



Soft Computing Methods with Phase Space Reconstruction for Wind Speed Forecasting—A Performance Comparison

Juan. J. Flores ^{1,*} ^(D), José R. Cedeño González ¹, Héctor Rodríguez ² ^(D), Mario Graff ³ ^(D), Rodrigo Lopez-Farias ⁴ ^(D) and Felix Calderon ¹ ^(D)

- ¹ Division de Estudios de Posgrado, Facultad de Ingenieria Electrica, Universidad Michoacana de San Nicolas de Hidalgo, Avenida Francisco J. Mugica S/N, Ciudad Universitaria, Morelia 58030, Mexico; jose.cedeno@umich.mx (J.R.C.G.); calderon@umich.mx (F.C.)
- ² Division de Estudios de Posgrado e Investigacion, Tecnologico Nacional de Mexico campus Culiacan, Juan de Dios Batiz 310 pte, Culiacan 80220, Mexico; hrodriguez@itculiacan.edu.mx
- ³ CONACYT—INFOTEC Centro de Investigacion e Innovacion en Tecnologias de la Informacion y Comunicacion, Circuito Tecnopolo Sur No 112, Fracc. Tecnopolo Pocitos II, Aguascalientes 20313, Mexico; mario.graff@infotec.mx
- ⁴ CONACYT—CentroGeo, Centro de Investigación en Ciencias de Información Geoespacial, Contoy 137, Col. Lomas de Padierna, Delegación Tlalpan, CDMX 14240, Mexico; rlopez@centrogeo.edu.mx
- * Correspondence: juanf@umich.mx; Tel.: +52-443-107-7834

Received: 15 July 2019; Accepted: 6 September 2019; Published: 16 September 2019



Abstract: This article presents a comparison of wind speed forecasting techniques, starting with the Auto-regressive Integrated Moving Average, followed by Artificial Intelligence-based techniques. The objective of this article is to compare these methods and provide readers with an idea of what method(s) to apply to solve their forecasting needs. The Artificial Intelligence-based techniques included in the comparison are Nearest Neighbors (the original method, and a version tuned by Differential Evolution), Fuzzy Forecasting, Artificial Neural Networks (designed and tuned by Genetic Algorithms), and Genetic Programming. These techniques were tested against twenty wind speed time series, obtained from Russian and Mexican weather stations, predicting the wind speed for 10 days, one day at a time. The results show that Nearest Neighbors using Differential Evolution outperforms the other methods. An idea this article delivers to the reader is: what part of the history of the time series to use as input to a forecaster? This question is answered by the reconstruction of phase space. Reconstruction methods approximate the phase space from the available data, yielding *m* (the system's dimension) and τ (the sub-sampling constant), which can be used to determine the input for the different forecasting methods.

Keywords: time series forecasting; wind speed forecasting; machine learning

1. Introduction

The world's population growth has taken the planet to unsustainable levels of pollution, mainly caused by the industrialization of the developing world. To solve this problem we need technological changes; aiming to solve this problem, humans have developed alternative ways of producing electrical and mechanical power, to be used in the industry. One direction where these sort of policies can be applied is to alternative sources of electrical energy, ones that limit carbon emissions [1].

In this direction, one of the most important renewable, less polluting, sources of electrical energy is wind energy. One of the problems in using this energy is its incorporation into the electrical network, due to the intermittency [2,3] and the difficult controllability of the main power source, turbulent

wind [4]. On the other hand, the amount of energy produced by wind power is a function of the wind speed. The challenge is to integrate this intermittent power source into the electricity grid.

We find two main streams of work in the literature regarding wind speed forecasting. One is the meteorology (or physical) approach to model and predict the state of the atmosphere [5–7]. This approach is based on the spatial (i.e., geographical) distribution of the wind features. Those models take into consideration factors like shelter from obstacles, local surface roughness, effects of orography, wind speed change and scaling, etc. (see [8,9]). Those works, mainly based on partial differential equations (PDE), were developed since the early 50's; J. Charney called those PDE Primitive Equations [10].

The other main stream of work is based on what has been defined as Time Series Analysis. This stream of work was initiated by Yule [11], Slutsky [12], and Wold [13], and made popular by Box and Jenkins [14]. A time series is a series of data points (commonly expressing the magnitude of a scalar variable) indexed in time order. A time series is a discrete sequence taken at successive equally spaced points in time.

Within the area of Time Series Analysis, there is plenty of work based on statistics, and more recently, based on artificial intelligence and machine learning techniques. Techniques like Artificial Neural Networks [15–18], Support Vector Machines [19–22], Nearest Neighbors [23,24], Fuzzy Systems [25–28], and recently Deep Learning [29,30], have been used to model and forecast Time Series.

The study of probabilistic forecasting methods is also an important research of interest that would allow to implement wind forecasting methods in real energy production market scenarios [31]. It could provide useful information to decision-making to maximize the rentability of wind power production, e.g., information about the future error probability distribution to know the risk of bidding certain power output. Zhang et al. [31] classifies the probabilistic approaches in three categories according to the uncertainty representation: probabilistic (e.g., [4,32,33]), risk index (e.g., [34,35]), and space time scenario forecasts [36–38].

One contribution of this article is to present a performance comparison of Time Series forecasting techniques using Statistical and Artificial Intelligence methods. The methods included in this study were tested and their performances compared using two data sets, 10 Russian time series and 10 data sets from a network of weather stations located in the state of Michoacan, Mexico. The forecasting horizon used in the experiments was to predict the wind speed one day ahead for 10 days, using those time series. Note that methods that won with this data sets under this forecasting scenario may not be the winners using a different data set or forecasting horizon.

We compare one of the simplest non-linear forecasting models (NN—Nearest Neighbors), with its equivalent statistical counterpart such as ARIMA (AutoRegressive Integrated Moving Average). Based on experimental results, we show that the simple Non-linear model outperforms the statistical model ARIMA when we compare forecast accuracy for non-linear time series. Taking NN as the base model, we report the accuracy of a set of NN based models composed by a deterministic version of NN. We present a non-deterministic version (NNDE—Nearest Neighbors using Differential Evolution), that uses the Differential Evolution Algorithm to find the optimum time lag, τ , embedding dimension, m, and neighborhood radius size, ϵ , that produce the minimum forecasting error. Fuzzy Forecast (FF), which can be considered as a kind of Fuzzy NN, produces forecasts using fuzzy rules applied to the delay vectors. ANN-cGA (Artificial Neural Network using Compact Genetic Algorithms) evolves the architecture of ANN's to minimize the forecasting error. As part of the evolved architecture, it determines the relevant inputs to the neural network forecaster. The input selection is closely related to the phase space reconstruction process. Finally, Evolving Directed Acyclic Graph (EvoDAG) evolves forecasting functions that take as arguments part of the history of the time series.

A second contribution of this article is the use of phase space reconstruction to determine the important inputs to the forecasting models. Phase space reconstruction determines the sub-sampling interval or time delay, τ , and the system dimension, *m*, using mutual information and false nearest

neighbors, respectively; these parameters are used to produce the characteristic vectors known as delay vectors. These delay vectors are the input to the different models presented in this article.

