Wind Speed Forecasting using a Portfolio of Forecasters

Mario Graff[a], Rafael Peña[b], Aurelio Medina[a], Hugo Jair Escalante[c]

[a] *División de Estudios de Postgrado*
Facultad de Ingeniería Eléctrica
Universidad Michoacana de San Nicolás de Hidalgo, México
[b] *Facultad de Ingeniería*
Universidad Autónoma de San Luis Potosí, México
[c] *Computer Science Department*
Instituto Nacional de Astrofísica, Óptica y Electrónica, México

Abstract

This contribution presents the application of a portfolio of forecasters to the problem of wind speed forecasting. This portfolio is created using a single time series and it is based on a number of time series characteristics, previously proposed, and a set of novel time series features. The results show that the proposed portfolio produces accurate predictions, and, has better performance than the forecasters composing it. In addition to this, the forecast values are used to determine the power generation capacity of a wind turbine driven a permanent magnet synchronous generator.

Keywords: Algorithm Portfolio, Algorithm Selection Problem, Time Series Forecasting, Power Generation Capacity, Wind Turbines

1. Introduction

The study of trends and patterns of complex systems is of great interest since the results obtained from these studies support the decision-making process in many activities. For the case of wind farms, the wind speed is the variable that decides the amount of electrical energy that can be delivered by the farm. Given the fact that the wind speed is a random variable [1], it is necessary to develop techniques capable of forecasting this variable. The forecast results are in particular important to the optimal day-to-day operation of the power systems [2].

The forecast of the wind speed has been previously addressed and reported in the literature; for example in [3, 4] an Auto-Regressive Integrate Moving Average (ARIMA) model was used, in [5, 6] an Artificial Neural Network (ANN) was developed, closely related, in [7, 8] an ensemble of ANN with empirical mode decomposition was proposed, in [9] a support vector machine (SVM) with wavelet transform was presented, a Bayesian structural model was used in [10], and, in [11, 12], a Genetic Programming technique was applied, just to mention a few works. Clearly, there are other variables of interest to the electrical scientific community, such as electric load forecasting [13, 14],

Email addresses: `mgraffg@dep.fie.umich.mx` (Mario Graff), `rafael.pena@uaslp.mx` (Rafael Peña), `amedina@dep.fie.umich.mx` (Aurelio Medina), `hugojair@inaoep.mx` (Hugo Jair Escalante)

economic forecasting [15], water flow [16], wave behaviour [17, 18], maximum temperature of the next day [19], and solar radiation [20, 21, 22, 23, 24], among others. For a recent review in forecasting of wind power generation we refer the reader to Foley *et al.* [1].

The first problem to face, when forecasting a time series, is to decide among all the different forecasters which one to use. This problem is known as the *algorithm selection problem* [25], the idea is to select among a set of different algorithm which one would perform the best given a particular problem. In this context, the problem is to select the forecaster that would perform the more accurate predictions. Closely related to this problem is the algorithm portfolio. An *algorithm portfolio* is a collection of algorithms that are run in parallel or in sequence to solve a particular problem. That is, the portfolio would perform more accurately predictions than any of the forecasters included in it.

Recently, an algorithm portfolio composed by traditional forecasters was developed in [26], namely ARIMA (see [27, 28]), and the Exponential smoothing state space model with Box-Cox transformation, Auto-Regressive Moving Average (ARMA) errors, Trend and Seasonal components (BATS) (see [29]). This portfolio was tested on different time series, and, for each time series, it selects the forecaster that would perform the best for that particular case. Clearly, the approach reported in [26] is not adequate when there is only one time series. This is because the portfolio would be used only once, consequently, the cost of building it would not be justified.

The idea of automatically selecting a forecaster has been studied previously; one of the first attempts to automate the process of selecting and/or tuning a forecaster was presented in [30]. The authors proposed a number of rules that were used to combine the forecasts performed by four techniques: random walk, linear regression, Holt's linear exponential smoothing [31], and Brown's linear exponential smoothing [32]. This work was then extended and improved in [33, 34], where the authors reduced the number of rules and proposed an automatic process to identify the most prominent time series features.

The first fully automated selection procedure was presented in [35] (and later extended in [36]). In these works, an induction-based expert system was used to select the most promising forecasting technique based on time series features. Almost during the same period, a discriminant analysis was used to select the most appropriate forecasting technique (see [37]). It is interesting that neither of them used the terms meta-learning or an algorithm selection problem (a common issue across different domains, further discussed in [38]).

More recently, this problem has been addressed by different researchers (see [39, 40, 41, 42]) using a variety of machine learning techniques, including linear combination of features, and decision trees, among others; and proposing novel time series characteristics that aim to enrich the set of features describing the time series. All these works have in common that all the features used to tackle the algorithm selection problem are based on time series characteristics. That is, these features are tailored specifically to describe time series, a few examples of these are: trend, seasonality, serial correlation, and periodicity, among others.

The aforementioned procedures, including our previous work [26], perform the next steps to decide which forecaster will be used given a time series, i.e., to solve the algorithm selection problem. Firstly, these techniques require a set of different time series, for each of them their characteristics are computed. Secondly, a machine learning technique is trained to relate the characteristics with the performance of the different forecasters composing the portfolio. Finally, given a new time series, the machine learning technique, trained previously, decides which forecaster will be used to predict the next points of the given time series.

To the best of our knowledge, a procedure, in a portfolio, that solves the algorithm selection problem using a single time series has not been proposed in the literature. This contribution tries to fill this gap by extending our previous work [26] to the case of a single time series. The idea is to solve the algorithm selection problem given a time series, and forecast n points ahead of that time series.

For a completeness of this contribution, the wind speed forecast is used to determine power generation capacity of a stand-alone wind turbine driven by a Permanent Magnet Synchronous Generator (PMSG) in the next hours. We have focus on this research topic due to the fact that wind power is today's most rapidly growing renewable energy source. This as a result of the generalized growing concern regarding the collaboration between fossil fuels and environmental pollution, the depletion of fossil fuels, and in consequence the increase of fossil fuels prices.

From the different types of wind turbines available in the market, in this paper a PMSG wind turbine is used, because this type of wind turbines have drawn great interest to wind turbine manufactures due to the advance of power electronics technology, improved designs and fabrication procedures [43, 44]. The model used in this contribution was implemented in the software Simulink/Matlab [45], this with the aim of having a more accurate power generation representation.

The rest of this paper is organized as follows: Section 2 presents the time series characteristics proposed by Wang *et al.*, Lemke *et al.*, and our own. These time series features are extended in Section 3 to be used in the portfolio when there is only one time series. Section 4 presents the details of the PMSG wind generation system. In Section 5, the results obtained in the forecast of the wind speed time series are presented and discussed; finally, Section 6 draws the main conclusions of this research work.

2. Portfolio of Forecasters

In general, an algorithm portfolio is built using a machine learning technique that selects the algorithm used to solve the problem at hand. Commonly, supervised learning algorithm [46] are used, and the portfolio is treated as a classification problem or as many regression problems. In a classification problem each different algorithm in the portfolio is given a unique label, and the classification algorithm is trained using a set of pairs of $\mathcal{T} = \{(\mathbf{x}_i, \ y_i) \ | i = 1 \dots n\}$ where $\mathbf{x}_i$ is a vector of features —this is related to the problems being solved — and y_i corresponds to the algorithm that solved the best the i-th problem.

