Learning and Nonlinear Models (L&NLM) – Journal of the Brazilian Neural Network Society, Vol. 10, Iss. 3, pp. 181-191, 2012.


# ECHO STATE NETWORKS IN SEASONAL STREAMFLOW SERIES PREDICTION

**Hugo Valadares Siqueira[a]**
**Levy Boccato[b]**
**Romis Attux[b]**
**Christiano Lyra Filho[a]**

University of Campinas – UNICAMP
[a]Systems Engineering Department (DENSIS)
[b]Department of Computer Engineering and Industrial Automation (DCA),
School of Electrical and Computer Engineering
emails: {hugo, chrlyra}@densis.fee.unicamp.br, {lboccato, attux}@dca.fee.unicamp.br
Av. Albert Einstein, 400, 13083-852, Campinas, SP, Brazil

**Abstract**- Echo state networks (ESNs) are promising options for performing time series prediction, as they establish an attractive tradeoff between dynamical processing capability and simplicity in the training process. The recurrent character of these structures is essentially concentrated on a nonlinear processing stage known as dynamic reservoir, whereas the output layer – also termed *readout* – is allowed to assume the form of a combiner that is, as a rule, linear with respect to its free parameters. Recently, Butcher et al. and Boccato et al. proposed readout design paradigms – based, respectively, on Volterra filtering and extreme learning machines – that were shown to enhance the information extraction potential of ESNs without compromising the intrinsic simplicity of their training process. In this work, these approaches are comparatively analyzed, using different reservoir configurations, in the context of monthly streamflow series forecasting. The analysis is based on data from the hydroelectric plant of Furnas, covering three different periods with distinct hydrologic profiles and including the canonical linear approach for readout design as a benchmark. The results reveal that the aforementioned proposals may lead to a relevant performance improvement, thus indicating that the reservoir is not *per se* capable of fully exploring the nonlinear relationships underlying the focused data.

**Keywords**- Echo State Networks, Volterra Filtering, PCA, Extreme Learning Machine, Seasonal Streamflow Series Forecasting

## 1 Introduction

Feedforward and recurrent artificial neural networks (ANN) have been widely used for time series forecasting in several application contexts (Zhang et al., 1998) (Haykin, 1999). From a purely structural standpoint, recurrent neural networks (RNNs) would be very natural choices for performing prediction tasks, as their dynamical character allows, *in abstracto*, a parsimonious modeling of the time evolution of a sequence of samples. Nevertheless, in practice, the training process of these networks can be associated with a number of challenging factors, like the menace of instability and a relatively complex process of calculating derivatives of a given cost function.

In 2001, Jaeger introduced an elegant strategy for designing recurrent neural networks that established a remarkable balance between performance and complexity. The proposed structure, known as echo state network (ESN), is characterized by the presence of a RNN with fixed parameters, which corresponds to a reservoir of nonlinear dynamics, and of an adaptive linear output layer (Jaeger, 2001). The attractive feature of ESNs is associated with the perspective of exploiting, to a certain extent, the benefits of recurrence alongside with a simple training procedure, which essentially consists in finding the optimum coefficients of the output linear combiner in the least-squares sense.

ESNs, along with the so-called liquid state machines (LSMs), proposed by Maass (2002), have directly contributed to the emergence and establishment of a new research field called reservoir computing (Lukoševičius, 2009). In this context, recent works have brought interesting ideas aiming to improve the ESN processing capability. For instance, Ozturk et al. (2007) focused on the study of the reservoir characteristics and proposed a new reservoir design method that explores the Kautz principle of linear systems to increase the diversity of dynamical behaviors at the reservoir, something that was quantified based on the concept of average entropy.

Instead of dealing with the reservoir design issue, Butcher et al. (2010) and Boccato et al. (2011) (2012) have addressed the possibility of using alternative readout structures. In the former work, the output linear combiner was replaced by an extreme learning machine (ELM), whereas, in the latter proposal, a Volterra filtering structure was employed together with the principal component analysis (PCA) technique. These readout models offer the possibility of using a nonlinear and, thus, more flexible processing structure to create complex nonlinear mappings. Moreover, in both cases, the task of determining the coefficients of the readout remains linear with respect to the free parameters, which means that, as occurs with the original ESN training process, an optimum solution can be obtained by means of linear regression methods.

The aforementioned features of ESNs qualify these structures as interesting approaches to cope with monthly streamflow series forecasting, a very important operations research problem, especially in countries like Brazil, where the power generation is mostly based on hydroelectric plants. The application of ESNs to this particular problem has been firstly addressed by Sacchi et al. (2007), who reported promising results. In this work, we aim to expand the repertoire of analyzed ESN architectures, bringing the following main contributions: (*i*) the use of two different reservoir design methods – the original idea of Jaeger and that of Ozturk et al. (2007) and (*ii*) the application of ESNs using nonlinear readouts – the architectures proposed by Butcher et al. and Boccato et al. All the considered networks shall play the role of one-step ahead predictors in the context of three scenarios associated with the seasonal streamflow series of the Furnas Hydroeletric Plant, located in Brazil.

This work is organized as follows: Section 2 presents the echo state networks proposed by Jaeger and Ozturk et al.; Sections 3 and 4 describe the nonlinear ESN readouts introduced by Boccato et al. and Butcher et al., respectively; the experimental results are reported and analyzed in Section 5, while Section 6 presents the main conclusions and future perspectives of this work.

## 2 Echo State Networks

Recurrent neural networks represent an important class of tools within the research field of neurocomputing due to their natural capability of dealing with dynamical / temporal problems. By virtue of their being endowed with feedback connections, these structures benefit from the emergence of an internal memory, which can be very useful to time series processing. However, well-known difficulties associated with standard RNN training approaches, such as the threat of reaching unstable configurations during the adaptation process, the difficulty of computing the derivatives of the cost function and the risk of converging to local optima (Haykin, 1997) pose relevant obstacles to their practical application. In this context, echo state networks (ESN), proposed by Jaeger (2001), represent an interesting path towards circumventing these difficulties and exploring the RNN structural advantages.