When reviewing the literature, there are differences in the ranges of the time scale between different works. For example, Soman et al. [39] use the term very short term to denote forecasting for a few seconds to 30 min in advance, short-term for 30 min up to 6 h ahead, medium-term for 6 h to one day ahead, and long-term from 1 day to a week ahead. According to their classification, this article addresses medium- and long-term forecasting; applications for these time scales, according to Soman et al., include generator online/offline decisions, operational security in day-ahead electricity market, unit commitment decisions, reserve requirements decisions, and maintenance and optimal operational cost, among others. They also present a study of different methods applied to different time scales; according to their work they conclude that statistical approaches and hybrid methods are "very useful and accurate" for medium- and long-term forecast. Therefore a performance comparison between AI methods for this particular forecast horizon offers guidance for the practitioners of the field.

The rest of the article is organized as follows: Section 2 provides a brief analysis of the state of the art in time series analysis using the methods included in this study. Section 3.2, describes the forecasting techniques used in the comparative analysis presented here. Section 4 describes the data sets used in the comparative analysis, the experimental setup, and the results of the comparison. Finally, Section 5 draws this empirical study conclusions.

2. Related Work

The study and understanding of wind speed dynamics for prediction purposes affects the performance of wind power power generation. Wind speed dynamics present a high level of potential harmful uncertainty to the efficiency of energy dispatch and management. Therefore, one of the biggest goals in wind speed forecasting is to reduce and manage the uncertainty with accurate models with the aim of increasing added value to the wind power generation.

Nevertheless, wind speed prediction is difficult to perform accurately and requires more than the traditional linear correlation-based models (i.e., Auto-Regressive Integrating Moving Average). Wind speed dynamics presents both strong chaotic and random components [40], which must be modeled and explained from the non-linear dynamics perspective. We found in the literature a diverse collection of wind power and speed forecasting models to meet the requirements predicting at specific time horizons.

The work of Wang et al. [2] is a comprehensive survey that organizes the forecasting models according to prediction horizons, e.g., immediate short time (8-h ahead), short term (one-day ahead), and long term (greater than one-day ahead).

L. Bramer states [41] that for short term forecasting horizons (from 1 to 3 h ahead), statistical models are more suitable than other kinds of models. In contrast, for longer horizons, the alternative methods perform better than the pure statistical models. Our forecasting horizon (one day ahead) calls for other methods, capable to produce forecasts deeper than that into the future. That is the reason to use AI-based techniques.

Okumus and Dinler [42] present a comprehensive review where they cite an important number of wind power and speed forecasting models, compared by standard error measures and length of the prediction horizon. The paper compares ANFIS, ANN, and ANFIS+ANN model, presenting better performance improving by 5% on average. The hybrid model presents a MAPE improvement of 25% for 24-h step ahead forecasts. Table 1 of Okumus et al.'s article provides errors obtained in the works included in their survey; unfortunately, those errors are reported for different forecasting tasks, using different error metrics. Under those conditions we cannot really compare the performance of the methods we present in this article with those presented in the articles included in the survey. It is not even clear to us the main characteristics of the data they use (e.g., sampling period, time series length, etc.); we do not know how complex the data is in those examples and forecasting accuracy depends on those factors. Another multivariate model is proposed by Cadenas et al. [43] for one step ahead forecasting, using Nonlinear Auto-regressive Exogenous Artificial Neural Networks (NARX). The NARX model considers meteorological variables such as wind speed, wind direction, solar radiation, atmospheric pressure, air temperature, and relative humidity. The model is compared against Auto-regressive Integrated Moving Average (ARIMA) model; NARX reports a precision improvement between 5.5% and 10%, and 12.8% for one-hour and ten-minute sample period time series.

Croonenbroeck and Ambach [44] compare and analyze the effectiveness of WPPT (Wind Power Prediction Tool) with GWPPT (Generalized Wind Power Prediction Tool) using wind power data of 4 wind turbines located in Denmark. WPPT is also compared with Non-parametric Kernel Regression, Mycielsky's Algorithm [45], Auto-regressive (AR), and Vector Auto-regressive (VAR). Their experiments were performed with one and 72-step (12 h) ahead forecasting horizons with a sampling period of 10 min.

Jiang et al. [46], present a one-day ahead wind power forecasting using a hybrid method based on the combination of the enhanced boosting algorithm and ARMA (ARMA-MS). An ARMA model is selected with parameters p = 1 and q = 1 for the AR and MA components, respectively. The ARMA-MS algorithm is based on the weighed combination of several ARMA forecasters. ARMA-MS is tested with wind power data from the east coast of the Jiangsu Province.

Despite the plethora of algorithms and models to forecast wind speed, the authors found few articles using non-linear time series theory [47,48]. Non-linear time series theory is useful to identify the structure and attractors, and predict time series with non-linear and potential chaotic behavior. An example is the Simple Non-Linear Forecasting Model based on Nearest Neighbors (NN) proposed by H. Kantz [23].

Previous experimental results with synthetic chaotic time series [49], indicate that using basic principles to reconstruct the non-linear time series in phase space, perform far better predictions than the basic statistic principles. They show that the prediction quality can be improved even more by using Differential Evolution.

3. Materials and Methods

This section describes the data sets used to test the forecasting methods presented in this article. Several forecasting techniques (developed by the authors) were selected to solve the wind speed forecasting problem. Those forecasting methods are then presented in the second subsection. Section 4 presents a comparative analysis of those forecasting techniques; the study was designed to be as exhaustive as possible.

3.1. Data Sets

The methods included in this study were tested and their performances compared using two data sets: a selection of 10 stations from the compilation "Six- and Three-Hourly Meteorological Observations from 223 Former U.S.S.R. Stations (NDP-048)" [50], and 10 are a subset of the network of weather stations located in the state of Michoacan, Mexico [51]. Every data set contains wind speed measurements collected at a height of 10 m. Russian time series are sampled at 3-h periods and Mexican time series are sampled every hour.

The forecasting horizon used in the experiments was to predict the wind speed one day ahead for 10 days, using those time series. Since we are forecasting for 10 days, the last 10 days of measures of each data set were saved in the validation set; the rest of the data was used as a training set. The number of samples used for training varied depending on the time series, ranging from 875 to 25,000 samples, while the length of the validation sets were 80 and 240 for the Russian and Mexican time series, respectively.

Figure 1 presents an example of the Mexican time series. For the sake of clarity, only a subset of the time series has been plotted. Given the range of wind speeds in the malpais area, it is clear that the time series presents an outlier near the end of the plotted data, where it goes well beyond 20 m/s.



Figure 1. Segment of the Malpais Time Series.

These data sets were selected since they present different problems encountered in non-synthetic experimentation, most importantly: noise, outliers, data precision, and missing values. In the particular case of the Russian stations, wind speed is measured as integers with no decimals, which adds a layer of noise since the data lacks granularity. The Mexican stations have one decimal digit.

While there are practically no missing values in the Russian data sets, the Mexican data sets lack several values. Because of this, some adjustments to train the forecasters and measure their performance were necessary. Those adjustments are described in Section 4.4.

Also, mostly in the Mexican data sets, some outliers were identified, both in the training and validation sets of some stations. As with the missing values, a few adequations were implemented in the forecasters to correctly treat these data sets.

The data and supplementary material is found at https://github.com/JRCGonzalez/Wind-Forecasting. The code for EvoDAG is provided at https://github.com/mgraffg/EvoDAG. The code for Fuzzy Forecasting is not provided because we are in the process of publishing additional results.

3.2. Forecasting Techniques

This section describes the techniques used in the performance comparison, presenting the equations and algorithms that composes them. Auto-Regressive Integrated Moving Average and Nearest Neighbors are well known forecasting techniques, while Nearest Neighbors with Differential Evolution Parameter Optimization, Artificial Neural Network with Compact Genetic Algorithm Optimization, Fuzzy Forecasting, and EvoDAG are the authors' recent contributions to the state of the art.

