Regular Echo State Networks: simple and accurate reservoir models to real-world applications

Lucas Z. Bissaro
Faculty of Computing
Federal University of Uberlandia
Uberlandia, Brazil
lucasbissaro@gmail.com

Yaochu Jin
Department of Computer Science
University of Surrey
Guildford, UK
yaochu.jin@surrey.ac.uk

Murillo G. Carneiro*
Faculty of Computing
Federal University of Uberlandia
Uberlandia, Brazil
mgcarneiro@ufu.br

ABSTRACT

Reservoir computing is a computational paradigm derived from recurrent neural network models. One of its most representative technique is the Echo State Network (ESN), which is usually composed by two salient components: reservoir and readout. The former is responsible by mapping temporal (or sequential) inputs into a high-dimensional space and the latter aims at learning the patterns in such a new space. Despite ESN has attracted a lot of attention nowadays, most works in the literature are focused on the development of additional features to the model, while there is a considerable lack of investigations related to understand and evaluate some of its inherent concepts as well as their relationships. In this paper we address such a limitation by investigating ESN components related to the reservoir structure and readout layer. To be specific, we evaluate regular and small-world network models besides the extensively adopted random one, and also analyze a total of eight classification techniques instead of considering just the few techniques largely adopted in the literature (mostly linear ones). In order to consistently evaluate the alternative ESN methods, we analyzed a wide range of parameters in both reservoir and readout layers through of several experiments with five real-world data sets. The results revealed that some problems can be considerably benefited from some level of organization in the reservoir, such as those provided by regular or small-world network models; and that the non-linear support vector machine classifier achieved the best predictive performance, although it was statistically comparable with the k-nearest neighbors one, which has much smaller time complexity. Interestingly, such findings may make the adoption of ESN methods more efficient from the point of view of embedded systems and large scale problems.

CCS CONCEPTS

• **Computing methodologies → Machine learning approaches**; **Supervised learning by classification**; *Machine learning algorithms*; **Neural networks**;

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SAC'21, March 22-March 26, 2021, Gwangju, South Korea
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8104-8/21/03…$15.00
https://doi.org/xx.xxx/xxx_x

KEYWORDS

Echo State Network, Reservoir Computing, Regular Network, Small-World Network, Complex Networks, Data Classification

ACM Reference Format:
Lucas Z. Bissaro, Yaochu Jin, and Murillo G. Carneiro. 2021. Regular Echo State Networks: simple and accurate reservoir models to real-world applications. In *Proceedings of ACM SAC Conference (SAC'21).* ACM, New York, NY, USA, Article 4, 7 pages. https://doi.org/xx.xxx/xxx_x

1 INTRODUCTION

A Recurrent Neural Network (RNN) is a very well-known kind of artificial neural network in which information is propagated through of neurons interconnected by links. Different of the famous feed-forward neural networks (e.g., Multilayer Perceptron), the salient aspect of RNNs is the presence of cycles in the network topology, which characterizes them as dynamical systems able to maintain nonlinear transformations of the input data [10, 17, 20]. Common applications of RNNs include domains related to sequential and temporal data, like speech recognition, machine translation and time series analysis [9, 18, 29].

Despite a considerable number of works have demonstrated the notable success of RNNs in many applications, there are also other works which have pointed out limitations related to convergence, scalability and parameterization [1, 23]. Motivated by such drawbacks, a new paradigm to design and train RNNs was proposed. Such a paradigm, which is referred to Reservoir Computing, is mostly based on two independent approaches which have essential features in common, the Echo State Network (ESN) [13] and the Liquid State Network (LSN) [22]. As the LSN model is intended to represent and replicate more sophisticated and biologically plausible neuronal theories into the reservoir, in this study we are particularly interested in the ESN model, which originally is more focused on machine learning theories [20].

The standard ESN model assures that an algebraic property named echo state property is satisfied by the reservoir structure [13]. The ESN is usually composed by two salient components: a reservoir structure and a readout layer. The former is responsible by mapping data inputs into a high-dimensional space and the latter aims at learning the patterns from such a new space. Regarding the reservoir structure, it contains cycles like conventional RNNs, but its main difference is that the internal weights of such a structure are always fixed, i.e., there is no training or weight updating scheme there. On the other hand, the readout layer is responsible to learn the associations between the reservoir and the desired output, which requires a training process. Although several works in the literature show that a simple linear readout is often enough to achieve good performance in many problems, other works have pointed out its drawbacks [14].

