

Article

# Energy Consumption Forecasting for Smart Meters Using Extreme Learning Machine Ensemble

Paulo S. G. de Mattos Neto <sup>1,\*</sup>, João F. L. de Oliveira <sup>2,t</sup>, Priscilla Bassetto <sup>3,t</sup>, Hugo Valadares Siqueira <sup>3,t</sup>, Luciano Barbosa <sup>1,t</sup>, Emilly Pereira Alves <sup>2,4,t</sup>, Manoel H. N. Marinho <sup>2,t</sup> and Guilherme Ferretti Rissi <sup>5,t</sup> and Fu Li <sup>5,t</sup>

<sup>1</sup> Centro de Informática, Universidade Federal de Pernambuco, Recife 50740-560, Brazil; luciano@cin.ufpe.br

<sup>2</sup> Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife 50720-001, Brazil; fausto.lorenzato@upe.br (J.F.L.d.O.); epa@poli.br (E.P.A.); marinho75@poli.br (M.H.N.M.)

<sup>3</sup> Graduate Program in Industrial Engineering, Federal University of Technology—Paraná, Ponta Grossa 84017-220, Brazil; pri\_bass@hotmail.com (P.B.); hugosiqueira@utfpr.edu.br (H.V.S.)

<sup>4</sup> Advanced Institute of Technology and Innovation (IATI), Recife 50751-310, Brazil

<sup>5</sup> CPFL Energia, Campinas, São Paulo 13087-397, Brazil; grissi@cpfl.com.br (G.F.R.); lifu@cpfl.com.br (F.L.)

\* Correspondence: psgmn@cin.ufpe.br

† These authors contributed equally to this work.



**Citation:** de Mattos Neto, P.S.G.; de Oliveira, J.F.L.; Bassetto, P.; Siqueira, H.V.; Barbosa, L.; Alves, E.P.; Marinho, M.H.N.; Rissi, G.F.; Li, F. Energy Consumption Forecasting for Smart Meters Using Extreme Learning Machine Ensemble. *Sensors* **2021**, *21*, 8096. <https://doi.org/10.3390/s21238096>

Academic Editors: Kyandoghere Kyamakya, Fadi Al-Machot, Ahmad Haj Mosa, Jean Chamberlain Chedjou, Zhong Li and Antoine Bagula

Received: 27 October 2021

Accepted: 1 December 2021

Published: 3 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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 (<https://creativecommons.org/licenses/by/4.0/>).

**Abstract:** The employment of smart meters for energy consumption monitoring is essential for planning and management of power generation systems. In this context, forecasting energy consumption is a valuable asset for decision making, since it can improve the predictability of forthcoming demand to energy providers. In this work, we propose a data-driven ensemble that combines five single well-known models in the forecasting literature: a statistical linear autoregressive model and four artificial neural networks: (radial basis function, multilayer perceptron, extreme learning machines, and echo state networks). The proposed ensemble employs extreme learning machines as the combination model due to its simplicity, learning speed, and greater ability of generalization in comparison to other artificial neural networks. The experiments were conducted on real consumption data collected from a smart meter in a one-step-ahead forecasting scenario. The results using five different performance metrics demonstrate that our solution outperforms other statistical, machine learning, and ensembles models proposed in the literature.

**Keywords:** energy consumption; smart metering; forecasting; Box and Jenkins models; neural networks; ensembles

## 1. Introduction

The interest in energy consumption in residential buildings has increased over the past years due to advances in home technology, economic technologies, and population growth [1]. Consumption profiles contribute with the elevated consumption since time indoors has increased due to the possibility of a home office [1]. Moreover, residential energy constitutes over 27% of global energy consumption [2,3] and over 40% of the consumption in United States and European Union [4].

Considering the amount of energy required in residential buildings, the employment of smart meters has become an important feature for planning and management of power generation systems [5]. Smart meters not only enable occupants to have insights of their own consumption patterns, but also provide useful information to energy suppliers in order to perform better planning of energy load. In this scenario, energy forecasting is considered an important tool for planning and decision making processes [6]. Its main challenge, however, is the high volatility of data concerning individual households. Consumption data can present different patterns since it can be influenced by external factors such as consumer profiles, weather, and the season of the year [7]. Moreover, the choice of an

appropriate model can also affect the quality of forecasts. In fact, according to [2], only a 1% reduction in forecasting errors can have positive impacts in the economy.

Traditional linear forecasting models such as the autoregressive integrated moving average (ARIMA) and exponential smoothing (ETS) have been explored in the context of energy forecasting in smart meters [8,9]. ARIMA models assume a linear correlation structure among past data. As a result, it presents reduced accuracy when dealing with nonlinear data. In contrast, nonlinear models such as artificial neural networks (ANNs) and support vector machines (SVMs) can deal with nonlinear patterns in time series, but it may not deal with all patterns equally well due to problems of model misspecification, overfitting, and underfitting [10].

In the light of the limitations of linear and nonlinear models when individually employed and the high volatility characteristics presented in smart meter data, hybrid systems have been proposed in order to overcome such limitations and produce more accurate forecasts [2,4]. In particular, ensemble models take into consideration a pool of forecasting models, where the forecasts are combined in order to improve the forecasting quality. However, in order to achieve an improvement in performance, the pool of forecasting models must be accurate, uncorrelated, and diverse [11]. The intuition is that the strengths of a model may compensate the weaknesses of another, mitigating the risk of selecting a single unsuitable model.

There are two important steps in building ensemble systems: model generation and forecast combination. An ensemble can be composed of multiple models of the same method (homogeneous ensemble) or by different methods (heterogeneous ensemble). However, different diversity generation approaches can be employed to improve the accuracy of the ensemble. Diversity generation methods, such as bagging, perform random sampling bootstraps of the original training data in order to train each model. They are often employed in homogeneous ensembles [12]. In heterogeneous ensembles, the diversity is achieved through the employment of different forecasting methods. The combination of forecasts can be trainable or non trainable. Nontrainable combination models comprehend several statistical operators such as mean, median, and mode. Combination through the median and mode is less sensible to the presence of outliers than by the mean [13].

Trainable combinations can explore the flexibility of nonlinear models such as ANNs and SVRs in such a way that a meta-data is created based on the predictions produced by the pool of forecasters. In this way, the combination based on nonlinear trainable methods is trained on training data, allowing the combinator to generalize to unseen data, performing combinations of future forecasts of the base models. This strategy maps nonlinear relations between forecasts, but also brings an overhead of parameters, which might increase the computational complexity of the overall hybrid system.

Taking into consideration the volatility in energy consumption data from smart meters, the limitations of the forecasting models proposed in the literature and the computational complexity required in the training phase of nonlinear trainable combinations, this work proposes a heterogeneous ensemble composed of a pool of models with different characteristics, combined using an extreme learning machine (ELM) [14] model. We use ELM in our solution since it presents less computational complexity and fewer configurable parameters than traditional machine learning methods such as ANNs and SVRs [14].

More specifically, the proposed method presents the following advantages:

- The diversity of the ensemble is introduced by the employment of different forecasting methods such as autoregressive (AR), multilayer perceptron (MLP), extreme learning machine (ELM), radial basis function (RBF), and echo state network (ESN).
- The combination step employs an ELM model in order to map nonlinear relations between forecasts and to perform more accurate combinations.
- The proposed method is versatile, since different forecasting methods can be used in the pool, and then combined by the ELM.

Our solution is employed in the context of smart metering data and compared with traditional models and different forecasting combination methods proposed in the liter-

ature. The experiments are conducted taking into consideration different consumption patterns present in the data. The results demonstrated that the proposed method achieved better results than single models and other ensembles with different combinations.

The remainder of this article is structured as follows: Section 2 presents an overview of the related works, Section 3 details the proposed ensemble method and its components. The experimental setup and results are discussed in Section 4 and the Conclusion is presented in Section 5.

## 2. Related Work

The development of systems based on ML models has been highlighted in the energy forecasting area [15]. In this area, electricity load and energy consumption forecasts have received great attention due to their relationship to demand, supply, and environmental issues [16,17]. In general, electricity load forecasting tasks have a major impact on the planning, operating, and monitoring power systems. The accuracy of the forecasts can impact operation costs since an overestimation can increase the number of generators employed and produce an unnecessary reserve of electricity. The underestimation of electricity load can put at risk the system's reliability due to insufficient load required to attend the demanding market [18]. In the same way, electricity consumption forecasting models can improve energy efficiency and sustainability in diverse sectors such as in residential buildings [19–21] and in industry [22,23].

In order to achieve accurate electricity load forecasts several machine learning (ML) models have been employed in this task [24–26]. Models such as ANNs based on wavelets [24], long short-term memory (LSTM), random forests [25], and ensembles [26] have been investigated.

