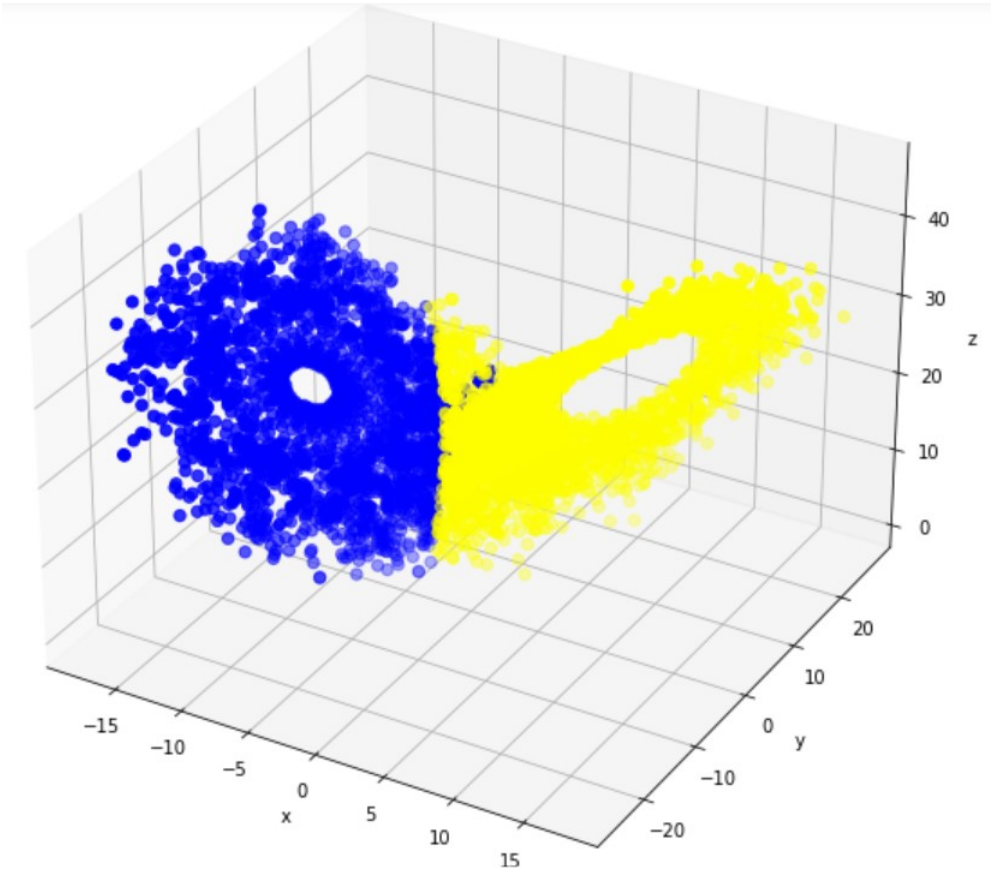


Figure 14 – Clustering internal units for two clusters using K-means. The data points (from a chosen dataset) are mapped to the grid on the plane. Each data point is related with a node of the generated network. Elaborated by the author.

3.2 ESN Design for Classification Task

In this section, we present the modified ESN for human tissue medical image classification. We develop two types of approaches: 1) using complex networks and 2) using clustered networks. Both approaches focus on modifying the natural state of the reservoir to generate an alternative topology to perform the same task. Figure 15 shows an scheme of the architecture of the proposed model.

Firstly, each color channel of the input human tissue image is transformed to a sequence of pixels or superpixels. This process is detailed below in section 3.2.2 by row concatenation from the matrices whose inputs are the RGB values respectively. Then, the sequences are fed to the reservoir.

1. **Input Layer:** The first layer consists of 3 input neurons to receive the RGB channel values of one sample (tile) from the training set. This layer is fully connected to the second layer (reservoir layer).

2. **Hidden Layer:** The reservoir layer has N neurons, called reservoir units with connection weights in the ith column of W_{in}. In this layer, for each sample, we calculate the states, $\mathbf{r_t}$, $t \in [0, 50176]$ of each neuron into the reservoir by applying the equation 2.3. After collect all the states for all $t \in [0, 50176]$, we obtain the reservoir state representation matrix for this sample. Thus, once all the samples have passed through the reservoir, we obtain a set of reservoir states representation matrix, one per sample. From this set, we generate the reservoir final state representation matrix, H, taking the average of the states of each reservoir state representation matrix. H is used to train the network and it is calculated twice, in the same way for training and testing phase. This process is illustrated in Figure 16.

3. **Output Layer:** The last layer contains 9 output neurons. Each output neuron will determine the probability that the image fed into the network belongs to one of the nine tissue classes, specifically, one neuron represents a type of tissues: Normal (NORM), Tumor (TUM), Mucosal (MUC), Muscular (MUS), Debrice (DEB), lymphatic (LYM), stromal (STR), Background (BACK), and Adipose (ADI). Each output neuron shows the probability that the tile (sample) presented to the network belongs to its corresponding tissue type. The neuron with the highest probability determines the type of tissue (class) of the sample.

In the proposed model, we implement a ESN using non-random topology as reservoir (ARROYO *et al.*, 2020). In this proposal, we change the traditional topology by the following different configurations:

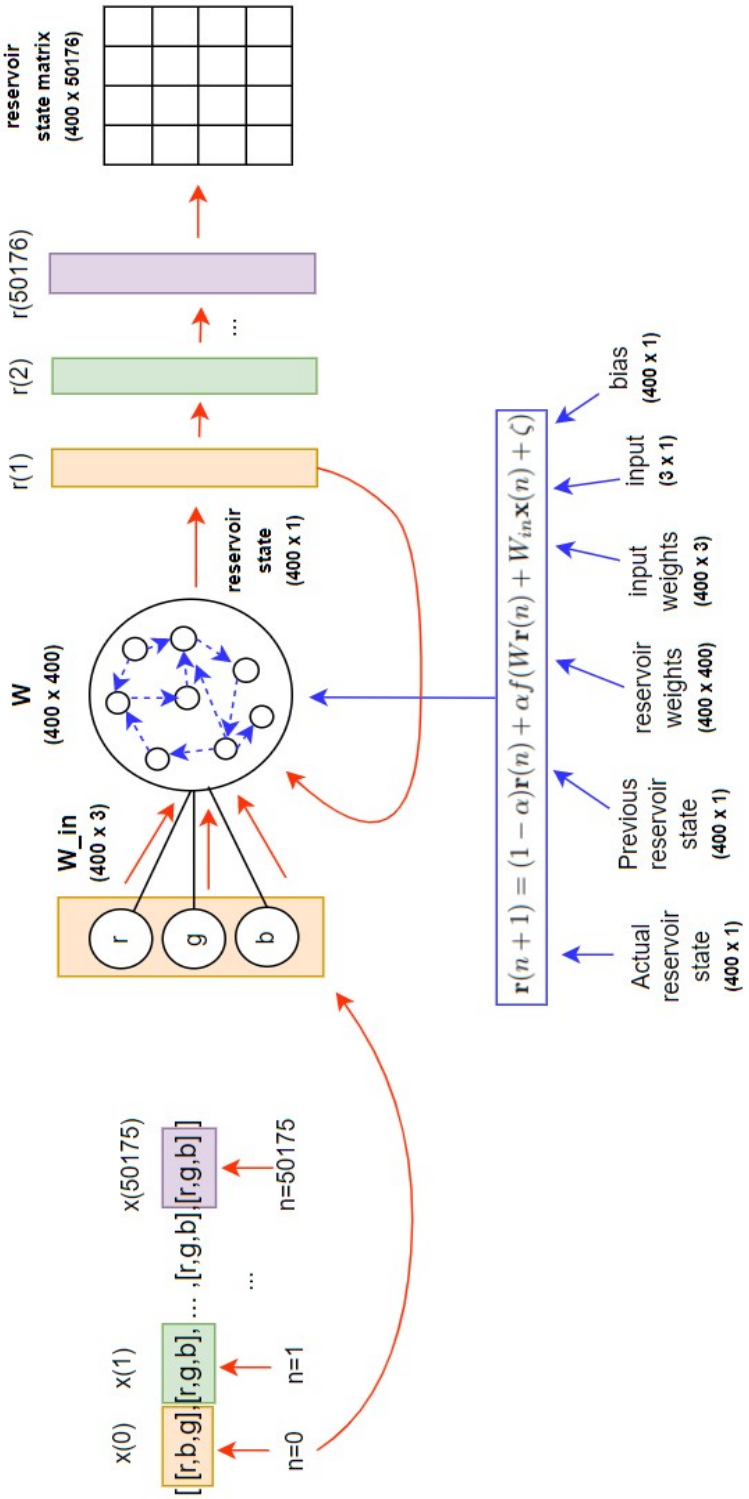


Figure 15 – Dynamic process of the proposed ESN for human tissue image classification. Each color channel of the input human tissue image is transformed to a sequence of pixels by row concatenation from the matrices whose inputs are the RGB values respectively. After, the sequences are fed to the reservoir. The reservoir layer, it was calculated the states for each sample, $\mathbf{r_t}$, $t \in [0, 50176]$. After collect all the states, we obtain the reservoir state representation matrix for this sample. Elaborated by the author.

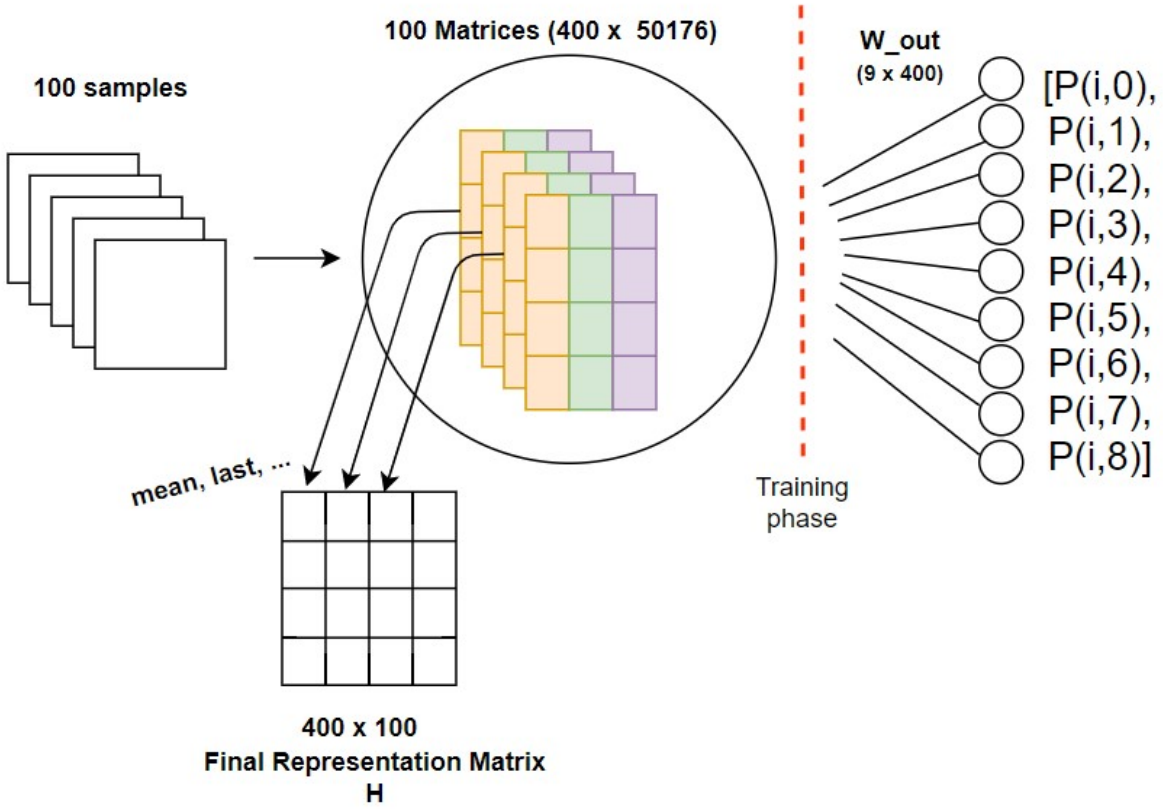


Figure 16 – Process to obtain final representation matrix, H. Once all the samples have passed through the reservoir, we obtain a set of reservoir states representation matrix, one per sample. From this set, we generate the reservoir final state representation matrix, H, taking the average of the states of each reservoir state representation matrix. H is used to train the network and it is calculated twice, in the same way for training and testing phase. Elaborated by the author.

