

Article

Neural-Based Ensembles and Unorganized Machines to Predict Streamflow Series from Hydroelectric Plants

Jônatas Belotti ^{1,2}, Hugo Siqueira ¹, Lilian Araujo ¹, Sérgio L. Stevan Jr. ¹, Paulo S.G. de Mattos Neto ³, Manoel H. N. Marinho ⁴, João Fausto L. de Oliveira ⁴, Fábio Usberti ², Marcos de Almeida Leone Filho ⁵, Attilio Converti ⁶ and Leonie Asfora Sarubbo ^{7,8,*}

¹ Graduate Program in Computer Sciences, Federal University of Technology–Parana (UTFPR), Ponta Grossa 84017-220, Brazil; jonatas.t.belotti@hotmail.com (J.B.); hugosiqueira@utfpr.edu.br (H.S.); lilian.araujo@ifpr.edu.br (L.A.); sstevanjr@utfpr.edu.br (S.L.S.J.)

² Institute of Computing, State University of Campinas (UNICAMP), Campinas 13083-852, Brazil; fusberty@ic.unicamp.br

³ Departamento de Sistemas de Computação, Centro de Informática, Universidade Federal de Pernambuco, (UFPE), Recife 50740-560, Brazil; psgmn@cin.ufpe.br

⁴ Polytechnic School of Pernambuco, University of Pernambuco, Recife 50100-010, Brazil; marinho75@poli.br (M.H.N.M.); fausto.lorenzato@upe.br (J.F.L.d.O.)

⁵ Venidera Pesquisa e Desenvolvimento, Campinas 13070-173, Brazil; marcos@venidera.com

⁶ Department of Civil, Chemical and Environmental Engineering, University of Genoa (UNIGE), 16126 Genoa, Italy; converti@unige.it

⁷ Department of Biotechnology, Catholic University of Pernambuco (UNICAP), Recife 50050-900, Brazil

⁸ Advanced Institute of Technology and Innovation (IATI), Recife 50751-310, Brazil

* Correspondence: leonie.sarubbo@unicap.br; Tel.: +55-81-2119-4084

Received: 2 July 2020; Accepted: 4 September 2020; Published: 12 September 2020



Abstract: Estimating future streamflows is a key step in producing electricity for countries with hydroelectric plants. Accurate predictions are particularly important due to environmental and economic impact they lead. In order to analyze the forecasting capability of models regarding monthly seasonal streamflow series, we realized an extensive investigation considering: six versions of unorganized machines—extreme learning machines (ELM) with and without regularization coefficient (RC), and echo state network (ESN) using the reservoirs from Jaeger’s and Ozturk et al., with and without RC. Additionally, we addressed the ELM as the combiner of a neural-based ensemble, an investigation not yet accomplished in such context. A comparative analysis was performed utilizing two linear approaches (autoregressive model (AR) and autoregressive and moving average model (ARMA)), four artificial neural networks (multilayer perceptron, radial basis function, Elman network, and Jordan network), and four ensembles. The tests were conducted at five hydroelectric plants, using horizons of 1, 3, 6, and 12 steps ahead. The results indicated that the unorganized machines and the ELM ensembles performed better than the linear models in all simulations. Moreover, the errors showed that the unorganized machines and the ELM-based ensembles reached the best general performances.

Keywords: monthly seasonal streamflow series forecasting; artificial neural networks; Box-Jenkins models; ensemble

1. Introduction

Planning the operation of a power generation system is defined by establishing the use of energy sources in the most efficient way [1–3]. Renewable sources are those with the lowest operation cost since the fuel is provided free of charge by nature. Good predictions of river streamflows allow resource management according to their future availability [4]. Therefore, this is mandatory for countries where there are hydroelectric plants [5–7].

The International Hydropower Association published the Hydropower Status Report 2020 [8], showing that 4306 TWh of electricity was generated in the world using hydroelectric plants in 2019. This amount represents the single most significant contribution from a renewable energy source in history. The document summarizes data from 13,000 stations in 150 countries. The top countries in hydropower installed capacity are China (356.40 GW), Brazil (109.06 GW), United States (102.75 GW), and Canada (81.39 GW).

In this context, it is important to predict accurate information about rivers' monthly seasonal streamflow, since it makes the turbines spin, transforming kinetic into electric energy [5,9]. These series present a specific seasonal behavior due to the volume of water throughout the year being mostly dependent on rainfall [10,11]. Ensuring efficient operation of such plants is needed, since it significantly impacts cost of production and suitable use of water [12,13]. Additionally, their operation leads to a smaller environmental impact than burning carboniferous fuel. Due to this, many pieces of research have presented investigations on such fields for countries such as China [14], Canada [15], Serbia [16], Norway [17], Malaysia [7], and Brazil [9].

Linear and nonlinear methodologies have been proposed to solve this problem. As discussed in [5,12], and [18], the linear methods of the Box-Jenkins family are widely used [19]. The autoregressive model (AR) is highlighted because its easy implementation process allows the calculation of its free coefficients in a simple and deterministic manner. An extended proposal for this task is the autoregressive and moving average model (ARMA), a more general methodology that uses the errors of past predictions to form the output response [19,20].

However, artificial neural networks (ANN) are prominent for this kind of problem [9,21–24]. They were inspired by the operation of the nervous system of superior organisms, recognizing data regularities and patterns through training and determining generalizations based on the acquired knowledge [18,25–27].

In recent times, some studies have indicated that the best results for time series forecasting can be achieved by combining different predictors using ensembles [28–30]. Many authors have applied these techniques to similar tasks [31–33]. However, the approaches commonly explore only the average of the single models output or the classic neural networks (multilayer perceptron (MLP) and radial basis function networks (RBF)) as a combiner. The specialized literature regarding streamflow forecasting shows that ANN approaches stand out, but some authors use linear models [16], support vector regression [14], and ensembles [15].

This work proposes using a special class of neural networks, the unorganized machines (UM), to solve the aforementioned forecasting task. The term UM defines the extreme learning machines (ELM) and echo state network (ESN) collectively. In this investigation, we addressed six versions of UMs: ELM with and without regularization coefficient (RC) as well as ESN using the reservoir designs from Jaeger's and Ozturk et al. with and without RC. Additionally, we addressed the ELM and the ELM (RC) as the combiner of a neural-based ensemble.

To realize an extensive comparative study, we addressed two linear models (AR and ARMA models); four well-known artificial neural networks (MLP, RBF, Elman network, Jordan Network); and four other ensembles, using as combiners the average, the median, the MLP, and the RBF. To the best of our knowledge, the use of ELM ensembles in this problem and similar repertoires of models is an investigation not yet accomplished. Therefore, we would like to fill this gap.

In this study, the database is from Brazil. In the country, electric energy is mostly generated by hydroelectric plants, these being responsible for 60% of all electric power produced in 2018 [8,34,35].

In addition, Brazil is the one of the largest producers of hydropower in the world. Therefore, the results achieved can be extensible for other countries.

The remainder of this work is organized as follows: Section 2 discusses the linear models from the Box-Jenkins methodology; Section 3 presents the artificial neural networks and the ensembles; Section 4 shows the case study, the details on the seasonal streamflow series, the computational results, and the result analysis; and Section 5 shows the conclusions.

2. Linear Forecasting Models

The definition of linear prediction models by Box-Jenkins makes use of linear filtering concepts [19]. The x_t element of a time series results from a linear filter Ψ on a Gaussian white noise a_t . Another form to represent a linear model is by weighing the previous signals ($x_{t-1}, x_{t-2}, \dots, x_1$) for the next forecast element (x_t). To do this, we add a noise a_t and the mean of the series μ :

$$x_t = \mu + a_t + \pi_1 x_{t-1} + \pi_2 x_{t-2} + \dots + \pi_t x_1 \quad (1)$$

where a_t is the noise of t -th term, and π_n is the weight assigned to the $(t - n)$ -th term of series.

2.1. Autoregressive Model

Given any value x_t of a time series, the delay p is defined with x_{t-p} . An autoregressive process of order p (AR(p)) is defined as the linear combination of p delays of observation x_t , with the addition of a white Gaussian noise a_t , as shown in Equation (2) [19]:

$$\tilde{x}_t = \phi_1 \tilde{x}_{t-1} + \phi_2 \tilde{x}_{t-2} + \dots + \phi_p \tilde{x}_{t-p} + a_t \quad (2)$$

where $\tilde{x}_t = x_t - \mu$, ϕ_p is the weighting coefficient for the delay p .

The term a_t is considered as the inherent error of the regression process. This is the error of the forecast when the model is used to predict future values. Thus, the optimum ϕ_p coefficients must be calculated to minimize the error a_t [36].

To determine the optimum values of ϕ_p it is necessary to solve a recurrence relation that emerges from its autocorrelation function, as presented in Equation (3):

$$\rho_j = \phi_1 \rho_{j-1} + \phi_2 \rho_{j-2} + \dots + \phi_p \rho_{j-p}, \quad \forall j > 0 \quad (3)$$

If we expand this relation to $j = 1, 2, \dots, p$ we obtain the set of linear equations denominated Yule–Walker equations, which define $\phi_1, \phi_2, \dots, \phi_p$ as a function of $\rho_1, \rho_2, \dots, \rho_p$ for a model AR(p), as in Equation (4) [19]:

$$\begin{aligned} \rho_1 &= \phi_1 \rho_0 + \phi_2 \rho_1 + \dots + \phi_p \rho_{p-1} \\ \rho_2 &= \phi_1 \rho_1 + \phi_2 \rho_0 + \dots + \phi_p \rho_{p-2} \\ \rho_3 &= \phi_1 \rho_2 + \phi_2 \rho_1 + \dots + \phi_p \rho_{p-3} \\ &\vdots \\ \rho_p &= \phi_1 \rho_{p-1} + \phi_2 \rho_{p-2} + \dots + \phi_p \rho_0 \end{aligned} \quad (4)$$

2.2. Autoregressive and Moving Average Model

Unlike the autoregressive, in the moving average model (MA), white noise signals are combined [19]. A model MA is said to be of order q if the prediction of x_t utilizes q samples of white noise signals, as in Equation (5):

$$x_t = -\theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q} + a_t \quad (5)$$

where $\theta_t, \forall t \in 1, 2, \dots, q$ are the parameters of model.