Likewise, energy consumption forecasting systems based on ML models have been used in the literature. Culaba et al. [19] employed a hybrid system based on clustering and forecasting using *K*-Means and SVR models, respectively. Deep learning models such as convolution neural networks (CNN) were employed by [20] for energy consumption forecasts in the context of new buildings with few historical data. Pinto et al. [21] used ensemble models to forecast energy consumption in office buildings. Walther and Weigold [22] performed a systematic review of the literature on energy consumption forecasting models in the industry.

Considering the literature of energy consumption forecasting on smart metering data, several ML methods have been investigated. In this context, Gajowniczek and Zabkowski [27] employed MLP and SVR models to forecast the consumption on individual smart meters. For that, their solution extracts features related to the meter's consumption history (e.g., average, maximum and minimum load) and the temperature inside the house. They argued that they do not perform a traditional time series modeling due to the high volatility of their data.

Zhukov et al. [28] investigated the effects of concept drift in smart grid analysis. A random forecast algorithm for concept drift was employed, and an ensemble using the weighted majority vote rule was used to combine the outputs of individual learners. The proposed method was compared to other algorithms in the concept drift detection context, obtaining promising results.

Electricity pricing and load forecasting are important tasks in smart grid structures due to the improvements of efficiency in the management of electric systems [17,29,30]. In this scenario, Heydari et al. [29] proposed a hybrid system based on variational mode decomposition (VMD), gravitational search algorithm (GSA), and general regression neural networks (GRNN). The VMD performs the series's decomposition into several intrinsic mode functions (IMFS), while the GSA performs a feature selection in the time series. Furthermore, considering the importance of electricity load forecasting in electric systems, this task can also be performed in individual households through the employment of smart metering technologies [31,32]. In this way, Li et al. [33] employed a convolutional long short-term memory-based neural network with selected autoregressive features to improve

forecasting accuracy. Fekri et al. [32] used deep learning models based on online adaptive recurrent neural networks, considering that energy consumption patterns may change over time. In addition, several load forecasting applications have been addressed, such as peak alert systems [34], where a modified support vector regression is employed, using smart meter data and weather data as input.

Another work that deals with smart metering forecast [7], investigated the effects of factors such as seasonality and weather condition for electricity consumption prediction using different ML models: regression trees, MLP and SVR. Their findings show that: regression trees obtain the lowest root mean squared error (RMSE) values in almost all evaluated scenarios; adding weather data does not improve the results; and a historical window of one year to train the models is enough to achieve low-error forecasts.

Sajjad et al. [35] propose a deep-learning model for hourly energy consumption forecast of appliances and houses. The input data is processed using min-max normalization or z-score standardization, which is fed into a convolutional neural network (CNN) followed by a recurrent neural network (RNN), specifically a gated recurrent unit (GRU). Finally, a dense layer on top of the GRU outputs the prediction. They do not provide, however, any details about their strategy of selecting the hyper-parameters of the network.

Similarly, Wang et al. [36] employ an long short-term memory (LSTM) model that outputs quantile probabilistic forecasts. For training, the network minimizes the average quantile loss for all quantiles. The input of the network is composed of the historical consumption, the day of the week and hour of the day of the data point to be predicted. Similar to [35], the process of selection of nodes and layers of the network is not presented.

In addition, hybrid systems have gained attention due to their ability to increase the accuracy of the single ML models [16,37]. These systems are developed aiming to overcome the limitations of single ML models regarding misspecification, overfitting, and underfitting [10]. In this sense, Somu et al. [38] employed the K-means clustering-based convolutional neural networks and long short term memory (KCNN-LSTM) to forecast energy consumption using data from smart meters. In this work, the K-means is employed to identify tendency and seasonal patterns in the time series, while the CNN-LSTM is used in the forecasting process.

Chou and Truong [39] proposed a hybrid system composed of four steps: linear time series modeling, nonlinear residual modeling, combination, and optimization. The parameter selection process for the models employed in the first three steps is performed through a Jellyfish Search (JS) optimization algorithm [40]. Bouktif et al. [41] employed a genetic algorithm (GA) and particle swarm optimization (PSO) to search for hyperparameters of the LSTM in load forecasting tasks.

The proposed hybrid system differs from the hybrid systems proposed in the literature since it employs a GA to perform the optimization of the residual forecasting model and the combination model. Furthermore, the optimization also selects the most relevant lags to reduce model complexity and enhance forecasting accuracy.

### 3. Proposed Ensemble Method

Ensembles are elaborated in order to improve the final response from of the single trained models (specialists) combining their outputs [42]. The idea is that each single model presents a better performance for some subset of the input data. Hence, a combination model can use each best single output to generate a more accurate final response [43,44]. Figure 1 summarizes the general idea of the proposed ensemble, presenting its two main steps: training and test.

In the training step, the single and combination models are adjusted to improve some performance measure. First, the single models ( $M_1, M_2, \dots, M_5$ ) are trained from training instances ( $X_{tr}$ ) that contain the time lags of the time series and the respective desired output. Then, the combination models are trained to fuse the single models' forecasts in order to minimize the difference between the desired output and the ensemble forecast according

to the performance measure. Each combination model receives a data set that combines the forecasts of the training pattern of the single models for each desired output of  $X_{tr}$ .

In the test step, given an unseen test pattern  $X_q$ , every single model generates one forecast  $\hat{X}_{q+1}^M$ . This set of forecasts is then passed to the combination model, which generates the final forecast  $\hat{X}_{q+1}$ .

In our solution, the pool of single models employed by the proposed ensemble method comprises the statistical linear AR and four well-known ML models: MLP, ELM, RBF, and ESN [45]. The AR model assumes a linear correlation structure in the data; therefore it can not perform nonlinear mappings. MLP, ELM, RBF, and ESN are flexible, data-driven, and able to perform nonlinear mappings. The MLP employs a multilayered architecture in order to learn from data, while the ELM uses a single hidden layer. The ESN has feedback loops of information in the hidden layer. The RBF is based on the locality learning principle since Gaussian functions are often employed as activation functions in the hidden units. Thus, these models represent different architectures in the energy consumption literature and were chosen due to promising results in time series forecasting tasks, especially those related to electricity [45–49].

In the proposal, the combination of the forecasts is performed using an ELM model. Therefore, since the combination is performed by a trainable method, the data set used for its training process is composed of the predictions of the pool ( $X_{tr}^{M1}, X_{tr}^{M2}, X_{tr}^{M3}, X_{tr}^{M4}, X_{tr}^{M5}$ ) and the target output value. After the training process is complete, the ELM model performs the combination of the forecasts achieved by the pool in the test set ( $\hat{X}_{q+1}^{M1}, \hat{X}_{q+1}^{M2}, \hat{X}_{q+1}^{M3}, \hat{X}_{q+1}^{M4}, \hat{X}_{q+1}^{M5}$ ) to generate the final forecast of the system  $\hat{X}_{q+1}$ .

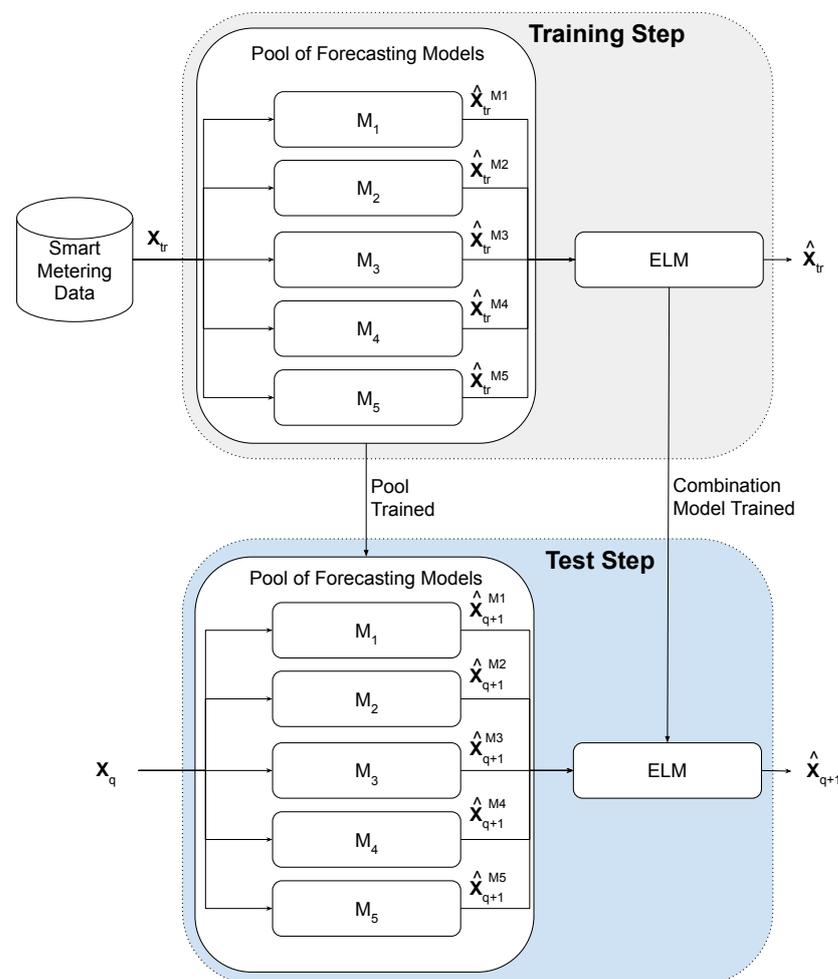


Figure 1. Model of the proposed ensemble.