On the other hand, when the portfolio uses a regression algorithm to make the selection, one needs to solve as many regression problems as algorithms are in the portfolio. More specifically, for the j-th algorithm in the portfolio, a regression problem is created using $\mathcal{T}$, as previously stated, with the difference that y_i is the performance of the j-th algorithm on the i-th problem, e.g., mean square error. This process can be seen as modeling performance of an algorithm (see [47]); and, for now on, we refer to the trained regression algorithm as being a model of performance, all these models are used to estimate the performance of all the algorithms in the portfolio given a new problem, then the algorithm selected is the one that has the best predicted performance. This approach is followed in this contribution.

As it can be observed, the first step to create a portfolio is to build $\mathcal{T}$. $\mathcal{T}$ is build using the pairs $(\mathbf{x}, \ y)$, where y is given by the performance of the algorithms, and $\mathbf{x} \in \Re^d$ is the vector of features that are related to the problem being solved.

To the best of our knowledge, all previous work designing portfolio of forecasters have followed the following procedure to compute $\mathbf{x}$. Let x_i be the i-th component of $\mathbf{x}$, then $x_i = f_i(\mathbf{y})$ where $\mathbf{y}$ is the time series of length ℓ, and $f_i : \Re^\ell \rightarrow \Re$. Consequently, $\mathbf{x}$ is computed using a set of functions $\mathcal{F} = \{f_i | i = 1 \dots d\}$ and a time series $\mathbf{y}$. Finally, $\mathcal{T}$ is built by applying $\mathcal{F}$ on a set of different time series, i.e., $\{\mathbf{y}_1, \dots, \mathbf{y}_n\}$.

In the rest of this section, we describe different time series characteristics, i.e. f, that have been proposed previously by other researchers and our own previous works. Subsection 2.1 shows the features proposed by Wang *et al.* [41]. The time series characteristics proposed by Lemke *et al.* are shown in Subsection 2.2, and the features proposed in our previous research work [47, 48, 26] are described in Subsection 2.3.

2.1. Wang et al.*'s Time Series Characteristics*

Wang *et al.* [41] proposed 13 time series features used to characterize univariate time series. Let us denote by w_i for $i = 1$ to 13 these time series characteristics, and w has the same signature than f.

Let $w_1 = 1 - \frac{\sigma^2(\mathbf{y}_f)}{\sigma^2(\mathbf{y}_p - \mathbf{y}_s)}$ and $w_2 = 1 - \frac{\sigma^2(\mathbf{y}_f)}{\sigma^2(\mathbf{y}_p - \mathbf{y}_t)}$ where $\mathbf{y}_f = \mathbf{y}_p - \mathbf{y}_t - \mathbf{y}_s$, $\mathbf{y}_p$ is the time series after the Box-Cox transformation[1] [50]; $\mathbf{y}_t$ corresponds to the trend of $\mathbf{y}_p$, $\mathbf{y}_s$ is the seasonal

[1] We used the BoxCox transformation implemented in [49]. λ is obtained using the loglik method with -1 and

component of $\mathbf{y}_p$, and $\sigma^2(\cdot)$ is the variance; w_3 is the periodicity of $\mathbf{y}_p - \mathbf{y}_t$ which is computed using the autocorrelation. The next two features, w_4 and w_5, correspond to the serial correlation of $\mathbf{y}$ and $\mathbf{y}_f$, respectively; w_6 and w_7 are the nonlinear autoregresive structure measured from $\mathbf{y}$ and $\mathbf{y}_f$, respectively, w_8 and w_9 are the skew of $\mathbf{y}$ and $\mathbf{y}_f$, respectively; and w_{10} and w_{11} are the kurtosis of $\mathbf{y}$ and $\mathbf{y}_f$, respectively. The self-similarity or long-range dependence corresponds to w_{12} and it is measured using $\mathbf{y}_f$. The last characteristic, i.e., w_{13}, is chaos, which is computed using the Lyapunov exponent on $\mathbf{y}$.

2.2. Lemke et al.*'s Time Series Characteristics*

Lemke *et al.* [42] used different machine-learning techniques such as neural networks, decision trees, support vector machines, and zoomed ranking, to tackle the algorithm selection problem. These machine learning techniques were trained with 27 time series features. Let l_i for $i = 1$ to 27 be the different time series characteristics.

The first ten features are based on traditional statistics, i.e., l_1 be the $\sigma(\mathbf{y}_p - \mathbf{y}_t)$, l_2 is the skewness of $\mathbf{y}$, l_3 is the kurtosis, $l_4 = \ell$, $l_5 = \frac{\mathbf{y}}{\mathbf{y}_p - \mathbf{y}_t}$, l_6 is the Durbin-Watson test of $\mathbf{y} - \mathbf{y}_p + \mathbf{y}_t$, l_7 counts the turning points of $\mathbf{y}$, l_8 is the step change, l_9 computes the non-linearity, and l_{10} is the largest Lyapunov exponent.

The next five features correspond to characteristics computed in the frequency domain, i.e., l_{11}, l_{12}, and l_{13} are the highest values of the power spectrum frequencies. l_{14} is maximum value of the power spectrum, and l_{15} count the number of peaks greater than 60% of the maximum.

The next four features correspond to the auto-correlation coefficients, namely *acf* (auto-correlation function) and *pacf* (partial auto-correlation function). l_{16} and l_{17} are the first two coefficients of *acf*, and l_{18} and l_{19} correspond to the first two coefficients of *pacf*.

The last eight characteristics measure the diversity of the algorithms being analyzed, i.e., l_{20}, l_{21} and l_{22} measure the difference between the error of each forecaster and the error of a forecaster composed by the average forecast of the whole algorithms; l_{23}, l_{24}, and l_{25} are the ratios between the previous values, i.e., error of forecasters and error of average forecast, and l_{26} and l_{27} are the mean and standard deviation of the correlation coefficients of the forecast made by the algorithms. Lemke *et al.* [42] proposed two more features which are the number of algorithms in the top performing cluster and the distance between the top performing cluster and the second best; however, these values are not included in this contribution because our forecasters are not grouped into clusters.

---

1 as its limits

2.3. Graff et al.*'s Features*

Our first set of features [47] were proposed to estimate the performance of Genetic Programming (GP) (see [51, 52]) and related techniques. This set of features uses the idea that the performance of a GP system on problem $\mathbf{y}$, i.e., $P(\mathbf{y})$, can be estimated using the performance of a set of reference algorithms in the problem $\mathbf{y}$. In formulae, the model is defined as:

$$P(\mathbf{y}) \approx a_0 + \sum_{s \in \mathcal{S}} a_s \cdot d_s(\mathbf{y}) \tag{1}$$

where $\mathcal{S}$ are the solutions given by the algorithms, $d_s(\mathbf{y})$ is the performance of the algorithm s on problem $\mathbf{y}$, and a_s are coefficients that need to be identified. In order to initialize Equation (1), one needs to define $\mathcal{S}$. In this contribution, $\mathcal{S}$ contains the algorithms composing the portfolio.

Our second set of features [48, 26] are inspired by the discrete derivative of a function. The discrete derivative of a function f w.r.t. a variable x is defined as:

$$\triangle_h f(x) = \frac{f(x+h) - f(x)}{h}, \tag{2}$$

where h is the step size. Normally, one tries to set h to the smallest possible value; however, counter intuitively, the difficulty indicators used are computed by varying the value of h and the order of the derivative i. In formulae:

$$\varsigma_i^h = \frac{1}{|I|} \sum_{x \in I} | \triangle_h f^{(i)}(x)|, \tag{3}$$

where i is the i-th order derivative, h is the step used to compute it, and I contains all the input patters. These features are easily explained with an example. Firstly, let f be a time series where $f(1)$ is the first measurement of the series, $f(2)$ is the second, and so on. In this case, I is $1, 2, \ldots, \ell$, where ℓ is the length of the time series. Secondly, let us define f as $f(1) = 12$, $f(2) = 2$, and $f(3) = -1$, consequently, $\varsigma_1^1(f) = | \triangle_1 f^{(1)}(1)| + | \triangle_1 f^{(1)}(2)|$, $\triangle_1 f^{(1)}(1) = f(2) - f(1) = -10$, $\triangle_1 f^{(1)}(2) = f(3) - f(2) = -3$, and this yields to $\varsigma_1^1(f) = \frac{13}{3}$.