An ARMA model is the union of AR and MA. To predict using the ARMA of order p, q , it is necessary to address p prior signals (AR) and q white noise signals (MA). Mathematically, an ARMA(p, q) model is described as in Equation (6):

$$x_t = \phi_1 \tilde{x}_{t-1} + \phi_2 \tilde{x}_{t-2} + \dots + \phi_p \tilde{x}_{t-p} - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q} + a_t \quad (6)$$

where ϕ_t and θ_t are the model parameters.

Unlike the AR, the calculation of the ARMA coefficient is done by solving nonlinear equations. However, it is possible to achieve an optimal linear predictor if the choice of these coefficients is adequate [18,19].

3. Artificial Neural Network

Artificial neural networks (ANN) are distributed and parallel systems composed of simple data processing units. These units are denominated artificial neurons and are capable of computing mathematical functions, which, in most cases, are nonlinear [37,38]. Artificial neurons are connected by normally unidirectional connections and can be arranged in one or more layers [25].

The ANNs present a learning ability through the application of a training method. They can generalize the knowledge acquired through the solution of problem instances for which no answer is known [39]. Neural networks are widely used in many areas of science, engineering computing, medicine, and others [9,25,40–45].

3.1. Multilayer Perceptron (MLP)

The multilayer perceptron (MLP) consists of a set of artificial neurons arranged in multiple layers so that the input signal propagates through the network layer by layer [25]. It is considered one of the most versatile architectures for applicability and is used in the universal approximation of functions, pattern recognition, process identification and control, time series forecasting, and system optimization [25,29].

The training process consists of adjusting the synaptic weights of the artificial neuron to find the set that achieves the best mapping of the desired event [46,47]. The most known training method for MLP is the steepest descent in which the gradient vector is calculated using the backpropagation algorithm [48,49].

The error signal of a neuron j in iteration t is given by Equation (7):

$$e_j(t) = d_j(t) - y_j(t) \quad (7)$$

where $e_j(t)$ is the error, $d_j(t)$ is the expected result (desired output), and $y_j(t)$ is the output of the network.

Finally, the rule for updating the synaptic weights of each neuron is done using Equation (8):

$$w_{ij}^m(t+1) = w_{ij}^m(t) - \alpha \frac{\partial E(t)}{\partial w_{ij}^m(t)} \quad (8)$$

where $w_{ij}^m(t)$ is the synaptic input weight i of the neuron j of the layer m in iteration t , and $\partial E(t)$ is the partial derivative of the error.

The training algorithm consists of two phases. Initially, the input data are propagated by the network to obtain its outputs. These values are then compared with the desired ones to obtain the error. In the second step, the opposite path is performed from the output layer to the input layer. In this case, all the synaptic weights are adjusted according to the rule of error correction assumed, so that the output given by the network in the following iteration is closer to the expected one [25].

3.2. Radial Basis Function Network (RBF)

Radial basis function networks (RBF), unlike MLPs, have only two layers, one hidden and one output layer. In the first, all the kernel (activation) functions are radial-based [25]. One of the functions most used is Gaussian function expressed in Equation (9):

$$\varphi(u) = e^{-\frac{(u-c)^2}{2\sigma^2}} \quad (9)$$

in which c is the center of Gaussian and σ^2 its variance as a function of the center.

The training of RBFs is performed in two stages. First, the weights of the intermediate layer are calculated, and the center is adjusted to the value of the base variance of each function. Subsequently, the weights of the output layer are tuned in a supervised process similar of the MLP [25,50].

3.3. Elman and Jordan Networks

The Elman Network is a recursive neural architecture created by Elman [51] based on an MLP. The author divided the input layer into two parts: the first comprises the network inputs and the second, denominated context unit, consists of the outputs of the hidden layer. An Elman network is shown in Figure 1.

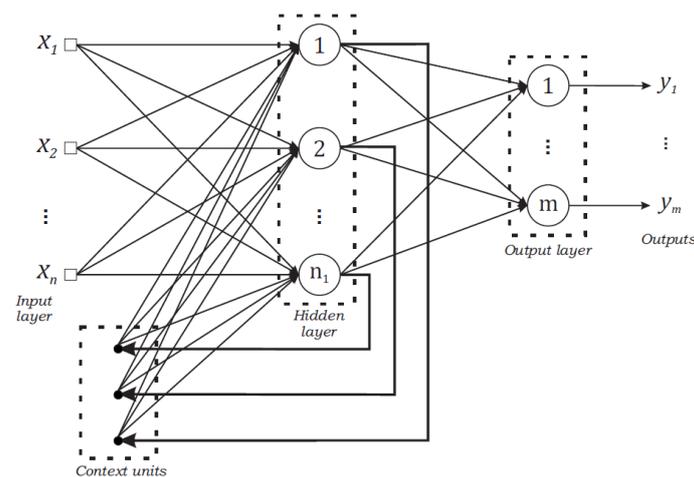


Figure 1. Elman network.

As the context units of an Elman network are treated as inputs, they also have network-associated synaptic weights and can be adjusted by the backpropagation through time algorithm. In this work, we used the truncated backpropagation through time version for one delay [25].

Jordan [52] created the first recurrent neural network based on similar premises. This neural network was initially used for time series recognition but is currently applied to all kinds of problems. Here, the context units are fed by the outputs of the output layer neurons instead of the hidden layer. Figure 2 illustrates this model.

As in the Elman network, the context units are treated as network inputs, also having associated synaptic weights, which allows the use of truncated backpropagation through time [25].

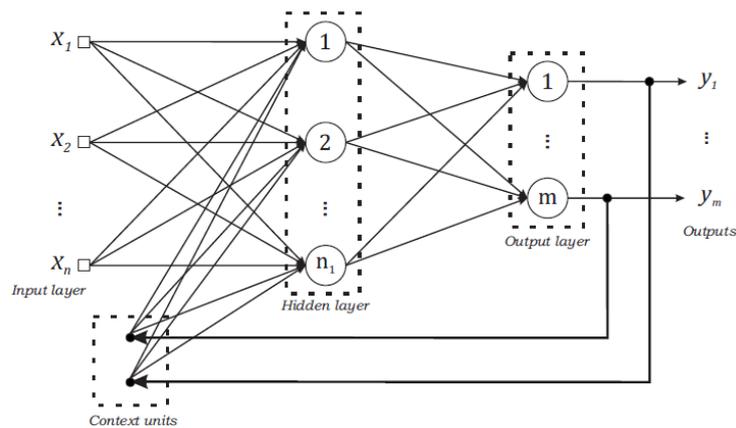


Figure 2. Jordan network.

3.4. Extreme Learning Machines (ELM)

Extreme learning machines (ELM), introduced by Huang et al. [53], are an architecture of feedforward neural network with only a hidden layer. The main difference between them and the traditional MLP is that the synaptic weights of the hidden layer are chosen randomly and remain untuned during the training process. For this reason, the ELM and the ESN are classified as unorganized machines.

Adjusting an ELM consists of determining the matrix with the synaptic weights of the output layer \mathbf{W}^{out} that generates the smallest error for the desired output vector \mathbf{d} , which can be done through an analytic solution. This process is summarized in utilizing the Moore-Penrose pseudo-inverse operator, which ensures the minimum mean square error and confers to the ELM a fast training process. This solution is demonstrated in Equation (10):

$$\mathbf{W}^{\text{out}} = (\mathbf{X}_{\text{hid}}^T \mathbf{X}_{\text{hid}})^{-1} \mathbf{X}_{\text{hid}}^T \mathbf{d} \quad (10)$$

where $\mathbf{X}_{\text{hid}} \in \mathbb{R}^{n \times m}$ is the matrix with all the outputs of the hidden layer for the training set, and m is the number of neurons in the output layer.

This operator ensures that the ELM training is much more computationally efficient than the application of the backpropagation. However, network performance can be improved by inserting a regularization coefficient C , as in Equation (11) [54]:

$$\mathbf{W}^{\text{out}} = \left(\frac{1}{C} + \mathbf{X}_{\text{hid}}^T \mathbf{X}_{\text{hid}} \right)^{-1} \mathbf{X}_{\text{hid}}^T \mathbf{d} \quad (11)$$

where $C = 2^\lambda$, where $\lambda \in \{-25, -24, \dots, 25, 26\}$.

To determine the best value of C , it is necessary to test all 52 possible values [55].

3.5. Echo State Networks (ESN)

Jaeger proposed echo state networks (ESN) as a new type of recurrent neural network. Recursive networks allow different outputs for the same input since it depends on the internal state of the network. The idea of using the term echo is based on the perception that the most recent samples and the previous states influence more strongly the output [56]. The theoretical proof of the existence of an echo state is denominated echo state propriety [9]

In the original proposal from Jaeger, the ESN presents three layers. The hidden layer is denominated dynamic reservoir and consists of fully interconnected neurons, which generate a nonlinear characteristic. The output layer is responsible for combining the outputs of the dynamic reservoir. The subsequent layer, in turn, corresponds to the linear portion of the network. Unlike other

RNNs, which can present feedback in any layer, the original proposal only presents feedback loops in the dynamic reservoir. Figure 3 shows a generic ESN.