Considering that ML models may present problems such as model misspecification, overfitting, and underfitting [10], ensemble methods can further increase the accuracy of the base models through of combination approaches. The errors could be decomposed to assure an adequate performance of the method, it is a crucial condition that the single models also present accurate performance and diversity in the response [50–52].

Let  $\bar{\mathbf{f}}$  be the ensemble and  $\mathbf{X}_{q+1}$  be the target output, the overall forecasting error of the ensemble can be decomposed as presented in Equation (1), where the first term of the right hand side of the equation represents the difference between the forecast and the target output, which is often referred to as bias, whereas the latter term represents the stability of the model in unseen samples, and is often referred to as variance.

$$E\{(\bar{\mathbf{f}}(\mathbf{X}) - \mathbf{X}_{q+1})^2\} = (E\{\bar{\mathbf{f}}(\mathbf{X})\} - \mathbf{X}_{q+1})^2 + E\{(\bar{\mathbf{f}}(\mathbf{X}) - E\{\bar{\mathbf{f}}(\mathbf{X})\})^2\}. \quad (1)$$

The employment of ensemble methods through averaging forecasts allows a decrease of the variance without increasing the bias term, therefore increasing the generalization capacity of the forecasting system [53]. The literature presents many different combiners, such as average, weighted voting, and using ML methods [50–52].

The following sections describe in further details the single and combination models employed in the proposed ensemble method.

### 3.1. Single Model: Autoregressive Model

The autoregressive Model belongs to the framework of the linear Box and Jenkins methodology. It is one of the most widely known approaches due to its good results presented in the literature and simple parameters' adjustment, with is based on a closed form solution [54].

Let  $x_t$  be a sample of a time series, and  $x_{t-p}$  its  $p$ -th lag (delay). We define the autoregressive model of order  $p$  (AR( $p$ )) as the weighted combination of  $p$  lags of observation  $x_t$ , as in Equation (2) [54]:

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 showed in Equation (2):

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

where  $a_t$  are white Gaussian noises (shocks) or the inherent error of the prediction,  $\tilde{x}_t = x_t - \mu$  ( $\mu$  is the average of the series),  $\phi_p$  is the weighting coefficient for the lag  $p$ .

The solution of the Yule–Walker equations is given in matrix form by Equation (3):

$$\Phi_p = \mathbf{P}_p^{-1} \rho_p \quad (3)$$

in which we can expand the elements as in Equation (4):

$$\mathbf{P}_p = \begin{bmatrix} 1 & \rho_1 & \dots & \rho_{p-1} \\ \rho_1 & 1 & \dots & \rho_p \\ \dots & \dots & \dots & \dots \\ \rho_{p-1} & \rho_{p-2} & \dots & 1 \end{bmatrix} \rho_p = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_p \end{bmatrix} \Phi_p = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix} \quad (4)$$

where  $\rho_p$  is the coefficients of the autocorrelation function of the series.

### 3.2. Single Model: Multilayer Perceptron (MLP)

Undoubtedly, the multilayer perceptron (MLP) is the most used artificial neural network architecture for nonlinear mapping due to its versatility and applicability [43]. Endowed of a set of artificial neurons organized in at least three multiple layers (input, hidden, and output layers), the MLP is a feedforward neural model.

The training of an MLP consists of tune the weights of the neuron to provide an adequate mapping between the inputs and the desired response [55,56]. The literature presents many methods to adjust an MLP, usually based on unconstrained nonlinear optimization. The most known method is the steepest decent algorithm in which the gradient vector is calculated using the backpropagation algorithm [57,58]. However, in this work we address the modified scaled conjugated gradient, a second order method [47].

The steps involved in the training of the MLP consists of two main iterative phases. The first one is a forward progression, in which the inputs data is propagated in the neural model until the achievement of the outputs. Next, the instant error regarding such outputs is calculated, using the desired output response. In the second step, the weights are adjusted from the output layer to the input layer, following the assumed optimization rule. In this sense, the error provided by the MLP in the next iteration is smaller. The kind of adjustment that uses a desired signal lies in the framework of the supervised training [59].

### 3.3. Single Model: Echo State Networks (ESN)

Designed in 2001, the echo state networks (ESN) are a kind of recurrent neural network since they have feedback loops of information in the hidden layer. This intrinsic characteristic may bring gains in the performance of the neural models when the inputs present temporal dependence, as in time series forecasting [60,61].

Recurrent models present different response depending on their internal state. The convergence proof of the ESN shows that the most recent inputs and the previous states influences more the output response of the network. Hence, Jaeger [60] used the term *echo* to describe the *echo state propriety* [62], which demonstrates the conditions for the network to present echo states.

Similar to MLP, the original ESN presents three layers. The hidden layer is named dynamic reservoir. This layer presents fully interconnected neurons, which are responsible to generate the nonlinear characteristic. The output layer combines the responses of the reservoir. Only the reservoir presents feedback loops in the original proposal. For each new input  $\mathbf{u}_{t+1}$  the internal states of the ESN are updated following Equation (5):

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{W}^{\text{in}}\mathbf{u}_{t+1} + \mathbf{W}\mathbf{x}_t) \quad (5)$$

where  $\mathbf{x}_{t+1}$  are the states in time  $t + 1$ ,  $\mathbf{f}(\cdot) = (f_1(\cdot), f_2(\cdot), f_3(\cdot), \dots, f_N(\cdot))$  the activations of reservoir neurons, and  $\mathbf{W}^{\text{in}}$  the weights of the input layer.

The output response  $\mathbf{y}_{t+1}$  is according to Equation (6):

$$\mathbf{y}_{t+1} = \mathbf{W}^{\text{out}}\mathbf{x}_{t+1} \quad (6)$$

in which  $\mathbf{W}^{\text{out}} \in \mathbb{R}^{L \times N}$  is the matrix containing all weights of the output layer, and  $L$  is the number outputs.

The weights of an ESN in the dynamic reservoir are not adjusted in the training phase. The Moore–Penrose pseudo-inverse operator (Equation (9)) is used to determine the coefficients of  $\mathbf{W}^{\text{out}}$ .

To create the dynamic reservoir, we use the original idea from Jaeger [60]. In this case, the weight matrix is composed of 3 possible values, which are randomly chosen according to the following probabilities:

$$W_{ki}^{\text{in}} = \begin{cases} 0.4 & \text{with a probability of } 0.025 \\ -0.4 & \text{with a probability of } 0.025 \\ 0 & \text{with a probability of } 0.95 \end{cases} \quad (7)$$

### 3.4. Single Model: Radial Basis Function Network (RBF)

The radial basis function networks (RBF) is another classic ANN architecture. It has only one hidden and one output layer. In the hidden layer, all kernel functions (activation)

are radial-based, and the Gaussian function is the most used [59]. This function is in Equation (8):

$$\varphi(u) = e^{-\frac{(u-c)^2}{2\sigma^2}} \quad (8)$$

where  $c$  the center of Gaussian, and  $\sigma^2$  the corresponding variance which is a function of the center position.

To adjust an RBF, it is necessary to follow two steps. In the first stage the synaptic weights of the intermediate layer are determined, and the center is adjusted to the value of the base variance of each function. This stage is adjusted by an unsupervised clustering approach [59]. In this work, we used the K-Medoids method. The second step involves the calculation of the weights of the output layer [63]. There are some possible approaches to perform this task. The most usual is the use of the backpropagation algorithm, as in MLP. Another possibility is the direct application of the Moore–Penrose pseudo-inverse operator (Equation (9)) [59], which is the one we chose in this work.

### 3.5. Single Model: Extreme Learning Machine (ELM)

Extreme learning machine (ELM) is a feedforward neural network architecture similar to the traditional MLP. The main difference lies in the training procedure, while we adjust all neural weights in the MLP, in the ELM, just the output layer is trained using a supervised approach. In addition, the ELM presents just one hidden layer [14].

In this sense, the neurons in the hidden layer are randomly generated and stands untuned. The training of an ELM is summarized in finding the weights of the output layer  $\mathbf{W}^{\text{out}}$  that lead the networks response to the smallest error regarding the desired output  $\mathbf{d}$  [14]. The usual way to solve this task is use a closed form solution, the Moore–Penrose pseudo-inverse operator. Besides the small computational cost involved in its application, the operator ensures minimum mean square error between the network response and the desired output. This solution is in Equation (9):

$$\mathbf{W}^{\text{out}} = (\mathbf{X}_{\text{hid}}^T \mathbf{X}_{\text{hid}})^{-1} \mathbf{X}_{\text{hid}}^T \mathbf{d} \quad (9)$$

in which  $\mathbf{X}_{\text{hid}} \in \mathbb{R}^{|\mathbf{x}| \times NN}$  is the matrix containing all outputs of the hidden layer for the training set, and  $NN$  is the number of neurons in the output layer [14].