Several methods and variants of the original ESN [13] have been presented in the literature, most of them covering changes in neuron properties or behaviors which resulted in different non-linear transformations [11, 27]. There are also ESN methods that work with multiples reservoirs instead of one [8, 21] and some works in which the focus is on improving the random structure of the reservoir by using heuristics [3, 7, 12, 24, 25] or mathematical models [28]. On the other hand, complex network models have been a few explored topic into the ESN context, with a related contribution published recently in [15]. In that work, regular, small-world and random network models were investigated as reservoir structures in order to evaluate their ability to process information in a specific context which was inspired by the human cortical neural connectivity. Those same network models were also investigated in [16] and strategies based on scale-free properties and hierarchically distributed structures in [5]. Nevertheless, to the best of our knowledge, there is no study in the literature which evaluate systematically the reservoir structure in function of the readout learning as we propose in this paper.

Different of the most current ESN works which are focused on the development of additional features to the model or inspired by biological modeling, we evaluate here alternative ESN components related to the reservoir structure, readout learning and parameter settings with a special focus on machine learning aspects. From the structure point of view, we investigate regular and small-world network models besides the widely adopted random one. Regarding the readout layer, we analyze a total of eight classification techniques in the task of mapping the reservoir memory to the corresponding desired output. In relation to the parameters, we systematically consider a reasonable number of settings in both reservoir and readout layers.

In order to consistently evaluate the alternative ESN components investigated here, we conduct experiments with five real-world applications which encompass very distinct characteristics in terms of domain, samples size, and number of features and class labels. As a glimpse, our statistical results indicated important aspects related to each one of the topics covered by our investigation, such as that some problems may be benefited from some level of organization in the reservoir, especially those provided by regular network models.

The remainder of the paper is organized as follows. Sect. 2 presents a quick background about ESN. Sect. 3 describes the methods we propose for both reservoir structure and readout layer. Sect. 4 discusses the experimental results obtained and Sect. 5 concludes the paper.

2 BACKGROUND

Reservoir computing methods are usually applied in the supervised learning context to address learning tasks which involve temporal or sequential data. Particularly, an ESN is a specific kind of reservoir computing created from the echo state property, that is related to the reservoir weight matrix and requires that its biggest autovalue is smaller than one, which allows that any data input to the reservoir vanish after a time. Notice that such a property do not ensure the vanish, but in practice it is rare not work. As a supervised technique, ESN aims at learning a function $f(.)$ such that for any given input $u(t)$ returns an output $y(t) \in \mathbb{R}^N$ which approximates the most of a ground-truth output $y^{target}(t)$. Formally, in a given temporal task, each object $u(t) \in \mathbb{R}^M$ is associated to an output $y^{target}(t) \in \mathbb{R}^N$, where time $t = 1, 2, \ldots, X$, with X denoting the number of sequenced objects in the data set [19].

Fig. 1 presents a general overview about the standard ESN architecture. In the figure, one can see three major parts related to the Input, Reservoir and Readout. The Input is represented by a fixed weight matrix $W^{in}_{M \times N}$, in which M denotes the number of features of $u(t)$ and N the number of neurons in the reservoir, which is responsible to distribute the input data $u(t)$ along all reservoir neurons. The Reservoir aims at processing the W^{in} output in your structure represented by a weight matrix $W_{N \times N}$ to obtain an activation vector denoted by $x(t)$, which is the transformed data. The Readout layer receive the output from the reservoir which can be used through of any supervised learning technique to obtain the predictive output y(t). Formally, the ESN is defined by the following equations:

$$\widetilde{x}(t) = tanh(W^{in}[1; u(t)] + W[x(t-1)]) , \quad (1)$$

$$x(t) = (1 - \alpha)x(t-1)) + \alpha \widetilde{x}(t) , \quad (2)$$

$$y(t) = f(x(t)) , \quad (3)$$

where $x(t) \in \mathbb{R}^N$ is the activation neurons of reservoir and $\widetilde{x}(t) \in \mathbb{R}^N$ is the update state vector in time n, $tanh(\cdot)$ is applied in each interaction, $[\cdot, \cdot]$ means a vertical concatenation in vector or matrix, $W^{in} \in \mathbb{R}^{N \times (1+M)}$ and $W \in \mathbb{R}^{N \times N}$ are respectively the input matrix and the recurrent one, and $\alpha \in (0, 1]$ is the leaky rate. A sigmoid function can be used instead of *tanh*. The model also can work without the leaky rate, which is the special case when $\alpha = 1$ and $x(t) \equiv \widetilde{x}(t)$. The $y(t) \in \mathbb{R}$ represents an approximation function applied in $x(t)$ trying to map with real output $y^{target}(t)$. Furthermore, in case of $y^{target}(t)$ is a discrete output we then have a classification task.