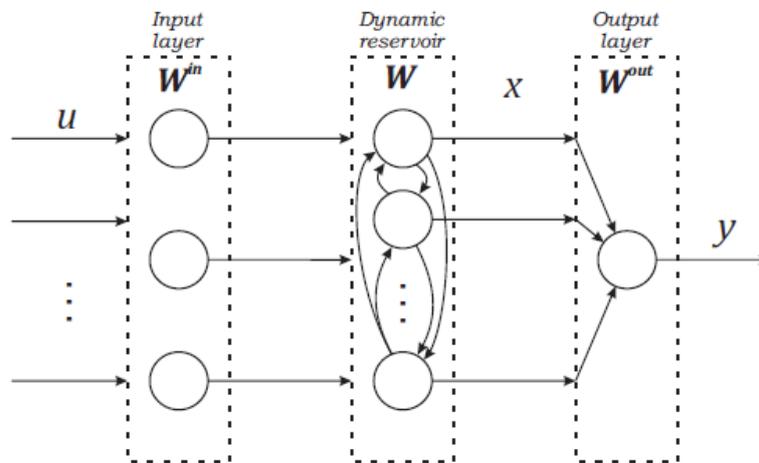


Figure 3. Echo state network (ESN).

For each new input on time $t + 1$, the states of the network are updated according to Equation (12):

$$x_{t+1} = f(W^{in}u_{t+1} + Wx_t) \tag{12}$$

in which x_{t+1} are the states in the input $t + 1$, $f(\cdot)$ represents the activations of reservoir neurons $f(\cdot) = (f_1(\cdot), f_2(\cdot), f_3(\cdot), \dots, f_N(\cdot))$, W^{in} are the coefficients of the input layer, and u_t is the input vector at time t .

In turn, the network output vector y_{t+1} is given by Equation (13):

$$y_{t+1} = W^{out}x_{t+1} \tag{13}$$

where $W^{out} \in \mathbb{R}^{L \times N}$ is the matrix with the synaptic weights of the output layer, and L is the number of network outputs.

As occurs in the ELM, the synaptic weights of the ESN dynamic reservoir are not adjusted during training. The Moore-Penrose pseudo-inverse operator is also used to determine the weights of W^{out} . Additionally, the performance can be improved by using the regularization coefficient.

In this work, we consider two forms of creating the dynamic reservoir. Jaeger et al. [56] proposed the weight matrix by setting three possible values, which are randomly chosen according to the probabilities described in Equation (14):

$$W_{ki}^{in} = \begin{cases} 0.4 & \text{with a probability of 0.025} \\ -0.4 & \text{with a probability of 0.025} \\ 0 & \text{with a probability of 0.95} \end{cases} \tag{14}$$

Ozturk et al. [57] elaborated a reservoir rich in mean entropy of the echo states. The eigenvalues respect a uniform distribution in the unit circle, creating a canonical matrix as presented in Equation (15):

$$W_{ki}^{in} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & -r^N \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \tag{15}$$

in which r is the spectral radius, set in the range $[0, 1]$, and N is the number of neurons present in the dynamics reservoir.

3.6. Ensemble Methodology

An ensemble combines the results of several individually adjusted models with the aim of improving the final response of the system [58]. The idea behind this methodology is that different methods, such as neural networks, produce different behaviors when the same inputs are applied. Therefore, a methodology can present better responses for a given range of data, while another works better in another band. A combination method (average, voting, or another neural network) is applied to produce the final ensemble output [29,30]. Figure 4 presents an example of this model.

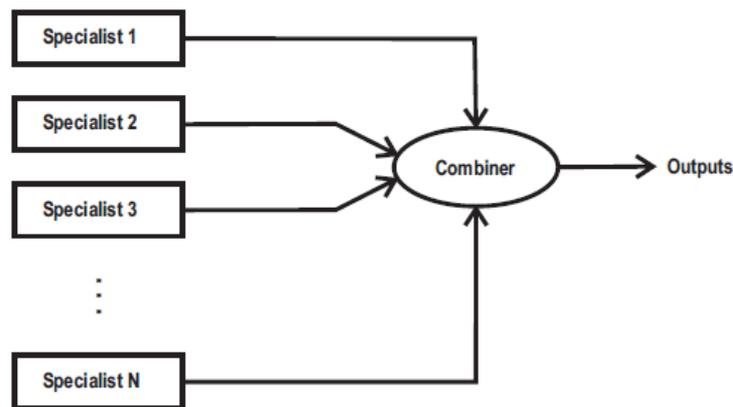


Figure 4. Schematic of an ensemble.

The combination of several specialists through an ensemble does not exclude predictors' need to show good individual performance. The purpose of an ensemble is to improve upon existing good results. Therefore, the essential condition for its accuracy is that its models be accurate and diverse [59]. Over time, ensembles have been used to solve many problems [60–63].

In this work, we used some distinct combiners. First, we addressed some non-trainable methods: the mean and the median of the outputs [29]. Additionally, we applied feedforward neural models: MLP, RBF, and ELM with and without the regularization coefficient [29,30,64]. We highlight that these methodologies are not used often, especially for seasonal streamflow series forecasting.

4. Case Study

Streamflow series are a kind of time series in which each observation refers to monthly, weekly, daily, or hourly average flow. This work addresses the monthly average flow. These series present seasonality since they follow the rain cycles that occur during the year [9]. Seasonality changes the standard behavior of the series, which must be adjusted to improve the response of predictors [5].

Deseasonalization is an operation that removes the seasonal component of the monthly series. They become stationary, presenting zero mean and unitary standard deviation. The new deseasonalized series is given by Equation (16) [50]:

$$z_{i,m} = \frac{x_{i,m} - \hat{\mu}_m}{\hat{\sigma}_m} \quad (16)$$

in which $z_{i,m}$ is the new standardized value of the i element of the series, $x_{i,m}$ is the value of the element i in the original series, $\hat{\mu}_m$ is the average of all elements of the series in the month m , and $\hat{\sigma}_m$ is the standard deviation of all elements of the series in the month m .

This investigation addressed the series of the following Brazilian hydroelectric plants: Agua Vermelha, Belo Monte, Ilha Solteira, Paulo Afonso, and Tucuru. All data were obtained from the National Electric System Operator (ONS) and are available on its website [65]. These plants were

selected because of their location in different regions of the country, and, as shown in Table 1, they have different hydrological behavior. Therefore, it is possible to accomplish a robust performance analysis.

Table 1. Mean and standard deviation of the sets.

Series	Mean	Standard Deviation
Agua Vermelha	2077.33 m ³ /s	1295.71 m ³ /s
Belo Monte	8045.84 m ³ /s	7769.75 m ³ /s
Ilha Solteira	5281.94 m ³ /s	3100.62 m ³ /s
Paulo Afonso	2698.80 m ³ /s	2026.48 m ³ /s
Tucuruí	10,935.23 m ³ /s	9182.29 m ³ /s

All series comprise samples from January of 1931 to December of 2015, a total of 85 years, or 1020 months. The data were divided into three sets: from 1931 to 1995 is the training set, used to adjust the free parameters of the models; from 1996 to 2005 is the validation set, used in the cross-validation process and to adjust the value of the regularization coefficient of ELMs and ESNs; and from 2006 to 2015 is the test set, utilized to measure the performance of the models. The mean squared error (MSE) was adopted as the performance metric, as done in other works of the area [12].

Predictions were made using up to the last six delays of the samples as inputs. These delays were selected using the wrapper method [66,67]. The predictions were performed for 1 (next month), 3 (next season), 6 (next semester), and 12 (next year) steps ahead, using the recursive prediction technique [68,69].

During the preprocessing stage, the series' seasonal component was removed through deseasonalization to make the behavior almost stationary. Therefore, the predicted values required an additional postprocessing step, where the data had the seasonal component reinserted. Figure 5 shows step by step the entire prediction process.

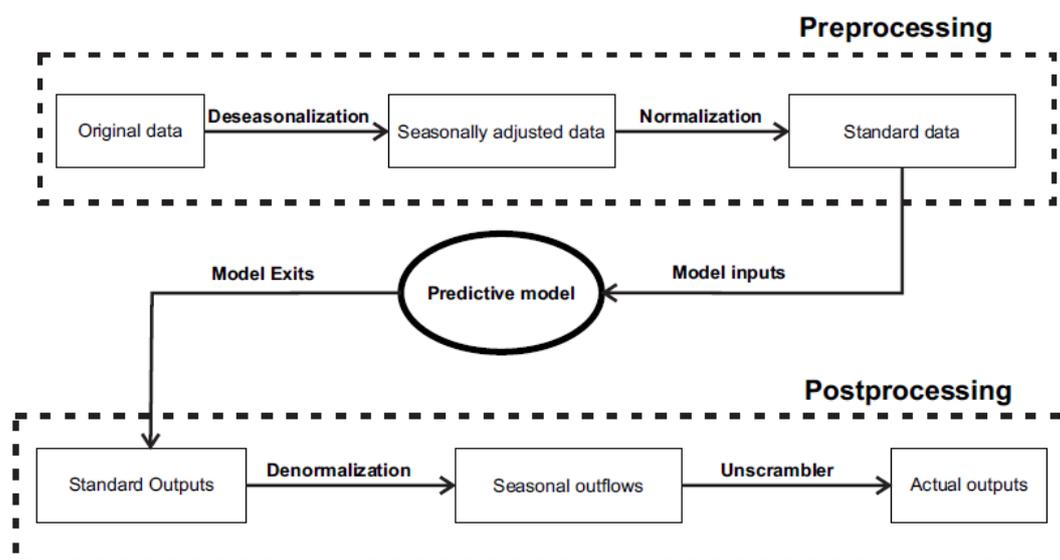


Figure 5. Preprocessing and postprocessing stages.