- **Based on complex networks**: To do this, we generate different complex networks and create the weight matrix, W from adjacency matrix of these configurations of connections between neurons. This process is the same to the one described in 3.1.

- **Based on clusters**: This approach consists in generate a fixed number of cluster with three type of configuration of connections based on complex networks: scale free, small world and random. These clusters can be isolated or connected. After the generation of clusters, we obtain the adjacency matrix of these configurations of connections to create the reservoir weights which is inserted into the reservoir layer to calculate the activation of reservoir units of one sample at given time. Figure 17 shows the configuration of the architecture of the proposed model. The networks are generated using Networkx modules[4]. Also, we insert a new parameter, num_inter. This parameter correspond to the number of inter-cluster connections (connections between clusters into the reservoir) and determine the "mixture level" between clusters. The "mixture level" is represent by ϕ and

[4] <https://networkx.github.io/documentation/networkx-1.9.1/reference/generators.html>

is calculated by equation 3.1.

$$\phi(num\_inter) = \frac{num\_inter}{mn^2(m-1)} \tag{3.1}$$

where $mn^2(m-1)$ is a total number of possible connections in a reservoir with m clusters and n reservoir nodes in each cluster. According to the value of num_inter, we create randomly a fixed number of connections between clusters. The weights of these connections are stored in the non diagonal sub-matrices of W with a uniform distribution in the interval $[-1,1]$.

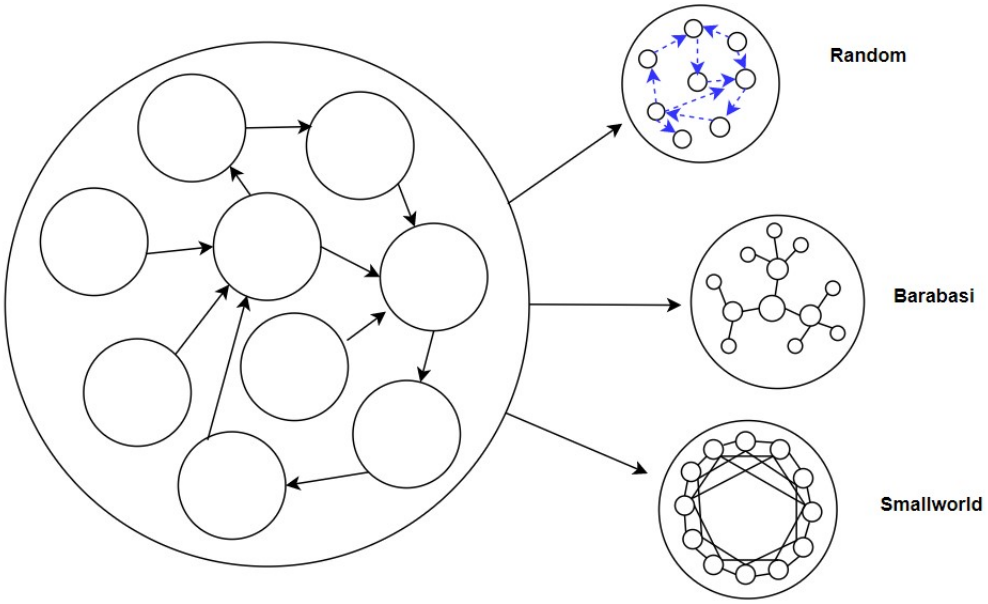


Figure 17 – An illustration about the reservoir topology of the proposed model. The design and generation of each type reservoir used depends on the structure. It was considered two types of structure: inspired in complex and clustered networks. Elaborated by the author.

3.2.1 Design and Generation of the Reservoir

In this section we detail how the topologies used to the classification task are generated. The design and generation of each type reservoir used in this work depends on the structure in which it is inspired. For this work, it was considered two types of structure: inspired in complex and clustered networks. As it was mentioned before in section 3.1.1, to generate the topology of connections between neurons inside of the reservoir, we need to generate the adjacency matrix, which determine how the neurons are connected. Thus, once the adjacency matrix is generated, the reservoir matrix (weights matrix) is obtained by assigning weights (values) to the connections previously established. These weights are uniformly distributed in the interval of $[-1,1]$, as described in section 5.3.

3.2.1.1 Based on Complex Network Structure

The topology is inspired in complex networks such as scale free and small-world, chosen for its inherent non-random characteristics: preferential attachment and short path between any two pairs of nodes. These topology are the same as those described in 3.1.1. Details about the generation of the networks are described as follows:

1. **Topology based on scale-free configuration:** The adjacency matrix required to obtain the reservoir matrix is originated by generating a scale-free network. A network is initialized with m_0 nodes, at first, randomly connected as long as each node has at least one link. Then, at each timestep, a new node is added which decides where to connect its m, $(m < m_0)$, links using preferential attachment, i.e., nodes with high degree. For the implementation, we generate a Barabási-Albert graph using Networkx modules[5].

2. **Topology based on small-world configuration:** The adjacency matrix required to obtain the reservoir matrix is originated by generating a small-world network. Initially, a ring with n nodes is created. Then each node in the ring is connected with its k nearest neighbors ($k-1$ neighbors if k is odd). To introduce the small-world property, some edges are replacing by the following rule: for each edge $u-v$ in the underlying n-ring with k nearest neighbors with probability p replace it with a new edge $u-w$ with uniformly random choice of existing node w. For the implementation, we generate a Watts Strogatz small-world network using Networkx modules[6].

3.2.1.2 Based on Clustered Structure

In this work, to generate clustered topology, we first generate clusters, each one with a topology inspired in the following complex networks: random, scale-free and small-world. The details about generation of the reservoir for each case are the same, only change the complex network. For clustered topology, the generation of the reservoir matrix is described as follows:

1. We first setup the number of clusters of the network. Then, the networks are generated according to the chosen connection configuration (random, scale-free, small-world) and the characteristics previously defined, such as reservoir size and average degree. In the case of defining intercluster connections, the mixing level parameter is set to establish how the connections between the clusters are defined. This parameter can also be set to zero.

2. The adjacency matrix of each network is extracted and the weights are assigned according to the connections previously established. These weights are uniformly values distributed in the interval of $[-1,1]$.

[5] <https://networkx.org/documentation/stable/reference/generated/networkx.generators.random_graphs.barabasi_albert_graph.html>

[6] <https://networkx.org/documentation/stable/reference/generated/networkx.generators.random_graphs.watts_strogatz_graph.html>

3. The adjacency matrix obtained from each generated network represents a submatrix of the reservoir matrix.

4. The reservoir matrix is formed by concatenation of these submatrices. The resulting matrix, the reservoir matrix, is obtained by concatenation of all submatrices. The matrix is a sparse matrix which diagonal are the submatrices.

5. The reservoir matrix is re-scaled in order to the spectral radius be a value in the interval $(0,1]$.

In Table 2, the characteristics of clustered and complex networks used in the ESN models are presented. In the simulations of the image classification task, the clustered reservoirs, generated using the random, scale-free and small-world configurations, are named as Cluster 1, Cluster 2, and Cluster 3, respectively.

3.2.2 Data Preprocessing

Before feeding the training samples to the ESN, the datasets are preprocessed in order to fit the ESN' requirements. The preprocessing process consists of reducing the dimensionality of tiles (in section 5.2, a detailed description on the meaning of tiles in this context is given). In this work, we process the images as a vector of elements, converting it to a time series, where for each instant of time t, $t > 0$, the network receives a vector of 3 elements corresponding to the color channels of the image. For each sample (tile), we convert three-channel images into an $[50176,3]-$array by concatenation. In order to do this, we have studied and implemented many options of concatenation to reduce the dimensionality of the tiles. For illustrating purpose, let's consider a dataset with 100 samples, $N = 400$ reservoir nodes and 224×224 image size (3 channels).

Figures 18 and 19 show the respective row and column concatenation processes. The row and column concatenation methods are very similar. The only difference between them is the direction in which the vector is concatenated. Figures 20 and 21 show the two options of row-column concatenation process, considering both row and column concatenations in the preprocessing process. In Figure 20, the number of samples entering the network is doubled, since each vector is concatenated in both ways (by row and by column). Instead, in the concatenation method shown by Figure 21, we obtain two reservoir state matrices for each sample, one for each type of concatenation. So, at the end, the average of these matrices is calculated to obtain a reservoir state matrix for each sample.

Table 3 shows the ESN performance for different type of data preprocessing applied to $7K$ dataset. In second column, the execution time was registered. This value is given in hours, without processor or GPU accelerator. In this work, we decide to implement the row concatenation process for preparing the tiles to feed to the network. Using this approach, the

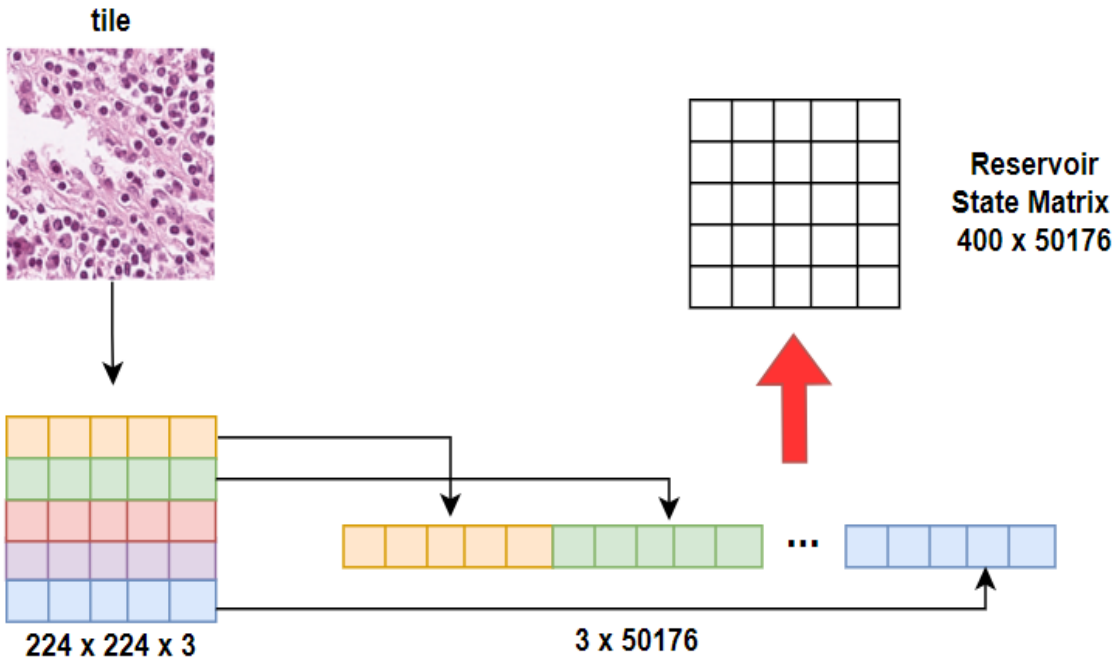


Figure 18 – Row concatenation process. The sample is divided by row and concatenated forming a vector. This process is made three times, one per channel. Elaborated by the author. Elaborated by the author.

network is able to obtain the same information about the input image at a lower computational cost. The data preprocessing is illustrated by flowchart presented in Figure 22.

Process	7K	time (hrs)
row concat	.92	1.83
column-concat	.90	1.84
mix concat 1	.92	3.40
mix concat 2	.93	3.51

Table 3 – ESN Performance for different type of data preprocessing applied to 7K dataset.