As one can see, the weights of the matrices W^{in} and W are fixed, which means that ESN does not suffer with the convergence problems of other recurrent methods based on gradient descent, for example. Consequently, the ESN training and application is very fast, which also makes the technique highly scalable. In this sense, the investigation of models able to exploit efficient network models and readout strategies in order to improve the ESN performance without losing such salient features (convergence and scalability) is a promising direction.

3 MODEL DESCRIPTION

In the following, we describe the ESN models investigated in this paper. In addition to present a general overview about the standard ESN architecture, Fig. 1 also shows the specific topics in which this paper aims to contribute. Regarding the reservoir layer, our hypothesis says that network properties, like the regular and small-world ones, can be efficient to save and process data inputs of problems which may be benefited from some level of organization. Complementary, we also investigate the influence of such a level of organization with eight different classification techniques in the readout layer, and with different parameter settings particularly

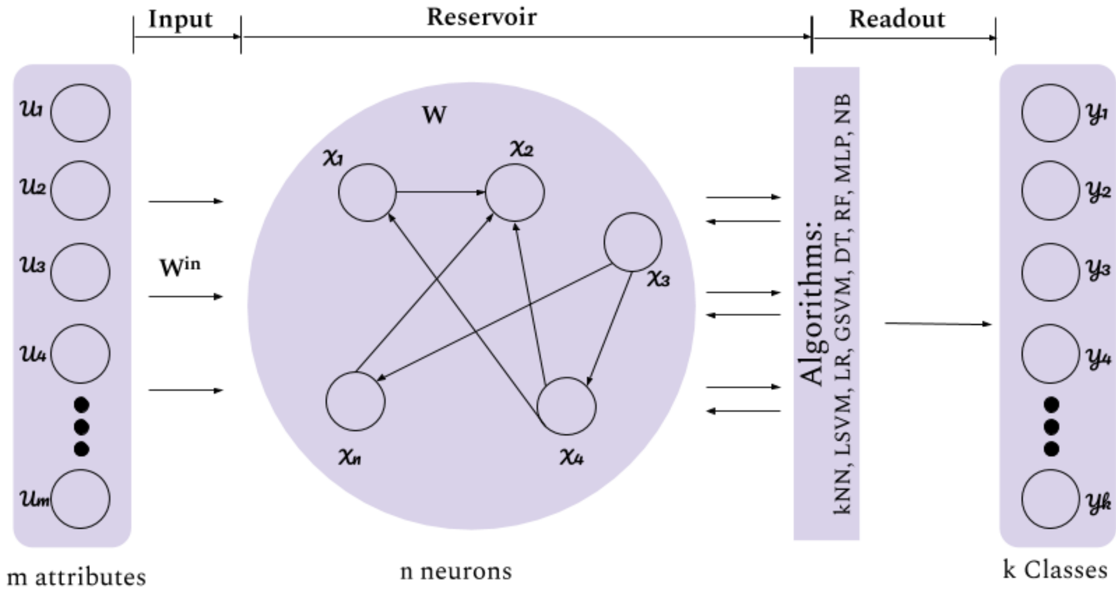


Figure 1: General overview about the components of an Echo State Network. The figure illustrates the ESN structure in terms of input, reservoir and readout layer which are respectively represented by W^{in}, W and many classification algorithms.

related to the number of neurons in the reservoir and the generation of the random weights.

3.1 Reservoir network models

In this study we consider the structure provided by three network models: regular, small-world and random. Fig. 2 presents an illustrative example about each one of the network models in the context of a reservoir, in which each vertex denotes a neuron and each edge a connection between neurons. In the figure, one can see a changing from well-defined network connections to disorganized ones: the regular model provides strictly organized network connections; the small-world provides a network with a bit of randomness in its connections, although it also presents a good level of organization; and the random model provides a chaotic network with all connections randomly generated. Next we briefly discuss each of these network models.

- In the regular model (Reg.), showed by Fig. 2(a), each network neuron has the same number of connections k, i.e., the same degree.
- In the small-world model (SW) the connections of a k-regular network are modified by considering a random rate p to which the values are usually on the following range $0 < p \leq 0.1$ [26]. Fig. 2(b) exhibits a SW network generated with $p = 0.1$.
- In the random model (Rand.) the network connections are randomly defined in function of a density parameter d, which refers to the percentage of existing links in the network. This is the principal reservoir structure, widely adopted in the literature. Fig 2(c) shows a random network.