In total, this work applied 18 predictive methods, two of which were linear methods of the Box-Jenkins family, 10 artificial neural networks, and six ensembles:

- AR(q)—autoregressive model of order q , optimized by the Yule–Walker equations;
- ARMA(p, q)—autoregressive moving-average model of order p, q , optimized by maximum likelihood estimators;
- MLP—MLP network with a hidden layer;

- RBF—RBF network;
- ELM—ELM network;
- ELM (RC)—ELM with the regularization coefficient;
- Elman—Elman network;
- Jordan—Jordan network;
- Jaeger ESN—ESN with reservoir from Jaeger;
- Jaeger ESN (RC)—ESN from Jaeger and regularization coefficient;
- Ozturk ESN—ESN with reservoir from Ozturk et al.;
- Ozturk ESN (RC)—ESN with reservoir from Ozturk et al. and regularization coefficient;
- Average Ensemble—ensemble with mean as arithmetic combiner;
- Median Ensemble—ensemble with median as arithmetic combiner;
- MLP Ensemble—ensemble with MLP as combiner;
- RBF Ensemble—ensemble with an RBF as combiner;
- ELM Ensemble—ensemble with ELM as combiner;
- ELM Ensemble (RC)—ensemble with an ELM with regularization coefficient as combiner.

All neural networks used the hyperbolic tangent as activation function with $\beta = 1$. The learning rate adopted for the models, which use backpropagation, was 0.1. As a stopping criterion, a minimum improvement in MSE of 10^{-6} or a maximum of 2000 epochs was considered. The networks were tested for the number of neurons from 5 to 200, with an increase of 5 neurons. All these parameter's values were determined after empirical tests.

The regularization coefficients were evaluated with all the 52 possibilities mentioned in Section 3.4. The one with the lowest MSE in the validation set was chosen for ELM and ESN approaches. The wrapper method was used to select the best lags to the single models as well as which experts were the best predictors when using the ensembles. The holdout cross-validation was also applied to all fully neural networks and ensembles to avoid overtraining and to determine the C of ELM and ESN.

All proposed models were executed 30 times to obtain sample output data for each input configuration and number of neurons, these having been chosen as the best executions. Additionally, following the methodology addressed in [9] and [70], we adjusted 12 independent models, one for each month, for all proposed methods. It is allowed since the mean and the variance of each month are distinct, and this approach can lead to better results [9].

Experimental Results

In this section, we present and discuss the results obtained by all models and four forecasting horizons. Tables 2–6 show the results achieved for Agua Vermelha, Belo Monte, Ilha Solteira, Paulo Afonso, and Tucuruí hydroelectric plants, for both real and deseasonalized domains. The best performances are highlighted in bold.

The Friedman [71,72] test was applied to the results for the 30 runs of each predictive model proposed, regarding the MSE in the test set. The p -values achieved were smaller than 0.05. Therefore, we can assume that changing the predictor leads to a significant change in the results.

To analyze the dispersion of the results obtained after 30 executions [73], Figure 6 presents the boxplot graphic [72,74].

Table 2. Computational results for Água Vermelha.

Model	P = 1		P = 3	
	MSE	MSE (d)	MSE	MSE (d)
AR(3)	468,513.81	0.4290	635,623.14	0.6901
ARMA(2,2)	459,868.32	0.4161	652,368.14	0.6562
MLP	435,047.55	0.4321	592,399.82	0.6701
RBF	433,015.37	0.4903	616,761.67	0.7871
ELM	413,488.42	0.3893	548,179.42	0.5522
ELM (RC)	393,462.78	0.3901	548,521.40	0.5862
Elman	391,192.04	0.4013	590,235.98	0.6513
Jordan	417,081.01	0.4181	642,432.88	0.7443
Jaeger ESN	383,174.66	0.3760	576,896.70	0.5940
Jaeger ESN (RC)	379,043.86	0.3781	565,927.30	0.5891
Ozturk ESN	401,843.39	0.3772	745,343.33	0.9742
Ozturk ESN (RC)	397,982.11	0.3772	833,991.16	1.1010
Average Ensemble	374,741.57	0.3683	543,746.08	0.5590
Median Ensemble	379,043.86	0.3782	574,877.03	0.5901
MLP Ensemble	381,381.74	0.3951	595,528.64	0.6240
RBF Ensemble	378,694.99	0.3910	582,641.85	0.6120
ELM Ensemble	360,574.55	0.3771	598,571.40	0.6311
ELM Ensemble (RC)	362,264.36	0.3870	596,396.64	0.6390

Model	P = 6		P = 12	
	MSE	MSE (d)	MSE	MSE (d)
AR(3)	920,536.06	0.9560	1,032,390.38	1.3101
ARMA(2,2)	857,987.67	0.8951	844,191.84	1.0820
MLP	705,060.11	0.8360	717,730.32	0.8534
RBF	1,003,756.28	1.3010	2,139,100.52	3.1500
ELM	602,358.84	0.6664	695,473.23	0.8123
ELM (RC)	643,385.78	0.7177	735,096.12	0.8840
Elman	719,246.60	0.8161	733,779.79	0.8746
Jordan	726,929.58	0.8792	711,201.57	0.8455
Jaeger ESN	650,640.11	0.7129	679,957.56	0.8021
Jaeger ESN (RC)	632,523.34	0.6990	690,611.82	0.8196
Ozturk ESN	671,232.69	0.8152	726,758.25	0.8643
Ozturk ESN (RC)	695,143.95	0.8861	749,393.17	0.8869
Average Ensemble	718,287.32	0.7420	736,530.62	0.9079
Median Ensemble	806,882.94	0.8172	761,767.76	0.9891
MLP Ensemble	796,168.68	0.8097	779,261.85	0.9861
RBF Ensemble	774,132.08	0.7849	738,982.18	0.9638
ELM Ensemble	834,303.17	0.8658	800,998.37	1.0501
ELM Ensemble (RC)	837,861.29	1.0021	1,252,464.38	1.7002

Table 3. Computational results for Belo Monte.

Model	P = 1		P = 3	
	MSE	MSE (d)	MSE	MSE (d)
AR(3)	5,136,070.02	0.3681	9,481,744.90	0.7818
ARMA(4,3)	5,560,980.02	0.3978	10,338,028.24	0.7803
MLP	4,611,515.04	0.3956	6,610,976.00	0.7674
RBF	4,189,519.18	0.3976	7,374,107.90	0.7756
ELM	3,932,745.86	0.4455	5,211,473.53	0.7751
ELM (RC)	4,263,133.64	0.3620	5,977,477.68	0.6716
Elman	4,348,164.47	0.4310	12,461,369.17	1.6201
Jordan	4,841,287.43	0.3795	8,029,604.93	0.7721
Jaeger ESN	3,990,917.22	0.3619	5,744,660.75	0.6405
Jaeger ESN (RC)	4,063,963.41	0.3703	6,268,470.74	0.6559
Ozturk ESN	4,135,861.20	0.3986	6,121,782.43	0.6100
Ozturk ESN (RC)	4,054,930.75	0.3769	5,576,790.17	0.6437
Average Ensemble	3,664,436.73	0.3738	6,583,650.15	0.6255
Median Ensemble	3,654,331.87	0.3379	7,892,943.44	0.6883
MLP Ensemble	3,683,236.20	0.3495	9,631,438.35	0.7448
RBF Ensemble	3,716,658.52	0.4082	6,756,906.30	0.7051
ELM Ensemble	3,496,325.52	0.3583	7,360,295.02	0.6748
ELM Ensemble (RC)	3,537,238.59	0.3522	7,038,931.44	0.6529

Model	P = 6		P = 12	
	MSE	MSE (d)	MSE	MSE (d)
AR(3)	13,220,246.31	1.0701	13,856,487.04	1.3102
ARMA(4,3)	14,598,946.90	1.0701	10,811,736.13	1.3103
MLP	8,503,685.65	0.9158	8,834,763.12	0.8732
RBF	9,560,433.48	0.9534	10,802,574.68	1.1402
ELM	5,390,642.11	0.8129	6,258,443.28	0.7798
ELM (RC)	6,266,406.91	0.7252	6,209,801.53	0.7287
Elman	6,837,675.11	1.0401	7,177,995.01	0.9005
Jordan	10,176,393.17	0.9748	10,698,937.70	1.0302
Jaeger ESN	5,778,132.12	0.7207	5,938,616.01	0.7261
Jaeger ESN (RC)	5,681,748.34	0.7446	5,942,372.26	0.7407
Ozturk ESN	5,906,402.73	0.6860	6,391,806.49	0.7184
Ozturk ESN (RC)	5,764,898.14	0.7325	5,911,871.83	0.7356
Average Ensemble	10,217,687.60	0.8305	9,363,944.04	1.1201
Median Ensemble	12,782,750.96	0.9963	11,326,906.60	1.3901
MLP Ensemble	14,838,424.95	1.1600	12,489,587.18	1.6103
RBF Ensemble	11,651,888.44	1.0301	12,834,385.61	1.2302
ELM Ensemble	11,834,559.77	0.9613	10,523,981.04	1.3001
ELM Ensemble (RC)	11,409,184.74	0.9323	10,465,521.50	1.2700

Table 4. Computational results for Ilha Solteira.