The standard ESN architecture is depicted in Figure 1:

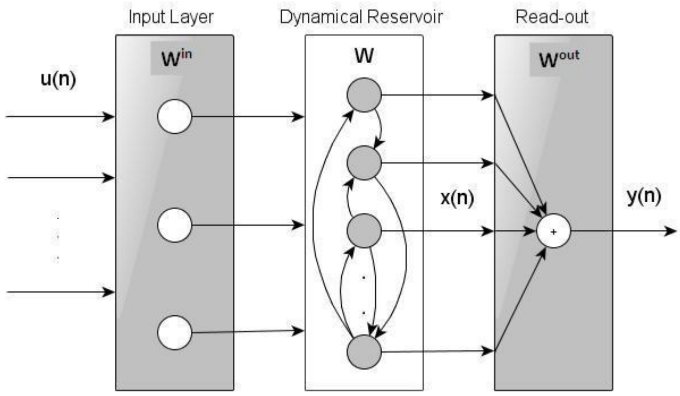



**Figure 1**: Echo State Network

The following variables are defined as:

*i*- $\mathbf{u}(n)$ – input vector, $\in \Re^{M}$;

*ii*- $\mathbf{x}(n)$ – echo states vector, $\in \Re^{N}$;

*iii*- $\mathbf{y}(n)$ – output of the network;

*iv*- $\mathbf{d}(n)$ – desired output;

*v*- $\mathbf{W^{in}}$ – matrix of weights of the input layer, $\in \Re^{NxK}$ ;

*vi*- $\mathbf{W}$ – matrix of recurrences of the hidden layer, $\in \Re^{NxN}$ ;

*vii*- $\mathbf{W^{out}}$ – matrix of weights of the output layer, $\in \Re^{LxN}$ .

The vector $\mathbf{u}(n) = [u(n), u(n-1), \ldots u(n-K+1)]^T$ contains the input samples, which are transmitted to the internal neurons by means of linear combinations, being $\mathbf{W^{in}}$ the matrix with the coefficients of such combinations. The recurrent layer, usually referred to as dynamical reservoir, is composed of highly interconnected nonlinear processing units. The vector $\mathbf{x}(n)$ specifies the network states, i.e., the activation of the internal neurons, and is updated according to the following expression:

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{W^{in}}.\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n)) \tag{1}$$

where $\mathbf{f}(.) = (f_1(.), f_2(.), .., f_N(.))$ specifies the activation functions of the neurons within the reservoir. The network output vector $\mathbf{y}(n) = [y_1(n), y_2(n), .., y_L(n)]^T$ is given by:

$$\mathbf{y}(n+1) = \mathbf{f^{out}}(\mathbf{W^{out}}\mathbf{x}(n+1)) \tag{2}$$

where $\mathbf{f^{out}}(.) = (f_1^{out}(.), f_2^{out}(.), .., f_L^{out}(.))$ stands for the activation functions of all neurons in the output layer. Throughout this work, we will assume that these activation functions are simply identity functions.

In his pioneering work, Jaeger (2001) observed that, under certain circumstances, the network states $\mathbf{x}(n)$ tend to become asymptotically independent of the initial conditions. In other words, starting from two different initial conditions $x_1(0)$ and $x_2(0)$, and with the same sequence of input stimuli, the corresponding network states converge to close values. Therefore, the effect of the input history is preponderant in the long-term reservoir dynamical behavior, and the network is said to have echo states (Jaeger, 2001). Additionally, Jaeger demonstrated that the existence of echo states is closely related to specific properties of the reservoir weight matrix. More specifically, in the case of an ESN without output feedback and whose internal neurons have hyperbolic tangent activation functions, the existence of echo states is guaranteed as long as the largest absolute singular value of $\mathbf{W}$ lies within the unit circle (Jaeger, 2001).

Since the presence of a useful memory is ensured by the echo state property, the training process of the network can be carried out according to the following steps (Jaeger, 2001):

1- Create a reservoir weight matrix $\mathbf{W}$ with spectral radius less than or equal to one;

2- Randomly define the input weights ($\mathbf{W^{in}}$), as these parameters do not affect the echo state property;

3- Determine the optimum solution for the output linear combiner in the least-squares sense.

By virtue of the linear character of the readout, the problem of choosing the ESN output weights can be directly solved in the least-squares sense, and the optimum solution can be expressed as follows:

$$\mathbf{h} = pinv(\mathbf{x}')\mathbf{d} \tag{3}$$

where $pinv(\mathbf{x}')$ denotes the Moore-Penrose generalized inverse of matrix $\mathbf{x}$, and $\mathbf{d}$ is the desired signal.

A desired feature for the ESN dynamical reservoir is that it be capable of generating a repertoire of dynamics as rich as possible in order to enable an adequate approximation of the desired signal, which means that the choice of the reservoir weights, albeit being performed separately to the network training, has to take this aspect into account. Having this in mind, Jaeger proposed an interesting strategy: to create a random sparse reservoir weight matrix according to a predefined distribution. The idea of having sparse connections within the reservoir aims to decouple groups of neurons and, hopefully, to produce a diverse repertoire of dynamical behaviors.

A different strategy, introduced by Ozturk et al. (2007), indicates that the eigenvalues of the reservoir weight matrix should be uniformly spread over a circle with radius $R < 1$. These reservoir models shall be considered in this work and are exemplified in (4) and (5), respectively:

$$w_{ij}^{Ja} = \begin{cases} +0.4 - .probability. = 0.025 \\ -0.4 - .probability. = 0.025 \\ 0 - .probability. = 0.95 \end{cases} \tag{4}$$

where $w_{ij}^{Ja}$ represents each element of $\mathbf{W}^{\mathbf{Ja}}$, $i,j = 1,\ldots,N$, and

$$\mathbf{W}^{\mathbf{Oz}} = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & -R^N \\ 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 & 0 \end{bmatrix} \tag{5}$$

In the next section, we present an extension of the standard ESN architecture proposed by Boccato et al. (2011) (2012).