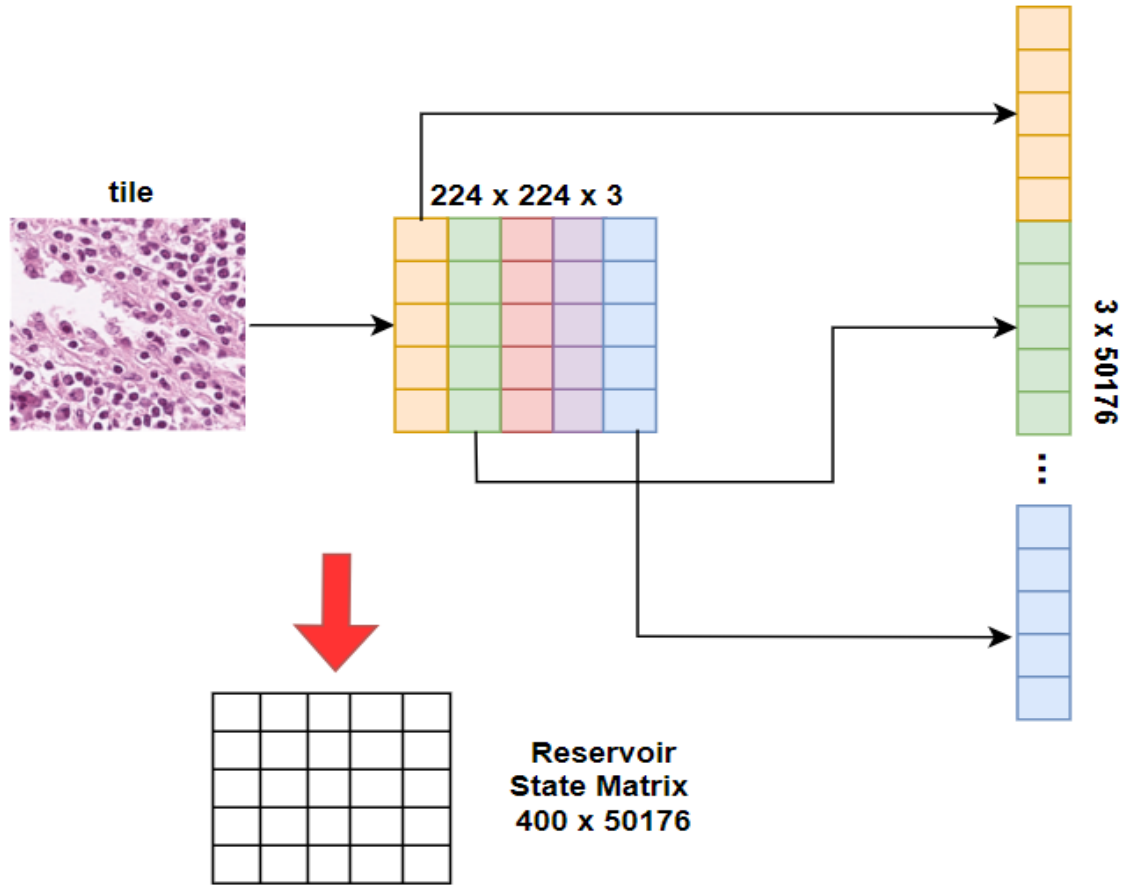


Figure 19 – Column concatenation process. The sample is divided by column and concatenated forming a vector. This process is made three times, one per channel. Elaborated by the author.

Figure 20 – First option of row-column concatenation process. The number of samples entering the network is doubled, since each vector is concatenated in both ways (by row and by column). Elaborated by the author.

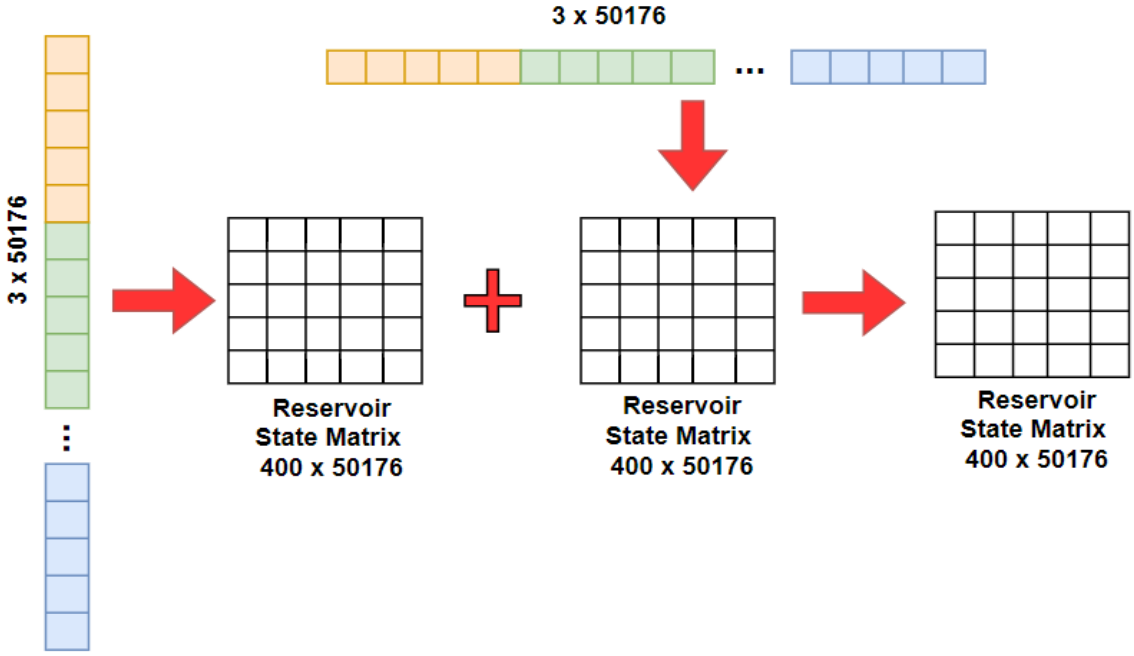


Figure 21 – Second option row-column concatenation process. It was obtained two reservoir state matrices for each sample, one for each type of concatenation. So, at the end, the average of these matrices is calculated to obtain a reservoir state matrix for each sample. Elaborated by the author.

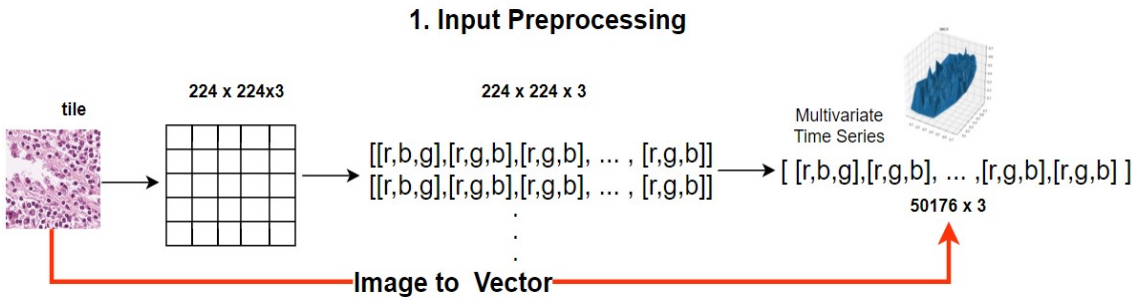


Figure 22 – Data preprocessing flowchart. Elaborated by the author.

CHAPTER

4

THE EXTENDED ECHO STATE NETWORKS APPLIED TO PREDICTION TASKS

In this section, we show the results obtained based on the proposed modifications on ENS models. For this purpose, the classical ESN model is compared with the clustered and complex reservoir topology, considering two study cases: Rössler and Lorenz systems to predict time series. The Rössler and Lorenz dynamic systems are known to possess chaotic attractors or called strange attractors, in the sense that they are highly sensitive to the initial conditions. These systems have been explored for many researches. The Lorenz system is a 3-dimensional ordinary differential equations firstly studied by Edward Lorenz (LORENZ, 1963) and the Rössler system is also composed of three non-linear ordinary differential equations originally created by Otto Rössler (RöSSLER, 1976). The Rössler attractor is similar to the Lorenz attractor, but it is simpler and has only one manifold. Initially, the study cases for the prediction task are described and details of the preparation and implementation of the model are shown. The numerical results are presented in tables and figures.

4.1 Problem Statement

Let's consider two dynamic systems defined in 4.1 and 4.2. The input and output vectors of the system: $x(t)$, $y(t)$, $z(t) \in \mathbb{R}$, $\mathbf{s}(t) \in \mathbb{R}^2$ are known during the interval of time $[0,T]$ para $T \in \mathbb{R}$. The main objective is to estimate the values of $\mathbf{s}(t) = (\mathbf{y}(t), \mathbf{z}(t))$ for a time $t > T$, from knowledge of variable x.

- Rössler System

$$\begin{cases} dx/dt = -y - z, \\ dy/dt = x + ay, \\ dz/dt = b + z(x - c) \end{cases} \tag{4.1}$$

where $a, b, c > 0$ are constants known as bifurcation parameters. Some results about the influence of these parameters are: if a is increased, new periodic orbits are created and the chaotic attractor increases in size. In the case that the parameter b approaches 0, the attractor approaches infinity. For low values of c, the attractor is periodic and it turns out to be chaotic when the value c increases (RöSSLER, 1976).

- Lorenz System

$$\begin{cases} dx/dt = r(y-x), \\ dy/dt = x(s-z), \\ dz/dt = xy - pz \end{cases} \tag{4.2}$$

where $r, s, p > 0$ are constants known as bifurcation parameters and they are system parameters proportional to the Prandtl number [1], Rayleigh number [2], and certain physical dimensions of the layer itself. In the case that $s < 1$ then there is only one equilibrium point, which is the origin.

In this work, we have chosen the parameter values according to those found in the literature because the system exhibits chaotic behavior with these parameter values.

4.2 Dataset Preparation

For the implementation model, we generate two sampling datasets by the implementation of 4^{th} Runge Kutta (using MATLAB):

- **Rossler_delta_0.1.csv**
- **Lorenz_delta_0.1.csv**

In the Rössler case, we use the equation 4.1 where $a = 1/2$, $b = 2$ and $c = 4$. For the Lorenz case, we use the equation 4.2 where $r = 10$, $s = 28$ and $p = -8/3$. The time step used for both implementations is $delta = 0.1$ over the interval $[0, 500]$, resulting in 5000 samples without the initial condition for each dataset. Both datasets are preprocessed (standardize) for each study case. The training and testing sets for two cases are defined as follows:

Training set: the first 1000 samples of each set are discarded in both datasets in order to make the reservoir state essentially independent of its initial state by time $t = 0$, as it has been done in (LU *et al.*, 2017). The training set is formed for two subsets: the subset of samples of the available variable, x, and the subset of samples of the variables to be predicted, y, z. Regarding the x variable, we have 4000 samples as available knowledge and respect to the variables y, z, which we wish to predict, 2000 samples are taken.

[1] <https://www.sciencedirect.com/topics/chemistry/prandtl-number>
[2] <https://www.sciencedirect.com/topics/engineering/rayleigh-number>

Testing set: the remaining 2000 samples of the variables y, z, which we wish to predict, are taken from each dataset.

4.3 Implementation Setup

We implement the ESN model that consists of three layers with 1 input node and 2 output nodes. Connections of the neurons between layers are stored in W_{in}, W, W_{out}, respectively. For the input layer, the ith input signal is connected to all the N reservoir nodes with connection weights in the ith column of W_{in}. Each reservoir node receives input from exactly one input signal. The non-zero elements of W_{in} are randomly chosen from a uniform distribution in $[-1,1]$. The hidden layer is a reservoir of neurons and we are interested in analyzing its connectivity configurations to improve the performance of the model. For this purpose, we incorporate the topology based on clusters and complex networks as connectivity configurations between the neurons into reservoir. The weights stored in W are randomly chosen from an uniform distribution in $[-1,1]$. Also, we scale W matrix in order to guarantee the condition for the echo property. The activation function for the state of neurons is according to the equation 2.3 where $\alpha = 1$ and **bias** $= 1$. For the implementation, we use the same parameters for both datasets. The parameters used are defined in the Table 4. To make comparisons between the models studied, we have fixed the value corresponding to the random factor (seed) in the generation of complex networks and clusters using the sklearn modules and data clustering algorithms.

Parameter	Value
seed	50
initial condition	$[1,1,0]$
number of reservoir nodes	$N = 400$
spectral radius	$\rho = 1$
average degree	$D = 20$
bias constant	$\zeta = 1$
leakage rate	$\alpha = 1$
time step	delta $t = 0.1$
initial time	$T_0 = 100$
initial time of training phase	$T_1 = 260$
final time of predicted phase	$T_2 = 500$
backbone connections	$nc\_inter = 1$
local neuron connections	$nc\_intra = 2$

Table 4 – Configuration setup.