3.2.1. Auto-Regressive Integrated Moving Average

In time series statistical analysis, the Auto-Regressive Moving Average models (ARMA) describes a (weakly) stationary stochastic process in terms of two polynomials. The Auto-Regressive Integrated Moving Average models (ARIMA), are a generalization of ARMA models. These models fit to time series data to obtain insights of the data or to forecast future data points in the series. In some cases, these models are applied to data where there is evidence of non-stationarity (this is, where the joint probability distribution of the process is time variant). In those cases, an initial differentiation step (which corresponds to the integrated part of the model) can be applied to reduce the non-stationarity [13].

Much empirical time series behave as if they did not have a bounded mean. Yet, they exhibit homogeneity in the sense that, parts of the time series are similar to other parts in different time lapses. The models that serve to identify non-stationary behavior can be obtained by applying adequate differences on the time series. An important class of models for which the d-th difference is a mixed stationary auto-regressive moving average process is called the ARIMA models.

The non-stationary ARIMA models are described as ARIMA(p,d,q), where p, d, and q are non-negative integer parameters; p is the order of the auto-regressive model, d is the differentiation degree, and q is the order of the moving average model [52].

Let us define the time lag operator *B* such that, when applied to a time series element, it produces the previous element. I.e., $Bs_t = s_{t-1}$. It is possible to call a time series $\mathbb{S} = [s_1, s_2, \dots, s_t, \dots, s_N]$ homogeneous, non-stationary. If it is not stationary, but its first difference, $w_t = s_t - s_{t-1} = (1 - B)s_t$, or any high-order differences $w_t = (1 - B)^d s_t$ produces a stationary time series, then \mathbb{S} can be modeled by an Auto-Regressive Integrated Moving Average process (ARIMA).

Hence, an ARIMA(p, d, q) model can be written as in Equation (1)

$$\Phi(B)(1-B)^d s_t = \delta + \Theta(B)\epsilon_t \tag{1}$$

where δ is an independent term (a constant), $\Phi(B)$ and $\Theta(B)$ are polynomials on B, s_t is the time series at time t, and ϵ_t is the error at time t.

After differentiation, which produces a new stationary time series, this results in an auto-regressive moving average model ARMA, which has the form shown in Equation (2), which can be expressed using polynomials of the lag operator, *B*, as shown in Equation (3)

$$s_{t} = \delta + \phi_{1}s_{t-1} + \phi_{2}s_{t-2} + \dots + \phi_{p}s_{t-p} + \epsilon_{t} - \theta_{1}\epsilon_{t-1} - \theta_{2}\epsilon_{t-2} - \dots - \theta_{q}\epsilon_{t-q}$$

$$= \delta + \sum_{i=1}^{p} \phi_{i}s_{t-1} + \epsilon_{i} - \sum_{i=1}^{q} \theta_{i}\epsilon_{t-i}$$
(2)

$$\Phi(B)s_t = \delta + \Theta(B)\epsilon_t \tag{3}$$

where ϕ_i and θ_i are the coefficients of the polynomials Φ and Θ .

3.2.2. Nearest Neighbors with Differential Evolution Parameter Optimization

Let $S = \{s_1, s_2, \dots, s_t, \dots, s_N\}$ be a time series, where s_t is the value of variable s at time t. It is desired to obtain the forecast of Δn consecutive values, $\{s_{N+1}, s_{N+2}, \dots, s_{N+\Delta n}\}$ by employing any observation available in S.

By using a τ delay and an embedding dimension m, it is possible to build delay vectors of the form $S_t = [s_{t-(m-1)\tau}, s_{t-(m-2)\tau}, \dots, s_{t-\tau}, s_t]$, where m > 0 and $\tau > 0$. The nearest neighbors are those S_t whose distance to S_N is at most ϵ .

For each vector S_t that satisfies Equation (4), the individual values $s_{t+\Delta n}$ are retrieved.

$$|S_N - S_t| \qquad \forall t \in [m, N-1] \tag{4}$$

Each of these vectors form the neighborhood $v_r(S_N)$ with radius r around the point S_N . It is possible to use any vector distance function to measure the distance between possible neighbors.

The forecast is the mean of the values $s_{t+\Delta n}$ of every delay vector near S_N , expressed in Equation (5).

$$\hat{S}_{N+\Delta_n} = \frac{1}{\mid v_r(S_N) \mid} \sum_{S_t \in v_r(S_N)} s_{t+\Delta n}$$
(5)

The Nearest Neighbors algorithm requires that its parameters are fine tuned so it can produce accurate forecasts. Differential Evolution contributes to obtain the best parameters given an specific fitness function. This hybrid method that combines Nearest Neighbors with Differential Evolution is called NNDE [49].

What it does is, for a stochastically generated population of individuals, each of which is encoded by a vector $[m, \tau, r]$, a forecast is obtained for the given time series. Then, the forecasts are compared to the validation set of the time series with an error measure such as MAPE or MSE. The result is the fitness of each individual set of parameters, and the individual with the lowest fitness is the one used to evolve the population. Once this process is completed, the individual with the overall lowest error is retrieved and becomes the set of parameters to use to produce forecasts for that particular time series.

3.2.3. Fuzzy Forecasting

Fuzzy Forecasting (FF) learns a set of Fuzzy Rules (FR) designed to predict a time series behavior. FF traverses the time series analyzing contiguous windows with its next observations and formulating a rule from each one of them. Those rules take the form of Equation (6).

If
$$X_{n-m\tau}$$
 is $A_0 \wedge X_{n-(m-1)\tau}$ is $A_1 \wedge \dots \wedge X_{n-\tau}$ is $A_{m-1} \wedge X_n$ is A_m
then X_{n+1} is A_{m+1} (6)

where *m* and τ are the embedding parameters, and A_i are the Fuzzy Linguistic Terms (FLTs).

The FLTs are formed by dividing the time series range into overlapping intervals. By producing a higher number of FLTs, the resulting forecaster is more precise. On the other hand, the process of learning the FR is more expensive (time-wise). The FLTs overlap just enough so that every real value belongs to at least one FLT (i.e., at least one membership function is not zero). An FR (see Equation (6)) represents the behavior of the time series in the time window where the FR is extracted from. The FR is a low resolution version of the information contained in the time series window.

Figure 2 illustrates a delay window of a time series. The magnitudes in the time series chosen for this example range from 6 to 44. That range was evenly split to produce 5 overlapping FLTs, running along the vertical axis. The first 4 points of that delay window form the antecedents, and the last one constitutes the consequent of the produced FR. Assuming m = 4, $\tau = 1$, and that the rule will be applied at time *t*, the produced FR has the form *If* X_{t-3} *is* $LT_1 \wedge X_{t-2}$ *is* $LT_1 \wedge X_{t-1}$ *is* $LT_4 \wedge X_t$ *is* LT_2 *then* X_{t+1} *is* LT_0 .



Figure 2. Part of a time series and corresponding fuzzy values.

Fuzzy Forecasting has two phases: Learning and forecasting. The first phase learns the FR set from the time series. Figure 3 shows a flow chart of the learning algorithm. Function LearnRules takes X, a time series, m, and τ , and returns the FR set learned from X.

LearnRules has three main parts, shown in each of the columns in Figure 3. First, *X* is traversed using a sliding window, looking for patterns contained in the time series. Those sliding windows

(*W*) have size *m* with delay τ ; each window is fuzzified and added to the list of Linguistic Terms, *LTS*. In the second part, those rules are traversed again, gathering them by the fuzzy form of the rule. The rules are stored in a dictionary *H* of rules and their strengths. Finally, all collected rules of the same form are compiled into a single one, with the strength of the average of the set. The compiled (learned) set of rules is returned.