Model	$P = 1$		$P = 3$	
	MSE	MSE (d)	MSE	MSE (d)
AR(6)	2,966,223.64	0.5451	3,607,452.30	0.8363
ARMA(2,2)	2,820,456.34	0.5311	3,575,263.58	0.8337
MLP	2,673,739.64	0.5366	3,385,476.99	0.7607
RBF	2,646,716.44	0.5745	3,910,685.33	0.9068
ELM	2,545,002.11	0.4973	3,316,801.94	0.7265
ELM (RC)	2,498,945.94	0.5240	3,445,130.56	0.7750
Elman	5,689,735.15	0.5713	3,719,407.93	0.8549
Jordan	2,722,458.65	0.5524	3,557,423.14	0.8219
Jaeger ESN	2,511,512.57	0.4914	3,259,442.70	0.7219
Jaeger ESN (RC)	2,568,883.38	0.4999	3,294,093.73	0.7489
Ozturk ESN	2,337,398.44	0.4663	3,428,320.79	0.7850
Ozturk ESN (RC)	2,313,534.09	0.4768	3,646,528.58	0.8801
Average Ensemble	2,277,545.05	0.4593	3,316,183.09	0.7599
Median Ensemble	2,313,534.09	0.4768	3,230,939.25	0.7472
MLP Ensemble	2,312,233.01	0.4945	3,196,000.87	0.7135
RBF Ensemble	2,006,994.79	0.4903	3,468,793.60	0.8388
ELM Ensemble	2,084,515.90	0.4546	3,377,696.82	0.7866
ELM Ensemble (RC)	1,992,611.39	0.4491	3,265,153.09	0.7606

Model	$P = 6$		$P = 12$	
	MSE	MSE (d)	MSE	MSE (d)
AR(6)	4,511,751.90	1.1100	4,773,379.18	1.1900
ARMA(2,2)	4,473,867.42	1.1201	4,494,073.05	1.1301
MLP	3,920,672.52	0.9025	4,049,775.43	0.9433
RBF	4,307,512.83	1.0100	4,283,245.08	1.1101
ELM	3,761,082.94	0.8480	3,873,269.31	0.8967
ELM (RC)	3,823,661.64	0.8934	3,815,701.76	0.8860
Elman	4,154,039.41	0.9698	4,116,367.32	0.9615
Jordan	4,169,869.49	0.9662	4,070,205.28	0.9373
Jaeger ESN	3,838,690.48	0.8814	3,477,087.73	0.8497
Jaeger ESN (RC)	3,884,498.96	0.8727	3,969,786.68	0.9293
Ozturk ESN	3,938,797.73	0.9629	3,712,781.93	0.8854
Ozturk ESN (RC)	3,864,185.96	0.9479	4,670,725.87	0.9975
Average Ensemble	4,188,451.09	1.0503	4,920,660.69	1.1702
Median Ensemble	4,131,637.37	1.0601	4,970,254.98	1.1602
MLP Ensemble	3,982,036.00	0.9874	4,739,091.94	1.0910
RBF Ensemble	3,934,481.31	1.0403	6,385,242.27	1.4903
ELM Ensemble	4,419,019.18	1.1702	6,998,311.41	2.4601
ELM Ensemble (RC)	4,586,375.57	1.3101	6,612,087.53	1.4502

Table 5. Computational results for Paulo Afonso.

Model	P = 1		P = 3	
	MSE	MSE (d)	MSE	MSE (d)
AR(6)	726,145.70	0.2931	1,129,419.58	0.5697
ARMA(2,1)	694,185.05	0.2874	1,131,738.27	0.5453
MLP	635,673.56	0.3437	1,058,546.46	0.6355
RBF	719,377.51	0.4103	2,000,897.18	1.1801
ELM	602,176.48	0.3312	938,475.79	0.5768
ELM (RC)	607,078.51	0.3237	1,089,089.42	0.6389
Elman	617,664.77	0.2869	936,344.63	0.4566
Jordan	673,001.75	0.3398	1,166,257.86	0.6914
Jaeger ESN	566,929.67	0.2786	929,239.63	0.5230
Jaeger ESN (RC)	627,573.85	0.3415	1,490,572.96	0.9615
Ozturk ESN	576,338.26	0.2957	1,256,147.76	0.6778
Ozturk ESN (RC)	581,609.27	0.3095	1,047,548.67	0.5800
Average Ensemble	551,531.61	0.2908	1,230,760.56	0.5601
Median Ensemble	552,252.81	0.2521	1,169,891.73	0.5154
MLP Ensemble	548,535.13	0.2931	1,185,108.36	0.5330
RBF Ensemble	554,508.82	0.2986	987,644.32	0.4806
ELM Ensemble	535,210.83	0.2714	1,088,286.19	0.4940
ELM Ensemble (RC)	535,865.26	0.3016	1,193,241.92	0.5780
Model	P = 6		P = 12	
	MSE	MSE (d)	MSE	MSE (d)
AR(6)	1,088,288.61	0.7460	1,916,895.36	1.1101
ARMA(2,1)	1,024,438.22	0.6806	1,634,060.46	0.9137
MLP	1,113,602.56	0.6855	1,118,376.67	0.6888
RBF	3,779,964.35	2.6001	13,040,123.34	8.1102
ELM	980,510.07	0.6357	1,290,254.85	0.7766
ELM (RC)	1,129,832.90	0.7220	1,723,308.91	1.1100
Elman	981,800.01	0.5880	1,214,455.57	0.7626
Jordan	1,209,656.52	0.7708	1,258,279.09	0.7960
Jaeger ESN	1,106,672.85	0.6847	1,364,591.04	0.8514
Jaeger ESN (RC)	1,278,004.66	0.8767	1,852,101.70	1.1703
Ozturk ESN	1,040,841.92	0.6180	1,250,520.94	0.7806
Ozturk ESN (RC)	1,083,715.18	0.7617	1,348,961.53	0.8507
Average Ensemble	1,025,453.52	0.6182	1,420,331.33	0.7761
Median Ensemble	1,078,563.81	0.5853	1,334,761.67	0.7067
MLP Ensemble	1,063,431.44	0.5914	1,311,740.31	0.7058
RBF Ensemble	1,058,247.06	0.6367	1,393,776.65	0.7927
ELM Ensemble	989,046.79	0.5391	1,267,630.78	0.6619
ELM Ensemble (RC)	997,954.04	0.5994	1,346,719.34	0.7240

Table 6. Computational results for Tucuruí.

Model	P = 1		P = 3	
	MSE	MSE (d)	MSE	MSE (d)
AR(3)	7,446,258.52	0.3310	13,406,528.21	0.9131
ARMA(4,1)	7,991,096.15	0.3383	13,974,802.61	0.9714
MLP	7,100,370.96	0.3451	11,346,681.97	0.7834
RBF	6,721,796.96	0.3592	11,966,687.62	0.7750
ELM	6,824,285.76	0.3030	10,906,829.60	0.7399
ELM (RC)	6,829,099.84	0.3066	11,134,575.77	0.7359
Elman	7,351,576.91	0.4107	13,167,672.49	0.8019
Jordan	6,980,914.40	0.3251	11,752,842.37	0.8432
Jaeger ESN	6,936,607.01	0.3024	11,639,311.67	0.6543
Jaeger ESN (RC)	6,975,358.52	0.3035	13,382,939.20	0.7647
Ozturk ESN	6,757,516.84	0.3084	17,883,748.20	0.8628
Ozturk ESN (RC)	7,119,853.31	0.3310	11,180,015.34	0.6629
Average Ensemble	6,254,104.50	0.3089	12,708,717.47	0.7938
Median Ensemble	6,757,516.84	0.3084	14,733,468.98	0.8444
MLP Ensemble	6,317,164.23	0.3121	12,726,659.05	0.8268
RBF Ensemble	5,776,403.72	0.3120	14,878,993.97	0.8811
ELM Ensemble	5,923,084.05	0.2792	15,279,206.30	1.0902
ELM Ensemble (RC)	5,859,606.76	0.2810	14,830,513.86	1.0101
Model	P = 6		P = 12	
	MSE	MSE (d)	MSE	MSE (d)
AR(3)	15,727,475.82	1.2001	22,161,343.50	1.4602
ARMA(4,1)	16,315,642.78	1.3700	29,987,913.13	1.9100
MLP	11,658,727.90	0.8131	11,743,205.59	0.8254
RBF	12,696,569.79	0.9830	15,087,729.25	0.9688
ELM	11,989,453.79	0.8238	12,304,667.66	0.8764
ELM (RC)	12,143,842.92	0.8236	12,284,951.17	0.8614
Elman	14,163,188.22	1.1002	10,533,366.45	0.6894
Jordan	12,264,013.86	0.8860	12,341,774.82	0.8899
Jaeger ESN	12,040,332.15	0.7548	11,766,645.65	0.8119
Jaeger ESN (RC)	12,071,588.19	0.7345	12,279,058.45	0.8578
Ozturk ESN	16,299,760.49	0.8256	11,238,408.86	0.7952
Ozturk ESN (RC)	11,770,003.63	0.7062	10,807,257.91	0.6965
Average Ensemble	16,469,919.61	0.9983	19,601,009.99	1.0703
Median Ensemble	20,303,247.25	1.1200	20,367,666.36	1.1801
MLP Ensemble	15,554,587.87	0.9710	21,292,183.31	1.0702
RBF Ensemble	18,606,965.66	1.0201	22,318,965.50	1.2100
ELM Ensemble	21,731,212.23	1.2802	31,221,147.23	1.5203
ELM Ensemble (RC)	20,723,489.57	1.2601	34,497,724.89	1.5801

Note that AR, ARMA, average ensemble, and median ensemble models did not show dispersion since they presented close-form solution [75]. One can verify the highest dispersion was obtained by RBF, while the lowest was achieved by Elman network.

Many aspects of the general results presented in Tables 2–6 can be discussed. The first is that the best predictor in the real space sometimes was not the same in the deseasonalized domain. This occurred because the deseasonalization process considered all parts of the series as having the same importance. In the literature, the error in the real space is adopted as the most important measure to evaluate the results [12,18].