4.4 Experimental Results

The results are divided in two parts: In the first part, we study the performances of the modified models and compared to the results presented in (LU *et al.*, 2017) for predicting the

measures of the variables y, z over the interval $[T_1, T_2]$. In order to show the performance of the proposed model, we use Mean Square Error (MSE), defined in (4.3), like a benchmark measure.

$$MSE := \frac{1}{n}\sum_{i=1}^{n}((Y_{true}) - (Y_{esn}))^2 \tag{4.3}$$

where n is the number of samples entering the network, Y_{true}, Y_{esn} are the true value and predicted value of measured variables, respectively. In order to compare the performance of the proposed model with the classical model, we took the values of the parameters based on (LU *et al.*, 2017), detailed in the subsection 5.3. In the second part, we study how the parameters influence the performance of the proposed ESNs. According to (JAEGER, 2002), the parameter with the greatest impact on the reservoir's dynamics is the spectral radius ρ. So, in this work, we perform some experiments by varying ρ to visualize the error's behavior and confirm its impact on the performance of ESN models. Other parameters studied here are the number of reservoir nodes, N, ridge parameter, β, leakage rate (reservoir memory capacity), α, bias term, **bias**, and number of clusters (*n_clusters*) to check the optimal value of these parameters in the experiments. Table 5 shows the performance of the ESN models of each implemented topology. The main approximation capability of ESN model is concentrated in its reservoir, which evokes periodic signals inside the reservoir dynamic as shown in the Figure 23, some reservoir states are plotted in the testing phase $[260, 500]$. According to the results, it is possible to observe that in both study cases, a non-random topology shows a better performance than the classical ESN. As we can see, in the case of the Rössler dynamical system, better results were obtained using complex network topologies, while in the case of the Lorenz dynamical system, the best results were obtained using topologies with clusters. This is because in each case, the configuration of connections in the reservoir creates a better representation of the input sequence through the states of the neurons. These results confirm that a reservoir with clustered topology could improve the performance of the ESN model.

Topology / Case	Rössler	Lorenz
Erdös	4.17e-7	2.01e-5
Barabási	**3.24e-7**	2.12e-5
Small World	**4.07e-7**	1.87e-5
Cluster 1	1.06e-5	**5.06e-6**
Cluster 2	8.98e-6	**5.890e-6**
Cluster 3	1.08e-5	**4.320e-6**

Table 5 – MSE error for ESN models using network complex as reservoir.

The dynamics of the reservoir is influenced by the parameters of the model, affecting the stability and accuracy of the prediction. The following is an initial experimental analysis of two important and influential parameters according to the literature on the performance of the ESN Model. Figures 24a-24b show the influence of the spectral radius on the performance of the ESN model. From the results, it is possible to note that the approximation capability of the ESN

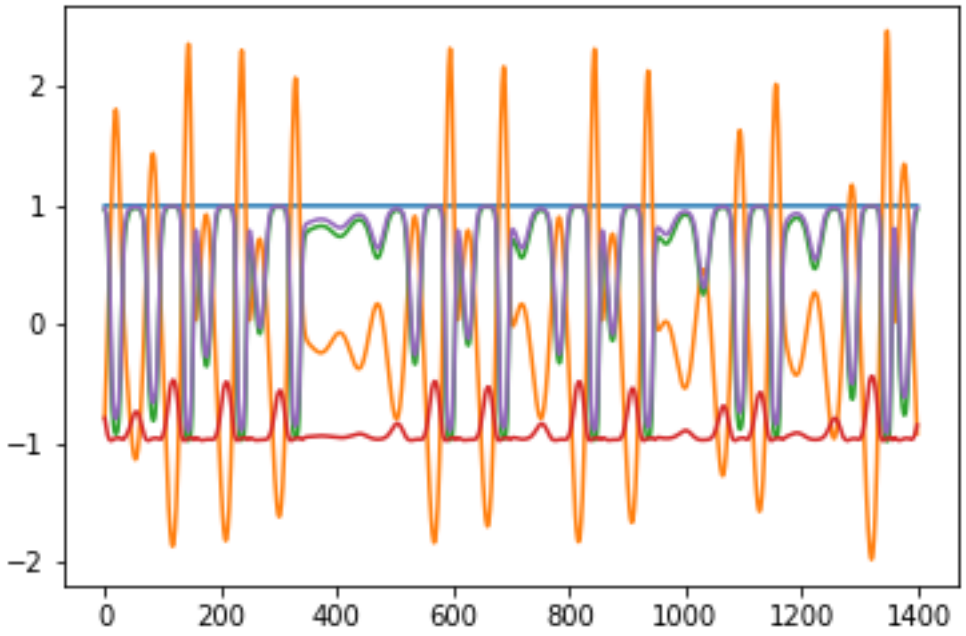


Figure 23 – The main approximation capability of ESN model is concentrated in its reservoir, which evokes periodic signals inside the reservoir dynamic. Illustration of the dynamic reservoir in the ESN model, some reservoir states are plotted in the testing phase $[260, 500]$. Elaborated by the author.

model is considerably affected by the spectral radius value. For complex network topology, if the spectral radius is small, the reservoir dynamic is fast and if it is larger (close to unity), we get a slow reservoir dynamic. For this reason, the best performance is obtained when ρ is closer than 1. An interesting result according to Figures 24a-24b is the fact that the echo property is less influential in clustered topology, allowing a higher range of values for this parameter.

In order to evaluate how much the reservoir's size can influence the performances of the ESN model, we vary the number of reservoir units, N. This information can help to obtain an optimal number of reservoir units to find an adequate estimation of the desired values to be predicted. Figures 24c-24d show the influence of the number of reservoir units in the performance of the ESN model. According to the obtained results, in general, it seems that we can obtain a better performance for all topology when N is large. However, we can obtain good performance with a low amount of neurons in the reservoir $N \sim [250, 750]$ too. In this way, we can avoid using an unnecessarily large amount of reservoir units, which, in turn, can reduce the computational cost.

In the case where we use the clustered topology in the reservoir, Figures 24e-24f show the error behavior of Rössler and Lorenz systems by varying number of clusters. According to the results, we can observe that it is possible to obtain good performance in both cases with a small number of clusters, avoiding computational cost in the weights reservoir matrix formation,

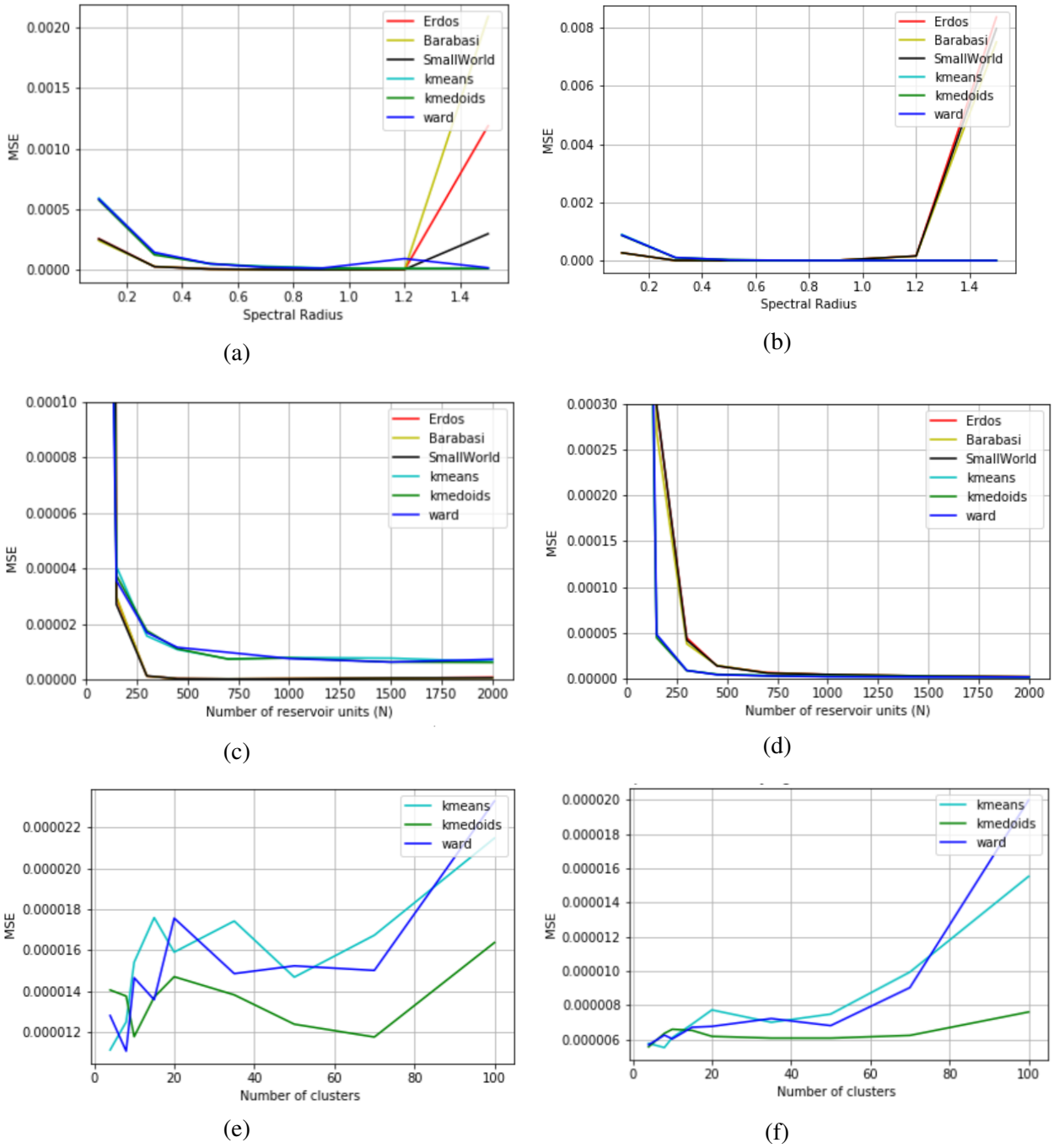


Figure 24 – In the order in which they are presented, ESN models performance analysis varying ρ: 24a - 24b, reservoir size: 24c - 24d, number of clusters: 24e - 24f for Rössler and Lorenz cases respectively. Elaborated by the author.