## 4. Experimental Evaluation

In the next sections, the experimental protocol and results are described. Section 4.1 presents the data set used in the experiments, Section 4.2 details the preprocessing and postprocessing stages used in the forecasting process, Section 4.3 describes the procedure of parameters selection and Section 4.4 shows the performance metrics used in the experimental evaluation. Sections 4.5 and 4.6 present the simulation results and some remarks are discussed, respectively.

### 4.1. Data Description

The energy consumption data used in this investigation were collected by a smart meter installed in a residential building located in New Taipei City (Taiwan) [64]. The residents are two adults and three children. The floor total area occupied is 350 m<sup>2</sup>.

The sampling used was 15 min for 30 days, from 22 June 2015 to 26 July 2015. Thus, four samples are recorded in one hour, with 96 points in one single day (24 h), and totaling 2880 points in 30 days. It is important to highlight that the original data set presented one missing sample, which was completed using the average of the neighbor points.

The data set was divide into three subsets, maintaining the temporal order: training (1824 samples or 19 days), validation (384 samples or four days), and test (672 samples or seven days). The AR model adjustment considered the first two subsets as one. Table 1 shows the statistical description of the whole series and the respective subsets.

**Table 1.** Mean and standard deviation of the sets.

Set	Number of Samples	Mean (kWh)	Standard Deviation
Whole Series	2880	0.20077	0.10115
Training	1824	0.20794	0.10238
Validation	384	0.19789	0.10065
Test	672	0.18296	0.09579

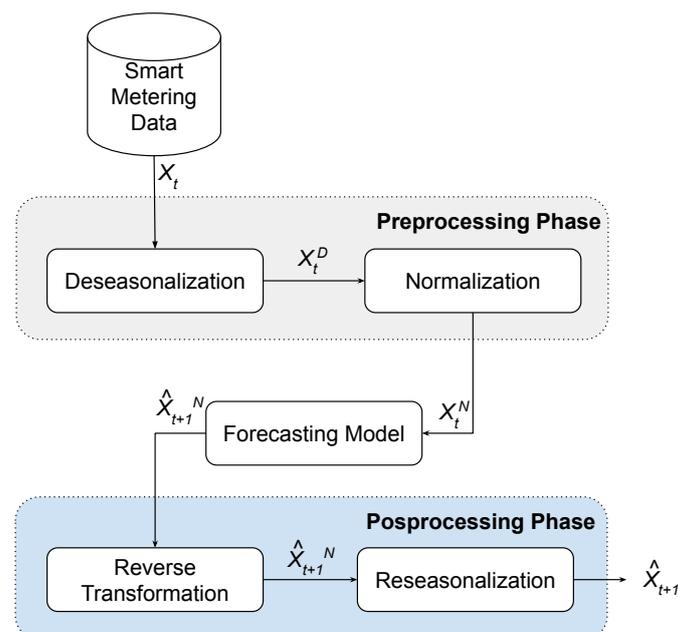
#### 4.2. Preprocessing and Postprocessing Stages

Energy consumption time series can be sampled weekly, daily, hourly, or minutely. As mentioned, the data used in this work was sampled every 15 min. This series presents a seasonal pattern every 96 points or an entire day. We performed a deseasonalization procedure that transforms the original series into approximately stationary, with zero mean and standard deviation close to one. This process changes the statistical behavior of the series, which may improve the output response of the forecasting models. In addition, the linear models from the Box and Jenkins methodology assume that the series is stationary [62]. The deseasonalization is given by Equation (10):

$$z_{i,s} = \frac{x_{i,s} - \hat{\mu}_s}{\hat{\sigma}_s} \quad (10)$$

in which  $z_{i,s}$  is the new standardized value of the  $i$  element of the original series  $x_{i,s}$ ,  $\hat{\mu}_s$  is the average of the elements of the series considering the seasonal pattern  $s$ , and  $\hat{\sigma}_s$  is the standard deviation.

All steps involved in the forecasting process of the consumption time series are summarized in Figure 2. The preprocessing stage is initiated after defining the time lags of the series, which are the inputs of the models. After that, deseasonalization is applied. Finally, a procedure to normalize the data into the interval  $[-1,+1]$  is performed. This normalization is mandatory for neural models that use hyperbolic tangent as activation function. After the preprocessing stage, the forecasting model generates the prediction based on the normalized inputs.



**Figure 2.** Stages of preprocessing and postprocessing employed in the modeling of the forecasting method.

In the postprocessing stage, the normalization and deseasonalization are reversed, leading to time series data to the original domain. With forecasts in the original domain at hand, the comparison with the actual series is performed to evaluate the models' errors.

#### 4.3. Experimental Setup

In this investigation, the experimental evaluation was performed using as single methods AR, MLP, ELM, ESN, and RBF. The proposed ensemble method also was compared with three distinct combination approaches: the mean and the median of the single models' outputs (non-trainable methods), and MLP [45,65].

As a baseline, we applied the seasonal autoregressive integrated moving average model (SARIMA) from the Box and Jenkins family [54], a classic linear model widely used in time series analysis. The parameters of the model were defined following the methodology proposed by Hyndman and Khandakar [66,67].

The parameters of the models addressed in this work were defined following some premises:

- The coefficients of the AR model were calculated using the Yule–Walker equations, a closed-form solution [54];
- All artificial neural networks used hyperbolic tangent as activation function of the hidden neurons [59];
- The number of neurons in the hidden layer was determined by previous empirical tests, considering a range of [3:500].
- All models were implemented in Matlab®.

Finally, The partial autocorrelation function (PACF) was applied to define the number of temporal lags used as inputs of the single models. Its application reveals that the first seven lags are significant, being adequate to solve the task. Therefore, all models employed a sliding window containing seven input lags in the forecasting of the electricity consumption dataset [54,68].

#### 4.4. Error Metrics

For performing a comparative analysis among the forecasting models we adopted five error metrics: mean squared error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE), root mean squared error (RMSE), and index of agreement (IA), which are described in Equations (11) to (15), respectively [67]:

$$MSE = \frac{1}{N} \sum (x_t - \hat{x}_t)^2, \quad (11)$$

$$MAE = \frac{1}{N} \sum |x_t - \hat{x}_t|, \quad (12)$$

$$RMSE = \sqrt{\frac{1}{N} \sum (x_t - \hat{x}_t)^2}, \quad (13)$$

$$MAPE = \frac{100}{n} \sum \left| \frac{x_t - \hat{x}_t}{x_t} \right|, \quad (14)$$

$$IA = 1 - \frac{\sum_{t=1}^N (x_t - \hat{x}_t)^2}{\sum_{t=1}^N (|\hat{x}_t - \bar{x}| + |x_t - \bar{x}|)^2}. \quad (15)$$

In all equations,  $N$  is the number of samples,  $x_t$  is the actual data, and  $\hat{x}_t$  is the predicted sample in time  $t$ . The MSE is a quadratic error measure which penalizes higher errors, but is sensible to outliers. The RMSE is less sensible to outliers, since the root square of the MSE is calculated. Likewise, MAE offers an error metric closer do the scale of the data. Furthermore, MSE, RMSE, and MAE are scale dependent error metrics [67]. In

contrast, MAPE and IA are not scale dependent. Note that all metrics must be minimized, except IA, which must be maximized in the range [0, 1].

#### 4.5. Results

Table 2 shows the values of the five error metrics (MSE, MAE, MAPE, RMSE, and IA) attained by the evaluated models for each day of the week. It is possible to observe that the approaches based on ensemble reached the best results in most of the cases (27 out of 35). The proposed ensemble ELM attained the best error values in 19 out of 35 comparisons. These results show the superiority of the proposal regarding statistical and ML models of the literature. The proposed ensemble obtained the best value in all weekdays in at least one performance metric. For instance, on Wednesday, Thursday, and Sunday, the ensemble ELM achieved the best values in majority of the performance measures. The single model ESN and the ensemble MLP reached the second-best result with the lowest error value in 5 out of 35 cases.

Considering the maximum value on all days of the week in Table 2, the ensemble ELM attained the best performance in terms of MAE, MAPE, and IA. Regarding the minimum value on all days of the week, the proposed ensemble achieved the best results in the MAE and IA metrics. The numbers show that the forecasts generated by the proposed ensemble presented stability on the different weekdays. These findings corroborate the hypothesis that supports the adoption of the ensemble in the forecasting task.

Table 3 shows the performance metrics values achieved by the models considering all days of the week. The results show that ensemble ELM attained the best values in all performance measures. The single ELM model obtained the second-best ranking in the MSE, MAE, RMSE, and IA metrics. The ensemble MLP attained the second-best MAPE value. The third-best value for the MSE, MAE, and RMSE were reached by the ensemble mean. The single models MLP and ESN achieved the third-best value for the MAPE and IA measures, respectively.