## 3 Echo State Network with PCA and Volterra Filter

As discussed in Section 1, ESNs are capable of allying the processing capability of recurrent structures to the simplicity of the training process of linear filters. However, these encouraging characteristics strongly depend on the reservoir design in order that an effective trade-off between performance and computational cost be reached.

Interestingly, a different path has been studied: instead of resorting to more sophisticated reservoir design methods, it may be possible to improve the ESN performance by replacing the linear combiner at the output layer with more robust processing structures. This was the motivation of the work of Boccato et al. (2011) (2012), which proposed the use of a nonlinear readout based on the Volterra filtering structure (Haykin, 1999), which means that each network output is obtained via linear combinations of polynomial terms, as described in Equation (6):

$$y_i(n) = h_0 + \sum_{p=1}^{N} h_1(p)x_p(n) + \sum_{p=1}^{N}\sum_{q=1}^{N} h_2(p,q)x_p(n)x_q(n) + \sum_{p=1}^{N}\sum_{q=1}^{N}\sum_{r=1}^{N} h_3(p,q,r)x_p(n)x_q(n)x_r(n) + \ldots \tag{6}$$

where $x_k(n)$ represents the k-th echo state in the instant $n$, $y_i(n)$ is the i-th ESN output, $h_i(.)$ are the coefficients of the filter (linearly related to the output) and $N$ is the number of echo states. This structure not only is capable of exploiting higher-order statistics of the network states but also preserves the simplicity of the ESN training process, as the adaptation of the output coefficients amounts to a least-squares problem.

Nevertheless, as the number of echo states is increased, the number of coefficients $h_i(.)$ grows very rapidly. Because of this menace, Boccato et al. (2011) employed a well-known compression technique - principal component analysis (PCA) (Hyvärinen et al, 2001) - to reduce the number of signals that are effectively transmitted to the output layer. Since there are redundancies among the echo states (Jaeger, 2001) (Ozturk et al., 2007), the use of PCA does not necessarily deteriorate the performance of the ESN.

Therefore, this new architecture is capable of expanding the processing capability of the original ESN and, at the same time, of maintaining the key features of the ESN training process. In the context of supervised channel equalization, Boccato et al. (2011) reported significant performance improvements achieved with the novel ESN architecture when compared with conventional ESNs. These promising evidences motivated the application of such network on streamflow series forecasting.

## 4 Hybrid Architecture with Echo State Network and Extreme Learning Machine

With the purpose of improving the capability of echo state networks to create complex nonlinear mappings, Butcher et al. (2010) proposed a hybrid architecture characterized by the use of extreme learning machines (ELMs) at the ESN output layer in lieu of the traditional linear combiner.

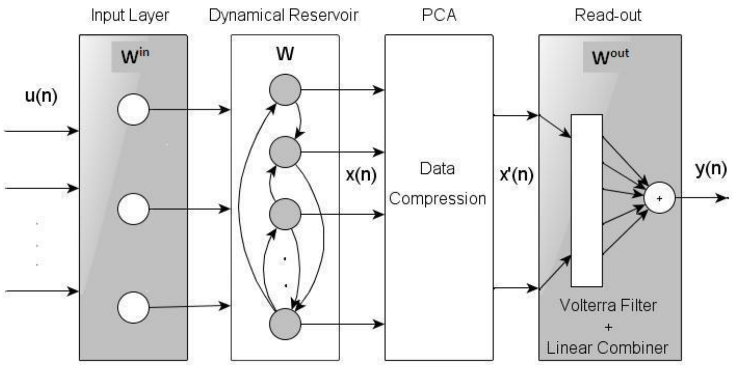



**Figure 2**: Echo State Network with PCA and Volterra Filter

ELMs are feedforward neural networks composed of two main structures: (*i*) a single hidden layer of nonlinear processing units, which is responsible for mapping data into high-dimensional spaces; (*ii*) a readout stage, which linearly combines the hidden neuron activations to generate the network outputs (Huang et al., 2004) (Huang et al., 2006). The distinctive feature of ELMs is that only the output weights are effectively adjusted with the aid of an error signal, while the hidden node parameters – input weights and biases – are randomly assigned. Hence, the ELM training process corresponds to the task of finding the optimum values of the output weights in the least-squares sense, which represents a significant simplification when compared to standard gradient-based learning algorithms (Huang et al., 2006).

Interestingly, this approach is supported by theoretical results that demonstrate that the hidden node parameters can be arbitrarily assigned as long as the activation function is infinitely differentiable (Huang et al., 2006). In addition to this aspect, due to the intrinsic properties of the Moore-Penrose generalized inverse operation, the optimum solution not only minimizes the mean squared error but also the norm of the output weight vector, which, according to Bartlett's standpoint (Bartlett, 1998), contributes in the sense of improving the generalization capability of the network.

In this work, we shall consider a simplified version of the model proposed by Butcher et al. (2010): here, a single ELM forms the readout, which receives only the reservoir dynamics to produce the network outputs, similarly to what occurs in the proposal of Boccato et al. (2011) (2012). Figure 3 depicts the hybrid architecture explored in this work.