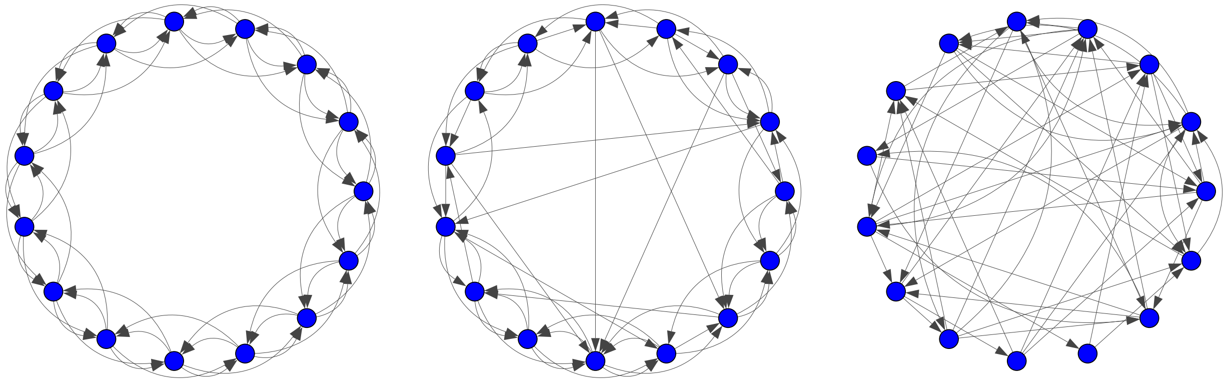
3.2 Readout classifiers

The information processed by the reservoir structure is further analyzed by a classification technique which also takes into account the temporal information intrinsic to each sample. The output can usually be separated by a linear classifier, but this do not mean that other methods cannot achieve more success [14]. In this work, the following eight algorithms are evaluated in the readout layer: k-Nearest Neighbors(kNN), Linear Support Vector Machine (LSVM), Logistic Regression (LR), RBF Support Vector Machine (GSVM), Decision Tree (DT), Random Forest (RF), Multilayer Perceptron (MLP) and Naive Bayes (NB). A salient feature of our study is that we evaluate the predictive performance of ESN considering the influence of both network models and readout techniques, which may also provide a better understanding about the relationship between complex network properties and classifiers inductive bias.

3.3 Parameter settings

In this study we rigorously evaluate two groups of parameters related to both reservoir structure and readout classifiers. Regarding the reservoir structure there are three parameters: the number of neurons given by N; the number of connections given by $k = d \cdot N$, with d referring to the percentage of existing connections in the network (a.k.a density), which also must be selected in order to preserve the echo state property; and the σ parameter, which refers to the standard deviation of the weight values (W^{in} and W), randomly generated following a normal distribution.

In relation to the readout classifiers, each technique has its parameter(s) carefully selected by a grid search method over a representative set of values. Next section presents in detail the values of



(a) Regular **(b) Small-Word** **(c) Random**

Figure 2: Example of reservoir structures generated from distinct network models: a) in the Regular, the neurons are connected with their k-nearest neighbors; b) In Small-Word, the neurons have a probability $p = 0.1$ to change their links after being connected with their k-nearest neighbors neurons; and c) in the Random, the connections are fully random.

parameters considered in both groups as well as the experimental setup.

4 EXPERIMENTAL RESULTS

In the following, we describe the experiments conducted over the reservoir structures presented before. Table 1 shows the metadata of the five data sets selected for this study [2, 6]. One can see that such a selection was made to encompass diversity on data domains as well as different number of classes, attributes and sizes (they vary from 3 to 378, 8 to 91 and 360 to 27619, respectively). A brief introduction about each data set is presented below.

- In the Accelerometer data set, each sample denotes the position of a patient in a bedroom taken by readings of an accelerometer sensor. The learned model should be able to identify the position of the patient regarding the bed to correctly detect when the patient is getting up in order to prevent some fall.
- LIBRAS is the Brazilian sign language, which is largely used by deaf people in Brazilian urban centers. Each word is converted to a sign. The sign can be seeing as a sequence of movements using the arms and hands. In the data set, each object means types of movement in LIBRAS. Notice that the sequence of movements form sentences, but the data set does not take this in consideration. The data were acquired mapping real movements of videos and for each frame was generated 90 features representing the coordinates of the movement.
- In the RSSI data set, the objects are described by 13 values, each one representing the object distance until a specific sensor. The sensors are scattered in the the first floor of Waldo Library, Western Michigan University. The floor was mapped in quadrants, thus the objective is to discover in which quadrant the object is.
- The Wall-Following robot data set aims to predict the movements of a SCITOS G5 robot using the readings of 24 ultrasound sensors. The data were collected as the robot navigates through the room following the wall in a clockwise direction for four rounds. The data set were designed to test the hypothesis that this apparently simple navigation task is indeed a non-linearly separable classification task.
- In the Ozone data set, the objects are denoted by 73 attributes related to day characteristics like temperature, wind speed, solar radiation and emissions stats, which are recommended by the Texas Commission on Environmental Quality (TCEQ) to monitor ozone peaks. The learned model should be able to predict the local ozone peaks into two groups: ozone day or normal day.