In order to verify if the proposed ensemble attained results statistically different from the other models, three hypothesis tests were used for this evaluation: Friedman test, Kruskal–Wallis test, and Wilcoxon test [69,70]. A significance level of 0.05 was employed in all hypothesis tests.

The statistical evaluation was performed from the MSE values obtained by the models in 30 independent executions, considering that some employed forecasting models, such as MLP, ELM, and ESN, have their parameters randomly initialized. In the literature, 30 samples are often considered sufficiently large and representative to perform the statistical analysis [71,72]. The  $p$ -values found were  $2.65 \times 10^{-39}$  and  $1.06 \times 10^{-43}$  for Friedman and Kruskal–Wallis tests, respectively. The Wilcoxon hypothesis test was employed to compare the results pairwise. In this case, the ensemble ELM (the best general model) and the ELM (the best single model) were compared with each forecasting model. Table 4 shows the  $p$ -values. In summary, considering the three tests addressed, we can assure that a change in the forecasting model led to distinct results since all  $p$ -values are smaller than 0.05.

#### 4.6. Discussion

Many aspects can be discussed about the forecasting responses and errors presented in this evaluation. Table 5 was elaborated to present a ranking regarding the values of the metrics achieved by the predictors. The column *Mean* presents the average of the positions in the ranking considering all metrics, while column *Rank* presents a ranking ordering the predictors.

The most important result is that the proposed ensemble ELM achieved the best overall results considering all metrics. This result corroborates with assumptions that support the use of the ELM as the combination model. In addition, among the first four best predictors, we have three ensembles. However, as the results show, a change in the combiner may lead to poor performances, as presented by the MLP.

Comparing the single ML models, the ELM attained the best value in 4 out of 5 performance metrics. Although the ELM is the “unorganized” version of the MLP neural network or a feedforward version of the ESN, the ELM model presented a superior accuracy among the single approaches. Furthermore, it is the second-best predictor, overcoming 3 of 4 ensembles employed in the experimental evaluation.

**Table 2.** The performance results in terms of the MSE, MAE, MAPE, RMSE, and IA metrics of the proposed Ensemble and literature models for each day of the week. The best values are highlighted in bold.

Model	Measure	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Max	Min
SARIMA	MSE ( $\times 10^{-3}$ kWh)	2.8720	2.7614	3.3822	2.9381	1.7513	2.0152	2.2561	3.3822	1.7513
	MAE (kWh)	0.0325	0.0332	0.0359	0.0353	0.0217	0.0288	0.0246	0.0359	0.0217
	MAPE (%)	15.8363	14.2251	17.1321	19.2733	13.2665	16.0358	14.9922	19.2733	13.2665
	RMSE (kWh)	0.0535	0.0525	0.0581	0.0542	0.0418	0.0448	0.0474	0.0581	0.0418
	IA	0.8400	0.8991	0.9272	0.8162	0.8970	0.9296	0.9419	0.9419	0.8162
AR	MSE ( $\times 10^{-3}$ kWh)	1.1708	1.8802	1.9812	1.8478	2.2187	3.4516	2.2879	3.4516	1.1708
	MAE (kWh)	0.0225	0.0297	0.0234	0.0300	0.0281	0.0355	0.0337	0.0355	0.0225
	MAPE (%)	14.3075	17.0181	14.9965	16.1628	<b>11.8905</b>	15.8668	19.7286	19.7286	<b>11.8905</b>
	RMSE (kWh)	0.0342	0.0434	0.0445	0.0430	0.0471	0.0588	0.0478	0.0588	0.0342
	IA	0.9355	0.9304	0.9568	0.9027	0.9202	0.9285	0.8827	0.9568	0.8827
MLP	MSE ( $\times 10^{-3}$ kWh)	1.1413	1.7036	1.8264	1.7168	2.1059	3.2103	1.9979	3.2103	1.1413
	MAE (kWh)	0.0217	0.0282	0.0216	0.0286	0.0289	0.0332	0.0304	0.0332	0.0216
	MAPE (%)	13.5270	15.7165	12.6500	15.4307	12.3867	14.9471	17.9853	17.9853	12.3867
	RMSE (kWh)	0.0338	0.0413	0.0427	0.0414	0.0459	0.0567	0.0447	0.0567	0.0338
	IA	<b>0.9367</b>	0.9375	0.9583	0.9100	0.9260	0.9325	0.9000	0.9583	0.9000
ELM	MSE ( $\times 10^{-3}$ kWh)	1.1526	1.6701	1.7890	1.7343	2.0591	3.1423	1.8294	3.1423	1.1526
	MAE (kWh)	<b>0.0209</b>	0.0271	0.0222	0.0285	0.0292	0.0329	0.0287	0.0329	0.0209
	MAPE (%)	12.7378	15.2818	13.9267	15.4566	12.6752	14.6467	17.0593	17.0593	12.6752
	RMSE (kWh)	0.0340	0.0409	0.0423	0.0416	0.0454	0.0561	0.0428	0.0561	0.0340
	IA	0.9356	0.9383	0.9594	0.9091	0.9284	0.9322	0.9079	0.9594	0.9079
ESN	MSE ( $\times 10^{-3}$ kWh)	1.1806	<b>1.5424</b>	1.7851	1.7948	1.9928	3.3204	2.0896	3.3204	1.1806
	MAE (kWh)	0.0213	0.0269	0.0220	0.0293	<b>0.0279</b>	0.0336	0.0305	0.0336	0.0213
	MAPE (%)	12.9235	15.4666	13.9666	15.9643	11.9069	14.9907	18.2586	18.2586	11.9069
	RMSE (kWh)	0.0344	<b>0.0393</b>	0.0423	0.0424	0.0446	0.0576	0.0457	0.0576	0.0344
	IA	0.9345	<b>0.9460</b>	0.9605	0.9044	<b>0.9320</b>	0.9320	0.8964	0.9605	0.8964
RBF	MSE ( $\times 10^{-3}$ kWh)	1.7832	1.7691	3.0669	2.1245	2.2380	3.4564	2.5668	3.4564	1.7691
	MAE (kWh)	0.0261	0.0287	0.0326	0.0313	0.0313	0.0368	0.0322	0.0368	0.0261
	MAPE (%)	15.2994	15.4352	22.0301	17.4932	13.7047	17.3610	20.1081	22.0301	13.7047
	RMSE (kWh)	0.0422	0.0421	0.0554	0.0461	0.0473	0.0588	0.0507	0.0588	0.0421
	IA	0.8957	0.9364	0.9243	0.8828	0.9245	0.9242	0.8633	0.9364	0.8633
Ensemble Mean	MSE ( $\times 10^{-3}$ kWh)	1.1632	1.6345	1.8150	1.7466	2.0379	3.1300	1.9460	3.1300	1.1632
	MAE (kWh)	0.0216	0.0277	0.0220	0.0289	0.0282	0.0325	0.0300	0.0325	0.0216
	MAPE (%)	13.0506	15.4767	13.5962	15.7273	12.1594	14.2690	17.7973	17.7973	12.1594
	RMSE (kWh)	0.0341	0.0404	0.0426	0.0418	0.0451	0.0559	0.0441	0.0559	0.0341
	IA	0.9342	0.9404	0.9583	0.9061	0.9289	<b>0.9336</b>	0.8992	0.9583	0.8992
Ensemble Median	MSE ( $\times 10^{-3}$ kWh)	<b>1.1378</b>	1.6290	1.8093	1.7949	2.0108	3.1865	1.9856	3.1865	<b>1.1378</b>
	MAE (kWh)	0.0215	0.0279	0.0214	0.0294	0.0281	0.0332	0.0303	0.0332	0.0214
	MAPE (%)	13.2260	15.7893	13.1504	15.9486	12.1569	14.8732	17.9705	17.9705	12.1569
	RMSE (kWh)	<b>0.0337</b>	0.0404	0.0425	0.0424	0.0448	0.0564	0.0446	0.0564	<b>0.0337</b>
	IA	0.9365	0.9409	0.9593	0.9046	0.9291	0.9329	0.8992	0.9593	0.8992
Ensemble MLP	MSE ( $\times 10^{-3}$ kWh)	1.1588	1.5856	1.7507	<b>1.7038</b>	1.9816	<b>3.0892</b>	1.7540	<b>3.0892</b>	1.1588
	MAE (kWh)	0.0210	0.0266	0.0219	<b>0.0278</b>	0.0292	0.0319	0.0275	0.0319	0.0210
	MAPE (%)	12.7435	<b>14.9182</b>	13.5774	15.2293	12.6736	14.1799	16.2368	16.2368	12.6736
	RMSE (kWh)	0.0340	0.0398	0.0418	0.0413	0.0445	<b>0.0556</b>	0.0419	<b>0.0556</b>	0.0340
	IA	0.9357	0.9428	0.9592	0.9095	0.9300	0.9328	0.9117	0.9592	0.9095
Ensemble ELM	MSE ( $\times 10^{-3}$ kWh)	1.2162	1.7898	<b>1.5592</b>	1.7071	<b>1.9356</b>	3.4034	<b>1.5598</b>	3.4034	1.2162
	MAE (kWh)	0.0217	<b>0.0261</b>	<b>0.0203</b>	0.0278	0.0303	<b>0.0296</b>	<b>0.0243</b>	<b>0.0303</b>	<b>0.0203</b>
	MAPE (%)	<b>12.5109</b>	15.0994	<b>12.1248</b>	<b>14.8702</b>	13.2287	<b>13.0881</b>	<b>13.8786</b>	<b>15.0994</b>	12.1248
	RMSE (kWh)	0.0349	0.0423	<b>0.0395</b>	<b>0.0413</b>	<b>0.0440</b>	0.0583	<b>0.0395</b>	0.0583	0.0349
	IA	0.9321	0.9382	<b>0.9624</b>	<b>0.9124</b>	0.9300	0.9249	<b>0.9263</b>	<b>0.9624</b>	<b>0.9124</b>

**Table 3.** MSE, MAE, MAPE, RMSE, and IA values for the evaluated models. The number of neurons used by each neural network is shown in the NN column. The performance corresponds to the whole test set of the energy consumption series. The best value for each metric is highlighted in bold.