To sum up, we have described the time series characteristics presented by Wang *et al.*, Lemke *et al.*, and our own. In the next section, we describe how these features can be used to create a portfolio of forecasters for a single time series.

3. Portfolios for a Single Time Series

As we have mentioned previously, in order to create a portfolio is needed a training set $\mathcal{T}$ formed by pairs of problems and performance of the algorithms [38]. So far, we have described that, in previous work, the problems are time series whose characteristics are used to create $\mathcal{T}$. Clearly, for the case where there exists only a time series, then, the previous approaches are not viable because $\mathcal{T}$ would have only one element.

In order to create $\mathcal{T}$ given only a time series, we follow the procedure used to initialize an auto-regressive model. An auto-regressive model has the form:

$$\hat{y}_t = \sum_{i=1}^{w} a_i y_{t-i}, \tag{4}$$

where $\hat{y}_t$ is the value of the time series at time t, a_i is a coefficient that needs to be identified, and w is the window width. In order to identify a, one can use ordinary least squares (OLS) as follows. Let $\mathbf{a} = (a_1, \ldots, a_w)'$, $\mathbf{b} = (y_{w+1}, y_{w+2}, \ldots, y_\ell)'$, and

$$\mathbf{W} = \begin{pmatrix} y_1 & y_2 & \cdots & y_w \\ \vdots & \vdots & \ldots & \vdots \\ y_{\ell-w} & \cdots & \cdots & y_{\ell-1} \end{pmatrix}, \tag{5}$$

then one needs to solve the equation $\mathbf{Wa} = \mathbf{b}$ whose solution is $\mathbf{a} = (\mathbf{W}'\mathbf{W})^{-1}\mathbf{W}'\mathbf{b}$.

A similar procedure is performed in [26] to initialize the models of performance of forecasters. The only difference is that i-th row of $\mathbf{W}$ is the result of applying $\mathcal{F}$ to the i-the time series, and w is set to d to be consistent with our notation. By noting that in this later case, each row corresponds to the features of a time series, it is reasonable to investigate whether the rows of W (see Equation (5)) can be treated as independent time series, and, as a consequence, $\mathcal{F}$ is applied to each of these rows. That is, the time series is transformed to $\mathbf{W}$, and, then, each of its rows is treated as a different time series where $\ell = w$.

Although the rows of $\mathbf{W}$ are treated as independent time series, these are not, and inspired on these relation we propose a set of novel time series characteristics. These characteristics measure the change of the time series based on the difference of the time series characteristics. These are defined as follows. Let Δ be the matrix formed by applying $\mathcal{F}$ to each row of $\mathbf{W}$, for example, Δ_{Wang} is created by applying the time series characteristics proposed by Wang *et al.* to $\mathbf{W}$, this lead to a $(\ell - w) \times 13$ matrix. Our novel features are computed from Δ using:

$$\delta_{i,j} = \sum_k \Delta_{i+v-j,k} - \Delta_{i+v,k}, \tag{6}$$

for $i = 1$ to $\ell - w - v$, where v corresponds to the number of characteristics, j goes from 1 to v, and k iterate over the number of columns of Δ. Let us illustrate our novel time series characteristics with an example. Let Δ be a $n \times 13$ matrix, e.g., this matrix could be the one computed with Wang *et al.*'s features. In this case, index i, of $\delta_{i,j}$, iterates from 1 to $n - v$, and j from 1 to v, this result in a $(n - v) \times v$ matrix which correspond to our novel features.

Inspired by the features proposed by Lemke *et al.* [42] to measure the diversity of the algorithms, we proposed a novel set of features. Lemke *et al.* measures the diversity of the algorithms in the training phase of the forecasters. That is, the error between the predicted values and actual values in the training of the forecaster is used to compute the correlation of all the algorithms.

Our novel set of features complements this, by computing the difference of the algorithm at the prediction step. Let $\hat{y}_a$ and $\hat{y}_b$ be the forecast made by algorithm a and b, then we compute the absolute difference of these predictions, i.e., $\gamma = |\hat{y}_a - \hat{y}_b|$. Please note that the measured value y is unavailable to the forecaster, and, consequently, it is not possible to compute the prediction error.

To sum up, we proposed features that complement Lemke's diversity characteristics, and Equation (6) was applied to Wang *et al.* time series characteristics, the first 19 features proposed by Lemke *et al.*, the diversity features proposed by Lemke *et al.*, the indicators described in Equation (1), and the characteristics defined in Equation (3). In total, we used 100 time series characteristics given that $v = 10$, $i = 8$, $h = 9$, and $|\mathcal{S}| = 3$.

3.1. Modeling Forecasters

Now that we have described the different time series characteristics, and the considerations made to include them in the case of a single time series, we are in the position to describe the model. As we mentioned previously, this problem can be tackled with a variety of machine learning techniques such as: ANN, SVM, mixture-of-experts, and linear regression, among others.

One of the simplest approaches, to create a model, is to combine all the features into a linear equation. This has shown success in solving the algorithm selection problem in other domains, e.g., [53, 54, 55, 56, 57, 58], and, we also used it in our previous works [47, 48, 26]. Based on the related work, and our own previous experience, we decided to follow this path.[2] As consequence, the model is defined as:

$$\begin{aligned} P(\mathbf{y}) = \sum_{i=1}^{13} a_i \cdot w_i(\mathbf{y}) + \sum_{i=1}^{27} a_i \cdot l_i(\mathbf{y}) + \sum_{s \in \mathcal{S}} a_s \cdot d_s(\mathbf{y}) + \sum_{h=1}^{n} \sum_{i=1}^{\frac{n}{h}} a_i^h \cdot \varsigma_i^h(\mathbf{y}) \\ + \sum_{j \in \{w,l,\varsigma,\gamma\}} \sum_{i=1}^{10} a_i \cdot \delta_i(\Delta_j) + \sum_{i=1}^{3} a_i \cdot \gamma_i(\mathbf{y}) \end{aligned}, \tag{7}$$

where w is Wang *et al.* features, l represents Lemke *et al.* characteristics; our previous characteristics are d, and ς. Finally, the novel time series features are δ and γ. The a's coefficients are identified using a training set $\mathcal{T}$ and Lars [59]. Note that the fourth and fifth term of Equation (7) are double summations, and the a coefficients have two indexes; these indicate that there is a coefficient for every pair of values (h, i) and (j, i), respectively.

The procedure used to initialize Equation (7) is similar to the one used to solve Equation (4). That is, first matrix $\mathcal{W}$ is defined, here each column corresponds to each different time series

[2] It is worth noting that the use of more complex techniques is worth exploring; however, we decided to leave this as future work.

feature. Then, vector **b** is constructed by measuring the performance of the forecaster being analyzed. That is, the forecaster is trained using the first measurements of the time series (e.g. the first 20 days), and then this forecaster is used to predict a number of points, e.g. the following 20 days, using a particular time horizon. These predicted points are compared against the ground values, and these errors are used as the response variable, i.e., **b**. Now that the system of equations is formed, we used Lars [59] and a cross-validation technique [46] (5-folds) to identify the coefficients a_i.