Figure 3. LearnRules (X, m, τ) .

The second phase of Fuzzy Forecasting is the production of the forecasts, based on the current observation and the set of fuzzy rules produced by the learning phase. Given a FR set, forecasting uses a fuzzy version of the current state of the time series, and sends this fuzzy state to the FR to produce the predicted value. In the forecasting process, more than one rule may fire; in those cases, the result (i.e., the forecast) is defuzzified using the center of gravity method. Figure 4 shows a flow chart of the forecasting algorithm. FuzzyForecasting produces one forecasted value, computed from the set of fuzzy rules and the delay vector formed by the last observations.



Figure 4. FuzzyForecasting (fuzzy rules (FR), *X*, *m*).

FuzzyForecasting takes the FR set, the time series, *X*, and the window size *m*. The Loop traverses FR, verifying if the fuzzy current state satisfies each fuzzy rule. When the fuzzy current state satisfies the antecedents of a rule, we say it fires. The membership of the conjunction of the memberships of the antecedents is called the rule fire strength. When a rule fires, its consequent and the firing strength are recorded in the list Fired. When all rules were traversed the fired rules' strength is combined and defuzzified using the Center of Gravity method. The defuzzufied value, which represents the forecast value, is returned.

3.2.4. Artificial Neural Network (ANN) with Compact Genetic Algorithm Optimization

ANNs are inspired are inspired by the functioning of the human brain. They have the capacity to model a non-linear relationship between input features and expected output. The ANN used in this work is a multi layer perceptron (MLP) with three layers. The first layer receives the inputs (*m* past observations), the hidden layer (one or more) which are the processing layers, and an output layer (the forecast). A sigmoid function is used as an activation function (as observed in Figure 5) [53].

A correct training process may result in a Neural Network Model that predicts an output value, classify an object, approximate a function, or complete a known pattern [54].

The ANN architecture used to forecast is defined by a compact Genetic Algorithm (cGA). The cGA algorithm and the chromosome description are shown in the work of Rodriguez et al. [55].



Figure 5. Artificial Neural Network (ANN) topology which starts with the input layer (*m* past observations), continues with the hidden layer (*h* hidden neurons), and ends with an output layer (a single output \hat{y}_{t+1}). For general Artificial Neural Network terminology, see [54].

3.2.5. EvoDAG

EvoDAG [56,57] is a Genetic Programming (GP) system designed to work on supervised learning problems. It employs a steady-state evolution with tournament selection of size 2. GP is an evolutionary search heuristic with the particular feature that its search in a program space. That is, the solutions obtained by GP are problems; however, EvoDAG restricts this search space to functions. The search space is obtained by composing elements from two sets: the terminal set and the function set. The terminal set contains the inputs and, traditionally, an ephemeral random constant. On the other hand, the function set includes operations related to solve the problem. In the case of EvoDAG, this set is composed by arithmetic, transcendental, trigonometric functions, among others.

EvoDAG uses a bottom-up approach in its search procedure. It starts by considering only the inputs, and then these inputs are joined with functions of the function set, creating individuals composed by one function and their respective inputs. Then these individuals are the inputs of a

function in the functions set, creating offspring which will be the inputs of another function and so on. The evolution continues until the stopping criteria are met.

Each individual is associated with a set of constants that are optimized using ordinary least squares (OLS), even the inputs are associated with a constant. For example, let x_1 be the first input in the terminal set, then the first individual created would be αx_1 where α is calculated using OLS. In order to depict the process of creating an individual using a function of the function set, let $\alpha_1 x_1$ and $\alpha_2 x_2$ be two individuals and be the addition the function selected from the function set, then the offspring is $\alpha_3 \alpha_1 x_1 + \alpha_4 \alpha_2 x_2$ where α_3 and α_4 are obtained using OLS.

EvoDAG uses as stopping criteria early stopping, which employs part of the training set as a validation set. The validation set is used to measure the fitness of all the individuals created, and the evolution stops when the best fitness in the validation set has not been improved for several evaluations, using by default 4000. Specifically, EvoDAG splits the training set in two; the first one acts as the training set and the second as the validation set. The training set guides the evolution and is used to optimize the constants in the individuals. Finally, the model, i.e., the forecaster, is the individual that obtained the best fitness in the validation set.

EvoDAG is a stochastic procedure having as a consequence high variance. In order to reduce the variance, it was decided to create an ensemble of forecasters using bagging. Bagging is implemented considering that the training set is split in two to perform early stopping; thus, it is only necessary to perform different splits, and for each one, a model is created. The final prediction is the median of each model's prediction.

4. Results

With the techniques described in Section 3.2, two experiments based on the same forecasting scenario were tested. In this section we discuss the performance of the forecasters measured by the Symmetric Mean Average Percentage Error.

4.1. Auto-Correlation Analysis of the Data Sets

To identify periodicity in a time series it is necessary to analyze its auto-correlation plot. The Auto-Correlation Function (ACF) is the correlation of a signal (in this case a time series) with a delayed copy of itself. The ACF plot shows ACF as a function of delay.

Since some of the data sets are missing a few samples, it was necessary to identify the longest sequences without missing values, in order to calculate the dominant Lyapunov exponents. For brevity, only the auto-correlation and partial auto-correlation graphs for the time series 24908 and lapiedad are shown in Figures 6 and 7, respectively.



(a) ACF for 24908 (b) PACF for 24908 Figure 6. Auto-Correlation Function (ACF) and partial ACF (PACF) for 24908.





Figure 7. ACF and PACF of lapiedad.

In the plots, the x-axis represents the time lag, and the y-axis shows the auto-correlation coefficient and the partial auto-correlation coefficient for the ACF and PACF figures, respectively. The definitions of ACF and PACF are described in [52]. In every plot there is an increased auto-correlation every 8 and 24 lags for the Russian and Michoacán stations, respectively. Many time lags exhibit auto-correlation values that exceed the 5% significance threshold; this fact indicates that the null hypothesis that there is no correlation for those time lags can be rejected.

4.2. Lyapunov Dominant Exponent Analysis of the Data Sets

Wind speed has been identified as a chaotic or as a non linear system [58–61]. To test these assertions we estimated the dominant Lyapunov exponents of the data sets used in this work.

To obtain the Lyapunov exponents we used an implementation of the algorithm described in [62]. Although there exist different ways to estimate the exponents (such as the ones described in [23,63], or [64]), we decided to use Rosenstein et al.'s implementation, since it is the more consistent in interpretation of the resulting exponent. Although it is desirable to use as much data as possible to obtain the exponents, only 1000 data points were used, in order to minimize the time consumed in this task. The Lyapunov exponents of the time series are shown in Table 1, which includes the Lyapunov exponents of two time series, the Logistic Map [65] and the Sine function. The table columns include the Time Series and the minimum, maximum, and average Lyapunov Exponents.

The calculated exponents are not as high as we expected, which indicates that wind speed is not dominated by noise. Those exponents indicates that the data contains predictable components (seen in part in their respective ACF), but they also explain the difficulty to predict this kind of data in the long term. The exponents obtained are consistent with the ones observed in the literature. In [58] the MLE obtained for their data set is 0.115, while the average MLE obtained by all of the data sets used in this work is 0.116. The Lyapunov Exponents computed for both sets of wind time series provide a strong indication that the underlying process that produce the data are chaotic.