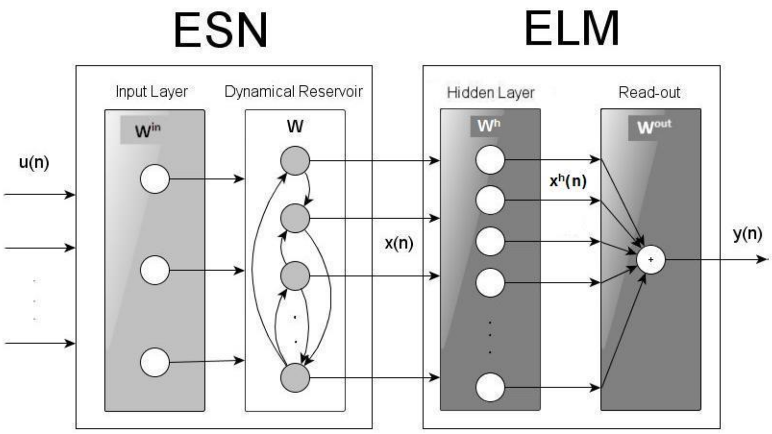



**Figure 3**: Hybrid Architecture: Echo State Network with Extreme Machine Learning

The activation of the hidden layer neurons is determined by the following expression:

$$\mathbf{x_h}(n) = \mathbf{f^h}(\mathbf{W^h}\mathbf{x}(n) + \mathbf{b}) \tag{9}$$

where $\mathbf{W^h} \in \Re^{N_{ext} x N}$ contains the input weights, **b** specifies the random bias of each hidden unit and $\mathbf{f^h}(.)$ stands for the activation function of the hidden neurons. Then, the ELM outputs are obtained by means of linear combinations of the hidden activations, as shown in the following expression:

$$\mathbf{y}(n) = \mathbf{W^{out}}\mathbf{x_h}(n) \tag{10}$$

where $\mathbf{W^{out}}$ contains the coefficients of such combinations. Since only the ELM output weights are adjusted, and the problem of obtaining the optimum solution is linear with respect to the output coefficients, the simplicity of the ESN training process is preserved in the context of the hybrid architecture.

# 5 Seasonal Streamflow Series Forecasting

## 5.1 Monthly Seasonal Streamflow Series and Preprocessing

Monthly seasonal streamflow series are non-stationary and present seasonal components that reflect the rainy and dry periods associated with the nearby rivers. Such components affect the performance of any predictor, linear or not (Luna and Ballini, 2011). Thus, it is quite useful to employ some kind of deseasonalization technique to remove the seasonal components, reinserting them only at the end of the forecasting process. Equation (11) displays the deseasonalization process adopted in this work:

$$z_{i,m} = \frac{s_{i,m} - \mu_m}{\sigma_m} \tag{11}$$

where $s_{i,m}$ are the samples of the original series $\mathbf{s}(n)$, which is transformed into a new deseasonalized series $\mathbf{z}(n)$ with zero mean and standard deviation equal to one. The average $\mu_m$ and the standard deviation $\sigma_m$ of each month $m$ are estimated by:

$$\mu_m = \frac{1}{N_y}\sum_{i=1}^{N_y} s_{i,m} \tag{12}$$

$$\sigma_m = \sqrt{\frac{1}{N_y}\sum_{i=1}^{N_y}(s_{i,m} - \mu_m)^2} \tag{13}$$

where $s_{i,m}$ denotes the streamflow in the year $i = 1,2,\ldots,N_y$, and in the month $m = 1,2,\ldots,12$.

In summary, we applied the previously described ESN architectures to the forecasting of the series $\mathbf{z}(n)$ in training and test sets. At the end of the prediction process, the seasonal components are reinserted, allowing the performance assessment in the original domain of the streamflow series.

## 5.2 Computational Results and Discussion

The prediction scenarios involved in this Section are associated with three periods of the monthly seasonal streamflow series of the Furnas Hydroeletric Plant, located at the Rio Grande river, Brazil. The data samples are available on the ONS - Electric System National Operator – website (*http://www.ons.org.br/operacao/vazoes_naturais.aspx*), and we considered the monthly streamflow values from 1931 to 1990. The test sets are composed of the samples in the following periods: (1) 1952 to 1956 – (dry), whose average equals to 656,41 $m^3/s$; 1972 to 1976 – (medium), whose average equals to 882,63 $m^3/s$; and 1981 to 1985, (wet), whose average equals to 942,04 $m^3/s$. All test periods comprise 5 years, or 60 samples, and are commonly used in this kind of study (Siqueira et al., 2011) due to their particular characteristics. The training set is built with all the samples available in the streamflow series, except those associated with the 5 years of each test set.

Initially, the deseasonalization process described on Section 5.1 was applied to each case and, then, the average of each set was subtracted, before the input data was presented to the ESN. All ESN architectures were trained with two inputs, i.e., using the samples associated with current month and the previous one. This choice was based on preliminary experiments and attempted to represent a compromise between performance and parsimony. The forecasting horizon was always one step ahead.

Based on preliminary tests, the proposal by Boccato et al. included only the first- and third-order terms of the Volterra expansion, and two principal components were considered in PCA. The reservoir of all architectures was designed according to the probability distribution described on Jaeger (2001), and the spectral radius was equal to 0.8 in the case of the reservoir method of Ozturk et al. (2007). With respect to the proposal of Butcher et al., the number of neurons in the ELM hidden layer was also defined based on preliminary tests.

Tables 1 to 3 present the performance of the ESNs based on the Mean Square Error (MSE) and Mean Absolute Error (MAE), considering an average of 20 independent simulations, with respect to both the training and test sets, in real and deseasonalized domains, where $N$ is the number of echo states and $N_{ext}$ the number of neurons in the hidden layer of the ELMs, to the hybrid architecture.

$$MSE = \frac{1}{N_s}\sum_{t=1}^{N_s}(d(n) - y(n))^2 \qquad (14)$$

$$MAE = \frac{1}{N_s}\sum_{t=1}^{Ns}|d(n) - y(n)| \qquad (15)$$

where $d(n)$ is the observed data, $y(n)$ is the output of the network and $N_s$ the number of samples.