4. PMSG Wind Turbine

Figure 1 illustrates the general structure of a stand-alone PMSG wind generation system used in this contribution. The system consists of a wind turbine connected to a permanent magnet synchronous machine with the stator windings connected to a converter. The converter is composed by a three-phase diode bridge rectifier, which is controlled to extract maximum power from the prevailing wind. A pulse-width modulation (PWM) inverter ensures the injection of the produced power to the AC load. Between the two converters, a capacitor is used as a DC voltage link. The system is connected to a three-phase load through a LC filter to improve the current quality.

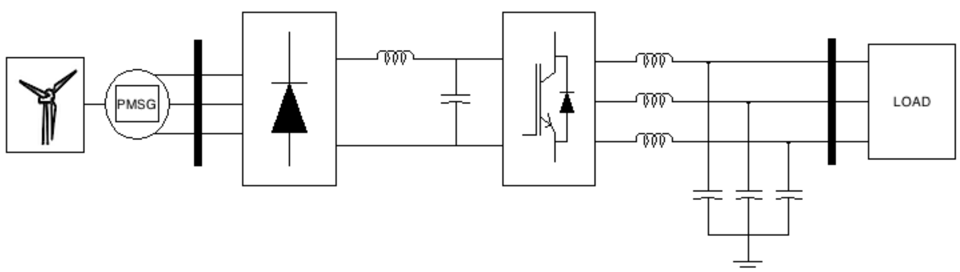

Figure 1: Simulation model components

The proposed model has been modeled and simulated using the Simulink/Matlab software [45], as shown in the Figure 2. The model includes an input for the wind speed sequence obtained in the forecast using the portfolio algorithm, this with the aim of having a more precise estimation of the electrical power that the system is generating at each instant of time. The control algorithms and models used to represent the system can be consulted in the references [60, 61]. The parameters of the system used in this work are given in Appendix A.

5. Results

Forecasting techniques are frequently used in wind generation systems to assess the power generation capacity, and the results obtained are used to make technical and economic decisions,

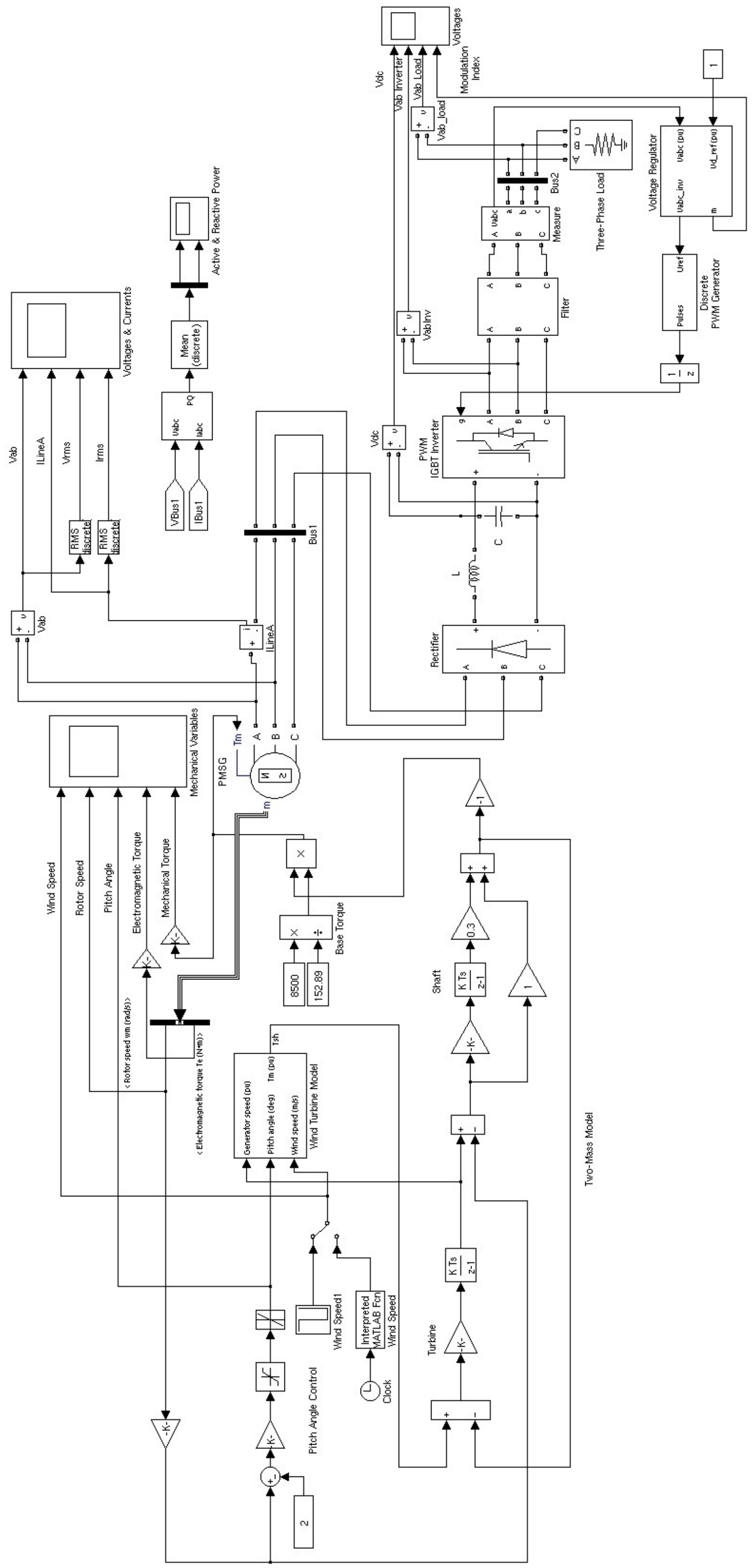

Figure 2: Simulink/Matlab representation of the test system

e.g., transmission system operators are interested in know the production capacity of a wind farm in order to maintain the balance of power transmitted through the network, while in a deregulated system, the wind farm owner is interested in know the production capacity at least 48 hours in advance to raise the necessary strategies to compete in the energy market [62].

In this case study, the time series used correspond to wind speed measurements made by the Comisión Federal de Electricidad (CFE) in the wind generation system, La Venta, located in Oaxaca, México. This time series contains 8759 measurements recorded on an hourly basis at a height of 30 m, in the year of 1999; the measurements can be consulted at the reference [63].

The first step in order to test our approach is to decide which forecasters would form the portfolio. In [26] we create a portfolio of three traditional forecasters namely, ARIMA (see [27, 28]), ETS (see [64, 65, 66]), and BATS (see [29]). Based on our previous experience, in this contribution, we decided to use these three forecasters; however, BATS was not competitive, and, we decided to replace it with an auto-regressive (AR) model (see Equation (4)) with a window of twenty four, i.e., $w = 24$. In summary, our portfolio is composed by ARIMA, ETS and AR. Clearly, one can include in the portfolio different techniques that have been successfully applied to forecast time series such as: ANN, GP, and SVM, among others. However, we decide to use traditional techniques, because these contain fewer parameters (e.g., ARIMA has 3 parameters) than, for example, ANN. In addition to this, there are fully automated procedures to use them, in particular, we used the implementation of the forecast package of R [49]. Furthermore, the computational cost require is lower than the other methodologies mentioned above. Nonetheless, the inclusion of more complex techniques is worth exploring, and we will pursue this avenue of research as future work.