Time Series	Minimum Lyapunov Exponent	Maximum Lyapunov Exponent	Average Lyapunov Exponent
20891	0.0557	0.1974	0.1035
22641	0.0773	0.1798	0.1290
22887	0.0444	0.1643	0.0885
23711	0.0889	0.1388	0.1111
24908	0.1026	0.1488	0.1263
27947	0.0197	0.1210	0.0662
28722	0.0561	0.1767	0.1083
29231	0.0814	0.1802	0.1140
30230	0.0789	0.1208	0.1064
37099	0.0959	0.1480	0.1176
aristeomercado	0.1038	0.1601	0.1298
cointzio	0.1132	0.2041	0.1520
corrales	0.0638	0.1330	0.1073
elfresno	0.0644	0.1457	0.1098
lapalma	0.1037	0.1195	0.1119
lapiedad	0.0601	0.1811	0.1174
malpais	0.1112	0.1573	0.1411
markazuza	0.0971	0.2094	0.1573
melchorocampo	0.0616	0.1864	0.1341
patzcuaro	0.0426	0.1590	0.0965
logistic map	0.3151	0.5921	0.4832
sine	0.0000	0.0000	0.0000

Table 1. Dominant Lyapunov exponents.

4.3. Experiments

An often required wind forecasting task is to forecast the wind speed for the following day. In order to gather more performance information than just one execution of the different forecasting algorithms, we perform a One Day Ahead forecast for the last 10 days of each time series. In performing One Day Ahead (ODA) forecasting, the forecaster generates the number of samples contained in one day of observations at a time.

After we have forecasted one whole day, we consider time advances and the next day of real wind speed observation is available. Since we are forecasting for 10 days, the last 10 days of measures of each data set were saved in the validation set; the rest of the data was used as training set. The sampling period of the Russian stations is three hours, while the sampling period of the Mexican stations is one hour. The number of samples used for training varied depending of the time series from 875 to 25,000 samples, while the length of the validation sets were 80 and 240 for the Russian and Mexican time series, respectively.

4.3.1. Experiment Settings

Each technique has different modelling considerations; those considerations are described in the following paragraphs, and the values used in the experiments are listed in Table 2.

ARIMA—The ARIMA implementation we used was the one included in the R statistical package [66]. The order of the ARIMA model was estimated using the auto.arima function.

NN—To determine the NN parameters (m, τ , and ϵ) we used the deterministic approach described by Kantz in [23]. The deterministic approach uses the Mutual Information algorithm to obtain τ and the False Nearest Neighbors algorithm to find an optimal m; ϵ is found by testing the number of neighbors found for an arbitrary ϵ value, which is updated by the rule $\epsilon \leftarrow \epsilon \times 1.2$ when not enough neighbors are found.

NNDE—In NNDE the NN parameters are found by a DE optimization, where the evolutionary individuals are vectors of the form $[m, \tau, \epsilon]$. Because of the stochastic nature of DE, this optimization process is executed 30 independent times. The set of parameters that yield the least error score is the one used to forecast.

FF—This technique compiles a set of fuzzy rules that describe the time series by using delay vectors of dimension *m* and time delay τ . These parameter values are set to the same as those obtained by the deterministic method used by NN [23,24]. Since the time series contains outliers, FF uses a simple filter which replaces any value greater than 6σ (σ is the standard deviation of the time series) with the missing value indicator.

ANN-cGA—This method determines the optimal topology of a MLP using Compact Genetic Algorithms. The optimization process consists in finding the optimal number of inputs (past observations), the number of hidden neurons, and the learning algorithm.

EvoDAG—EvoDAG uses its default parameters and *m* is set to three days behind.

Table 2 shows the parameters used by the techniques for ODA forecasting. With the exception of NNDE, all the forecasters use the same parameters. NNDE varies its parameters since the DE process can obtain different parameters depending of the forecasting scenario.

Once the different forecasting models were trained, we proceeded to test their forecasts for the proposed forecasting task.

Time Series	NN Deterministic [m, tau, epsilon]	NNDE MSE $[m, \tau, \epsilon]$	NNDE SMAPE $[m, \tau, \epsilon]$	ARIMA (p, d, q)	ANN [m, h, TM]	FF [m, τ]
20891	[8, 1, 6.318]	[23, 48, 6.121]	[43, 51, 12.713]	(1, 1, 5)	[57, 24, bfgs]	[8, 1, 20]
22641	[7, 6, 6.320]	[45, 72, 10.599]	[26, 95, 6.328]	(0, 1, 5)	[59, 39, bfgs]	[7, 6, 20]
22887	[8, 13, 4.389]	[36, 16, 7.653]	[9, 1, 3.346]	(3, 1, 2)	[61, 64, bfgs]	[8, 13, 20]
23711	[7, 5, 1.021]	[6, 8, 0.000]	[29, 69, 4.468]	(1, 1, 1)	[54, 35, bfgs]	[7, 5, 20]
24908	[7, 1, 2.116]	[28, 11, 2.941]	[14, 100, 2.155]	(3, 0, 3)	[62, 59, bfgs]	[7, 1, 20]
27947	[6, 1, 2.116]	[18, 46, 14.602]	[14, 50, 13.834]	(2, 1, 4)	[39, 37, bfgs]	[6, 1, 20]
28722	[6, 1, 3.048]	[42, 29, 9.348]	[16, 20, 6.194]	(3, 1, 1)	[45, 47, bfgs]	[6, 1, 20]
29231	[6, 5, 5.266]	[50, 7, 8.753]	[43, 7, 8.166]	(5, 1, 2)	[41, 35, bfgs]	[6, 5, 20]
30230	[6, 1, 3.048]	[17, 4, 9.849]	[39, 29, 8.056]	(1, 1, 5)	[33, 25, bfgs]	[6, 1, 20]
37099	[6, 1, 1.021]	[16, 45, 7.132]	[1, 49, 28.025]	(2, 1, 3)	[43, 63, bfgs]	[6, 1, 20]
aristeomercado	[8, 8, 10.920]	[13, 5, 28.544]	[7, 16, 22.724]	(1, 0, 2)	[16, 4, bfgs]	[8, 8, 20]
cointzio	[6, 6, 4.389]	[2, 45, 15.820]	[1, 29, 7.759]	(2, 1, 3)	[10, 2, bfgs]	[6, 6, 20]
corrales	[7, 6, 4.389]	[3, 23, 10.155]	[24, 1, 19.312]	(0, 1, 4)	[5, 60, rprop]	[7, 6, 20]
elfresno	[6, 9, 0.410]	[5, 75, 16.560]	[5, 36, 18.935]	(3, 1, 4)	[25, 16, gdx]	[6, 9, 20]
lapalma	[5, 5, 6.320]	[1, 57, 0.018]	[1, 40, 0.000]	(0, 1, 5)	[5, 21, cg]	[5, 5, 20]
lapiedad	[5, 10, 4.389]	[2, 22, 8.457]	[11, 2, 21.112]	(2, 0, 5)	[32, 6, gdm]	[5, 10, 20]
malpais	[9, 1, 116.842]	[23, 97, 12.113]	[41, 2, 19.531]	(0, 1, 2)	[64, 5, gdm]	[9, 1, 20]
markazuza	[5, 1, 2.540]	[26, 1, 10.878]	[24, 1, 10.878]	(3, 1, 4)	[53, 19, bfgs]	[5, 1, 20]
melchorocampo	[5, 1, 4.389]	[2, 5, 1.269]	[1, 59, 3.140]	(1, 0, 2)	[39, 15, rprop]	[5, 1, 20]
patzcuaro	[11, 1, 10.920]	[2, 22, 5.827]	[24, 1, 14.179]	(5, 1, 0)	[29, 3, rprop]	[11, 1, 20]

Table 2. Parameters of the forecasting techniques for ODA.