	Model	NN	MSE ( $\times 10^{-3}$ kWh)	MAE (kWh)	MAPE (%)	RMSE (kWh)	IA
Single	SARIMA	-	2.5675	0.0303	15.4004	0.0506	0.9129
	AR	-	2.1195	0.0290	15.7090	0.0460	0.9318
	MLP	200	1.9574	0.0275	14.5376	0.0442	0.9391
	ELM	120	1.9110	0.0271	14.5393	0.0437	0.9405
	ESN	40	1.9579	0.0274	14.7819	0.0442	0.9402
	RBF	60	2.4292	0.0310	17.1017	0.0493	0.9226
Ensemble	Ensemble Mean	-	1.9247	0.0273	14.5826	0.0439	0.9373
	Ensemble Median	-	1.9363	0.0274	14.7307	0.0440	0.9375
	Ensemble MLP	40	2.1671	0.0284	14.2228	0.0466	0.9358
	Ensemble ELM	60	<b>1.8817</b>	<b>0.0257</b>	<b>13.5424</b>	<b>0.0434</b>	<b>0.9410</b>

**Table 4.**  $p$ -values of the Wilcoxon statistical test comparing the Ensemble ELM and ELM with the other forecasting models.

Models	$p$ -Value (Ensemble ELM)	$p$ -Value (ELM)
Ensemble ELM	—	0.0013
ELM	0.0013	—
SARIMA	$1.21 \times 10^{-12}$	$1.21 \times 10^{-12}$
AR	$7.47 \times 10^{-10}$	0.0045
MLP	0.0241	0.0323
ESN	$1.72 \times 10^{-6}$	0.0478
RBF	$3.01 \times 10^{-11}$	$3.01 \times 10^{-11}$
Ensemble Mean	$1.91 \times 10^{-7}$	$3.35 \times 10^{-11}$
Ensemble Median	$2.05 \times 10^{-5}$	$3.35 \times 10^{-11}$
Ensemble MLP	$8.48 \times 10^{-9}$	$1.35 \times 10^{-7}$

**Table 5.** Ranking of the models for each performance metric in the energy consumption forecasting.

	Model	MSE (kWh)	MAE (kWh)	MAPE (%)	RMSE (kWh)	IA	Mean	Rank
Single	SARIMA	10	9	8	10	10	9.4	9
	AR	7	8	9	7	8	7.6	8
	MLP	5	6	3	5	4	4.6	5
	ELM	2	2	4	2	2	2.4	2
	ESN	6	4	7	6	3	5.2	6
	RBF	9	10	10	9	9	9.4	9
Ensemble	Ensemble Mean	3	3	5	3	6	4	3
	Ensemble Median	4	5	6	4	5	4.8	4
	Ensemble MLP	8	7	2	8	7	6.4	7
	Ensemble ELM	1	1	1	1	1	1	1

The autoregressive (AR) model attained the best result regarding the five performance metrics between the linear approaches. The AR also obtained a superior performance to the RBF model in all metrics and the MLP model in terms of MSE and RMSE. The results show that nonlinear ML models or ML-based ensembles are more appropriate for this kind of problem. The energy consumption time series can have nonlinear patterns [7] that are not properly modeled by linear techniques, such as AR or SARIMA. However, ML models' adoption can also lead to underperforming results due to problems regarding overfitting, underfitting, or misspecification [59]. These issues can be related to the poor performance of the RBF model or the result of the combination using the MLP model that was not able

to overcome the nontrainable ensembles (ensemble mean and ensemble median). It is also important to mention that the computational cost to adjust the ELM and ESN is smaller than the fully trained models, such as MLP and RBF.

Figure 3 presents the boxplot of 30 simulations of each predictor. As expected, the SARIMA, AR, ensemble mean, and ensemble median do not present dispersion, since they present closed form solutions. It is interesting to note that the MLP showed a small dispersion, followed by the ESN. Regarding the median of the values, it is important to mention that the 3 best ensembles presented the best results, followed by the MLP.

Finally, Figure 4 depicts the real energy consumption time series, and the forecasting provided by the ensemble ELM (the best overall predictor) and the ELM, the best single model.

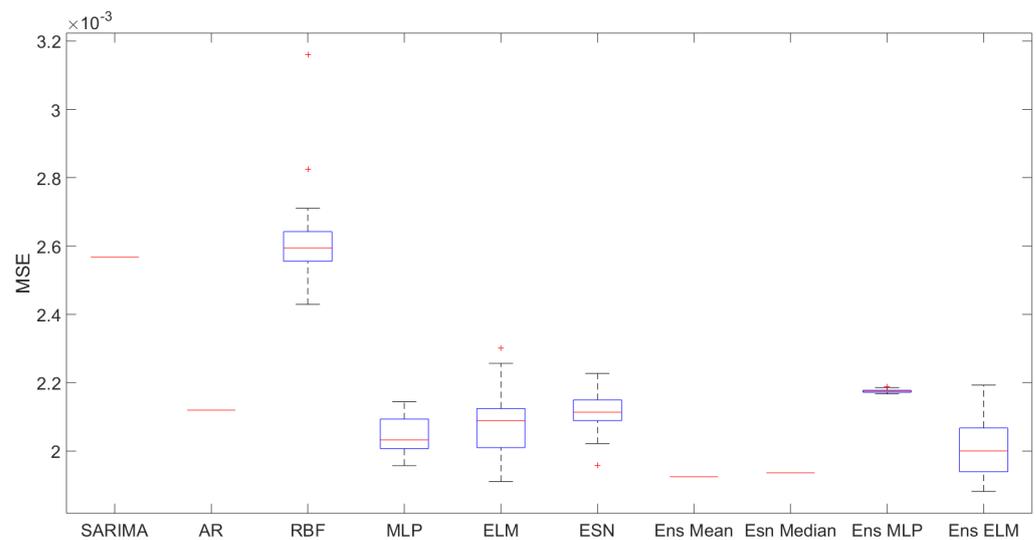


Figure 3. Boxplot graphic.

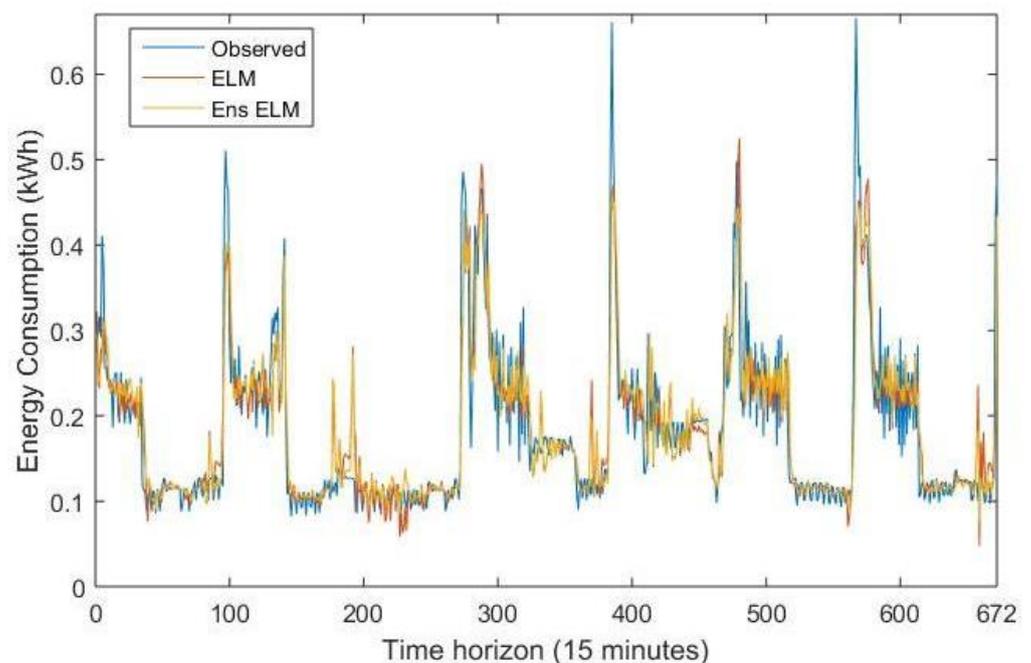


Figure 4. Energy consumption forecasting obtained by the ELM and ensemble ELM.