The second step is to create the training set $\mathcal{T}$ and a validation set $\mathcal{V}$. As mentioned previously, $\mathcal{T}$ is used to initialize the portfolio and $\mathcal{V}$ is used to asses its predictions capabilities, $\mathcal{V}$ is also known as out-sample data in other domains. In order to create $\mathcal{T}$ one needs to measure the performance of each forecaster in each problem in $\mathcal{T}$. To do this, one needs to train the forecaster on each problem in $\mathcal{T}$, and, then, one can measure its performance. It is worth mention that this step requires the majority of the computational resources of our procedure; however, given that traditional techniques are being used, this step can be perform in the order of minutes.

In order to test our portfolio, we decided to create 50 different training sets and 50 different validation sets. The idea, behind these values, is to include as many days as possible in the validation set, i.e., to test the generality of our approach, and, also, to have robust statistics, in order to compare the performance of our approach against the traditional forecasters included in the portfolio. The procedure used to create these 50 sets took into consideration that the forecasters need to be initialised, and, also, the coefficients of the model need to be identified. In summary, approximately 76 days were not used in the validation, and, so, for these days, we could not measure the capabilities of our model. Nonetheless, more that 78% of the available data was used to test our modelling technique.

Specifically, the following procedure was performed to create the 50 training and validation sets. The data was split into 61 points; uniformly distributed, starting from the 20-th day[3] and removing the last 23 hours of the year.[4] The i-th point marks the beginning of the i-th set and the point $i+1$ corresponds to the end of that particular set. In total, there are 60 different sets, all of them have in average 137.38 measurements.[5] Let us explain the construction of the i-th training set $\mathcal{T}_i$ and its corresponding validation set $\mathcal{V}_i$. The first step to create these sets is to initialize the forecasters, we decided to use all the measurements up to the $i-th$ point to do this, e.g., when $i=1$ the we used the first 20 days to train the forecasters. Now that the forecaster is trained, we used it to forecast all the measurements until the end of set $i+9$. That is, we used 10 sets. Using this information, we create $\mathcal{T}_i$.[6] Note that all the points below the $(i+10)$-th mark have been used for either training the forecaster or creating $\mathcal{T}$. Consequently, these cannot be used to create $\mathbf{V}_i$. Instead, $\mathbf{V}_i$ is created by all the predicted values in the $i+10$ set; these are computed by training again the forecaster using the actual measurements before the $(i+10)$-th mark. In summary, there are 60 different sets, namely S_i for $i=1$ to 60. Then the i-th pair of training and validation sets is defined by (S_i, S_{i+10}), respectively.

We used different time horizon h to test the portfolio, starting from 1 hour to 24 hours ahead. The procedure used to predict the values is as follow: let t be all the measurements obtained until time t then the value $t+h$ is forecast. To forecast the point $t+h+1$ we used all the information until $t+1$; we continue this process until the end of the set being predicted.

Table 1 shows the performance of the portfolio; the different forecasters of the portfolio, the performance of a traditional selection mechanism, and the performance of a portfolio enhance with a perfect model. Clearly, this last portfolio gets the best performance with these forecasters. The column label Tr. Sel. corresponds to the traditional selection, which is to select the forecaster based on its prediction in a set not seen during the training phase. Specifically, the performance of the forecasters is assessed with the data used to initialize the portfolio. That is, the forecaster used to predict set $\mathcal{V}_i$ is the one with the best performance in the training set. The performance is the mean absolute error, with the difference that all the data is in the range $(0,1)$. The parameters of this normalization are obtained using actual measurements.

From Table 1, it can be observed in bold the forecaster with best performance, in this com-

[3]These 20 first days are used to train the forecasters for the first pair of training and validation. Note that the AR model has 24 degrees of freedom, and, with these days, we have at least 20 measurements for each degree, so we are confident that this is a well-posed problem.

[4]We decided to remove these last hours, so that each day has 24 measurements.

[5]Clearly, some blocks contains 137 values an others 138.

[6]In average $|\mathcal{T}_i|$ has 1373 elements, we included that many elements in the training set to be confident that the identification of the model's coefficients is a well-posed problem, given that the model has 100 degrees of freedom.

parison we did not include the portfolio with the perfect model. The first column indicates the time horizon starting from one hour to twenty-four hours. As it can be seen, our portfolio has the best performance in almost all the cases, the exception is on one hour ahead. Furthermore, the table presents the p values of Student's t-test, it can be observed that our portfolio is statistically better in 17 cases with a confidence of 95 %; in two more cases with a confidence of 90 % and in the rest five time horizons is statistically equivalent, this includes the one hour ahead study. Please observe that our portfolio is better with a confidence of 99 % in all time horizons above 11 hours, i.e., $h = 12, \ldots, 24$.

The last column (label Gain) indicates the percentage of improvement of the portfolio. The best improvement is for $h = 17$ and around this time horizon is where our portfolio gets the best scores. On average, the improvement of the portfolio is 6.21 %. In order to compute the gain, we analyzed the performance of this procedure. Let us define an improvement of 100% to the portfolio that has the performance of the perfect model (column Perfect), and assign a 0% to the portfolio that has the same performance as the algorithm with best average performance; we include here the performance of the traditional selection. For example, for $h = 22$ the forecaster with best average performance is the traditional selection with 0.1265.

Regarding the forecasters, it can be observed that AR obtained the best average performance from one hour ahead to seventeen hours ahead; and ARIMA has the best average performance in the rest of the time horizons, i.e., $h = 18, \ldots, 24$. Comparing the traditional selection (Tr. Sel.) with the forecasters, it is observed that this selection technique has the best performance from 16 hours ahead to 22 hours ahead.

So far we have compared our portfolio against the forecasters composing the portfolio and a traditional selection technique. However, we have proposed a number of novel time series characteristics and we have not provided any insight on whether these features are useful. The results of comparing the performance of the portfolio with these novel features against the portfolio without them show that the portfolio with all the features has the best performance in all the cases, except when $h = 8$; however, in this case the p-value is 0.95. Furthermore, in 9 time horizons the p values are higher than 0.05, indicating similarity in performance. For the rest of the time horizons (15 cases) the portfolio shown in Table 1 has the best performance. Clearly, these results suggest that our features are improving the performance of the portfolio, and, in fact, based on the performance of the portfolio, these novel features are statistically significant.

The information presented on Table 1 allow us to compare our technique against the forecasters composing the portfolio, and a traditional selection technique. However, one of the aims of this work is to know whether the predictions made by the portfolio can be used to forecast the generated power. We decided to forecast the power generated from day 249 to day 306. Figure 3 shows the power generated obtained with the forecast made by the portfolio and the real power generated.

Table 1: Performance (in terms of the normalize mean absolute error) on different time horizons (h) of the portfolio, the different forecasters, and two selection techniques which are selecting with a perfect model and using the fitness in a validation set (label as Tr. Sel.). The table presents the p value of the Student's t-test, and the last row (Gain) shows the percentage of improvement of the portfolio.