**Table 1**: Mean Absolute Error and Mean Square Error for series FURNAS 1952/1956

| ESN | $N/N_{ext}$ | MSE ($10^4$) | MAE ($10^4$) | MSE deseas | MAE deseas | MSE ($10^4$) train | MAE ($10^4$) train | MSE deseas train | MAE deseas train |
|---|---|---|---|---|---|---|---|---|---|
| **Jaeger** | 15 | 7.3308 | 186.1698 | 0.2985 | 0.4379 | 10.9855 | 212.5689 | 0.6020 | 0.5432 |
| **Ozturk** | 25 | 7.2037 | 180.3335 | 0.2823 | 0.4320 | 10.8298 | 207.9432 | 0.5692 | 0.5241 |
| **Boc.+Jaeg.** | 30 | 5.4968 | 165.7800 | 0.2491 | 0.4198 | 11.1337 | 216.2583 | 0.5831 | 0.5491 |
| **Boc. +Ozt.** | 80 | 5.5751 | 164.4064 | 0.2471 | 0.41403 | 10.8662 | 212.8069 | 0.5632 | 0.5404 |
| **Jaeg. +ELM** | 10/40 | 4.4674 | 149.2833 | 0.2214 | 0.3914 | 10.1403 | 205.4675 | 0.5362 | 0.5253 |
| **Ozt. +ELM** | 7/50 | 4.8742 | 152.8290 | 0.2139 | 0.3760 | 10.4620 | 205.0174 | 0.5356 | 0.5184 |

**Table 2**: Mean Absolute Error and Mean Square Error for series FURNAS 1972/1976

| ESN | $N/N_{ext}$ | MSE ($10^4$) | MAE ($10^4$) | MSE deseas | MAE deseas | MSE ($10^4$) train | MAE ($10^4$) train | MSE deseas train | MAE deseas train |
|---|---|---|---|---|---|---|---|---|---|
| **Jaeger** | 15 | 6.6845 | 183.1000 | 0.4068 | 0.4995 | 11.1632 | 207.8543 | 0.5761 | 0.5179 |
| **Ozturk** | 25 | 7.5629 | 192.9986 | 0.4330 | 0.5300 | 10.3911 | 203.2150 | 0.5643 | 0.5148 |
| **Boc.+Jaeg.** | 30 | 7.0115 | 180.3267 | 0.3838 | 0.4917 | 11.1091 | 214.4129 | 0.5803 | 0.5421 |
| **Boc. +Ozt.** | 80 | 6.5607 | 171.5009 | 0.3603 | 0.4686 | 10.7798 | 211.2429 | 0.5602 | 0.5339 |
| **Jaeg. +ELM** | 5/50 | 5.0143 | 159.4289 | 0.3758 | 0.4635 | 10.2798 | 206.7741 | 0.5178 | 0.5176 |
| **Ozt. +ELM** | 7/60 | 6.0904 | 182.9762 | 0.4293 | 0.5238 | 10.1119 | 200.9086 | 0.5333 | 0.5105 |

**Table 3**: Mean Absolute Error and Mean Square Error for series FURNAS 1981/1985

| ESN | $N/N_{ext}$ | MSE ($10^4$) | MAE ($10^4$) | MSE deseas | MAE deseas | MSE ($10^4$) train | MAE ($10^4$) train | MSE deseas train | MAE deseas train |
|---|---|---|---|---|---|---|---|---|---|
| **Jaeger** | 110 | 25.3856 | 339.6919 | 1.6505 | 0.9090 | 6.6726 | 166.7190 | 0.30821 | 0.4223 |
| **Ozturk** | 100 | 23.1586 | 339.3481 | 1.6114 | 0.9158 | 7.2630 | 173.4588 | 0.32943 | 0.4378 |
| **Boc.+Jaeg.** | 120 | 22.8347 | 341.3128 | 1.6265 | 0.9164 | 8.5571 | 186.8554 | 0.38152 | 0.4662 |
| **Boc. +Ozt.** | 60 | 23.1160 | 347.8526 | 1.6342 | 0.9294 | 8.8430 | 189.3176 | 0.38931 | 0.4701 |
| **Jaeg. +ELM** | 100/60 | 22.4578 | 346.8280 | 1.7028 | 0.9425 | 7.7406 | 181.8424 | 0.36621 | 0.4617 |
| **Ozt. +ELM** | 10/40 | 24.1913 | 352.2593 | 1.7842 | 0.9635 | 8.2486 | 185.5712 | 0.38012 | 0.4678 |

Finally, the ANOVA Friedman's test was used to check whether the approaches actually provided different results (Luna and Ballini, 2011). The p-values achieved were: 6.52336e-009, for the period 1952/1956; 2.88366e-007 for 1972/1986; and 0.0975 to 1981/1985. This indicates that the prediction performances are, indeed, different or, in other words, that the prediction method directly affects the overall results.

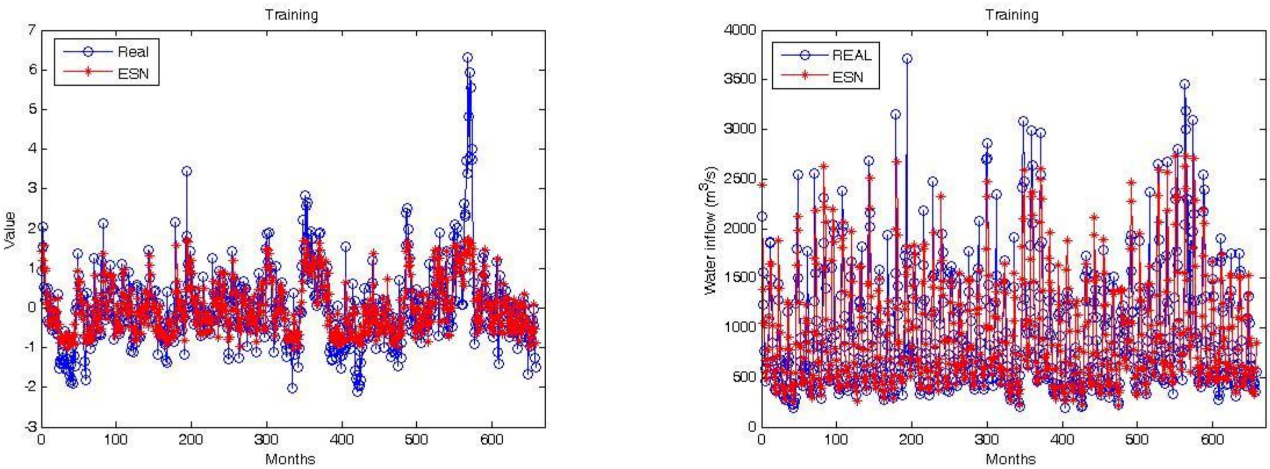
The results presented in Tables 1, 2 and 3 lead to the following important observations. Firstly, there are cases in which there is no direct correspondence between the best values of MSE and MAE. Secondly, achieving the best results in the deseasonalized domain does not necessarily mean that, in the real domain, *i.e.*, when the deseasonalization is reverted, the best performance is obtained as well. This problem is intrinsic to the prediction of this kind of series and occurs because of the deseasonalization process treats equally months with different standard deviations. With respect to the reservoir design methods, the obtained results do not indicate a clear preference in this task for the proposals of Jaeger or of Ozturk et al.