W. Figures 25a-25f show the influence of parameters α, β, and bias term inside the proposed ESN model.

From the results, we can observe that, in a general way, all the parameters studied have influence on the performance of the proposed model. Specifically, the memory capability in both cases is affected for small α values as can be seen in Figures 25a and 25b. The training process through the solution of linear regression obtained from the reservoir states matrix and

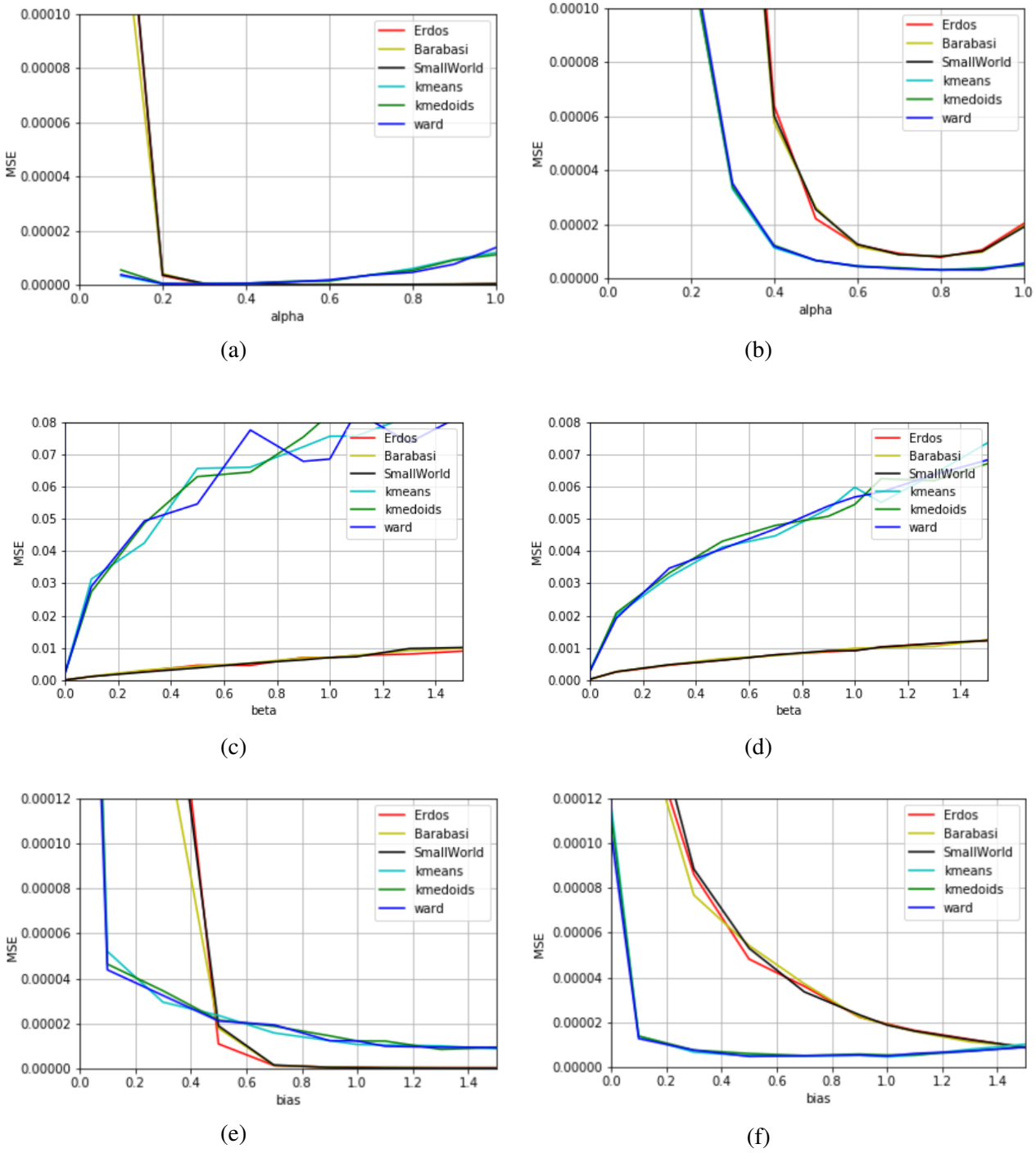


Figure 25 – In the order in which they are presented, ESN models performance analysis varying α: 25a - 25b, β: 25c - 25d, bias: 25e - 25f for Rössler and Lorenz cases respectively. Elaborated by the author.

target training set is suitable for small β values ($\beta \leq 1$) in the proposed model, as can be seen in Figures 25c and 25d. Finally, Figures 25e and 25f show that incorporating bias in the model can improve the performance of prediction. According to the presented results, the influence of these parameters within the performance of the model is evidenced. The choice of these parameters can vary from one set to another and depends on the nature of the data. Based on all the obtained

results, we can confirm that clusters inside the reservoir improve the ESN model performance.

CHAPTER

5

THE EXTENDED ECHO STATE NETWORKS APPLIED TO CLASSIFICATION TASKS

In this section, we describe the configurations of the experiments and show the obtained results on human tissue image classification using the extended ESNs. Each subsection details the parameters used in the classic and proposed model. The results are divided in three parts: 1) we compare the performance of the proposed model to classic ESN model in classification of digital medical images; 2) we contrast the performance of the proposed models to classic ESN model by varying parameters such as reservoir size (number of neurons in the reservoir), spectral radius, number of clusters, among others; 3) we compare the proposed models with the state-of-the-art classification techniques in digital images classification tasks.

5.1 Problem Statement

Let's consider $D = (X_1, Y_1), (X_2, Y_2), \ldots, (X_M, Y_M)$ a collection of pairs (X_i, Y_i) where X_i is a multivariate time series and Y_i as its corresponding label vector. Without loss of generality, for a dataset with K classes, the label vector Y_i is a vector of length K, where each element $i \in [1, K]$ is equal to 1 if the class of X_i is i and 0 otherwise. The purpose is to train a classifier on three datasets (see section 5.2), in order to map from the input space to a probability distribution over the class values (labels). More details about the datasets, see section 5.2.

5.2 Dataset Preparation

For all the experiments, we use the following three datasets:

- **CRC-VAL-HE-7K** (7K): The dataset is a set of histological images (tiles) of human colorectal cancer and healthy tissue. The dataset is composed for a set of 7180 image

patches from 50 patients with colorectal adenocarcinoma. All tissue samples are provided by the Natural Center for Tumor diseases(National Center for Tumor Diseases, Heidelberg, Germany) tissue bank.

- **NCT-CRC-HE-100K** (100K): This is a set of 100,000 non-overlapping image patches from hematoxylin & eosin (H&E) stained histological images of human colorectal cancer (CRC) and normal tissue. All images are normalized using Macenko's method[1]. These images are manually extracted from N=86 H&E stained human cancer tissue slides and formalin-fixed paraffin-embedded (FFPE) samples from the NCT Biobank and the UMM pathology archive (University Medical Center Mannheim, Mannheim, Germany). Tissue samples contain CRC primary tumor slides and tumor tissue from CRC liver metastases; normal tissue classes are augmented with non-tumorous regions from gastrectomy specimen to increase variability.

- **NCT-CRC-HE-100K-NONORM** (100K-NONORM): This is a slightly different version of the NCT-CRC-HE-100K image set: This set contains 100,000 images in 9 tissue classes at 0.5 MPP and is generated from the same raw data as NCT-CRC-HE-100K. However, no color normalization is applied to these images. Consequently, staining intensity and color slightly vary between the images. Although this image set is generated from the same data as NCT-CRC-HE-100K, the image regions are not completely identical because the selection of non-overlapping tiles from raw images is a stochastic process.

In the context of this work, a tile is a subdivision of a whole medical image, in this case, the tiles are non overlapping. These tiles (samples) are fed into the ESN trough three input nodes that each neuron receives the information of one color channel (R,G,B). From this information, the reservoir constructs a representation for each sample fed into the ESN. As described in section 3.2, the reservoir state matrix representation (RSM), that extract the main features from tile through the dynamic of reservoir nodes, governs the topology configuration. In this work, many topology configurations of the reservoir are studied and each one works in a similar form: Extracting features and storing in a final state matrix representation that varies in its structure (density of the matrix) according to the connectivity topology of the reservoir.

For the experiments, the datasets are divided into two sets as shows in Table 6. For more details, these datasets are available in (KATHER; HALAMA; MARX, 2018). The tile size is 224×224 pixels (px) at 0.5 microns per pixel (MPP) and the human tissue classes are shown in Figure 26, where illustrates an example for each class of tile. As we can see, there are classes, which are easily to be classified, such as the background (BACK). However, there are also classes, which contain complex patterns and are difficult to be classified, as shown in Figure 27. In the confusion matrix, we can see that, with an initial ESN classical configuration, the classifier does not manage very well to differentiate some types of human tissue like stroma, normal and tumor

[1] http://ieeexplore.ieee.org/abstract/document/5193250/

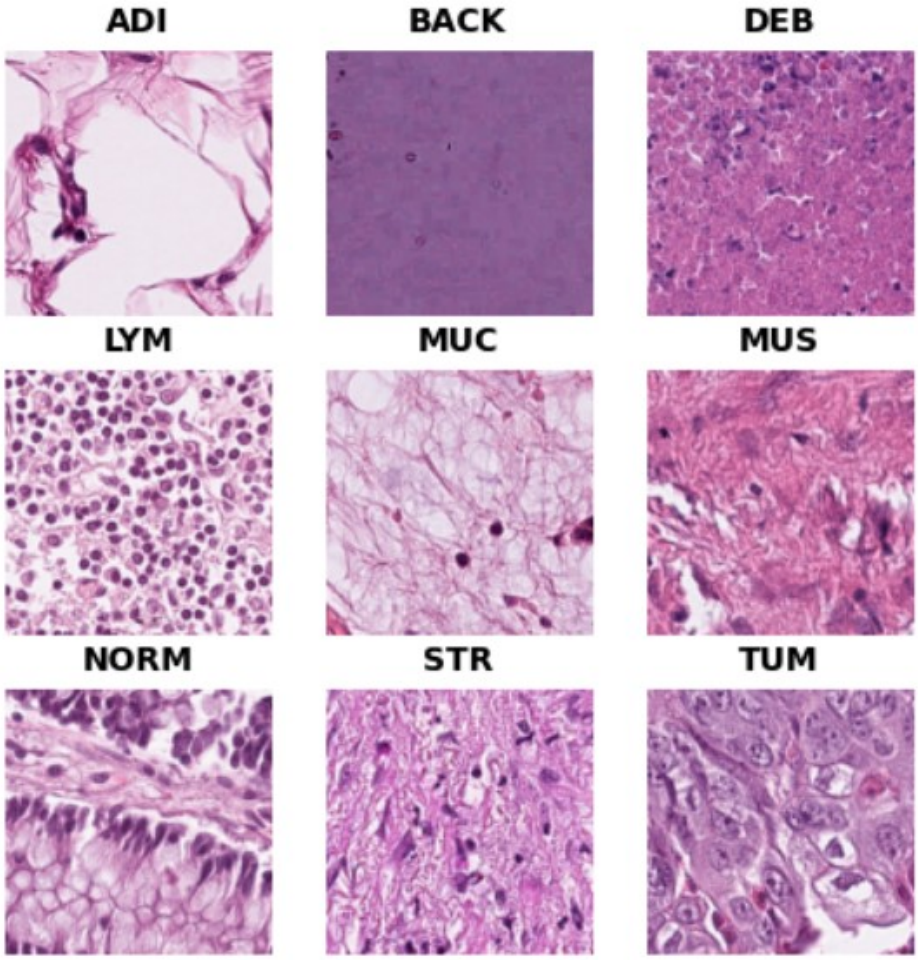


Figure 26 – Types of Human Tissues: Adipose (ADI), background (BACK), debris (DEB), lymphocytes (LYM), mucus (MUC), smooth muscle (MUS), normal colon mucosa (NORM), cancer-associated stroma (STR), colorectal adenocarcinoma epithelium (TUM). From (KATHER; HALAMA; MARX, 2018)

as easy as it does for background tissue. So it is necessary to make certain alterations in both the topology of connections and the inherits parameters of the model to analyze the performance of the ESN model.

	Training Set (80%)	Testing Set (20%)
7K	5744	1436
100K	80.000	20.000
100K-NORM	80.000	20.000

Table 6 – Quantity of samples in training and testing phase.