h	Portfolio	Tr. Sel.	AR	ARIMA	ETS	Perfect	p-value	Gain
1	0.0367	0.0367	**0.0366**	0.0370	0.0379	0.0319	0.1328	-2.15 %
2	**0.0547**	0.0553	0.0552	0.0561	0.0588	0.0468	0.0008	5.91 %
3	**0.0673**	0.0682	0.0678	0.0693	0.0744	0.0561	0.0312	4.02 %
4	**0.0771**	0.0783	0.0778	0.0798	0.0875	0.0632	0.0239	4.45 %
5	**0.0860**	0.0869	0.0863	0.0889	0.0988	0.0690	0.3979	1.66 %
6	**0.0924**	0.0934	0.0931	0.0960	0.1081	0.0733	0.0822	3.44 %
7	**0.0983**	0.0990	0.0989	0.1020	0.1157	0.0766	0.1813	2.49 %
8	**0.1031**	0.1038	0.1032	0.1068	0.1220	0.0790	0.8380	0.40 %
9	**0.1062**	0.1072	0.1066	0.1105	0.1268	0.0801	0.4351	1.53 %
10	**0.1083**	0.1101	0.1092	0.1135	0.1306	0.0811	0.0915	3.27 %
11	**0.1104**	0.1124	0.1115	0.1165	0.1344	0.0821	0.0457	3.90 %
12	**0.1121**	0.1147	0.1138	0.1191	0.1380	0.0828	0.0062	5.31 %
13	**0.1146**	0.1175	0.1165	0.1216	0.1413	0.0838	0.0033	5.80 %
14	**0.1166**	0.1196	0.1190	0.1238	0.1444	0.0847	0.0003	7.16 %
15	**0.1178**	0.1220	0.1215	0.1256	0.1465	0.0856	0.0000	10.40 %
16	**0.1188**	0.1237	0.1241	0.1270	0.1482	0.0863	0.0000	12.90 %
17	**0.1203**	0.1254	0.1263	0.1279	0.1487	0.0865	0.0000	12.99 %
18	**0.1213**	0.1253	0.1283	0.1282	0.1486	0.0863	0.0000	10.15 %
19	**0.1220**	0.1263	0.1305	0.1279	0.1479	0.0861	0.0000	10.87 %
20	**0.1219**	0.1262	0.1325	0.1274	0.1461	0.0853	0.0000	10.32 %
21	**0.1219**	0.1264	0.1347	0.1270	0.1445	0.0851	0.0000	10.80 %
22	**0.1233**	0.1265	0.1375	0.1267	0.1432	0.0847	0.0000	7.52 %
23	**0.1240**	0.1280	0.1406	0.1270	0.1423	0.0844	0.0003	7.19 %
24	**0.1239**	0.1289	0.1437	0.1277	0.1424	0.0840	0.0000	8.81 %
Average	0.1041	0.1067	0.1090	0.1089	0.1240	0.0769	0.0946	6.21 %

The wind speed forecast corresponds to a time horizon of six hours. It can be observed that the predicted generated power closely follows the generated power. Figure 4 complements this figure by showing the relative error between the predicted power and the actual measurements. The relative error is a number that compares how incorrect a quantity is from a number considered to be true, that is, this concept help us to determine how close the predicted data are in comparison to the measured data. The relative error is defined as [67]:

$$Relative\ error(\%) = \left| \frac{Measurement\ value - Predicted\ value}{Measurement\ value} \right| \cdot 100 \tag{8}$$

It can be observed that for most of the period, the relative error is well below 10 %. However, there is a period around the values 600 and 800 where the relative error is very high, reaching relative errors close to 80 %. This increment in the relative error is because in that period the wind speed value is close to zero, and the obtained forecast has a different value.

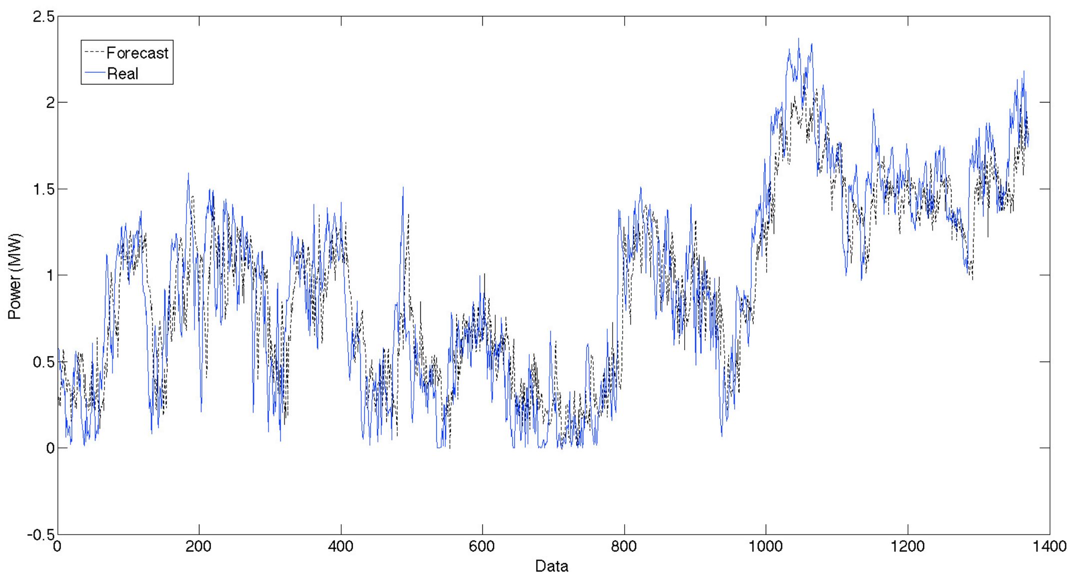

Figure 3: Electric power generated with the predictions made by the portfolio and the actual power generated.

6. Conclusions

A portfolio of forecasters for a single time series has been presented in this contribution. This portfolio is built by extending previous time series characteristics, specifically, the features proposed by Wang *et al.*, Lemke *et al.* and our owns features. In addition to this, we have proposed a number of novel time series characteristics. These novel features complement the previous ones and together create a better portfolio.

The performance of the portfolio has been compared against the performance of the forecasters composing it. The results have shown that our portfolio is statistically better in 19 of the 24 different time horizons tested. Furthermore, it only has a lower performance for the one step ahead forecast. It has been shown that on average our portfolio has an improvement of 6.21 % and the best improvement is presented when $h = 17$.

For completeness, we have also tested the practical application of the wind speed related with the field of electrical engineering. That is, we have performed simulations to illustrate the amount

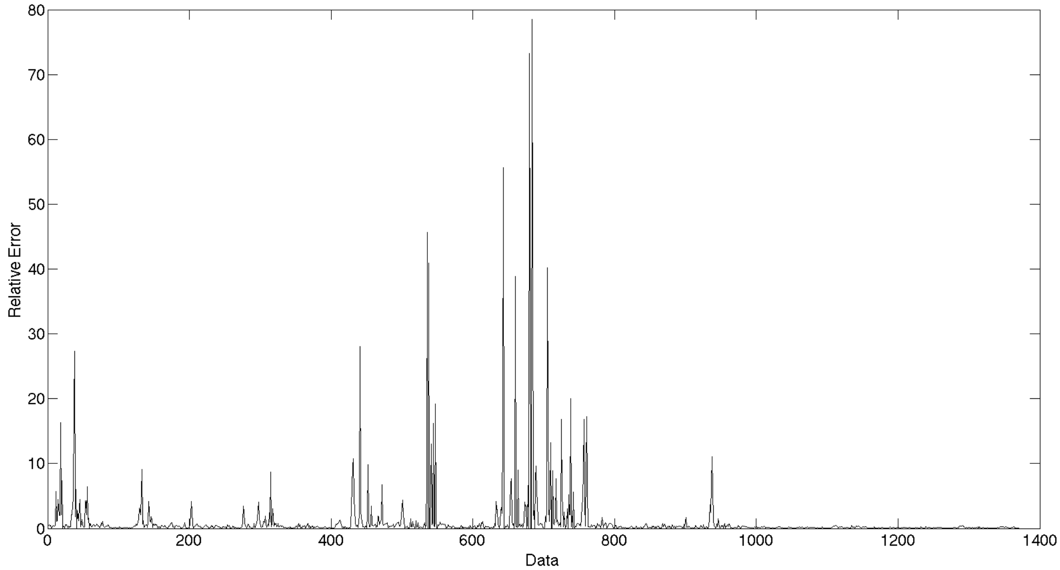

Figure 4: Relative Error between the predicted power generated and the actual power generated

of power generated by the forecast wind and the actual wind using a stand-alone PMSG wind generation system implemented in Simulink/Matlab. The simulations have shown that most of the time, the amount of power obtained using measurements and the forecast data are very similar. This indicates that our approach can be used to predict the amount of energy that a wind farm will produce in a short period of time (we presented the results for six hour).