Table 1: Metadata description of the data sets.

Data set	#Samples	#Features	#Classes
Accelerometer	27619	8	3
LIBRAS	360	91	48
RSSI	1420	15	378
Wall-Following Robot	5456	24	4
Ozone	2536	73	2

In the experiments we consider the setting of parameters related to both reservoir structure and readout classifiers. Regarding the parameters of the network models, we considered the number of neurons $n \in \{100, 200, 400\}$, the mean standard deviation $\sigma \in \{0.08, 0.15, 0.22\}$ and the density $d = 0.1$. In case of $n = 400$ or $\sigma = 0.22$, we decreased the density to 0.05 in order to assure that the reservoir satisfy the echo state property. The SW network was generated with $p = 0.1$ and the α parameter of equation 2 was fixed as 0.1. Regarding the parameters of the classifiers evaluated in the readout layer: kNN has the number of nearest neighbors k; LR has the regularization parameter C and the penalty norm pen; LSVM (linear) has the regularization parameter C; GSVM (rbf kernel) has the regularization parameter C and the kernel coefficient γ; RF has the number of trees t; MLP has the learning rate α and the number of neurons m; and CART and NB have no parameter selection step. Table 2 lists the range of parameter values evaluated for each technique. Each simulation is conducted through of a k-fold stratified cross-validation process, in which the whole data set is divided in k disjoint parts, with one being used as test and the others as training data, resulting in a total of k executions. The predictive performance is averaged over a repeated stratified cross-validation that averages three runs of 5-fold stratified cross-validation, taking the folds randomly each time. The predictive performance is given in terms of averaged weighted-F1. The parameter values selected for each classifier are also listed in Table 2.

Table 2: List of parameters for each technique used in readout and the parameter selected with the best result.

Algorithm	List of parameters	Selected
kNN	$k \in (1, 3, 5, 7, 9)$	$k = 1$
LSVM	$C \in (2^{-4}, 2^0, 2^4, 2^8)$	$C = 2^8$
LR	$C \in (2^{-4}, 2^0, 2^4, 2^8)$	$C = 2^8$
	$pen \in (l1, l2)$	$p = l1$
GSVM	$C \in (2^{-4}, 2^0, 2^4, 2^8)$	$C = 2^8$
	$\gamma \in (2^{-4}, 2^{-2}, 2^0, 2^2)$	$\gamma = 1$
RF	$t \in (2^6, 2^8, 2^{10})$	$t = 2^{10}$
MLP	$\alpha \in (0.0005, 0.002)$	$\alpha = 0.0005$
	$m \in (2n, 8n)$	$m = 2n$

Reservoir parameters analysis. In order to analyze statistically thousands of results, we adopted the Friedman test, which permits to compare multiple techniques over multiple data sets [4]. Firstly, we evaluate the n and σ configurations in the reservoir structure. The null hypothesis states that different configurations of (n, σ) are equivalent. Under the significance level at 0.05, the null hypothesis is rejected. Following the Nemenyi posthoc test is adopted to find the configurations that are not equivalent [4]. Fig. 3 is the critical difference diagram obtained from the test which compares all configurations against each other. It indicates that $(n = 200, \sigma = 0.22)$ and $(n = 400, \sigma = 0.15)$ outperform any other parameter configuration, with exception of $(n = 200, \sigma = 0.15)$ which is statistically equivalent to both. Therefore, one can see that higher values of parameters achieved better results. Notice that as thousands of simulations have been analyzed here, we omitted such results for sake of space.

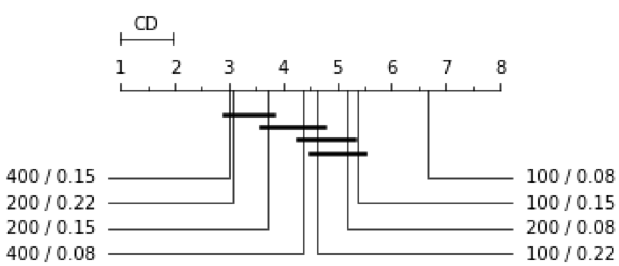


Figure 3: Critical difference diagram obtained by the Nemenyi post-hoc test considering different parameter configurations in the reservoir structure.