## 5. Conclusions

Energy consumption time series may present present linear and non-linear patterns, which hinders models to achieve accurate predictions. In this sense, the use of ensembles stood out in the current literature due to their high capability to increase the prediction power of stand-alone forecasting models. Based on that, we propose in this work neural-based ensembles for energy consumption forecasting.

More specifically, we use in this investigation as predictors the linear AR, and neural networks architectures: MLP, ELM, ESN, and RBF. As a combiner, we employ the non-trainable ensembles based on mean and median, and the MLP and ELM. As a baseline we address the SARIMA model.

The experimental evaluation was conducted by using a series from a residential building containing a installed smart grid network. Before the simulations, we applied a deseasonalization procedure in order to make the series stationary. The computational results showed that the ELM-based ensemble outperformed the other proposals in terms of five distinct error metrics. In addition, the single ELM stood out in comparison to the other single approaches. This is an important observation, since the ELM is a neural network with a simple training process, which confers a fast adjustment of the free parameters of the architecture.

As possible future directions, variable selection techniques can be applied to define the best lags as inputs of the forecasting models, and error correction hybrid models can be used to produce more accurate models. Finally, the propositions of this work should be tested in other databases related to energy consumption.

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

**Funding:** This work received funding and technical support from CPFL Energia within the scope of the project “PA3046—Development of Intelligent Measurement Platform with Cybersecurity, Business Intelligence and Big Data”, an R&D program regulated by ANEEL, Brazil” and “The APC was funded by CPFL Energia.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The authors would like to thank the database provided by Jui-Sheng Chou.

**Acknowledgments:** The authors thank the support of IATI—Advanced Institute of Technology and Innovation and Time Energy for providing the infrastructure and information necessary to carry out the tests. They also thank the Brazilian agencies Coordination for the Improvement of Higher Education Personnel (CAPES)—Financing Code 001, Brazilian National Council for Scientific and Technological Development (CNPq), processes number 40558/2018-5, 315298/2020-0, and Araucaria Foundation, process number 51497, for their financial support.

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

## References

1. Räsänen, T.; Voukantsis, D.; Niska, H.; Karatzas, K.; Kolehmainen, M. Data-based method for creating electricity use load profiles using large amount of customer-specific hourly measured electricity use data. *Appl. Energy* **2010**, *87*, 3538–3545. [[CrossRef](#)]
2. Khan, Z.A.; Hussain, T.; Ullah, A.; Rho, S.; Lee, M.; Baik, S.W. Towards Efficient Electricity Forecasting in Residential and Commercial Buildings: A Novel Hybrid CNN with a LSTM-AE based Framework. *Sensors* **2020**, *20*, 1399. [[CrossRef](#)]

3. Nejat, P.; Jomehzadeh, F.; Taheri, M.M.; Gohari, M.; Majid, M.Z.A. A global review of energy consumption, CO<sub>2</sub> emissions and policy in the residential sector (with an overview of the top ten CO<sub>2</sub> emitting countries). *Renew. Sustain. Energy Rev.* **2015**, *43*, 843–862. [[CrossRef](#)]
4. Divina, F.; Gilson, A.; Gómez-Vela, F.; García Torres, M.; Torres, J.F. Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting. *Energies* **2018**, *11*, 949. [[CrossRef](#)]
5. Kolokotsa, D. The role of smart grids in the building sector. *Energy Build.* **2016**, *116*, 703–708. [[CrossRef](#)]
6. Hadri, S.; Najib, M.; Bakhouya, M.; Fakhri, Y.; El Arroussi, M. Performance Evaluation of Forecasting Strategies for Electricity Consumption in Buildings. *Energies* **2021**, *14*, 5831. [[CrossRef](#)]
7. Lusi, P.; Khalilpour, K.R.; Andrew, L.; Liebman, A. Short-term residential load forecasting: Impact of calendar effects and forecast granularity. *Appl. Energy* **2017**, *205*, 654–669. [[CrossRef](#)]
8. Alberg, D.; Last, M. Short-term load forecasting in smart meters with sliding window-based ARIMA algorithms. *Vietnam. J. Comput. Sci.* **2018**, *5*, 241–249. [[CrossRef](#)]
9. Kumar Dubey, A.; Kumar, A.; García-Díaz, V.; Kumar Sharma, A.; Kanhaiya, K. Study and analysis of SARIMA and LSTM in forecasting time series data. *Sustain. Energy Technol. Assessments* **2021**, *47*, 101474. [[CrossRef](#)]
10. Taskaya-Temizel, T.; Casey, M.C. A comparative study of autoregressive neural network hybrids. *Neural Netw.* **2005**, *18*, 781–789. [[CrossRef](#)]
11. Kourentzes, N.; Barrow, D.; Petropoulos, F. Another look at forecast selection and combination: Evidence from forecast pooling. *Int. J. Prod. Econ.* **2019**, *209*, 226–235. [[CrossRef](#)]
12. Cassales, G.; Gomes, H.; Bifet, A.; Pfahringer, B.; Senger, H. Improving the performance of bagging ensembles for data streams through mini-batching. *Inf. Sci.* **2021**, *580*, 260–282. [[CrossRef](#)]
13. Kourentzes, N.; Barrow, D.K.; Crone, S.F. Neural network ensemble operators for time series forecasting. *Expert Syst. Appl.* **2014**, *41*, 4235–4244. [[CrossRef](#)]
14. 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](#)]
15. Ahmad, T.; Chen, H. A review on machine learning forecasting growth trends and their real-time applications in different energy systems. *Sustain. Cities Soc.* **2020**, *54*, 102010. [[CrossRef](#)]
16. Deb, C.; Zhang, F.; Yang, J.; Lee, S.E.; Shah, K.W. A review on time series forecasting techniques for building energy consumption. *Renew. Sustain. Energy Rev.* **2017**, *74*, 902–924. [[CrossRef](#)]
17. Heydari, A.; Keynia, F.; Garcia, D.A.; De Santoli, L. Mid-Term Load Power Forecasting Considering Environment Emission using a Hybrid Intelligent Approach. In Proceedings of the 2018 5th International Symposium on Environment-Friendly Energies and Applications (EFEA), Rome, Italy, 24–26 September 2018; pp. 1–5.
18. Chan, S.C.; Tsui, K.M.; Wu, H.C.; Hou, Y.; Wu, Y.; Wu, F.F. Load/Price Forecasting and Managing Demand Response for Smart Grids: Methodologies and Challenges. *IEEE Signal Process. Mag.* **2012**, *29*, 68–85. [[CrossRef](#)]
19. Culaba, A.B.; Del Rosario, A.J.R.; Ubando, A.T.; Chang, J.S. Machine learning-based energy consumption clustering and forecasting for mixed-use buildings. *Int. J. Energy Res.* **2020**, *44*, 9659–9673. [[CrossRef](#)]
20. Gao, Y.; Ruan, Y.; Fang, C.; Yin, S. Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data. *Energy Build.* **2020**, *223*, 110156. [[CrossRef](#)]
21. Pinto, T.; Praça, I.; Vale, Z.; Silva, J. Ensemble learning for electricity consumption forecasting in office buildings. *Neurocomputing* **2021**, *423*, 747–755. [[CrossRef](#)]
22. Walther, J.; Weigold, M. A Systematic Review on Predicting and Forecasting the Electrical Energy Consumption in the Manufacturing Industry. *Energies* **2021**, *14*, 968. [[CrossRef](#)]
23. Barzola-Monteses, J.; Espinoza-Andaluz, M.; Mite-León, M.; Flores-Morán, M. Energy Consumption of a Building by using Long Short-Term Memory Network: A Forecasting Study. In Proceedings of the 2020 39th IEEE International Conference of the Chilean Computer Science Society (SCCC), Coquimbo, Chile, 16–20 November 2020; pp. 1–6.
24. Rana, M.; Koprinska, I. Forecasting electricity load with advanced wavelet neural networks. *Neurocomputing* **2016**, *182*, 118–132. [[CrossRef](#)]
25. Zolfaghari, M.; Golabi, M.R. Modeling and predicting the electricity production in hydropower using conjunction of wavelet transform, long short-term memory and random forest models. *Renew. Energy* **2021**, *170*, 1367–1381. [[CrossRef](#)]
26. El-Hendawi, M.; Wang, Z. An ensemble method of full wavelet packet transform and neural network for short term electrical load forecasting. *Electr. Power Syst. Res.* **2020**, *182*, 106265. [[CrossRef](#)]
27. Gajowniczek, K.; Ząbkowski, T. Short term electricity forecasting using individual smart meter data. *Procedia Comput. Sci.* **2014**, *35*, 589–597. [[CrossRef](#)]
28. Zhukov, A.V.; Sidorov, D.N.; Foley, A.M. Random forest based approach for concept drift handling. In *International Conference on Analysis of Images, Social Networks and Texts*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 69–77.
29. Heydari, A.; Nezhad, M.M.; Pirshayan, E.; Garcia, D.A.; Keynia, F.; De Santoli, L. Short-term electricity price and load forecasting in isolated power grids based on composite neural network and gravitational search optimization algorithm. *Appl. Energy* **2020**, *277*, 115503. [[CrossRef](#)]
30. Heydari, A.; Garcia, D.A.; Keynia, F.; Bisegna, F.; Santoli, L.D. Hybrid intelligent strategy for multifactor influenced electrical energy consumption forecasting. *Energy Sources Part B Econ. Plan. Policy* **2019**, *14*, 341–358. [[CrossRef](#)]