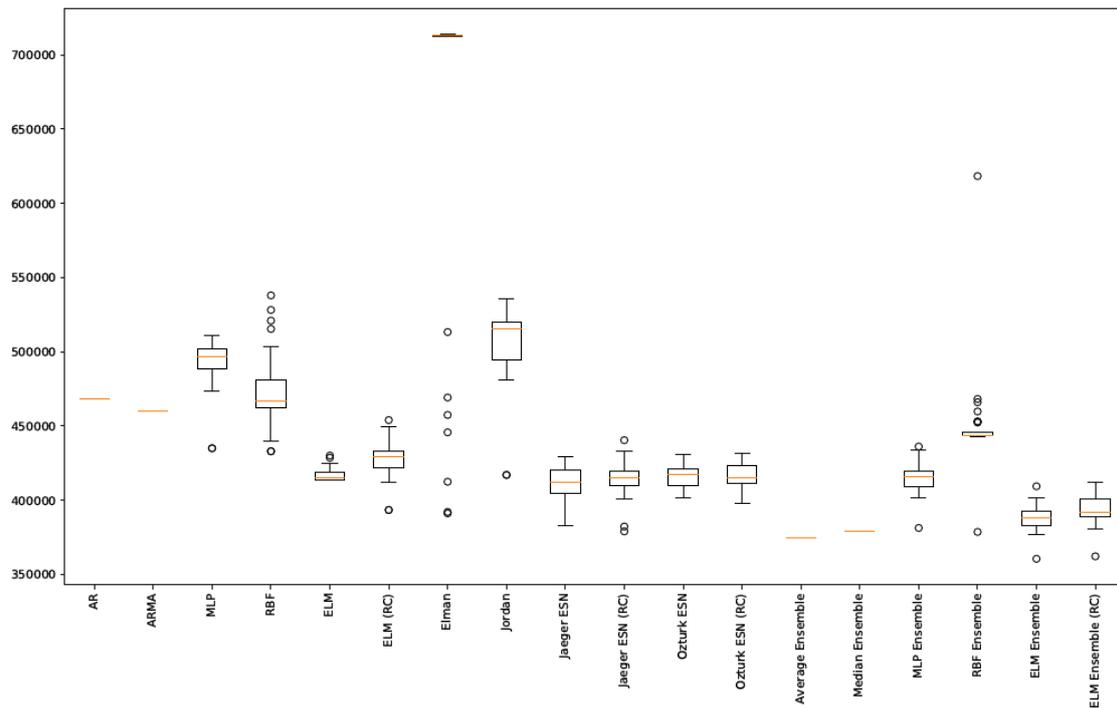


Figure 6. Boxplot Água Vermelha, 2006 at 2015–30 executions.

As the forecasting horizon grows, the prediction process becomes more difficult, and the errors tend to increase for all models. It is directly related to the decreased correlation between the input samples and the desired future response. Therefore, the output estimates tend to achieve the long term mean, or the historical mean [9].

We elaborated Table 7 using the results from Tables 2–6 to show a winner ranking to illustrate which models achieved the best general performance.

Table 7. Best model by horizon.

Model	P = 1	P = 3	P = 6	P = 12	Total	
AR(3)	-	-	-	-	0	0
ARMA(4,1)	-	-	-	-	0	
MLP	-	-	1	1	2	2
RBF	-	-	-	-	0	
ELM	-	2	4	-	6	6
ELM (RC)	-	-	-	-	0	
Elman	-	-	-	-	1	1
Jordan	-	-	-	-	0	
Jaeger ESN	-	1	-	2	3	4
Jaeger ESN (RC)	-	-	-	-	0	
Ozturk ESN	-	-	-	-	0	
Ozturk ESN (RC)	-	-	-	1	1	
Average Ensemble	-	1	-	-	1	7
Median Ensemble	-	-	-	-	0	
MLP Ensemble	-	1	-	-	1	
RBF Ensemble	1	-	-	-	1	
ELM Ensemble	3	-	-	-	3	
ELM Ensemble (RC)	1	-	-	-	1	

For $P = 1$ step ahead, for all series, the best predictor was always an ensemble, highlighting the ELM-based combiner, which was the best for three of five scenarios (60%). This result indicates that the use of ensembles can lead to an increase in the performance. Moreover, the application of an unorganized machine requires less computational effort than the MLP or the RBF since its training process is based on a deterministic linear approach. It also corroborates that their approximation capability is elevated, overcoming their fully trained counterpart [9,12].

The results varied for $P = 3$, for which several architectures—ELM, Jaeger ESN, average ensemble, and MLP ensemble—were the best at least once. We emphasize that the ELM network was better in two cases, and the ESN in one. The UMs were better in 60% of the cases. Regarding $P = 6$, the ELM was also the best predictor, achieving the smallest error in four cases (80%), followed by the MLP, which was the best only for Tucuruí.

Analyzing the last forecasting horizon, $P = 12$, four different neural architectures reached the best performance at least once: MLP, Elman, Jaeger ESN (twice), and Ozturk ESN (RC). An important observation is the presence of recurrent models among them. This horizon is very difficult to predict since the correlation between the input samples and the desired response is small. Therefore, there is an indication that the existence of model's internal memory is an advantage.

In summary, the unorganized networks (ESN and ELM), in stand-alone versions or as a combiner of an ensemble, provided the best results in 14 of 20 scenarios (70%). This is relevant since such methods are newer than the others and are simpler to implement.

Considering the reservoir design of ESNs, we achieved almost a draw; in nine cases, the proposal from Ozturk et al. was the best, and in 11, the Jaeger model achieved the smallest error. Therefore, we cannot state which one is the most adequate for the problem.

Regarding the feedforward neural models, one can observe for 16 of 20 cases (80%), the ELMs overcame the traditional MLP and RBF architectures. In the same way, the ESNs were superior to the traditional and the fully trained Elman and Jordan proposals in 17 of 20 scenarios (85%). This is strong evidence that the unorganized models are prominent candidates to carry out such problems.

Linear models did not outperform neural networks in any of the 20 scenarios. For the problem of forecasting monthly seasonal streamflow series, the results showed that ANNs were most appropriate. However, it is worth mentioning that linear models are still widely used in current days.

Finally, to provide a visual appreciation of the final simulation, Figure 7 presents the forecast made by the ELM ensemble for Água Vermelha plant with $P = 1$.

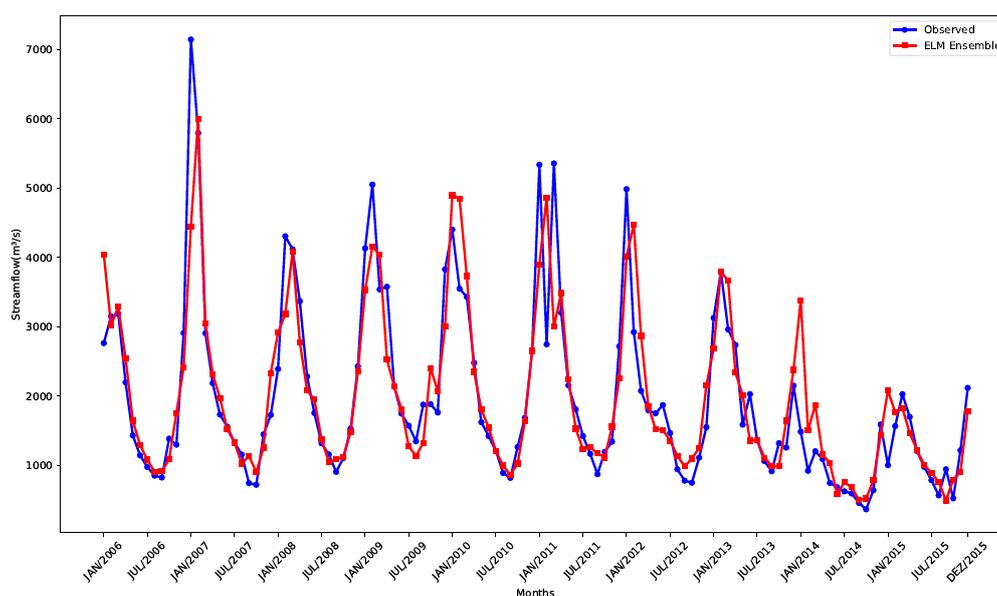


Figure 7. Best predicted to Água Vermelha of 2006 at 2015 and $P = 1$.

5. Conclusions

This work investigated the performance of unorganized machines—extreme learning machines (ELM), echo state networks (ESN), and ELM-based ensembles—on monthly seasonal streamflow series forecasting from hydroelectric plants. This is a very important task for countries where power generation is highly dependent on water as a source, such as Canada, China, Brazil, and the USA, among others. Due to the broad use of this kind of energy generation in the world, even a small improvement in the accuracy of the predictions can lead to significant financial resource savings as well as a reduction in the impact of using fossil fuels.

We also used many artificial neural network (ANN) architectures—multilayer perceptron (MLP), radial basis function networks (RBF), Jordan network, Elman network, and the ensemble methodology using the mean, the median, the MLP, and the RBF as combiner. Moreover, we compared the results with the traditional AR and ARMA linear models. We addressed four forecast horizons, $P = 1, 3, 6$, and 12 steps ahead, and the wrapper method to select the best delays (inputs).

The case study involves a database related to five hydroelectric plants. The tests showed that the neural ensembles were the most indicated for $P = 1$ since they presented the best performances in all the simulations of this scenario, especially those that employed the ELM. For $P = 3$ and 6, the ELM was highlighted. For $P = 12$, it was clear that the recurrent models were outstanding, mainly those with the ESN.

Regarding the linear models, this work showed its inferiority in comparison to the neural ones in all cases. Furthermore, the unorganized neural models (ELM and ESN), in their stand-alone versions or as combiners of the ensemble, prevailed over the others, presenting 14 of the lowest errors (70%).