4.4. Performance Analysis

For this comparison, 7 forecasters were used. Auto-Regressive Integrated Moving Average (ARIMA), and Nearest Neighbors (NN) are well known forecasting techniques in the time series area. Nearest Neighbors with Differential Evolution Optimization (NNDE), Fuzzy Forecasting (FF), Artificial Neural Network with Compact Genetic Algorithm Optimization (ANN-cGA), and EvoDAG are techniques proposed by the authors to tackle this forecasting problem.

As previously indicated, the Mexican Data Sets present missing values in both the training and validation sets. To preserve the integrity of the results, when forecasting, if the value to predict is a missing value that measure does not contribute to the error score of the forecaster.

To measure the performance of the forecasters, we used the Symmetric Mean Absolute Percentage Error (SMAPE).

SMAPE is defined in Equation (7).

$$SMAPE = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}$$
(7)

where *A* are the actual values, *F* are the forecasted values, and *n* is the number of samples in both sets.

SMAPE was selected as error measure because it allows to compare the error between data sets, since it is expressed as a percentage. This can be also be done with the non-symmetric version of this error measure (Mean Average Percentage Error—MAPE). However, the data sets used contain many zero valued samples, which in MAPE lead to undetermined error values.

It is important to clarify that undetermined values can also occur with SMAPE. However, if there is a sum of zeros in the denominator, it indicates that the actual and forecasted values are zero, making the denominator and numerator of the fraction zero, which makes the contribution to the error meaningless.

4.5. One Day Ahead Forecasting

The One Day Ahead forecasting scenario represents the forecasting task this article is addressing. For this scenario each forecaster generates a day of estimations at a time. Once these forecasts are made, one day of observations are taken from the validation set and are incorporated into the training set (replacing the values introduced by the forecaster), and the process is repeated until 10 days of forecasts are completed.

Table 3 shows the SMAPE scores of the 10 day forecasts produced by the different forecasting techniques.

Station	NNDE	EvoDAG	FF	ANNCGA	NN	ARIMA
20891	33.552	42.287	47.349	177.667	40.701	52.485
22641	70.466	73.837	76.620	138.389	71.605	74.538
22887	91.541	94.351	123.836	162.067	100.261	183.042
23711	86.782	106.386	120.273	144.784	111.488	91.577
24908	90.196	147.510	144.755	183.813	146.980	138.607
27947	53.114	57.073	64.736	166.419	57.268	56.960
28722	74.403	83.761	88.769	167.003	84.692	82.507
29231	51.058	56.976	64.375	160.847	55.995	60.928
30230	125.952	132.347	147.326	168.304	134.553	170.713
37099	40.640	41.770	49.845	116.413	42.350	42.964
aristeomercado	38.259	49.938	62.502	189.804	37.395	48.956
cointzio	25.181	39.590	61.984	189.027	45.312	76.323
corrales	30.205	44.388	54.242	179.650	37.860	145.898
elfresno	38.378	56.528	36.649	187.003	49.125	42.472
lapalma	31.136	34.783	39.919	181.602	36.503	141.309
lapiedad	50.624	64.312	31.002	180.052	56.940	200.000
malpais	28.856	46.742	62.484	198.142	46.046	40.755
markazuza	36.916	51.388	62.080	154.063	49.003	113.731
melchorocampo	27.885	35.134	40.899	185.632	29.965	117.441
patzcuaro	39.728	83.065	91.997	187.654	50.734	84.692

Table 3. SMAPE results for One Day Ahead (ODA) forecasting. Bold indicate the winning technique.

NNDE produces the best scores in most stations, which indicates that, for these time series, it is well suited to forecast many samples in advance, compared to the other forecasters (at least for this particular forecasting task). Figure 8 presents a plot of the forecast of the winning model, NNDE, for the malpais time series. As discussed earlier in the article, the quality of the data is not as one could expect. Time Series contain noise, outliers, and missing data, not to count the fact that they are chaotic. Those characteristics make them extremely difficult to forecast. Nonetheless, from the figure, we can observe that the model closely predicts the cyclic behavior of the data, not being able to account for the noise included in the validation set, nor the outliers.

Section 2 includes a discussion on Okumus et al.'s survey article, which provides errors obtained in several works in the area. Those errors are reported for different forecasting tasks, using different

error metrics, so we cannot objectively compare the performance of the methods we present in this article with those presented in the articles included in the survey.



Figure 8. Nearest Neoghbors with Differential Evolution (NNDE) Forecast for the Malpais Validation Set.

5. Conclusions

A set of forecasting methods based on Soft Computing and the comparison of their performances, using a set of wind speed time series available to the public, has been presented. The set of methods used in the performance comparison are Nearest Neighbors (the original method, and a version where its parameters are tuned by Differential Evolution—NNDE), Fuzzy Forecasting, Artificial Neural Networks (designed and tuned by Compact Genetic Algorithms), and Genetic Programming (EvoDAG). For the sake of comparison, we have included ARIMA (a non AI-based method).

The experiments were carried out using twenty time series with wind speed, ten of them correspond to Russian weather stations and the other ten come from Mexican weather stations. The Russian time series are sampled at intervals of 8 h and expressed as integers, while the Mexican time series were sampled at intervals of one hour, using one decimal digit. The maximum exponents of Lyapunov were calculated for each time series', which show that the time series are chaotic. In summary, the data we use for comparison are chaotic, contain noise, and have many missing values, therefore, these time series are difficult to predict in the long term.

The forecasting task was to predict one day ahead, repeated for 10 days. This forecasting task represents 80 measurements for the Russian and 240 for the Mexican time series.

In addition to comparing the performance of forecasting techniques, the use of phase space reconstruction to determine the important contributions of the past as input to forecasting models was presented. Two parameters from the reconstruction process can be used in time series forecasting methods: *m* (the embeding dimension) and τ (the time delay or sub-sampling constant). These parameters provide an insight into what part of the history of the time series can be used in the regression process. Nevertheless, most of the work on forecasting using ANN, just considers some window in the past of the history, normally given by the experience of experts in the application field, and no sub-sampling is used at all. Other statistical methods provide information about what part

of the history of the time series is of importance in the forecasting process. For instance, ARIMA frequently uses the information provided by the ACF and PACF functions.

The results of the performance comparison show that NNDE outperforms the other methods in a vast majority of cases for OSA forecasting and in more than half in ODA forecasting. It is well known that the results of a performance comparison depend heavily on the error criterion used to measure the performance of the forecasting techniques. In this case, we used MSE and SMAPE; NNDE won under both error functions, which places this technique as the most suitable for the forecasting tasks used in this comparison.

We expect the results of this study to be useful for the renewable energy and time series forecasting scientific communities.

Author Contributions: Conceptualization, J.J.F.; Methodology, J.R.C.G. and F.C.; Research on related work and ARIMA, R.L.-F.; Nearest Neighbors with Differential Evolution Parameter Optimization, J.R.C.G. and J.J.F.; Fuzzy Forecasting, J.J.F.; Artificial Neural Network with Compact Genetic Algorithm Optimization, H.R.; EvoDAG, M.G.

Funding: This research was funded by CONACYT Scholarship No. 516226/290379.