Readout classifier analysis. In the following, we evaluate the predictive performance of the eight classification techniques. The results of the ESN models considering the three parameter configurations with the best (and equivalent) statistical results are presented in the Table 3. The table shows the averaged weighted-F1 of each ESN in function of the network models and readout classifiers. In the table, "Reg.", "SW" and "Rand." denote respectively the regular, small-world and random network models. Taking into account the ESN network models, the best local results in the table are underlined and the best global results are boldfaced. By the table, one can see that GSVM provided the best results for most data sets, even considering the distinct network models evaluated. Interestingly, kNN also presented good results in comparison with other widely adopted readout classifiers like LR and LSVM. Furthermore, such a classifier seemed strongly related to the regular network model, obtaining 4/5 of their best results with that configuration. In that sense, LR achieved all its best results with such a network model too. In the opposite direction, LSVM achieved 3/5 of its best results with the random network.

Regarding the statistical analysis, the null hypothesis states that the results of the readout classifiers are equivalent. Under the significance level at 0.05, the null hypothesis is rejected. As shown in the diagram presented by Fig. 4, the Nemenyi posthoc test revealed that GSVM outperforms every classifier, with exception of kNN and LSVM, which the results are considered statistically equivalent to GSVM. In a few words, GSVM demonstrated consistent performance in every data set. For the Accelerometer one, GSVM with SW network model provided the best results, closely followed by GSVM with Rand. model. For the LIBRAS data set, the Reg. network model combination with GSVM provided the best results, closely followed by GSVM with SW model. For the RSSI data set, the best result was achieved when adopting GSVM with the Reg. network model. Regarding the Wall-Following robot data set, LR and GSVM with respectively Reg. and SW network models obtained the best result. Lastly, the SW network model with the LSVM readout classifier achieved the best predictive performance for the Ozone data set, closely followed by kNN and GSVM. Another technique which deserves positive attention is the kNN which achieved very good performance with much lower computational cost than GSVM: $O(N \log(N))$ against $O(N^3)$ in the worst case. Different of GSVM, kNN and LSVM, other classifiers had troubles to learn the patterns directly from the reservoir, especially the NB which presented the worse results for the Accelerometer, RSSI, Wall-Following robot and Ozone data sets.

Network models analysis. Now we move on to analyze statistically the network models investigated in this study. The null hypothesis of the Friedman test says that there are no significant difference between regular, small-word and random network models. Under the significance level at 0.05, the null hypothesis is rejected. After applying the Nemenyi posthoc test, the critical difference diagram exhibited by Fig. 5 is obtained. In the figure, one can see that the regular network model outperforms statistically the random model, although is statistically equivalent to the small-world one.

We also analyze statistically the network models in function of the three best (and statistically equivalent) readout classifiers: GSVM, kNN and LSVM. As our intention is to compare each two network models over multiple data sets, the Wilcoxon Signed Ranks

Table 3: Averaged weighted-F1 of the eight classifiers on the five real-world data sets. Boldfaced values denote the best results for a given data set and underlined ones the ESN which provided the better result for each network model.

	Accelerometer			LIBRAS			RSSI			Wall-Following Robot			Ozone		
Alg.	Rand.	Reg.	SW	Rand.	Reg.	SW	Rand.	Reg.	SW	Rand.	Reg.	SW	Rand.	Reg.	SW
KNN	94.27	94.66	94.46	93.31	94.93	94.53	73.45	74.01	74.88	90.16	90.62	90.47	93.32	93.64	93.05
LSVM	87.29	86.41	87.22	95.84	95.56	95.35	69.26	68.52	67.32	89.73	90.81	90.53	93.36	93.51	**93.79**
LR	78.83	79.62	78.43	70.43	70.92	70.54	23.8	24.78	24.16	90.27	**90.83**	90.69	89.04	89.23	89.23
GSVM	96.83	96.65	**96.96**	95.9	**97.19**	97.11	74.95	**77.01**	76.51	89.97	90.72	**90.83**	93.24	93.14	92.91
DT	86.55	87.63	87.65	54.37	54.35	59.44	16.32	15.79	15.57	90.45	90.56	90.41	78.89	80.46	80.88
RF	73.48	73.46	73.27	82.89	82.54	83.44	44.81	43.19	44.04	90.0	90.34	90.29	78.21	78.37	78.29
MLP	77.37	80.33	78.31	76.01	75.8	77.06	47.52	39.74	42.52	89.99	90.06	90.29	89.22	88.89	89.27
NB	69.9	69.72	69.21	39.4	44.84	41.54	28.55	29.42	28.85	44.18	44.85	44.85	62.66	64.88	62.28

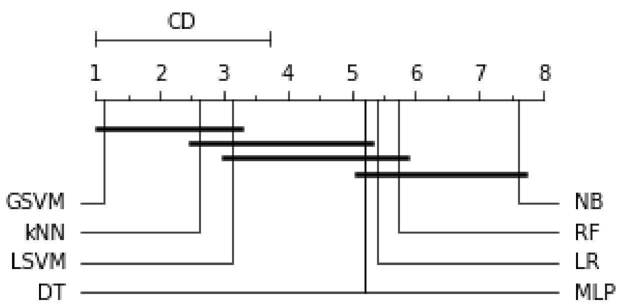


Figure 4: Critical difference diagram obtained by the Nemenyi post-hoc test considering different classification techniques in the readout layer.