5.3 Implementation Setup

In experiments to be presented in this chapter, we implement the classical and the proposed ESNs with the same parameter setting in order to get a fair comparison. For the proposed models, we implemented an Echo State Network model, described in 2.1 with following

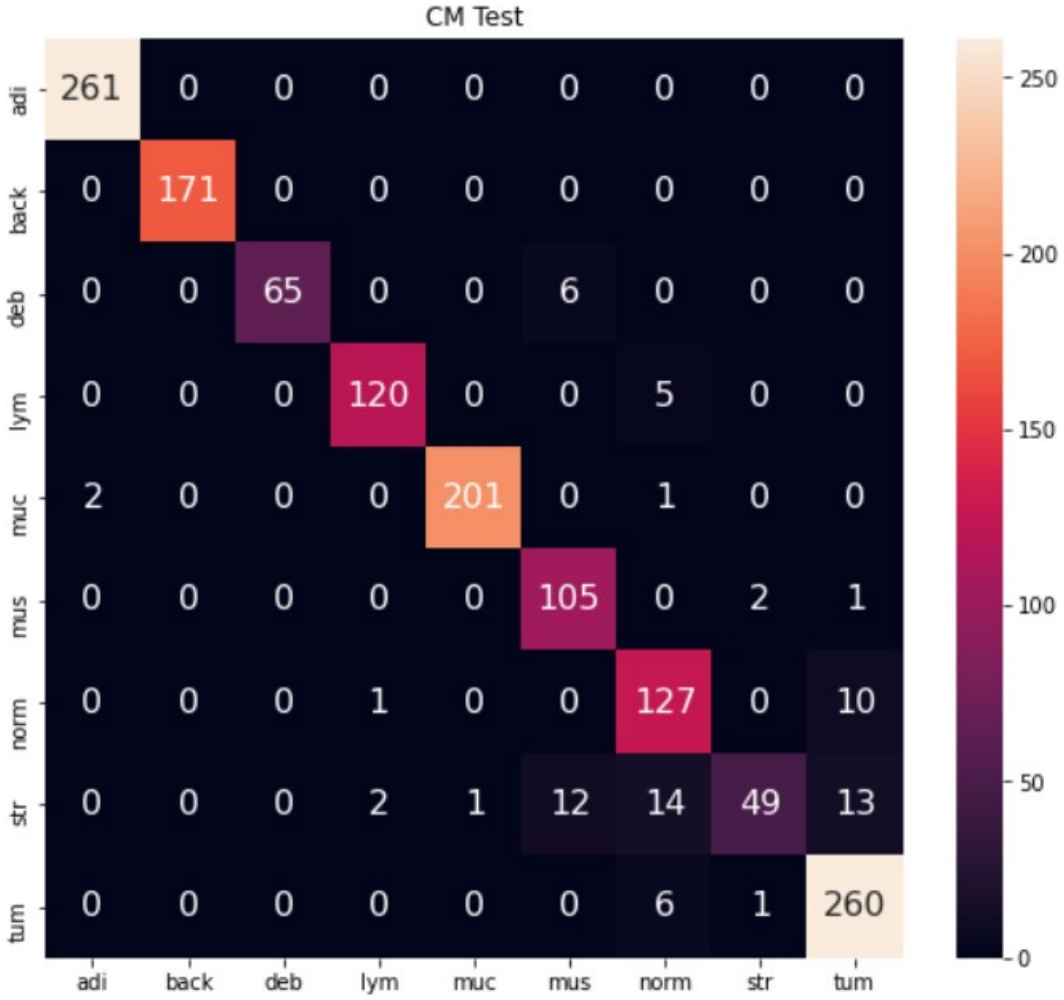


Figure 27 – Confusion matrix of classical ESN to 7K dataset using 400 reservoir nodes.

characteristics: The ESN has three layers and the weighted undirected connections of the neurons between layers are stored in the matrices W_{in}, W, W_{out} respectively. The non-zero elements of the W_{in} and W are randomly chosen from a uniform distribution in $[-1, 1]$. In order to guarantee the condition for the echo property described in 2.1, we scale W matrix using the spectral radius of a W. The last matrix is used to calculate the states (activation) of reservoir nodes in the training and testing phase. The activation function to calculate the state of neurons is according to the Eq. 2.3, where $\alpha = 1$ and **bias** $= 1$. For each sample in dataset, after collecting the states for each time, **r**, in the reservoir matrix states, **R**, the network was trained. For the implementation, the parameters settings used are defined in the Table 7. To make comparisons between the models studied, we have fixed the value corresponding to the random factor (seed) in the generation of complex networks and clusters using the sklearn modules.

Parameter	Value
number of reservoir nodes	$N = 400$
spectral radius	$\rho = 1$
average degree	$D = 20$
bias constant	$\zeta = 1$
leakage rate	$\alpha = 1$

Table 7 – Configuration setup.

5.4 Experimental Results

For the experiments, we set up the values of the parameters based on (LU *et al.*, 2017), detailed in the subsection 5.3. In the tables below, the first value in each cell is the classification accuracy for training phase and the second value corresponds to the accuracy for the testing phase. The first part of the results focus on comparing the performances (in terms of accuracy) between the proposed ESNs and the classical ESN by varying the parameter values based on (LU *et al.*, 2017).

Topology/Dataset	7K	100K	100K-NORM
Erdös	.94 - .92	.89 - .86	.89 - .86
Barabási	.93 - .92	.87 - .85	.87 - .85
Small World	.93 - .92	**.89 - .87**	.89 - .87
Cluster 1	**.93 - .93**	**.89 - .87**	**.89 - .88**
Cluster 2	.93 - .92	**.89 - .87**	.89 - .87
Cluster 3	**.94 - .93**	**.89 - .87**	**.89 - .88**

Table 8 – Accuracy for ESN models.

Table 8 shows the classification accuracy for each dataset described in 5.2. From the results, we see that the proposed ESN models have better performance than the classical ESN with 1% or 2% improvement, which, in this context, represents a difference of about 200 - 400 tissue images for the larger datasets. In the case of 7K dataset, 2 of the 5 proposed ESN models have a better performance than the classical ones; for 100K and 100K-NORM dataset, 4 of the 5 proposed ESN models have a better performance than the classical one. In all the cases, a non-random topology of the ESN model shows a better performance than the classical ESN with a random reservoir. From these experimental results, we see that the changes in the classical topology can really improve the performance of the ESN model, in particular, ESNs with clustered reservoirs get the best classification performance. Moreover, since the adjacent matrix of the clustered reservoir is usually sparse due to the small number of inter-cluster connections, sparse matrix numerical algorithms can be applied and, consequently, the computational cost of the clustered ESNs is low compared to the classical ESN.

The dynamics of the reservoir is influenced by the parameters of the model, consequently, affecting the accuracy of the image classification. Therefore, the second part of the experiments of this paper have focus on the study by varying parameter values. According to (JAEGER,

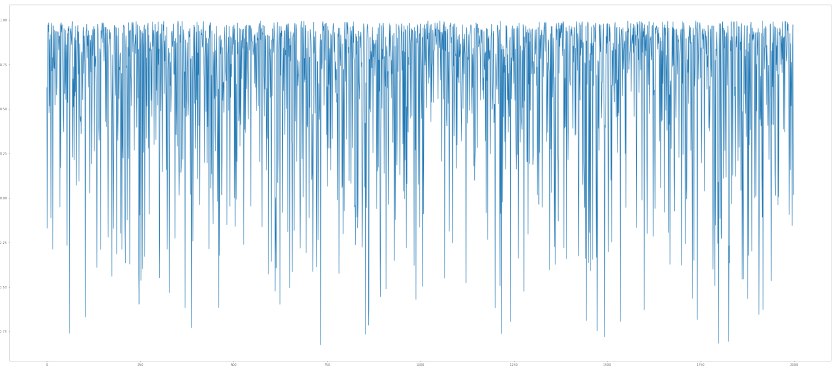


Figure 28 – Reservoir unit activation for $t \in [0, 5744]$ during the training phase.

2002), the parameter with the greatest impact on the reservoir's dynamics is the spectral radius ρ. So, in this work, we perform some experiments varying ρ to visualize the classification accuracy and confirm its impact on the performance of the ESN models. Other parameters under study are the number of reservoir nodes, N, ridge parameter, β, leakage rate (reservoir memory capacity), α, and number of clusters (*n_clusters*) to check the optimal value of these parameters in the experiments. We start the study with the N parameter, this parameter is very important because the main approximation capability of ESN model is concentrated in its reservoir, which evokes periodic signals inside the reservoir dynamic as shown in the Figure 28, some reservoir states are plotted for 2000 reservoir units in the training phase for $t \in [0, 5744]$.

The classification accuracy for $N = 100$ and $N = 3000$ reservoir units is registered in the Table 9. From these results, we see that the reservoir size has a significant impact in the performance for all models. Specifically, the classification accuracy is improved for all datasets as the reservoir size is increased. Figures 30d, 30e and 30f show the behavior of classification accuracy for different values of N. Internally, a larger number of reservoir units provides a higher dimensional representation of the input data. Consequently, the class locations are much separated in the data space, facilitating the classification tasks.

	Datasets					
	7K		**100K**		**100K-Norm**	
Topology	100	3000	100	3000	100	3000
Erdos	.91	.96	.81	.88	.82	.87
Barabasi	.91	.96	.82	.88	.82	.88
Small World	.91	.97	.82	.89	.83	.88
Clustering 1	.91	.97	.82	.89	.83	.88
Clustering 2	.91	.98	.82	.89	.83	.89
Clustering 3	.91	.97	.83	.89	.83	.88

Table 9 – Classification accuracy of ESN models with 100 and 3000 reservoir units.

The second parameter studied in this work is the spectral radius, as defined in 2.1. Table 10 shows the difference in the performances to $\rho = 0.3$ and $\rho = 1.3$, respectively. From the results, it is possible to note that the approximation capability of the ESN model is considerably affected by the spectral radius value. Note that for complex network topology, if the spectral radius is small, the reservoir dynamic is fast; on the other hand, if the value is larger (close to unity), it has a slow reservoir dynamic. For this reason, the best performance is obtained when ρ is closer than 1 as show in Figures 30a, 30b and 30c

	Datasets					
	7K		**100K**		**100K-Norm**	
Topology	0.3	1.3	0.3	1.3	0.3	1.3
Erdos	.88	.9	.78	.84	.82	.85
Barabasi	.89	.91	.77	.84	.78	.84
Small World	.89	.92	.78	.84	.79	.86
Clustering 1	.91	.92	.78	.84	.79	.86
Clustering 2	.90	.93	.78	.84	.79	.86
Clustering 3	.89	.92	.78	.85	.79	.86

Table 10 – Classification accuracy of ESN models for $\rho = 0.3$ and $\rho = 1.3$.

For a reservoir with clustered topology, we have varied the number of clusters in the reservoir to analyze the behavior of the model. For the implementation, the level of cluster mixture is set as $n = 0$, i.e. the clusters have no connection between them, those clusters are like mini isolated topology inside the reservoir. From the results, we can conclude that for the three datasets, the model achieves better performance with a higher number of clusters as shown by Figures 31a, 31b and 31c, respectively.

	Datasets					
	7K		**100K**		**100K-Norm**	
Topology	2	16	2	16	2	16
Clustering 1	.90	.92	.82	.84	.83	.84
Clustering 2	.90	.92	.82	.83	.82	.85
Clustering 3	.90	.92	.82	.83	.82	.84

Table 11 – Classification accuracy of ESN models using 2 and 16 clusters in the reservoir.

Another parameter studied is the leakage rate of the model, α, known as memory of capacity. According to 2.3, this parameter controls the level of information used from the previous state of the reservoir, i.e., its memory. Table 12 shows the classification accuracy for $\alpha = 0.3$ and $\alpha = 1$, respectively. From the results, we see that the performance of the model is better with larger values of α (close to the unity) as shown by Figures 29a, 29b and 29c.

Finally, the classification accuracy with two different values of β, ridge parameter, are registered in the Table 13. Based on the results, we see that the training of the model through the solution of linear regression task obtained from the reservoir states matrix and target training set is suitable for small β values ($\beta \leq 1$), as shown by Figures 29d, 29e and 29f.

	Datasets					
	7K		100K		100K-Norm	
Topology	0.3	1	0.3	1	0.3	1
Erdos	.90	.91	.81	.86	.82	.86
Barabasi	.90	.92	.81	.85	.81	.85
Small World	.91	.92	.81	.87	.82	.87
Clustering 1	.90	.92	.82	.87	.83	.88
Clustering 2	.91	.93	.81	.87	.83	.87
Clustering 3	.91	.92	.82	.87	.82	.88

Table 12 – Classification accuracy of ESN models with $\alpha = 0.3$ and $\alpha = 1$.

	Datasets					
	7K		100K		100K-Norm	
Topology	$1e^{-3}$	$1e^{-15}$	$1e^{-3}$	$1e^{-15}$	$1e^{-3}$	$1e^{-15}$
Erdos	.76	.91	.69	.86	.71	.86
Barabasi	.75	.91	.67	.85	.69	.85
Small World	.76	.91	.68	.87	.71	.87
Clustering 1	.76	.91	.69	.87	.71	.88
Clustering 2	.77	.92	.7	.87	.72	.87
Clustering 3	.78	.92	.71	.87	.73	.88

Table 13 – Classification accuracy of ESN models with $beta = 1e^{-3}$ and $beta = 1e^{-15}$.

According to the previous results, the influence of these parameters upon the model's performance is evident. The choice of these parameters varies from one dataset to another. Our studies provide some guidance for parameter selection of ESNs.