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

Abbreviations

The following abbreviations, listed in alphabetical order, were used in this manuscript:

ACF	Auto-Correlation Function
ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Network
ANN-cGA	ANN with cGA
AR	Auto-regressive
ARMA	AutoRegressive Moving Average
ARIMA	AutoRegressive Integrated Moving Average
cGA	Compact Genetic Algorithms
EvoDAG	Evolving Directed Acyclic Graph
FF	Fuzzy Forecast
FLT	Fuzzy Linguistic Terms
FR	Fuzzy Rules
GWPPT	Generalized Wind Power Prediction Tool
MAPE	Mean Average Percentage Error
MSE	Mean Square Error
NARX	Nonlinear Auto-regressive Exogenous Artificial Neural Networks
NN	Nearest Neighbors
NNDE	Nearest Neoghbors with Differential Evolution
ODA	One Day Ahead
OLS	Ordinary Least Squares
PDE	Partial Differential Equations
SMAPE	Symmetric Mean Average Percentage Error
WPPT	Wind Power Prediction Tool
RMSE	Root Mean Squared Error
VAR	Vector Auto-regressive

References

- 1. How Pollution is Threatening Our World. World Economic Forum. 2014. Available online: https://www. weforum.org/agenda/2014/11/pollution-top-concern-2015/ (accessed on 26 December 2017).
- Wang, X.; Guo, P.; Huang, X. A review of wind power forecasting models. *Energy Procedia* 2011, 12, 770–778. [CrossRef]
- 3. Calif, R.; Schmitt, F.G.; Huang, Y. Multifractal description of wind power fluctuations using arbitrary order Hilbert spectral analysis. *Phys. A Stat. Mech. Appl.* **2013**, *392*, 4106–4120. [CrossRef]

- 4. Calif, R.; Schmitt, F. Modeling of atmospheric wind speed sequence using a lognormal continuous stochastic equation. *J. Wind. Eng. Ind. Aerodyn.* **2012**, *109*, 1–8. [CrossRef]
- Dubus, L. Practices, needs and impediments in the use of weather/climate information in the electricity sector. In *Management of Weather and Climate Risk in the Energy Industry*; Springer: Berlin, Germany, 2010; pp. 175–188.
- 6. LemaÎtre, O. Meteorology, climate and energy. In *Management of Weather and Climate Risk in the Energy Industry;* Springer: Berlin, Germany, 2010; pp. 51–65.
- 7. Monteiro, C.; Bessa, R.; Miranda, V.; Botterud, A.; Wang, J.; Conzelmann, G. *Wind Power Forecasting: State-of-the-Art* 2009; Technical Report; Argonne National Laboratory (ANL): Lemont, IL, USA, 2009.
- 8. Landberg, L. A mathematical look at a physical power prediction model. *Wind Energy* 1998, 1, 23–28.
- 9. Hong, J.S. Evaluation of the high-resolution model forecasts over the Taiwan area during GIMEX. *Weather Forecast.* **2003**, *18*, 836–846.
- 10. Charney, J. The use of the primitive equations of motion in numerical prediction. Tellus 1955, 7, 22–26.
- 11. Yule, G.U. Why do we sometimes get nonsense-correlations between Time-Series?—A study in sampling and the nature of time-series. *J. R. Stat. Soc.* **1926**, *89*, 1–63.
- 12. Slutzky, E. The summation of random causes as the source of cyclic processes. *Econom. J. Econom. Soc.* **1937**, *5*, 105–146.
- 13. Wold, H. A Study in the Analysis of Stationary Time Series; Wiley: Hoboken, NJ, USA, 1939.
- 14. Box, G.E.; Jenkins, G.M. Time Series Analysis: Forecasting and Control; Holden-Day. San Francisco, CA. 1970.
- 15. Weigend, A.S.; Huberman, B.A.; Rumelhart, D.E. Predicting the future: A connectionist approach. *Int. J. Neural Syst.* **1990**, *1*, 193–209.
- 16. Mozer, M.C. Neural net architectures for temporal sequence processing. In *Santa Fe Institute Studies on the Sciences of Complexity*; Addison-Wesley Publishing Co.: Boston, MA, USA, 1993; Volume 15, p. 243.
- 17. Connor, J.T.; Martin, R.D.; Atlas, L.E. Recurrent neural networks and robust time series prediction. *IEEE Trans. Neural Netw.* **1994**, *5*, 240–254.
- 18. Allende, H.; Moraga, C.; Salas, R. Artificial neural networks in time series forecasting: A comparative analysis. *Kybernetika* **2002**, *38*, 685–707.
- Müller, K.R.; Smola, A.J.; Rätsch, G.; Schölkopf, B.; Kohlmorgen, J.; Vapnik, V. Predicting time series with support vector machines. In Proceedings of the International Conference on Artificial Neural Networks, Lausanne, Switzerland, 8–10 October 1997; Springer: Berlin, Germany, 1997; pp. 999–1004.
- Mukherjee, S.; Osuna, E.; Girosi, F. Nonlinear prediction of chaotic time series using support vector machines. In *Neural Networks for Signal Processing VII, Proceedings of the 1997 IEEE Signal Processing Society Workshop*; IEEE: New York, NY, USA, 1997; pp. 511–520.
- 21. Cao, L. Support vector machines experts for time series forecasting. *Neurocomputing* 2003, 51, 321–339.
- 22. Sapankevych, N.I.; Sankar, R. Time series prediction using support vector machines: A survey. *IEEE Comput. Intell. Mag.* **2009**, *4*, 24–38.
- 23. Kantz, H.; Schreiber, T. *Nonlinear Time Series Analysis*; Cambridge University Press: Cambridge, UK, 2004; Volume 7.
- 24. Abarbanel, H. Analysis of Observed Chaotic Data; Springer Science & Business Media: Berlin, Germany, 2012.
- 25. Sullivan, J.; Woodall, W.H. A comparison of fuzzy forecasting and Markov modeling. *Fuzzy Sets Syst.* **1994**, 64, 279–293.
- 26. Heshmaty, B.; Kandel, A. Fuzzy linear regression and its applications to forecasting in uncertain environment. *Fuzzy Sets Syst.* **1985**, *15*, 159–191.
- 27. Frantti, T.; Mähönen, P. Fuzzy logic-based forecasting model. Eng. Appl. Artif. Intell. 2001, 14, 189–201.
- Flores, J.J.; González-Santoyo, F.; Flores, B.; Molina, R. Fuzzy NN Time Series Forecasting. In Scientific Methods for the Treatment of Uncertainty in Social Sciences; Gil-Aluja, J., Terceño-Gómez, A., Ferrer-Comalat, J.C., Merigó-Lindahl, J.M., Linares-Mustarós, S., Eds.; Springer: Berlin, Germany, 2015; pp. 167–179.
- 29. Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning; MIT Press: Cambridge, MA, USA, 2016.
- 30. Gamboa, J.C.B. Deep Learning for Time-Series Analysis. *arXiv* **2017**, arXiv:1701.01887.
- 31. Zhang, Y.; Wang, J.; Wang, X. Review on probabilistic forecasting of wind power generation. *Renew. Sustain. Energy Rev.* **2014**, *32*, 255–270. [CrossRef]
- 32. Lange, M. On the Uncertainty of Wind Power Predictions—Analysis of the Forecast Accuracy and Statistical Distribution of Errors. *J. Sol. Energy Eng.* **2005**, *127*, *2*, 177–184. . [CrossRef]