31. Yu, C.N.; Mirowski, P.; Ho, T.K. A sparse coding approach to household electricity demand forecasting in smart grids. *IEEE Trans. Smart Grid* **2016**, *8*, 738–748. [[CrossRef](#)]
32. Fekri, M.N.; Patel, H.; Grolinger, K.; Sharma, V. Deep learning for load forecasting with smart meter data: Online Adaptive Recurrent Neural Network. *Appl. Energy* **2021**, *282*, 116177. [[CrossRef](#)]
33. Li, L.; Meinrenken, C.J.; Modi, V.; Culligan, P.J. Short-term apartment-level load forecasting using a modified neural network with selected auto-regressive features. *Appl. Energy* **2021**, *287*, 116509. [[CrossRef](#)]
34. Komatsu, H.; Kimura, O. Peak demand alert system based on electricity demand forecasting for smart meter data. *Energy Build.* **2020**, *225*, 110307. [[CrossRef](#)]
35. Sajjad, M.; Khan, Z.A.; Ullah, A.; Hussain, T.; Ullah, W.; Lee, M.Y.; Baik, S.W. A novel CNN-GRU-based hybrid approach for short-term residential load forecasting. *IEEE Access* **2020**, *8*, 143759–143768. [[CrossRef](#)]
36. Wang, Y.; Gan, D.; Sun, M.; Zhang, N.; Lu, Z.; Kang, C. Probabilistic individual load forecasting using pinball loss guided LSTM. *Appl. Energy* **2019**, *235*, 10–20. [[CrossRef](#)]
37. Zhang, L.; Wen, J.; Li, Y.; Chen, J.; Ye, Y.; Fu, Y.; Livingood, W. A review of machine learning in building load prediction. *Appl. Energy* **2021**, *285*, 116452. [[CrossRef](#)]
38. Somu, N.; Raman M R, G.; Ramamritham, K. A deep learning framework for building energy consumption forecast. *Renew. Sustain. Energy Rev.* **2021**, *137*, 110591. [[CrossRef](#)]
39. Chou, J.S.; Truong, D.N. Multistep energy consumption forecasting by metaheuristic optimization of time-series analysis and machine learning. *Int. J. Energy Res.* **2021**, *45*, 4581–4612. [[CrossRef](#)]
40. Chou, J.S.; Truong, D.N. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Appl. Math. Comput.* **2021**, *389*, 125535. [[CrossRef](#)]
41. Bouktif, S.; Fiaz, A.; Ouni, A.; Serhani, M. Multi-sequence LSTM-RNN deep learning and metaheuristics for electric load forecasting. *Energies* **2020**, *13*, 391. [[CrossRef](#)]
42. 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; Volume 2, pp. 1625–1630.
43. 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](#)]
44. 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](#)]
45. Belotti, J.; Siqueira, H.; Araujo, L.; Stevan, S.L.; de Mattos Neto, P.S.; Marinho, M.H.; de Oliveira, J.F.L.; Usberti, F.; Leone Filho, M.d.A.; Converti, A.; et al. Neural-Based Ensembles and Unorganized Machines to Predict Streamflow Series from Hydroelectric Plants. *Energies* **2020**, *13*, 4769. [[CrossRef](#)]
46. de Mattos Neto, P.S.; de Oliveira, J.F.L.; Júnior, D.S.d.O.S.; Siqueira, H.V.; Marinho, M.H.D.N.; Madeiro, F. A Hybrid Nonlinear Combination System for Monthly Wind Speed Forecasting. *IEEE Access* **2020**, *8*, 191365–191377. [[CrossRef](#)]
47. Siqueira, H.; Boccato, L.; Attux, R.; Lyra, C. Unorganized machines for seasonal streamflow series forecasting. *Int. J. Neural Syst.* **2014**, *24*, 1430009. [[CrossRef](#)] [[PubMed](#)]
48. Domingos, S.d.O.; de Oliveira, J.F.; de Mattos Neto, P.S. An intelligent hybridization of ARIMA with machine learning models for time series forecasting. *Knowl.-Based Syst.* **2019**, *175*, 72–86.
49. Siqueira, H.; Luna, I.; Alves, T.A.; de Souza Tadano, Y. The direct connection between box & Jenkins methodology and adaptive filtering theory. *Math. Eng. Sci. Aerosp. (MESA)* **2019**, *10*, 27–40.
50. 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](#)]
51. Yang, D. Spatial prediction using kriging ensemble. *Sol. Energy* **2018**, *171*, 977–982. [[CrossRef](#)]
52. Kim, D.; Hur, J. Short-term probabilistic forecasting of wind energy resources using the enhanced ensemble method. *Energy* **2018**, *157*, 211–226. [[CrossRef](#)]
53. Berardi, V.; Zhang, G. An empirical investigation of bias and variance in time series forecasting: Modeling considerations and error evaluation. *IEEE Trans. Neural Netw.* **2003**, *14*, 668–679. [[CrossRef](#)]
54. Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
55. Zhang, G.P.; Qi, M. Neural network forecasting for seasonal and trend time series. *Eur. J. Oper. Res.* **2005**, *160*, 501–514. [[CrossRef](#)]
56. 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](#)]
57. Werbos, P.J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. Thesis, Harvard University, Cambridge, UK, 1974.
58. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Cogn. Model.* **1986**, *5*, 1. [[CrossRef](#)]
59. Haykin, S.S. *Neural Networks and Learning Machines/Simon Haykin*; Prentice Hall: New York, NY, USA, 2009.
60. Jaeger, H. The “echo state” approach to analysing and training recurrent neural networks—with an erratum note. *Ger. Natl. Res. Cent. Inf. Technol.* **2001**, *148*, 13.

61. Siqueira, H.; Boccato, L.; Attux, R.; Lyra Filho, C. Echo state networks for seasonal streamflow series forecasting. In *International Conference on Intelligent Data Engineering and Automated Learning*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 226–236.
62. 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](#)]
63. Siqueira, H.; Luna, I. Performance comparison of feedforward neural networks applied to stream flow series forecasting. *Math. Eng. Sci. Aerosp.* **2019**, *10*, 41–53.
64. Chou, J.S.; Ngo, N.T. Time series analytics using sliding window metaheuristic optimization-based machine learning system for identifying building energy consumption patterns. *Appl. Energy* **2016**, *177*, 751–770. [[CrossRef](#)]
65. de Mattos Neto, P.S.; Firmino, P.R.A.; Siqueira, H.; Tadano, Y.D.S.; Alves, T.A.; De Oliveira, J.F.L.; Marinho, M.H.D.N.; Madeiro, F. Neural-Based Ensembles for Particulate Matter Forecasting. *IEEE Access* **2021**, *9*, 14470–14490. [[CrossRef](#)]
66. Hyndman, R.; Khandakar, Y. Automatic Time Series Forecasting: The forecast package for R. *J. Stat. Software* **2008**, *27*, 1–22. [[CrossRef](#)]
67. Hyndman, R.J.; Koehler, A.B. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. [[CrossRef](#)]
68. Siqueira, H.; Macedo, M.; Tadano, Y.d.S.; Alves, T.A.; Stevan, S.L.; Oliveira, D.S.; Marinho, M.H.; Neto, P.S.; de Oliveira, J.F.; Luna, I.; et al. Selection of temporal lags for predicting riverflow series from hydroelectric plants using variable selection methods. *Energies* **2020**, *13*, 4236. [[CrossRef](#)]
69. Diebold, F.X.; Mariano, R.S. Comparing predictive accuracy. *J. Bus. Econ. Stat.* **2002**, *20*, 134–144. [[CrossRef](#)]
70. Harvey, D.; Leybourne, S.; Newbold, P. Testing the equality of prediction mean squared errors. *Int. J. Forecast.* **1997**, *13*, 281–291. [[CrossRef](#)]
71. Brownlee, J. *Statistical Methods for Machine Learning: Discover How to Transform Data into Knowledge with Python*; Machine Learning Mastery: San Francisco, CA, USA, 2018.
72. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.