The third and the last part of the experiments has focus on comparing the performance of the new ESN models with other state-of-the-art techniques in human tissue image classification task. Tables 14 and 15 show the classification accuracy of a Convolutional Neural Network (CNN) and Support Vector Machines (SVM) technique using different kernels. Again, we see that the proposed ESNs can achieve competitive classification accuracy.

Optimizer / Dataset	7K	100K	100K-Norm
SGD	.83	.89	.9
RMSprop	.88	.9	.91
Adam	.9	.91	.91
Adadelta	.33	.51	.57
Adagrad	.67	.66	.75

Table 14 – Classification accuracy of CNN.

The CNN architecture implemented is detail in Figure 32. The activation function used in the CNN implemented was "Relu". For the implementation of the SVM, the same images described in the datasets (see section 5.2) are used, but using only one channel. Based on the obtained results, we can conclude again that the performance of the ESN models is superior

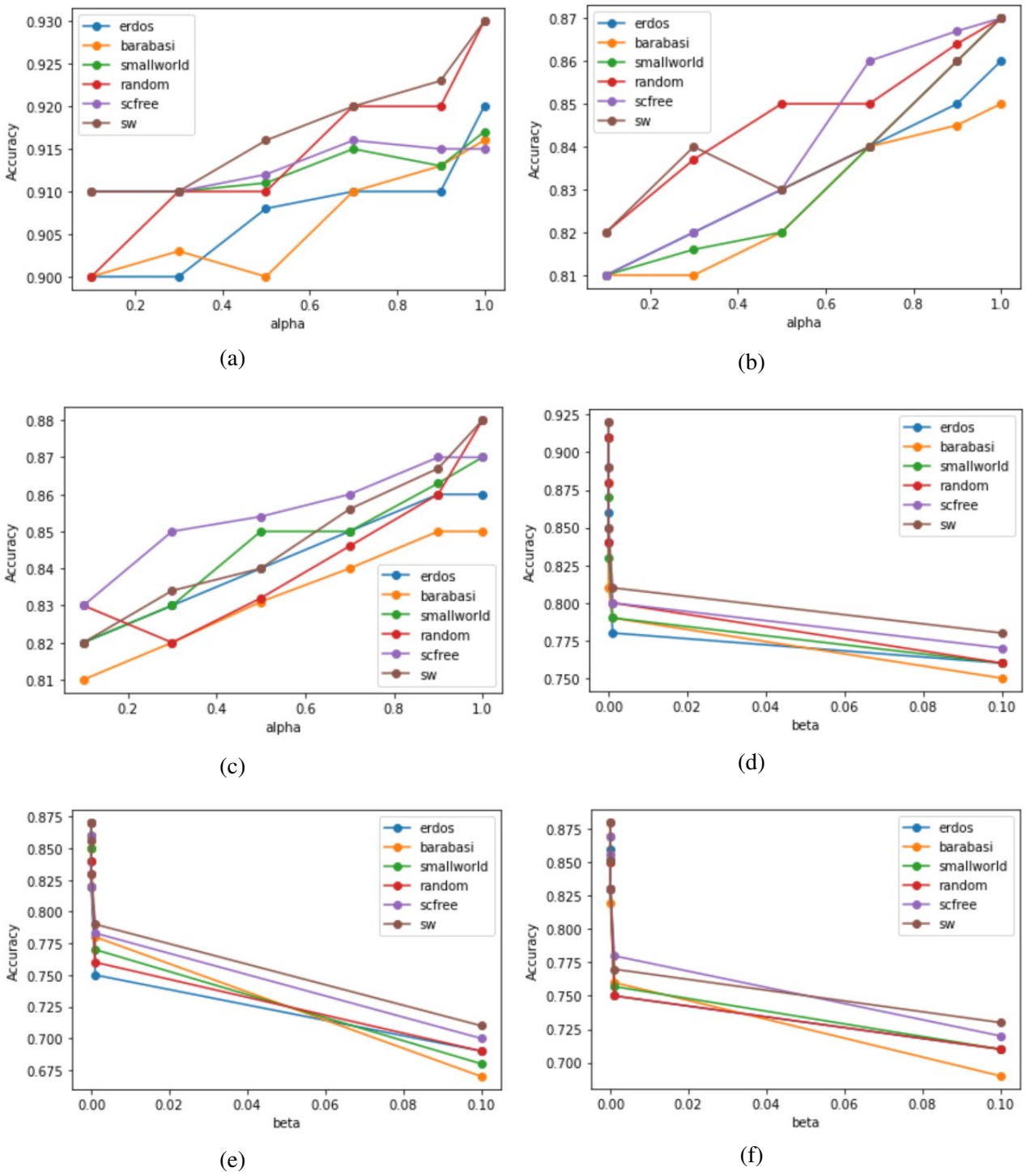


Figure 29 – In the order in which they are presented, ESN models performance analysis varying α: 29a - 29c and β: 29d - 29f for $7K$, $100K$ and $100K$-Norm datasets respectively. Elaborated by the author.

compared to the state-of-the-art techniques.

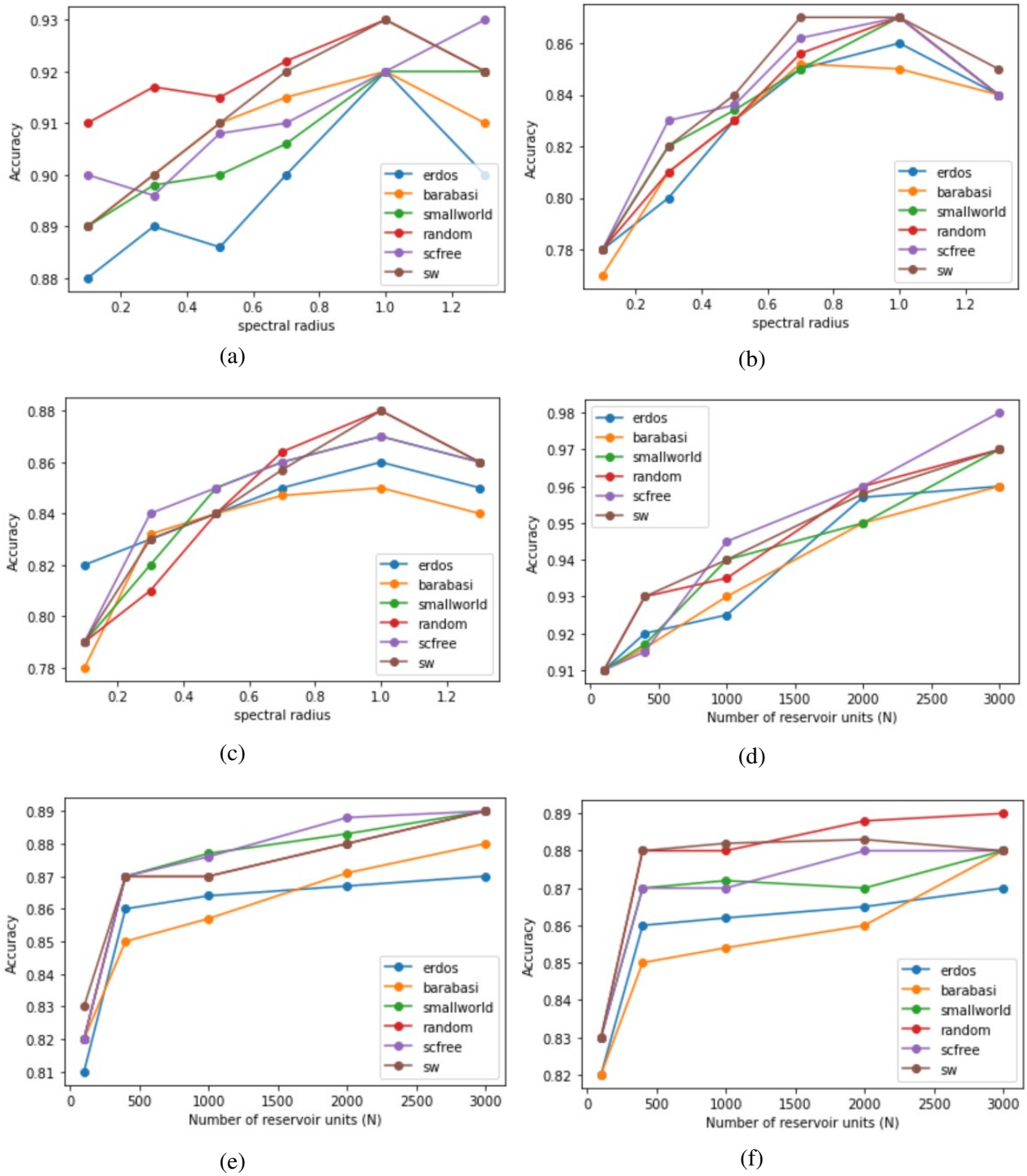
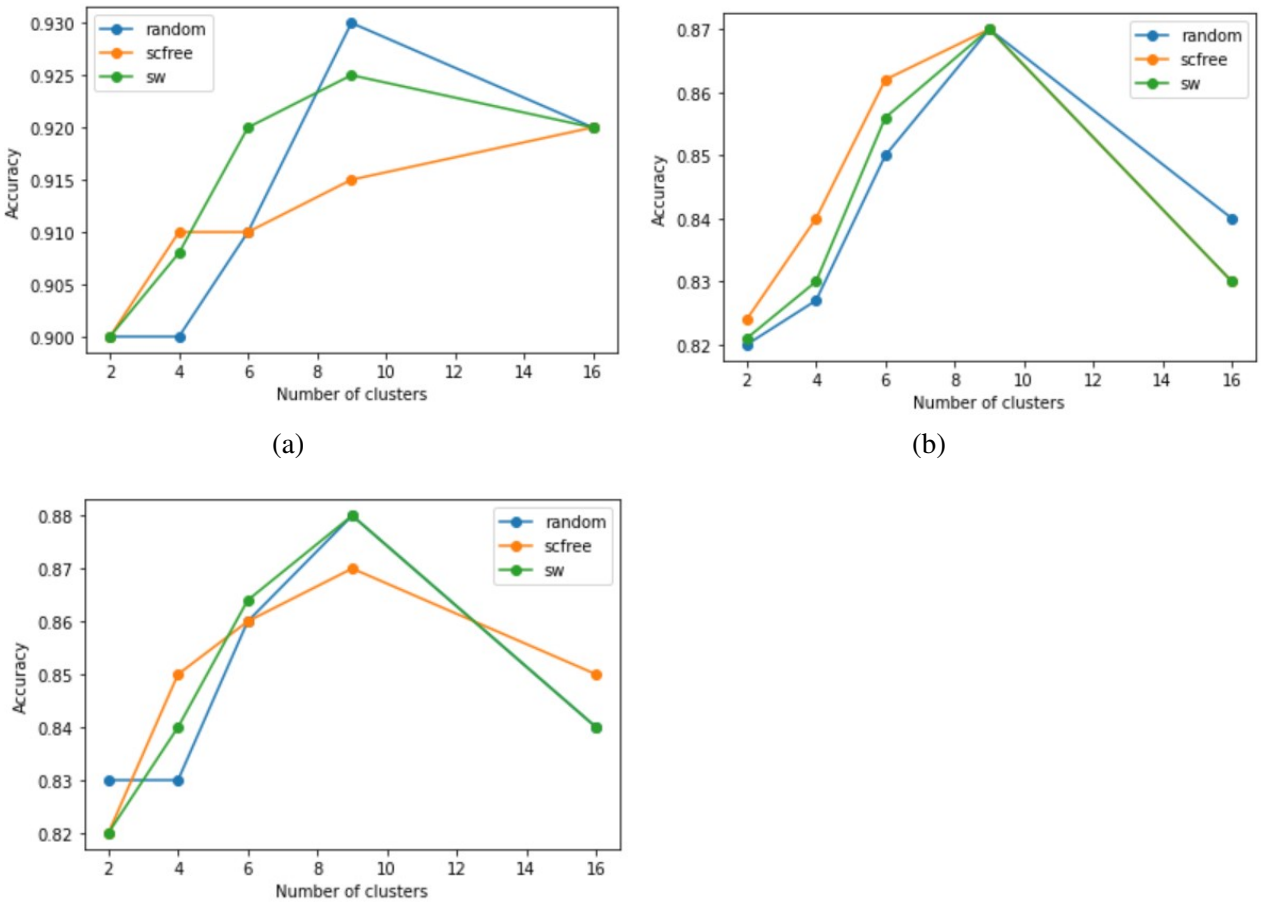


Figure 30 – In the order in which they are presented, ESN models performance analysis varying ρ: 30a - 30c and reservoir size: 30d - 30f for $7K$, $100K$ and $100K$-Norm datasets respectively. elaborated by the author.