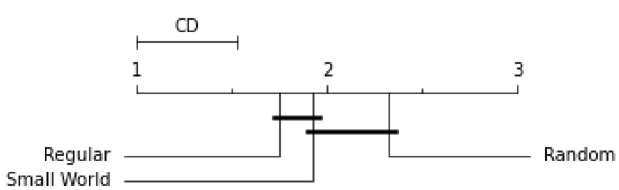


Figure 5: Critical difference diagram obtained by the Nemenyi post-hoc test considering the random, regular and small-world network models.

test has been adopted [4]. Basically, the test calculates the differences in the results of two methods for each data set and compares the ranks for the positive and negative differences. The null hypothesis says that the ranks obtained by each two network models (random vs regular, random vs small-world, and regular vs small-world) are similar. The p-values found by the Wilcoxon test are exhibited in Table 4. Under the significance level at 0.05, the test failed to reject the null hypothesis. However, under a significance level at 0.1, one can see that the regular network model is statistically superior than the random one. In addition, analyzing the results presented in Table 3, one can see that the best results were mostly obtained by the regular or small-world network models, especially with kNN and GSVM classifiers. This suggests that a higher level of organization in the reservoir structure such as those provided by the regular network may benefit many real-world applications and also allow the creation of the reservoir in a simple, easy and fast way.

Table 4: P-values found by the Wilcoxon test when analyzing the results of the three best readout classifiers (GSVM, kNN and LSVM) in function of the random (Rand.), regular (Reg.) and small-world (SW) network models.

#	Rand.	Reg.
Reg.	**0.069**	-
SW	0.139	0.410

The reservoir contribution. Finally, we also evaluate the relevance of the reservoir structure in terms of predictive performance by comparing its results against those obtained by learning the data sets directly from the readout classifiers, i.e., without the reservoir structure. Table 5 presents the averaged weighted-F1 results of GSVM, kNN and LSVM with and without the reservoir. As one can see, with exception of the Ozone data set in which the reservoir structure decreases the results provided directly from the readout classifiers, the ESN model achieved the best results in the other four data sets. In the RSSI data set, for example, their results were at least twice as high as those obtained without the reservoir. This is a very attractive result which emphasizes the salient features of ESN, especially because both models (with and without reservoir) had their parameters properly tuned. Moreover, the Wilcoxon test also attested that the models with the reservoir structure provided statistically better results than those without it (p-value < 0.01), which demonstrates the relevance of the reservoir in mapping the data inputs into a high-dimensional space for the learning process.

5 CONCLUSION

In this paper we investigated alternative methods for standard components of Echo State Networks. In a few words, we evaluated regular and small-world network models in the reservoir structure besides the extensively adopted random one; analyzed a total of eight classification techniques instead of considering just the few techniques largely adopted in the literature (most linear ones); and evaluated systematically a considerable number of parameters in

Table 5: Averaged weighted-F1 of the three best readout classifiers on the five real-world data sets considering both models: with (Res.) and without (No-Res.) the reservoir structure. Boldfaced values denote the best results for a given data set and underlined ones the model which provided the better result.

	KNN		LSVM		GSVM	
Data sets	Res.	No-Res.	Res.	No-Res.	Res.	No-Res.
Accelerometer	94.66	90.66	86.41	77.32	**96.65**	91.63
LIBRAS	94.93	91.86	95.56	91.50	**97.19**	93.43
RSSI	74.01	32.74	68.52	26.38	**77.01**	32.98
Wall-Following Robot	90.62	88.30	**90.81**	73.23	90.72	87.25
Ozone	93.64	95.55	93.51	94.10	93.16	**95.84**

both reservoir and readout layers. Experimental results with five real-world data sets showed that (i) the non-linear support vector machine classifier achieved the best predictive performance, although statistically comparable with the k-nearest neighbors one, which has much smaller time complexity; and (ii) that some problems can considerably be benefited from some level of organization in the reservoir structure, such as those provided by regular or small-world network models, which can keep the reservoir simple, well behaved and fast. Interestingly, such findings may make the adoption of ESN methods more efficient from the point of view of embedded systems and large scale problems. In the future we aim at investigating other complex networks models in the reservoir structure as well as extending our experiments to a higher number of real-world applications.