The classical ESN models of Jaeger and of Ozturk et al., in general, presented better results in the training set when compared with the architecture of Boccato et al., and with the architecture from Butcher et al. in the 81/85 test set. However, this observation does not hold for the test sets. We have checked the possibility of overfitting, but a reduction in the number of neurons within the reservoir tends to deteriorate the overall results of these classical ESNs.

It is possible to observe in Tables 1 to 3 that the introduction of nonlinear output layers – a Volterra Filter along with PCA or an ELM – led to a reduction in the error on the test sets – in all cases in the real domain, and in two out of three in the deseasonalized domain. This means that, albeit the best results with respect to the training set were obtained with the classical ESN approaches, the ESNs using nonlinear readouts were capable to absorb, in a balanced way, the characteristics of the time series, extracting the essential information of the training set without compromising the generalization capability.

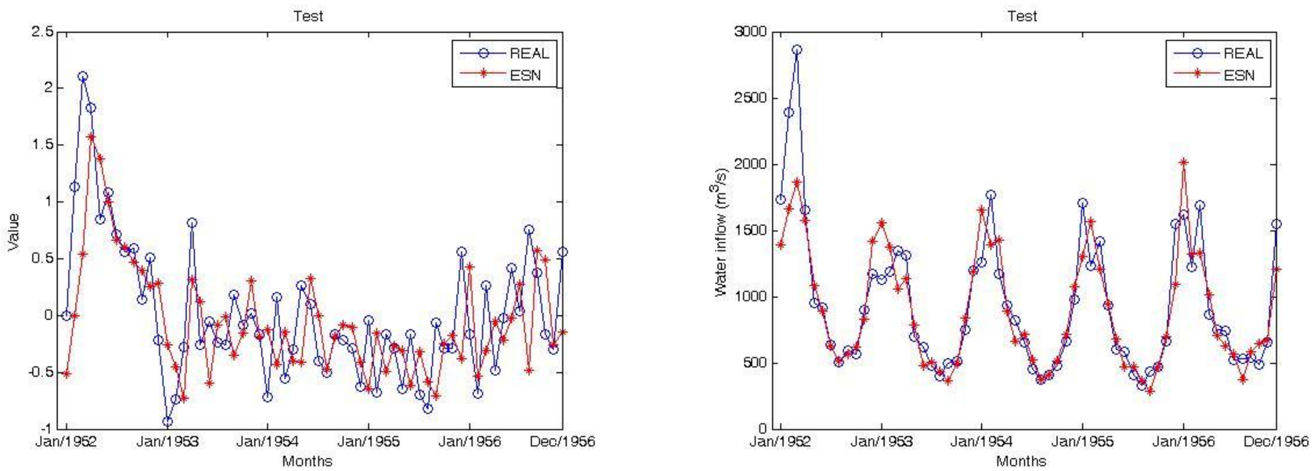
It is also important to notice that the approach from Boccato et al. required a higher number of internal neurons, which suggests that it is necessary to use more echo states to attain a sufficiently adequate compression without losing a significant amount of information about the input signal. The number of principal components assumed small values in order to impose a parsimonious situation to the network. The architecture from Butcher et al. presented the best overall results, and, in most of the cases, we employed more neurons in the ELM hidden layer than in the ESN reservoir.

Based on all these observations, it is possible to affirm that the obtained results show the benefits achieved with the introduction of a more flexible output layer in the time series prediction. Aiming to illustrate the performance of each network, Figures 4 to 9 exhibit the original streamflow series along with the best predicted values, both in real and deseasonalized spaces.



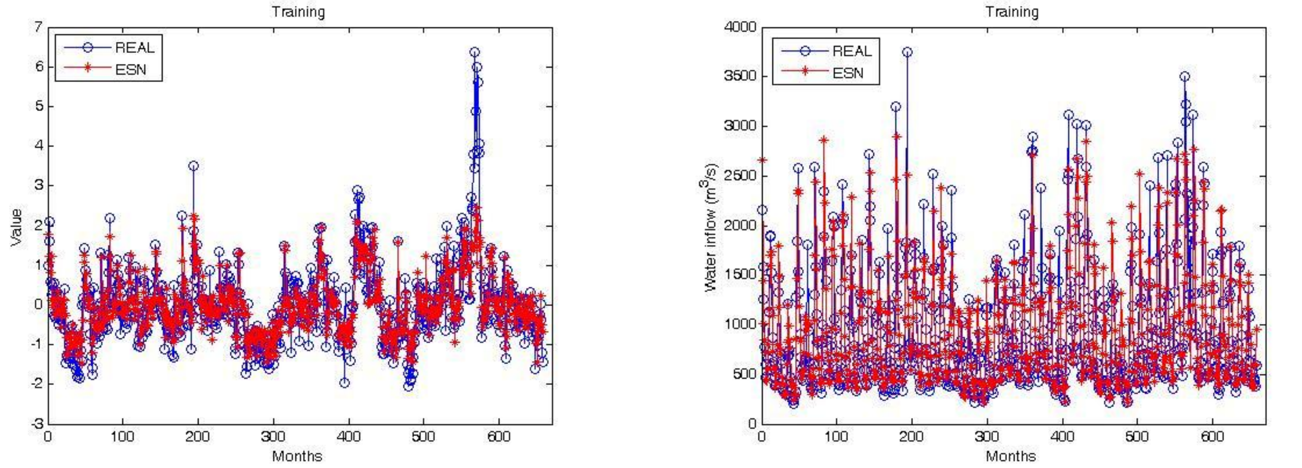

(a) (b)