Kernel / Dataset	7K	100K	100K-Norm
Linear	.49	.26	.31
Poly	.55	.26	.32
RBF	.56	.26	.26

Table 15 – Classification accuracy of SVM.



(a) (b)

(c)

Figure 31 – In the order in which they are presented, ESN models performance analysis varying number of clusters: 31a - 31c for $7K$, $100K$ and $100K$-Norm datasets respectively. Elaborated by the author.

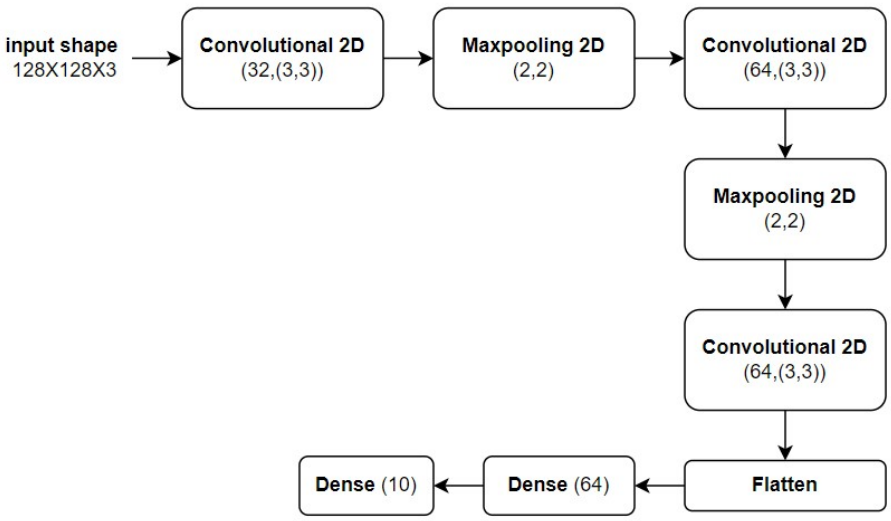


Figure 32 – CNN architecture implemented.

CHAPTER

6

CONCLUSIONS

In this work, we have proposed to extend the reservoir topology of ESNs using clustered non-random complex networks. Good numerical results have been obtained for two applications: time series prediction and classification task.

6.1 Concluding Remarks

In the time series prediction task, we apply the proposed ESNs in two study cases: Rössler and Lorenz systems. From the results obtained in the experiments, we can conclude that the reservoirs with clustered non-random topology can improve the performance of ESNs. Also, we have conducted extensive study the parameter selection of the models. Specifically, we determine the selection ranges for the number of the reservoir units, the spectral radius, and others parameters in the model: $\alpha, \beta, n\_clusters$, and bias term.

In the human tissue medical image classification task, we have presented the application of the modified ESNs to three datasets. From the results obtained in the experiments, we can conclude again that the clustered non-random reservoirs can improve the ESN model performance. Also, we confirm the influence of the number of the reservoir units, the spectral radius, and other parameters in the model: β, and $n\_clusters$.

Our study shows that ESNs are suitable tools for time series processing. For other types of data, such as the image data treated in this work, we can firstly convert it to time series form and then apply ESN model.

The interesting point of ESNs is the flexibility to design the reservoirs in various network topology. In this way, we believe that complex network theory can make further contributions to ESN study.

6.2 Future Works

The reservoir design and the experiments presented in this work show the potential of ESNs in solving complex real problems. As future works, we would like to continue developing new ESN models and applications. Specifically, we suggest the following tasks:

- We will design reservoirs in ESNs using other complex network topology, such as the core-periphery networks. We will calculate more network measures and study the relationship between the models' performance and the network measures of the corresponding reservoirs.
- We believe that the ESN performance could be improved by optimizing the cluster formation, based on the distribution of the neurons in the human brain. In other words, we will conduct deep study on human brain organization to propose bio-inspired reservoirs.
- Inspired from deep learning, we will develop multi-layer reservoirs to automatically extract time series features before training. In this case, a reservoir can be modeled as a multi-layer complex network.
- We will continue to develop human tissue image classification application. We hope to develop a complete computational system to detect tumors for helping cancer diagnosis.

BIBLIOGRAPHY

ARROYO, D. C. R.; FLOREZ, A. C.; FLORES, D. M.; ROMERO, R.; ZHAO, L. Echo state network performance analysis using non-random topologies. In: . [S.l.: s.n.], 2020. Citations on pages 22, 24, 29, and 49.

Barabási, A.; Pósfai, M. **Network science**. [S.l.]: Cambridge University Press, 2016. Citations on pages 30, 31, 32, 33, 34, 35, and 36.

CHATURVEDI, S.; SRINIVAS, A. S.; R, K.; M, V.; KUMAR, N.; M, S. Medical image denoising and classification based on machine learning: A review. **ECS Transactions**, 2022. Citation on page 41.

CUCCHI, M.; ABREU, S.; CICCONE, G.; BRUNNER, D.; KLEEMANN, H. Hands-on reservoir computing: a tutorial for practical implementation. **Neuromorphic Computing and Engineering**, 2022. Citations on pages 21, 22, and 28.

DING, L.; BAI, Y.-L.; FAN, M.-H.; YU, Q.-H.; ZHU, Y.-J.; CHEN, X.-Y. Serial-parallel dynamic echo state network: A hybrid dynamic model based on a chaotic coyote optimization algorithm for wind speed prediction. **Expert Systems with Applications**, 2023. Citation on page 30.

FITZGERALD, J.; HIGGINS, D.; MAZO, C.; WATSON, W.; MOONEY, C.; RAHMAN, A.; ASPELL, N.; CONNOLLY, A.; GONZALEZ, C.; GALLAGHER, W. Future of biomarker evaluation in the realm of artificial intelligence algorithms: application in improved therapeutic stratification of patients with breast and prostate cancer. **J Clin Pathol. 2021**, 2021. Citation on page 39.

GAO, R.; LI, R.; HU, M.; SUGANTHAN, P. N.; YUEN, K. F. Dynamic ensemble deep echo state network for significant wave height forecasting. **Applied Energy**, 2023. Citation on page 30.

GHIMIRE, S.; NGUYEN-HUY, T.; AL-MUSAYLH, M. S.; DEO, R. C.; CASILLAS-PéREZ, D.; SALCEDO-SANZ, S. A novel approach based on integration of convolutional neural networks and echo state network for daily electricity demand prediction. **Energy**, 2023. Citation on page 30.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation (3rd Edition)**. [S.l.]: Prentice-Hall, Inc., 2007. Citation on page 21.

____. **Neural Networks and Learning Machines**. [S.l.]: Pearson, 2009. Citation on page 21.

HILGETAG, C. C.; GOULAS, A. 'hierarchy' in the organization of brain networks. **Philos. Trans. R. Soc. Lond. B Biol. Sci.**, 2020. Citation on page 22.

JAEGER, H. Tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the echo state network approach. **GMD-Forschungszentrum Informationstechnik, 2002.**, 2002. Citations on pages 22, 27, 28, 29, 62, and 72.

JAMSHIDI, M. B.; DANESHFAR, F. A hybrid echo state network for hypercomplex pattern recognition, classification, and big data analysis. In: . [S.l.: s.n.], 2022. Citation on page 22.

JIANG, H.; DIAO, Z.; SHI, T.; ZHOU, Y.; WANG, F.; HU, W.; ZHU, X.; LUO, S.; TONG, G.; YAO, Y.-D. A review of deep learning-based multiple-lesion recognition from medical images: classification, detection and segmentation. **Computers in Biology and Medicine**, 2023. Citations on pages 40 and 41.

JUNIOR, L. O.; STELZER, F.; ZHAO, L. Clustered echo state networks for signal observation and frequency filtering. In: . [S.l.]: SBC, 2020. Citations on pages 22, 24, and 29.

JYOTHI, P.; SINGH, A. R. Deep learning models and traditional automated techniques for brain tumor segmentation in mri: a review. **Artificial Intelligence Review**, 2023. Citation on page 40.

KATHER, J. N.; HALAMA, N.; MARX, A. **100,000 histological images of human colorectal cancer and healthy tissue**. [S.l.]: Zenodo, 2018. Citations on pages 68 and 69.

KITAYAMA, K.-i. Guiding principle of reservoir computing based on "small-world" network. **Scientific Reports**, 2022. Citations on pages 22, 24, and 29.

KSHATRI, S. S.; SINGH, D. Convolutional neural network in medical image analysis: A review. **Archives of Computational Methods in Engineering**, 2023. Citation on page 41.

LI, Z.; LIU, Y.; TANAKA, G. Multi-reservoir echo state networks with hodrick–prescott filter for nonlinear time-series prediction. **Applied Soft Computing**, 2023. Citation on page 29.

LIU, Z.; TONG, L.; CHEN, L.; JIANG, Z.; ZHOU, F.; ZHANG, Q.; ZHANG, X.; JIN, Y.; ZHOU, H. Deep learning based brain tumor segmentation: a survey. **Complex & Intelligent Systems**, 2023. Citation on page 41.

LORENZ, E. Deterministic nonperiodic flow. **J. Atmos. Sci.**, 1963. Citation on page 59.

LU, Z.; PATHAK, J.; HUNT, B.; GIRVAN, M.; BROCKETT, R.; OTT, E. Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. **Chaos: An Interdisciplinary Journal of Nonlinear Science**, 2017. Citations on pages 60, 61, 62, and 71.

NAZIR, S.; DICKSON, D. M.; AKRAM, M. U. Survey of explainable artificial intelligence techniques for biomedical imaging with deep neural networks. **Computers in Biology and Medicine**, 2023. Citation on page 40.

Newman, M. **Networks: An Introduction**. [S.l.]: Oxford University Press, Inc., 2010. Citations on pages 30, 34, and 36.

REHMER, A.; KROLL, A. On the vanishing and exploding gradient problem in gated recurrent units. **IFAC-PapersOnLine**, 2020. Citations on pages 22 and 27.

ROSTAMI, M.; OUSSALAH, M.; BERAHMAND, K.; FARRAHI, V. Community detection algorithms in healthcare applications: A systematic review. **IEEE Access**, 2023. Citation on page 37.

RöSSLER, O. An equation for continuous chaos. **Physics Letters A**, 1976. Citations on pages 59 and 60.

SCHWEDERSKY, B. B.; FLESCH, R. C. C.; DANGUI, H. A. S. Nonlinear mimo system identification with echo-state networks. **Journal of Control, Automation and Electrical Systems**, 2022. Citation on page 22.

SHAMSHAD, F.; KHAN, S.; ZAMIR, S. W.; KHAN, M. H.; HAYAT, M.; KHAN, F. S.; FU, H. Transformers in medical imaging: A survey. **Medical Image Analysis**, 2023. Citation on page 40.

SOLTANI, R.; BENMOHAMED, E.; LTIFI, H. Echo state network optimization: A systematic literature review. **Neural Processing Letters**, 2023. Citations on pages 21, 22, and 30.

STEINER, P.; JALALVAND, A.; BIRKHOLZ, P. Cluster-based input weight initialization for echo state networks. **IEEE transactions on neural networks and learning systems**, 2021. Citation on page 29.

SUN, C.; SONG, M.; CAI, D.; ZHANG, B.; HONG, S.; LI, H. A systematic review of echo state networks from design to application. **IEEE Transactions on Artificial Intelligence**, 2022. Citation on page 22.

VIEHWEG, J.; WORTHMANN, K.; MäDER, P. Parameterizing echo state networks for multi-step time series prediction. **Neurocomputing**, 2023. Citation on page 30.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of 'small-world' networks. **Nature**, 1998. Citation on page 35.

ZHENG, K.; QIAN, B.; LI, S.; XIAO, Y.; ZHUANG, W.; MA, Q. Long-short term echo state network for time series prediction. **IEEE Access**, 2020. Citation on page 22.


ICMC
USP
SÃO CARLOS