ACKNOWLEDGMENTS

Authors thank the financial support given by the Brazilian National Council for Scientific and Technological Development - CNPq (grant number 439556/2018-0, and LZB scholarship). Authors also acknowledge the support from the NVIDIA Corporation with the donation of a Titan V GPU used in this research.

REFERENCES

[1] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–166.

[2] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. 2015. The UCR Time Series Classification Archive. www.cs.ucr.edu/~eamonn/time_series_data/.

[3] Naima Chouikhi, Boudour Ammar, Nizar Rokbani, and Adel M Alimi. 2017. PSO-based analysis of Echo State Network parameters for time series forecasting. *Applied Soft Computing* 55 (2017), 211–225.

[4] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research* 7, Jan (2006), 1–30.

[5] Zhidong Deng and Yi Zhang. 2006. Complex systems modeling using scale-free highly-clustered echo state network. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 3128–3135.

[6] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[7] Aida A Ferreira and Teresa B Ludermir. 2009. Genetic algorithm for reservoir computing optimization. In *2009 International Joint Conference on Neural Networks*. IEEE, 811–815.

[8] Claudio Gallicchio, Alessio Micheli, and Luca Pedrelli. 2017. Deep reservoir computing: A critical experimental analysis. *Neurocomputing* 268 (2017), 87–99.

[9] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 6645–6649.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[11] Masanobu Inubushi and Kazuyuki Yoshimura. 2017. Reservoir computing beyond memory-nonlinearity trade-off. *Scientific reports* 7, 1 (2017), 1–10.

[12] K Ishu, Tijn van Der Zant, Vlatko Becanovic, and Paul Ploger. 2004. Identification of motion with echo state network. In *Oceans' 04 MTS/IEEE Techno-Ocean'04 (IEEE Cat. No. 04CH37600)*, Vol. 3. IEEE, 1205–1210.

[13] Herbert Jaeger. 2001. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148, 34 (2001), 13.

[14] Herbert Jaeger. 2005. Reservoir riddles: Suggestions for echo state network research. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, Vol. 3. IEEE, 1460–1462.

[15] Yuji Kawai, Jihoon Park, and Minoru Asada. 2019. A small-world topology enhances the echo state property and signal propagation in reservoir computing. *Neural Networks* 112 (2019), 15–23.

[16] Yuji Kawai, Tatsuya Tokuno, Jihoon Park, and Minoru Asada. 2017. Echo in a small-world reservoir: Time-series prediction using an economical recurrent neural network. In *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE, 126–131.

[17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[18] Decai Li, Min Han, and Jun Wang. 2012. Chaotic time series prediction based on a novel robust echo state network. *IEEE Transactions on Neural Networks and Learning Systems* 23, 5 (2012), 787–799.

[19] Mantas Lukoševičius. 2012. A practical guide to applying echo state networks. In *Neural networks: Tricks of the trade*. Springer, 659–686.

[20] Mantas Lukoševičius and Herbert Jaeger. 2009. Reservoir computing approaches to recurrent neural network training. *Computer Science Review* 3, 3 (2009), 127–149.

[21] Zeeshan Khawar Malik, Amir Hussain, and Qingming Jonathan Wu. 2016. Multilayered echo state machine: a novel architecture and algorithm. *IEEE Transactions on cybernetics* 47, 4 (2016), 946–959.

[22] Thomas Natschläger, Wolfgang Maass, and Henry Markram. 2002. The" liquid computer": A novel strategy for real-time computing on time series. *Special issue on Foundations of Information Processing of TELEMATIK* 8, ARTICLE (2002), 39–43.

[23] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*. 1310–1318.

[24] Md Jubayer Alam Rabin, Md Safayet Hossain, Md Solaiman Ahsan, Md Abu Shahab Mollah, and Md Tawabur Rahman. 2013. Sensitivity learning oriented nonmonotonic multi reservoir echo state network for short-term load forecasting. In *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*. IEEE, 1–6.

[25] Anderson Tenório Sergio and Teresa B Ludermir. 2012. PSO for reservoir computing optimization. In *International Conference on Artificial Neural Networks*. Springer, 685–692.

[26] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of 'small-world' networks. *nature* 393, 6684 (1998), 440.

[27] Yili Xia, Cyrus Jahanchahi, and Danilo P Mandic. 2014. Quaternion-valued echo state networks. *IEEE transactions on neural networks and learning systems* 26, 4 (2014), 663–673.

[28] Jun Yin, Yan Meng, and Yaochu Jin. 2012. A developmental approach to structural self-organization in reservoir computing. *IEEE transactions on autonomous mental development* 4, 4 (2012), 273–289.

[29] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* (2014).