These results are important since the unorganized machines are easy to implement and require less computational effort than the fully trained approaches. This is related to the use of the Moore–Penrose inverse operator to train their output layer, since it ensures the optimum value of the weights in the mean square error sense. The use of the backpropagation could lead the process to a local minimum, indicating how difficult the problem is. In at least 80% of the cases, the unorganized proposals (ELM and ESN) overcame the fully trained proposals (MLP, RBF, Elman, and Jordan).

Other deseasonalization processes should be investigated in future works. Additionally, the streamflow from distinct plants must be predicted and the results evaluated. Moreover, the use of bio-inspired optimization methods [76–78] is encouraged to optimize the ARMA model and the application of the support vector regression method.

Author Contributions: Conceptualization, H.S., S.L.S.J., J.B., L.A.; methodology, H.S., S.L.S.J., J.B., L.A., software, H.S., J.B.; validation, H.S., P.S.G.d.M.N., J.F.L.d.O.; formal analysis, P.S.G.d.M.N., M.H.N.M., J.F.L.d.O.; investigation, J.B., L.A.; resources: M.H.N.M.; writing original draft preparation, H.S., J.B., L.A.S. and I.C.; writing, review and editing, M.d.A.L.F., F.U., L.A.S., A.C.; visualization, P.S.G.d.M.N., J.F.L.d.O.; supervision, A.C. and M.H.N.M. project administration, M.d.A.L.F., M.H.N.M.; funding acquisition, L.A.S., M.d.A.L.F. and M.H.N.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work received funding and technical support from AES and Associated Companies (of the CPFL, Brookfield and Global group) as part of the ANEEL PD-0610-1004/2015 project, “IRIS—Integration of Intermittent Renewables: A simulation model of the operation of the Brazilian electrical system to support planning, operation, commercialization, and regulation”, which is part of an R&D program regulated by ANEEL, Brazil. The authors also thank the Advanced Institute of Technology and Innovation (IATI) for its support, the National Institute of Meteorology (INMET) for providing the data, and the Coordination for the Improvement of Higher Education Personnel—Brazil (CAPES)—Financing Code 001, for support.

Acknowledgments: This work was partially supported by the Brazilian agencies CAPES and FACEPE. The authors thank the Brazilian National Council for Scientific and Technological Development (CNPq), process number 40558/2018-5, and Araucaria Foundation, process number 51497, for their financial support. The authors also thank the Federal University of Technology.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Marczas, G.; Uniejewski, B.; Weron, R. Beating the Naïve—Combining LASSO with Naïve Intraday Electricity Price Forecasts. *Energies* **2020**, *13*, 1667. [[CrossRef](#)]
2. Sigauke, C.; Nemukula, M.M.; Maposa, D. Probabilistic hourly load forecasting using additive quantile regression models. *Energies* **2018**, *11*, 2208. [[CrossRef](#)]
3. Malfatti, M.G.L.; Cardoso, A.O.; Hamburger, D.S. Linear Empirical Model for Streamflow Forecast in Itaipu Hydroelectric Dam—Parana River Basin. *Rev. Bras. Meteorol.* **2018**, *33*, 257–268. [[CrossRef](#)]
4. Moscoso-López, J.A.; Turias, I.; Jiménez-Come, M.J.; Ruiz-Aguilar, J.J.; del Cerbán, M.M. A two-stage forecasting approach for short-term intermodal freight prediction. *Int. Trans. Oper. Res.* **2019**, *26*, 642–666.
5. Francelin, R.; Ballini, R.; Andrade, M.G. Back-propagation and Box & Jenkins approaches to streamflow forecasting. *Simpósio Bras. Pesqui. Oper.-SBPO Lat.-Iber.-Am. Congr. Oper. Res. Syst. Eng.-CLAIO* **1996**, *3*, 1307–1312.
6. Sacchi, R.; Ozturk, M.C.; Principe, J.C.; Carneiro, A.A.F.M.; Silva, I.N. Water inflow forecasting using the echo state network: A Brazilian case study. In Proceedings of the 2007 International Joint Conference on Neural Networks, Orlando, FL, USA, 12–17 August 2007; pp. 2403–2408.
7. Yaseen, Z.; Sulaiman, S.; Deo, R.; Chau, K.W. An enhanced extreme learning machine model for river flow forecasting: State-of-the-art, practical applications in water resource engineering area and future research direction. *J. Hydrol.* **2018**, *569*. [[CrossRef](#)]
8. International Hydropower Association. *Hydropower Status Report*; International Hydropower Association: London, UK, 2019.
9. Siqueira, H.; Boccato, L.; Luna, I.; Attux, R.; Lyra, C. Performance analysis of unorganized machines in streamflow forecasting of Brazilian plants. *Appl. Soft Comput.* **2018**, *68*, 494–506. [[CrossRef](#)]
10. Wu, C.; Chau, K.W. Data-driven models for monthly streamflow time series prediction. *Eng. Appl. Artif. Intell.* **2010**, *23*, 1350–1367. [[CrossRef](#)]
11. Kisi, O.; Cimen, M. Precipitation forecasting by using wavelet-support vector machine conjunction model. *Eng. Appl. Artif. Intell.* **2012**, *25*, 783–792. [[CrossRef](#)]
12. Siqueira, H.; Boccato, L.; Attux, R.; Lyra, C. Unorganized machines for seasonal streamflow series forecasting. *Int. J. Neural Syst.* **2014**, *24*, 1430009. [[CrossRef](#)]
13. Taormina, R.; Chau, K.W. ANN-based interval forecasting of streamflow discharges using the LUBE method and MOFIPS. *Eng. Appl. Artif. Intell.* **2015**, *45*, 429–440. [[CrossRef](#)]
14. Zhu, S.; Zhou, J.; Ye, L.; Meng, C. Streamflow estimation by support vector machine coupled with different methods of time series decomposition in the upper reaches of Yangtze River, China. *Environ. Earth Sci.* **2016**, *75*, 531. [[CrossRef](#)]
15. Arsenault, R.; Côté, P. Analysis of the effects of biases in ensemble streamflow prediction (ESP) forecasts on electricity production in hydropower reservoir management. *Hydrol. Earth Syst. Sci.* **2019**, *23*, 2735–2750. [[CrossRef](#)]
16. Stojković, M.; Kostić, S.; Prohaska, S.; Plavšić, J.; Tripković, V. A New Approach for Trend Assessment of Annual Streamflows: A Case Study of Hydropower Plants in Serbia. *Water Resour. Manag.* **2017**, *31*, 1089–1103. [[CrossRef](#)]
17. Hailegeorgis, T.T.; Alfredsen, K. Regional statistical and precipitation–runoff modelling for ecological applications: Prediction of hourly streamflow in regulated rivers and ungauged basins. *River Res. Appl.* **2017**, *33*, 233–248. [[CrossRef](#)]
18. Siqueira, H.V.; Boccato, L.; Attux, R.R.F.; Lyra Filho, C. Echo State Networks in Seasonal Streamflow Series Prediction. *Learn. Nonlinear Models* **2012**, *10*, 181–191. [[CrossRef](#)]
19. Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*; John Wiley & Sons: New York, NY, USA, 2015.
20. Hwang, W.Y.; Lee, J.S. A new forecasting scheme for evaluating long-term prediction performances in supply chain management. *Int. Trans. Oper. Res.* **2014**, *21*, 1045–1060. [[CrossRef](#)]
21. Singh, P.; Deo, M. Suitability of different neural networks in daily flow forecasting. *Appl. Soft Comput.* **2007**, *7*, 968–978. [[CrossRef](#)]
22. Jain, A.; Kumar, A.M. Hybrid neural network models for hydrologic time series forecasting. *Appl. Soft Comput.* **2007**, *7*, 585–592. [[CrossRef](#)]