Appendix A. Test System Parameters

The parameters of a PMSG wind turbine used in this contribution are the following [68]:

Table 2: Wind Turbine Parameters

Parameter	Value
Air density	1.205 kg/m^3
Rotor radius	38 m
Rated wind speed	11.8 m/s
Maximum power coefficient	0.4412

Table 3: Drive Train Parameters

Parameter	Value
Gear ratio	1
Rotational damping coefficient	0
Equivalent inertia (turbine and generator)	0.3 kg m^3

Table 4: Generator Parameters

Parameter	Value
Rated generated power	2 MW
Rated mechanical speed	2.18 rad/s
Stator resistance	0.08
Stator d-axis inductance	0.334 H
Stator q-axis inductance	0.217 H
Stator leakage inductance	0.0334 H
Permanent magnet flux	0.4832 Wb
Pole pairs	3

Acknowledgements

The authors want to acknowledge the Universidad Michoacana de San Nicolas de Hidalgo (UMSNH) through the Division de Estudios de Posgrado of the Facultad de Ingeniería Eléctrica, and the Facultad de Ingeniería of the Universidad Autónoma de San Luis Potosí (UASLP) for the facilities granted to carry-out this investigation. Rafael Peña acknowledges the financial support granted by the FAI (Fondo de Apoyo a la Investigación) of the UASLP to finance this research.

[1] Aoife M. Foley, Paul G. Leahy, Antonino Marvuglia, and Eamon J. McKeogh. Current methods and advances in forecasting of wind power generation. *Renewable Energy*, 37(1):1–8, January 2012.

[2] Heiko Hahn, Silja Meyer-Nieberg, and Stefan Pickl. Electric load forecasting methods: Tools for decision making. *European Journal of Operational Research*, 199(3):902–907, December 2009.

[3] O. A. Jaramillo and M. A. Borja. Wind speed analysis in La Ventosa Mexico: a bimodal probability distribution case. *Renewable Energy*, 29:1613–1630, 2004.

[4] Erasmo Cadenas and Wilfrido Rivera. Wind speed forecasting in the south coast of Oaxaca, Mexico. *Renewable Energy*, 32(12):2116 − 2128, 2007.

[5] Juan J. Flores, Roberto Loaeza, Hector Rodriguez, and Erasmo Cadenas. Wind speed forecasting using a hybrid neural-evolutive approach. In Arturo Hernández Aguirre, Raúl Monroy Borja, and Carlos A. Reyes García, editors, *MICAI*, volume 5845 of *Lecture Notes in Computer Science*, pages 600–609. Springer, 2009.

[6] Juan J. Flores, Mario Graff, and Hector Rodriguez. Evolutive design of ARMA and ANN models for time series forecasting. *Renewable Energy*, 44(0), 2011.

[7] Zhenhai Guo, Weigang Zhao, Haiyan Lu, and Jianzhou Wang. Multi-step forecasting for wind speed using a modified EMD-based artificial neural network model. *Renewable Energy*, 37(1):241–249, January 2012.

[8] Hui Liu, Chao Chen, Hong-qi Tian, and Yan-fei Li. A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural networks. *Renewable Energy*, 48:545–556, December 2012. 00006.

[9] Da Liu, Dongxiao Niu, Hui Wang, and Leilei Fan. Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm. *Renewable Energy*, 62:592–597, February 2014. 00000.

[10] Yu Jiang, Zhe Song, and Andrew Kusiak. Very short-term wind speed forecasting with bayesian structural break model. *Renewable Energy*, 50:637–647, February 2013. 00008.

[11] Ekaterina Vladislavleva, Tobias Friedrich, Frank Neumann, and Markus Wagner. Predicting the energy output of wind farms based on weather data: Important variables and their correlation. *Renewable Energy*, 50:236–243, February 2013. 00004.

[12] Juan J. Flores, Mario Graff, and Erasmo Cadena. Wind prediction using genetic programming and gene expression programming. In *Techniques and Methodologies for Modelling and Simulation of Systems, Lyon France - Mexico, (AMSE), ISBN:970-703-323-1*, pages 34 − 40. AMSE, International Association for Advanced of Modelling and Simulation, 2005.

[13] Heiko Hahn, Silja Meyer-Nieberg, and Stefan Pickl. Electric load forecasting methods: Tools for decision making. *European Journal of Operational Research*, 199(3):902–907, December 2009.

[14] Y.V. Makarov, P.V. Etingov, Z. Huang, J. Ma, B.B. Chakrabarti, K. Subbarao, C. Loutan, and R.T. Guttromson. Integration of wind generation and load forecast uncertainties into power grid operations. In *Transmission and Distribution Conference and Exposition, 2010 IEEE PES*, pages 1–8, April 2010.

[15] Lu Fang-yuan and Gao Feng-you. Economic forecasting research based on artificial neural network technology. In *Control and Decision Conference, 2008. CCDC 2008. Chinese*, pages 1151–1155, July 2008.

[16] Rafael Peña, Aurelio Medina, Olimpo Anaya-Lara, and James R. McDonald. Capacity estimation of a minihydro plant based on time series forecasting. *Renewable Energy*, 34(5):1204–1209, May 2009.

[17] D. Q. Truong and K. K. Ahn. Wave prediction based on a modified grey model MGM(1,1) for real-time control of wave energy converters in irregular waves. *Renewable Energy*, 43:242–255, July 2012. 00004.

[18] J. Ross Halliday, David G. Dorrell, and Alan R. Wood. An application of the fast fourier transform to the short-term prediction of sea wave behaviour. *Renewable Energy*, 36(6):1685–1692, June 2011.

[19] A. Paniagua-Tineo, S. Salcedo-Sanz, C. Casanova-Mateo, E.G. Ortiz-Garca, M.A. Cony, and E. Hernndez-Martn. Prediction of daily maximum temperature using a support vector regression algorithm. *Renewable Energy*, 36(11):3054–3060, November 2011.

[20] Christophe Paoli, Cyril Voyant, Marc Muselli, and Marie-Laure Nivet. Forecasting of preprocessed daily solar radiation time series using neural networks. *Solar Energy*, 84(12):2146–2160, December 2010.

[21] Ji Wu, Chee Keong Chan, Yu Zhang, Bin Yu Xiong, and Qing Hai Zhang. Prediction of solar radiation with genetic approach combing multi-model framework. *Renewable Energy*, 66:132–139, June 2014. 00000.

[22] Dazhi Yang, Chaojun Gu, Zibo Dong, Panida Jirutitijaroen, Nan Chen, and Wilfred M. Walsh. Solar irradiance forecasting using spatial-temporal covariance structures and time-forward kriging. *Renewable Energy*, 60:235–245, December 2013. 00003.

[23] S. X. Chen, H. B. Gooi, and M. Q. Wang. Solar radiation forecast based on fuzzy logic and neural networks. *Renewable Energy*, 60:195–201, December 2013. 00002.

[24] R. Yacef, M. Benghanem, and A. Mellit. Prediction of daily global solar irradiation data using bayesian neural network: A comparative study. *Renewable Energy*, 48:146–154, December 2012. 00006.

[25] John R. Rice. The algorithm selection problem. *Advances in Computers*, 15:65–118, 1976.