- 33. Bludszuweit, H.; Dominguez-Navarro, J.A.; Llombart, A. Statistical Analysis of Wind Power Forecast Error. *IEEE Trans. Power Syst.* **2008**, *23*, 983–991. [CrossRef]
- 34. Pinson, P.; Kariniotakis, G. On-line assessment of prediction risk for wind power production forecasts. *Wind Energy* **2004**, *7*, 119–132. [CrossRef]
- 35. Pinson, P.; Nielsen, H.; Madsen, H.; Kariniotakis, G. Skill forecasting from ensemble predictions of wind power. *Appl. Energy* **2009**, *86*, 1326–1334. [CrossRef]
- 36. Pinson, P.; Madsen, H.; Nielsen, H.A.; Papaefthymiou, G.; Klöckl, B. From probabilistic forecasts to statistical scenarios of short-term wind power production. *Wind Energy* **2009**, *12*, 51–62. [CrossRef]
- Papaefthymiou, G.; Pinson, P. Modeling of Spatial Dependence in Wind Power Forecast Uncertainty. In Proceedings of the 10th International Conference on Probablistic Methods Applied to Power Systems, Rincon, PR, USA, 25–29 May 2008; pp. 1–9.
- 38. Morales, J.; Mínguez, R.; Conejo, A. A methodology to generate statistically dependent wind speed scenarios. *Appl. Energy* **2010**, *87*, 843–855. [CrossRef]
- Soman, S.S.; Zareipour, H.; Malik, O.; Mandal, P. A review of wind power and wind speed forecasting methods with different time horizons. In Proceedings of the North American Power Symposium 2010, Arlington, TX, USA, 26–28 September 2010; pp. 1–8. [CrossRef]
- 40. Luo, H.y.; Liu, T.q.; Li, X.y. Chaotic Forecasting Method of Short-Term Wind Speed in Wind Farm. *Power Syst. Technol.* **2009**, *9*, 19.
- 41. Bramer, L. Methods for Modeling and Forecasting Wind Characteristics. Ph.D. Thesis, Iowa State University, Ames, IA, USA, 2013.
- 42. Okumus, I.; Dinler, A. Current status of wind energy forecasting and a hybrid method for hourly predictions. *Energy Convers. Manag.* **2016**, *123*, 362–371. [CrossRef]
- 43. Cadenas, E.; Rivera, W. Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA–ANN model. *Renew. Energy* **2010**, *35*, 2732–2738.
- 44. Croonenbroeck, C.; Ambach, D. A selection of time series models for short- to medium-term wind power forecasting. *J. Wind. Eng. Ind. Aerodyn.* **2015**, *136*, 201–210. [CrossRef]
- 45. Hocaoğlu, F.O.; Fidan, M.; Ömer N. Gerek. Mycielski approach for wind speed prediction. *Energy Convers. Manag.* **2009**, *50*, 1436–1443. [CrossRef]
- 46. Jiang, Y.; Xingying, C.; Kun, Y.; Yingchen, L. Short-term wind power forecasting using hybrid method based on enhanced boosting algorithm. *J. Mod. Power Syst. Clean Energy* **2017**, *5*, 126–133. [CrossRef]
- 47. Santamaría-Bonfil, G.; Reyes-Ballesteros, A.; Gershenson, C. Wind speed forecasting for wind farms: A method based on support vector regression. *Renew. Energy* **2016**, *85*, 790–809.
- 48. Skittides, C.; Früh, W.G. Wind forecasting using Principal Component Analysis. *Renew. Energy* 2014, 69, 365–374. [CrossRef]
- 49. Flores, J.J.; González, J.R.C.; Farias, R.L.; Calderon, F. Evolving nearest neighbor time series forecasters. *Soft Comput.* **2017**. [CrossRef]
- 50. Razuvaev, V.; Apasova, E.; Martuganov, R.; Kaiser, D. *Six-and Three-Hourly Meteorological Observations from* 223 USSR Stations; Technical Report; Oak Ridge National Lab.: Oak Ridge, TN, USA, 1995.
- 51. Camargo, G.S.; Barriga, N.G. Preliminary identification study of the wind resource at the State of Michoacán. In Proceedings of the 2014 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, 14–16 November 2014; pp. 1–7. [CrossRef]
- 52. Montgomery, D.C.; Jennings, C.L.; Kulahci, M. *Introduction to Time Series Analysis and Forecasting*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
- 53. Wang, S.C. Artificial neural network. In *Interdisciplinary Computing in Java Programming*; Springer: Berlin, Germany, 2003; pp. 81–100.
- 54. Demuth, H.B.; Beale, M.H.; De Jess, O.; Hagan, M.T. *Neural Network Design*, 2nd ed.; Martin Hagan: Stillwater, OK, USA, 2014.
- 55. Rodriguez, H.; Flores, J.J.; Puig, V.; Morales, L.; Guerra, A.; Calderon, F. Wind speed time series reconstruction using a hybrid neural genetic approach. *IOP Conf. Ser. Earth Environ. Sci.* **2017**, *93*, 012020.
- Graff, M.; Tellez, E.S.; Miranda-Jiménez, S.; Escalante, H.J. EvoDAG: A semantic Genetic Programming Python library. In Proceedings of the 2016 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, 9–11 November 2016; pp. 1–6. [CrossRef]

- 57. Graff, M.; Tellez, E.S.; Escalante, H.J.; Miranda-Jiménez, S. Semantic Genetic Programming for Sentiment Analysis. In NEO 2015; Schütze, O.; Trujillo, L.; Legrand, P., Maldonado, Y., Eds.; Number 663 in Studies in Computational Intelligence; Springer International Publishing: Berlin, Germany, 2017; pp. 43–65. [CrossRef]
- Guo, Z.; Chi, D.; Wu, J.; Zhang, W. A new wind speed forecasting strategy based on the chaotic time series modelling technique and the Apriori algorithm. *Energy Convers. Manag.* 2014, 84, 140–151.
- Hongfei, X.; Tao, D. Chaotic Prediction Method of Short-Term Wind Speed. In Proceedings of the 2011 International Conference on Informatics, Cybernetics, and Computer Engineering (ICCE2011), Melbourne, Australia, 19–20 November 2011; Springer: Berlin, Germany, 2012; pp. 479–487.
- Chen, P.; Chen, H.; Ye, R. Chaotic wind speed series forecasting based on wavelet packet decomposition and support vector regression. In Proceedings of the 2010 Conference Power Engineering Conference IPEC, Singapore, 27–29 October 2010; pp. 256–261.
- 61. Huffaker, R.; Bittelli, M. A Nonlinear Dynamics Approach for Incorporating Wind-Speed Patterns into Wind-Power Project Evaluation. *PLoS ONE* **2015**, *10*, e0115123.
- 62. Rosenstein, M.T.; Collins, J.J.; De Luca, C.J. A practical method for calculating largest Lyapunov exponents from small data sets. *Phys. D Nonlinear Phenom.* **1993**, *65*, 117–134.
- 63. Wolf, A.; Swift, J.B.; Swinney, H.L.; Vastano, J.A. Determining Lyapunov exponents from a time series. *Phys. D Nonlinear Phenom.* **1985**, *16*, 285–317.
- 64. Maus, A.; Sprott, J. Evaluating Lyapunov exponent spectra with neural networks. *Chaos Solitons Fractals* **2013**, *51*, 13–21.
- 65. May, R.M. Simple mathematical models with very complicated dynamics. Nature 1976, 261, 459–467.
- 66. R Development Core Team. *R: A Language and Environment for Statistical Computing;* R Foundation for Statistical Computing: Vienna, Austria, 2008; ISBN 3-900051-07-0.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).