23. Toro, C.H.F.; Meire, S.G.; Gálvez, J.F.; Fdez-Riverola, F. A hybrid artificial intelligence model for river flow forecasting. *Appl. Soft Comput.* **2013**, *13*, 3449–3458. [[CrossRef](#)]
24. Kazemi, S.; Seied Hoseini, M.M.; Abbasian-Naghneh, S.; Rahmati, S.H.A. An evolutionary-based adaptive neuro-fuzzy inference system for intelligent short-term load forecasting. *Int. Trans. Oper. Res.* **2014**, *21*, 311–326. [[CrossRef](#)]
25. Haykin, S.S. *Neural Networks and Learning Machines*; Prentice Hall: New York, NY, USA, 2009.
26. Singh, P.; Borah, B. An efficient time series forecasting model based on fuzzy time series. *Eng. Appl. Artif. Intell.* **2013**, *26*, 2443–2457. [[CrossRef](#)]
27. Parsa, N.; Keshavarz, T.; Karimi, B.; Moattar Hussein, S.M. A hybrid neural network approach to minimize total completion time on a single batch processing machine. *Int. Trans. Oper. Res.* **2020**. [[CrossRef](#)]
28. Sharkey, A.J.C. *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*; Springer: London, UK, 1999.
29. De Mattos Neto, P.S.; Madeiro, F.; Ferreira, T.A.; Cavalcanti, G.D. Hybrid intelligent system for air quality forecasting using phase adjustment. *Eng. Appl. Artif. Intell.* **2014**, *32*, 185–191. [[CrossRef](#)]
30. Firmino, P.R.A.; de Mattos Neto, P.S.; Ferreira, T.A. Correcting and combining time series forecasters. *Neural Netw.* **2014**, *50*, 1–11. [[CrossRef](#)]
31. Kasiviswanathan, K.S.; Sudheer, K.P. Quantification of the predictive uncertainty of artificial neural network based river flow forecast models. *Stoch. Environ. Res. Risk Assess.* **2013**, *27*, 137–146. [[CrossRef](#)]
32. Fan, F.M.; Schwanenberg, D.; Alvarado, R.; Reis, A.A.; Collischonn, W.; Naumann, S. Performance of Deterministic and Probabilistic Hydrological Forecasts for the Short-Term Optimization of a Tropical Hydropower Reservoir. *Water Resour. Manag.* **2016**, *30*, 3609–3625. [[CrossRef](#)]
33. Thober, S.; Kumar, R.; Wanders, N.; Marx, A.; Pan, M.; Rakovec, O.; Samaniego, L.; Sheffield, J.; Wood, E.F.; Zink, M. Multi-model ensemble projections of European river floods and high flows at 1.5, 2, and 3 degrees global warming. *Environ. Res. Lett.* **2018**, *13*, 014003. [[CrossRef](#)]
34. EPE—Energy Research Company. *National Energy Balance 2019*; Ministry of Mines and Energy: Rio de Janeiro, Brazil, 2019.
35. Siqueira, H.V.; Boccato, L.; Attux, R.R.F.; Lyra Filho, C. Echo state networks and extreme learning machines: A comparative study on seasonal streamflow series prediction. *Lect. Notes Comput. Sci.* **2012**, *7664*, 491–500.
36. Haykin, S.O. *Adaptive Filter Theory*; Pearson Higher: London, UK, 2013.
37. Cao, Q.; Ewing, B.T.; Thompson, M.A. Forecasting wind speed with recurrent neural networks. *Eur. J. Oper. Res.* **2012**, *221*, 148–154. [[CrossRef](#)]
38. Barrow, D.; Kourentzes, N. The impact of special days in call arrivals forecasting: A neural network approach to modelling special days. *Eur. J. Oper. Res.* **2018**, *264*, 967–977. [[CrossRef](#)]
39. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst. (MCCS)* **1989**, *2*, 303–314. [[CrossRef](#)]
40. Polezer, G.; Tadano, Y.S.; Siqueira, H.V.; Godoi, A.F.; Yamamoto, C.I.; de André, P.A.; Pauliquevis, T.; de Fatima Andrade, M.; Oliveira, A.; Saldiva, P.H.; et al. Assessing the impact of PM2.5 on respiratory disease using artificial neural networks. *Environ. Pollut.* **2018**, *235*, 394–403. [[CrossRef](#)] [[PubMed](#)]
41. Reifman, J.; Feldman, E.E. Multilayer perceptron for nonlinear programming. *Comput. Oper. Res.* **2002**, *29*, 1237–1250. [[CrossRef](#)]
42. Feng, S.; Li, L.; Cen, L.; Huang, J. Using MLP networks to design a production scheduling system. *Comput. Oper. Res.* **2003**, *30*, 821–832. [[CrossRef](#)]
43. Dahl, G.E.; Stokes, J.W.; Deng, L.; Yu, D. Large-scale malware classification using random projections and neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 3422–3426.
44. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc.: South Lake Tahoe, NV, USA, 2012; pp. 1097–1105.
45. Kristjanpoller, W.; Minutolo, M.C. Gold price volatility: A forecasting approach using the Artificial Neural Network–GARCH model. *Expert Syst. Appl.* **2015**, *42*, 7245–7251. [[CrossRef](#)]
46. Zhang, G.P.; Qi, M. Neural network forecasting for seasonal and trend time series. *Eur. J. Oper. Res.* **2005**, *160*, 501–514. [[CrossRef](#)]

47. Rendon-Sanchez, J.F.; de Menezes, L.M. Structural combination of seasonal exponential smoothing forecasts applied to load forecasting. *Eur. J. Oper. Res.* **2019**, *275*, 916–924. [[CrossRef](#)]
48. Werbos, P.J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. Thesis, Harvard University, Cambridge, MA, USA, 1974.
49. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Cogn. Model.* **1986**, *5*, 1. [[CrossRef](#)]
50. Siqueira, H.; Luna, I. Performance comparison of feedforward neural networks applied to stream flow series forecasting. *J. MESA* **2019**, *10*, 41–53.
51. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [[CrossRef](#)]
52. Jordan, M. Attractor Dynamics and Parallelism in a Connectionist Sequential Machine. In Proceedings of the Eighth Annual Conference of the Cognitive Science Society, Amherst, MA, USA, 15–17 August 1986; Erlbaum: Hillsdale, NJ, USA, 1986; pp. 531–546.
53. Huang, G.H.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 25–29 July 2004; Volume 2, pp. 985–990.
54. Huang, G.H.; Zhou, H.; Ding, X.; Zhang, R. Extreme learning machine for regression and multiclass classification. *Trans. Syst. Man Cybern.-Part B Cybern.* **2012**, *42*, 513–529. [[CrossRef](#)] [[PubMed](#)]
55. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [[CrossRef](#)]
56. Jaeger, H. The “echo state” approach to analysing and training recurrent neural networks—with an erratum note. *Ger. Natl. Res. Cent. Inf. Technol. GMD Tech. Rep.* **2001**, *148*, 13.
57. Ozturk, M.C.; Xu, D.; Príncipe, J.C. Analysis and Design of Echo State Networks for Function Approximation. *Neural Comput.* **2007**, *19*, 111–138. [[CrossRef](#)]
58. Wichard, J.D.; Ogorzalek, M. Time series prediction with ensemble models. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 25–29 July 2004; IEEE: Piscataway, NJ, USA, 2004; Volume 2, pp. 1625–1630.
59. Hansen, L.K.; Salamon, P. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 993–1001. [[CrossRef](#)]
60. Yu, L.; Wang, S.; Lai, K.K. A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates. *Comput. Oper. Res.* **2005**, *32*, 2523–2541. [[CrossRef](#)]
61. West, D.; Dellana, S.; Qian, J. Neural network ensemble strategies for financial decision applications. *Comput. Oper. Res.* **2005**, *32*, 2543–2559. [[CrossRef](#)]
62. Yang, D. Spatial prediction using kriging ensemble. *Sol. Energy* **2018**, *171*, 977–982. [[CrossRef](#)]
63. Kim, D.; Hur, J. Short-term probabilistic forecasting of wind energy resources using the enhanced ensemble method. *Energy* **2018**, *157*, 211–226. [[CrossRef](#)]
64. Kumar, N.K.; Savitha, R.; Al Mamun, A. Ocean wave height prediction using ensemble of Extreme Learning Machine. *Neurocomputing* **2018**, *277*, 12–20. [[CrossRef](#)]
65. ONS—Electric System Operator. Database of “Hydrological Data/Streamflows”. 2018. Available online: <http://ons.org.br/Paginas/resultados-da-operacao/historico-da-operacao/dados-hidrologicos-vazoes.aspx> (accessed on 1 February 2018).
66. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
67. Wang, H.; Sun, J.; Sun, J.; Wang, J. Using random forests to select optimal input variables for short-term wind speed forecasting models. *Energies* **2017**, *10*, 1522. [[CrossRef](#)]
68. Dong, W.; Yang, Q.; Fang, X. Multi-step ahead wind power generation prediction based on hybrid machine learning techniques. *Energies* **2018**, *11*, 1975. [[CrossRef](#)]
69. Sorjamaa, A.; Hao, J.; Reyhani, N.; Ji, Y.; Lendasse, A. Methodology for long-term prediction of time series. *Neurocomputing* **2007**, *70*, 2861–2869. [[CrossRef](#)]
70. Stedinger, J.R. *Report on the Evaluation of Cepel's Par Models: Technical Report*; School of Civil and Environmental Engineering-Cornell University: Ithaca, NY, USA, 2001.
71. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [[CrossRef](#)]

72. Araujo, N.A.; Belotti, J.T.; Antonini Alves, T.; Tadano, Y.S.; Siqueira, H. Ensemble method based on Artificial Neural Networks to estimate air pollution health risks. *Environ. Model. Softw.* **2020**, *123*, 104567. [[CrossRef](#)]
73. Siqueira, H.V.; Boccato, L.; Attux, R.R.F.; Lyra Filho, C. Echo state networks for seasonal streamflow series forecasting. *Lect. Notes Comput. Sci.* **2012**, *7435*, 226–236.
74. Kachba, Y.; Chirolí, D.M.; Belotti, J.T.; Alves, T.A.; de Souza Tadano, Y.; Siqueira, H. Artificial neural networks to estimate the influence of vehicular emission variables on morbidity and mortality in the largest metropolis in South America. *Sustainability* **2020**, *12*, 2621. [[CrossRef](#)]
75. Siqueira, H.; Luna, I.; Antonini Alves, T.; de Souza Tadano, Y. The direct connection between Box & Jenkins methodology and adaptive filtering theory. *Math. Eng. Sci. Aerosp. (MESA)* **2019**, *10*, 27–40.
76. Santos, P.; Macedo, M.; Figueiredo, E.; Santana, C.S., Jr.; Soares, F.; Siqueira, H.; Maciel, A.; Gokhale, A.; Bastos-Filho, C.J.A. Application of PSO-based clustering algorithms on educational databases. In Proceedings of the 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Arequipa, Peru, 8–10 November 2017; pp. 1–6.
77. Puchta, E.; Lucas, R.; Ferreira, F.R.V.; Siqueira, H.V.; Kaster, M.S. Gaussian adaptive PID control optimized via genetic algorithm applied to a step-down DC-DC converter. In Proceedings of the 2016 12th IEEE International Conference on Industry Applications (INDUSCON), Curitiba, Brazil, 20–23 November 2016; pp. 1–6.
78. Santana, C.J., Jr.; Macedo, M.; Siqueira, H.; Gokhale, A.; Bastos-Filho, C.J.A. A novel binary artificial bee colony algorithm. *Future Gener. Comp. Syst.* **2019**, *98*, 180–196. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).