[26] Mario Graff, Hugo Jair Escalante, Jaime Cerda-Jacobo, and Alberto Avalos Gonzalez. Models of performance of time series forecasters. *Neurocomputing*, 122:375–385, December 2013. 00001.

[27] R. Shanmugam. Introduction to time series and forecasting. *Technometrics*, 39(4):426426, 1997.

[28] J. Durbin, S. J. Koopman, and A. C. Atkinson. *Time series analysis by state space methods*, volume 15. Oxford University Press Oxford, 2001.

[29] Alysha M. De Livera, Rob J. Hyndman, and Ralph D. Snyder. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496):1513–1527, December 2011.

[30] Fred Collopy and J. Scott Armstrong. Rule-based forecasting: Development and validation of an expert systems approach to combining time series extrapolations. *Management Science*, 38(10):1394–1414, October 1992.

[31] Charles C. Holt, Franco Modigliani, John F. Muth, Herbert A. Simon, Charles P. Bonini, and Peter R. Winters. *Planning Production, Inventories, and Work Force.* Prentice Hall, 1960.

[32] Robert Goodell Brown. *Statistical forecasting for inventory control.* McGraw-Hill, 1959.

[33] Monica Adya, J.Scott Armstrong, Fred Collopy, and Miles Kennedy. An application of rule-based forecasting to a situation lacking domain knowledge. *International Journal of Forecasting*, 16(4):477–484, October 2000.

[34] Monica Adya, Fred Collopy, J.Scott Armstrong, and Miles Kennedy. Automatic identification of time series features for rule-based forecasting. *International Journal of Forecasting*, 17(2):143–157, April 2001.

[35] B Arinze. Selecting appropriate forecasting models using rule induction. *Omega*, 22(6):647–658, November 1994.

[36] Bay Arinze, Seung-Lae Kim, and Murugan Anandarajan. Combining and selecting forecasting models using rule based induction. *Computers and Operations Research*, 24(5):423–433, May 1997.

[37] Chandra Shah. Model selection in univariate time series forecasting using discriminant analysis. *International Journal of Forecasting*, 13(4):489–500, December 1997.

[38] Kate A. Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Comput. Surv.*, 41(1):1–25, 2008.

[39] N. Meade. Evidence for the selection of forecasting methods. *Journal of forecasting*, 19(6):515535, 2000.

[40] Ricardo B.C. Prudencio and Teresa B. Ludermir. Meta-learning approaches to selecting time series models. *Neurocomputing*, 61(0):121–137, October 2004.

[41] Xiaozhe Wang, Kate Smith-Miles, and Rob Hyndman. Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series. *Neurocomputing*, 72(10-12):2581–2594, June 2009.

[42] Christiane Lemke and Bogdan Gabrys. Meta-learning for time series forecasting and forecast combination. *Neurocomputing*, 73(1012):2006–2016, June 2010.

[43] Aarti Gupta, DK Jain, and Surender Dahiya. *Some investigations on recent advances in wind energy conversion systems*. IPCSIT Press, Singapore, 2012.

[44] P. Brndsted and R. Nijssen. *Advances in Wind Turbine Blade Design and Materials*. Elsevier, October 2013. 00000.

[45] Hydro-Québec TransÉnergie Technologies. Matlab user's guide, version 3: Simpowersystems. for use with simulink. http://www.mathworks.com/help/releases/R13sp2/pdf_doc/physmod/powersys/powersys.pdf.

[46] Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, Cambridge, MA, USA, 2004.

[47] Mario Graff and Riccardo Poli. Practical performance models of algorithms in evolutionary program induction and other domains. *Artificial Intelligence*, 174(15):1254–1276, October 2010.

[48] Mario Graff, Riccardo Poli, and Juan J. Flores. Models of performance of evolutionary program induction algorithms based on indicators of problem difficulty. *Evolutionary Computation*, 21(4):533–560, November 2012. 00002.

[49] Rob J. Hyndman and Yeasmin Khandakar. Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, 27(3):1–22, 2008.

[50] G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, page 211252, 1964.

[51] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*. The MIT Press, 1992.

[52] Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee. *A field guide to genetic programming*. Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008. (With contributions by J. R. Koza).

[53] Eugene Nudelman, Kevin Leyton-Brown, Holger H. Hoos, Alex Devkar, and Yoav Shoham. Understanding random SAT: Beyond the clauses-to-variables ratio. In Mark Wallace, editor, *CP*, volume 3258 of *Lecture Notes in Computer Science*, pages 438–452. Springer, 2004.

[54] Kevin Leyton-Brown, Eugene Nudelman, and Yoav Shoham. Empirical hardness models: Methodology and a case study on combinatorial auctions. *J. ACM*, 56(4), 2009.

[55] Kevin Leyton-Brown, Eugene Nudelman, and Yoav Shoham. Empirical hardness models for combinatorial auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 19, pages 479–504. MIT Press, 2006.

[56] Kevin Leyton-Brown, Eugene Nudelman, and Yoav Shoham. Learning the empirical hardness of optimization problems: The case of combinatorial auctions. In Pascal Van Hentenryck, editor, *CP*, volume 2470 of *Lecture Notes in Computer Science*, pages 556–572. Springer, 2002.

[57] Kevin Leyton-Brown, Eugene Nudelman, Galen Andrew, Jim McFadden, and Yoav Shoham. A portfolio approach to algorithm selection. In Georg Gottlob and Toby Walsh, editors, *IJCAI*, pages 1542–1542. Morgan Kaufmann, 2003.

[58] Kevin Leyton-Brown, Eugene Nudelman, Galen Andrew, Jim McFadden, and Yoav Shoham. Boosting as a metaphor for algorithm design. In Francesca Rossi, editor, *CP*, volume 2833 of *Lecture Notes in Computer Science*, pages 899–903. Springer, 2003.

[59] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[60] Mahmoud Hussein, Tomonobu Senjyu, Mohamed Orabi, Mohamed Wahab, and Mohamed Hamada. Control of a stand-alone variable speed wind energy supply system . *Applied Sciences*, 3(2):437–456, April 2013.

[61] C.N. Bhende, S. Mishra, and S.G. Malla. Permanent magnet synchronous generator-based standalone wind energy supply system. *IEEE Transactions on Sustainable Energy*, 2(4):361–373, 2011.

[62] G.N. Bathurst, J. Weatherill, and G. Strbac. Trading wind generation in short term energy markets. *IEEE Transactions on Power Systems*, 17(3):782–789, August 2002.

[63] México Comisión Federal de Electricidad. Wind speed of La Venta Oaxaca México. http://dep.fie.umich.mx/∼mgraffg/WindSpeedVentaOaxaca.zip.

[64] Rob J Hyndman, Anne B Koehler, Ralph D Snyder, and Simone Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454, July 2002.

[65] Rob Hyndman, Muhammad Akram, and Blyth Archibald. The admissible parameter space for exponential smoothing models. *Annals of the Institute of Statistical Mathematics*, 60(2):407–426, 2008.

[66] Rob Hyndman, Anne B. Koehler, J. Keith Ord, and Ralph D. Snyder. *Forecasting with Exponential Smoothing: The State Space Approach*. Springer, August 2008.

[67] Yanlin Jiang, Alina A. von Davier, and Haiwen Chen. Evaluating equating results: Percent relative error for chained kernel equating. *Journal of Educational Measurement*, 49(1):39–58, 2012.

[68] A. Rolan, A. Luna, G. Vazquez, D. Aguilar, and G. Azevedo. Modeling of a variable speed wind turbine with a permanent magnet synchronous generator. In *IEEE International Symposium on Industrial Electronics, 2009. ISIE 2009*, pages 734–739, 2009.