**Figure 4**: Best performance for series FURNAS 1952/1956 in deseasonalized (a) and real (b) spaces - Training



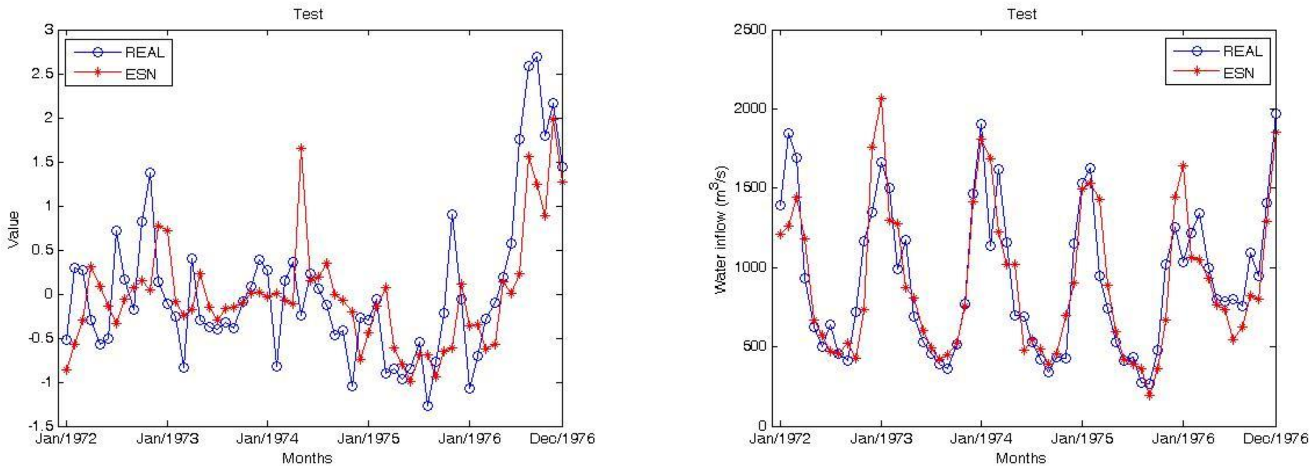

(a) (b)

**Figure 5**: Best performance for series FURNAS 1952/1956 in deseasonalized (a) and real (b) spaces - Test



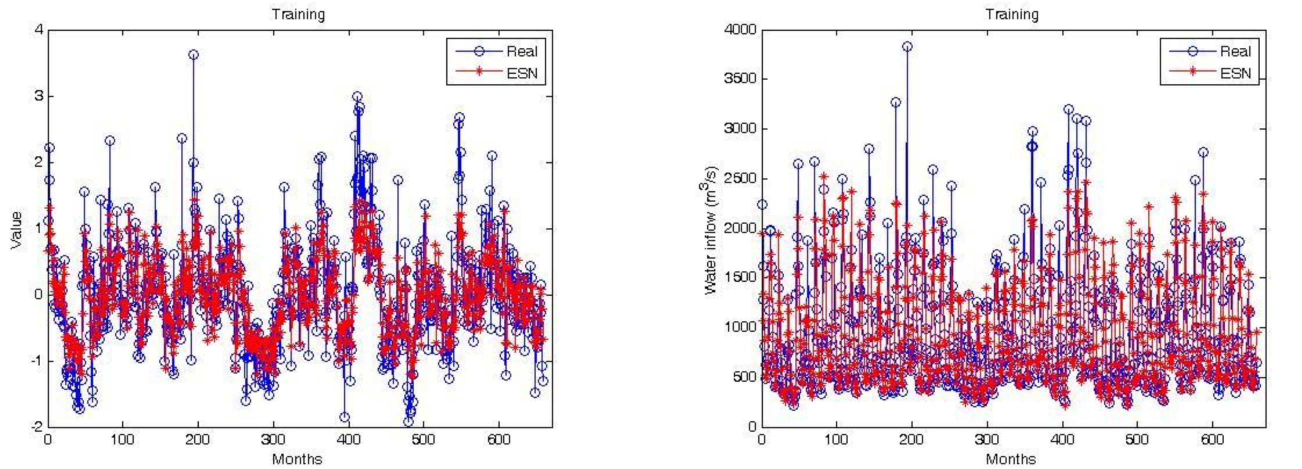

(a) (b)

**Figure 6**: Best performance for series FURNAS 1972/1976 in deseasonalized (a) and real (b) spaces - Training



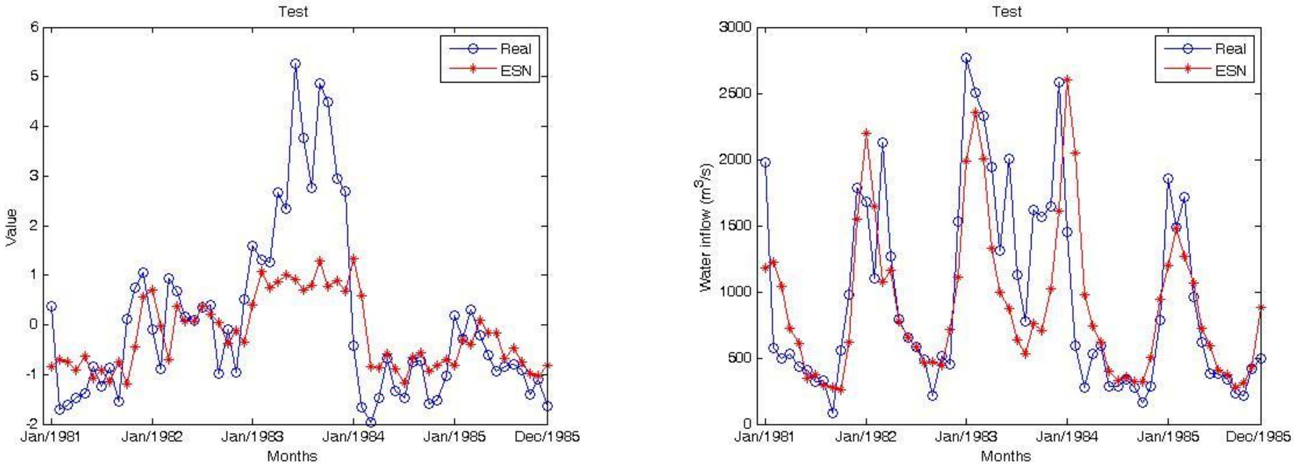

(a) (b)

**Figure 7**: Best performance for series FURNAS 1972/1976 in deseasonalized (a) and real (b) spaces - Test




(a) (b)

**Figure 8**: Best performance for series FURNAS 1981/1985 in deseasonalized (a) and real (b) spaces - Training




(a) (b)

**Figure 9**: Best performance for series FURNAS 1981/1985 in deseasonalized (a) and real (b) spaces - Test

## 6 Conclusion

This work presented a comparative study of the application of different architectures of echo state networks (ESNs) to seasonal streamflow series forecasting, covering proposals from Jaeger, Ozturk et al, Boccato et al. and Butcher et al., of which the last two are characterized by the use of nonlinear readouts. The main motivation underlying the application of ESNs to this problem is the perspective of combining the advantages of a recurrent structure to a training process that is kept as simple as possible.

The work performed a prediction study for three different periods of the monthly seasonal streamflow series from Furnas Hydroelectric Plant, always aiming at an estimate one step ahead. It showed that that the proposals of Boccato et al. and Butcher et al. brought effective performance gains both in the real and the deseasonalized domains. In essence, this means that the introduction of a nonlinear readout leads to a more effective use of the higher-order information present in hydrological time series. Hence, nonlinear output layers should be considered as relevant options to solve problems of this kind using ESNs, with the caveat that the training process must always be suitably conceived.

As perspectives for future work, we indicate: a) the application of the various ESN paradigms to prediction tasks with a time horizon transcending the single-step case; b) the use of a vaster repertoire of hydrological time series; c) investigations about improvements in the deseasonalization process, thereby enhancing the correlation between the deseasonalized and real domains.

## Acknowledgment

This work was supported by grants from CAPES, CNPq and FAPESP.

## References

Ballini, R. (2000). Análise e Previsão de Vazões Utilizando Séries Temporais, Redes Neurais e Redes Neurais Nebulosas, Tese de Doutorado, FEEC-Unicamp, Brasil.

Bartlett, P. L. (1998). The Sample Complexity of Pattern Classification with Neural Networks: the Size of the Weights is More Important than The Size of the Network. IEEE Transactions on Information Theory, 44(2), pp. 525-536.

Boccato, L., Lopes A., Attux, R., Von Zuben F. J. (2011). An Echo State Network Architecture Based on Volterra Filtering and PCA with Application to the Channel Equalization Problem, *IEEE* Proceedings of International Joint Conference on Neural Networks, San Jose - CA, USA.

Boccato, L., Lopes, A., Attux, R., Von Zuben, F. J. (2012). An Extended Echo State Network Using Volterra Filtering and Principal Component Analysis", Neural Networks, volume 32, pp. 292-302, 2012.

Box, G., Jenkins, G. e Reinsel, G. C. (1994). Time Series Analysis, Forecasting and Control, 3rd ed., Holden Day, Oakland, California, EUA.

Butcher, J., Verstraeten D., Schrauwen B., Day, C. and Haycock, P. (2010). Extending Reservoir Computing with Random Static Projections: a Hybrid Between Extreme Learning and RC, European Symposium on artificial Neural Networks - Computational Intelligence and Machine Learning, Bruges, Belgium.

Haykin, S. (1997). Adaptive Filter Theory, Prentice Hall.

Haykin, S. (1999). Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice-Hall.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2004). Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. Proceedings of the International Joint Conference on Neural Networks (IJCNN 2004), pp. 985-990.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme Learning Machine: Theory and Applications. Neurocomputing, 70 (1-3), pp. 489-501.

Hyvärinen A., Karhunen J. e Oja E. (2001). Independent Component Analysis, New York: John Wiley & Sons.

Jaeger, H. (2001). The echo state approach to analyzing and training recurrent neural networks, Bremem: German National Research Center for Information Technology, Tech. Rep. GMD Report 148.

Lukoševičius, M. and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training, Computer Science Review, v. 3, Issue 3, pp. 127-149 .

Luna, I. and Ballini, R. (2011). Top-Down Strategies Based on Adaptive Fuzzy Rule-Based Systems for Daily Time Series Forecasting, International Journal of Forecasting, p. 1-17.

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. Neural Computation, 14(11), pp. 2531-2560.

Ozturk, M. C., Xu, D. and  Principe J. C. (2007). Analysis and design of echo state networks, Neural Computation, vol. 19, pp. 111–138.

Sacchi, R., Ozturk, M. C., Príncipe, J. C., Carneiro, A. A. F. M., da Silva, I. N. (2007). Water Inflow Forecasting Using the Echo State Network: a Brazilian Case Study, *IEEE* Proceedings of International Joint Conference on Neural Network, Orlando FL, USA.

Siqueira, H. V., Boccato, L., Attux, R., Lyra Filho, C. (2011). Seasonal Streamflow Series Forecasting Using Echo State Networks, In Portuguese, 10th Brazilian Congress on Computational Intelligence, Fortaleza-CE, Brazil [Previsão de séries de vazões com redes neurais de estados de eco].

Zhang, G.., Patuwo, B. E., Hu, M. Y. (1998). Forecasting with Artificial Neural Networks: the State of the Art, International Journal of Forecasting, Vol. 14, pp. 35-62.