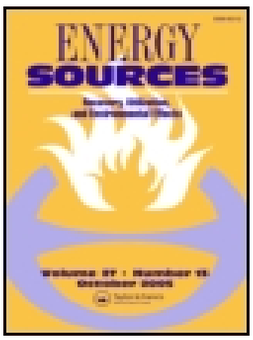

Energy Sources, Part A: Recovery, Utilization, and Environmental Effects

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ueso20

Double ensemble system for wind energy forecasting based on generalized autoregressive conditional heteroskedasticity and neural network models with variational mode decomposition

Angel Colmenares & Jianzhou Wang

To cite this article: Angel Colmenares & Jianzhou Wang (2021): Double ensemble system for wind energy forecasting based on generalized autoregressive conditional heteroskedasticity and neural network models with variational mode decomposition, Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, DOI: 10.1080/15567036.2021.1922550

To link to this article: https://doi.org/10.1080/15567036.2021.1922550

ENERGY SOURCES, PART A: RECOVERY, UTILIZATION, AND ENVIRONMENTAL EFFECTS
https://doi.org/10.1080/15567036.2021.1922550

Double ensemble system for wind energy forecasting based on generalized autoregressive conditional heteroskedasticity and neural network models with variational mode decomposition

Angel Colmenares and Jianzhou Wang

School of Statistics, Dongbei University of Finance and Economics, Dalian, China

ABSTRACT

With the steady integration of wind energy into electricity networks, precise wind speed forecasting is an essential element in the administration and management of power systems. However, wind energy forecasting research has focused increasingly on short-term forecasting, leaving aside the challenging horizons of medium- and long-term predictions. Therefore, this study proposes a wind speed forecasting methodology based on two types of ensembles, which addresses the nonlinearity and chaotic behavior of wind speed using decomposition-based models. With the results of the first ensemble of 90 ARMA-generalized autoregressive conditional heteroskedasticity (ARMA-GARCH) models, the second ensemble is established based on three types of neural networks and learning functions. Finally, we propose the application of variational mode decomposition (VMD) before or after the first ensemble. The experimental outcomes lead us to divide the prediction horizons into two broad groups, those where VMD inclusion did and did not improve the ensemble results. These horizons are classified as short-term (3, 4, and 5 steps) and mid- and long-term forecast horizons (6, 12, 24, and 48 steps), where the best performance arises with the VMD application after the first ensemble. The research contributes to the existing literature studying a wide variety of innovation distribution and optimization methods that can be implemented with GARCH-type models. Simultaneously, the VMD application is proposed in a novel way not seen in the literature by applying it to the predictions already made by other models, in this case, in ensembles of GARCH-type models.

ARTICLE HISTORY
Received 5 October 2020
Revised 31 March 2021
Accepted 20 April 2021

KEYWORDS
Wind speed forecasting; time-series analysis; variational mode decomposition; conditional heteroskedasticity; neural network

Introduction

The projected global wind total power volume by the end of 2022 is 840.9 GW (GWEC 2019). However, even with ecological benefits, wind power's intrinsic uncertainty carries challenges in administering power systems. Wind speed forecasting is essential to offer system managers convenient approximations of available wind power in future lapses.

Many researchers have given much attention to wind energy forecasting research methods during the past few years. These approaches can mainly be divided into the following four categories: physical models (i), conventional statistical models (ii), artificial intelligence models (iii), and hybrid forecasting models (iv). Physical models, which are suitable for wind energy forecasting, consider historical data and use physical parameters. Conventional statistical methods usually apply a linear function to predict the upcoming values. Some of the basic linear forecasting models include the classic autoregressive moving average (ARMA).

CONTACT Jianzhou Wang wangjz@dufe.edu.cn School of Statistics, Dongbei University of Finance and Economics, Dalian 116025, China

In contrast, in artificial intelligent techniques, forecast of future wind speed is through a nonlinear function for the input data. Nonlinear functions can learn the complex nature of wind speed series more precisely. One of the most famous models developed in this category is artificial neural networks (ANNs) or neural networks. Under ensemble techniques, also known as hybrid models or combined forecasting, the principle of solving a particular forecasting problem is based on "*M*" different decomposition and forecasting models.

In general, the principle of decomposition-based hybrid models is to separate the wind speed time series into subseries with more stationary features prior to other techniques' applications. In the study by Qian et al. (2019), a review of decomposition-based hybrid models for wind energy forecasting was conducted. In over 101 publications, they found wavelet (WD)-based and empirical mode (EMD)-based methods to be the most common decomposition approaches, with the fewer commons being the seasonal adjustment method (SAM) or the relatively novel variational mode decomposition (VMD) of 2014.

In recent publications (all since 2018), Li et al. (2018) proposed a hybrid model for short-term wind speed, and Liu et al. (2018) proposed a secondary decomposition, ensemble method, and error correction algorithm. Wang and Li (2018) offered multistep ahead wind speed prediction with a long short-term memory neural network, Tian, Hao, and Hu (2018) proposed hybrid data preprocessing and multiobjective optimization, and Liu, Mi, and Li (2018) applied singular spectrum analysis with neural networks for multistep wind speed forecasting. Zhang, Wei, and Tan (2019) proposed a hybrid model for short-term wind speed forecasting and Zhang et al. (2019) proposed a new prediction method considering wind speed characteristics. In all these studies, VMD is generally adopted as the first step before additional processing.

On the other hand, hybrid models for wind energy forecasting typically use baseline techniques such as ARMA and ARMA-generalized autoregressive conditional heteroskedasticity (ARMA-GARCH). In publications from 2010, such as in the study by Lojowska et al. (2010), the advantages of ARMA-GARCH wind speed modeling are presented, and in the study by Liu, Erdem, and Shi (2011), some approaches for modeling the mean and volatility of wind speed were evaluated. However, none of these extensive GARCH-based studies considered the application of decomposition methods. In the field of neural network models, the extreme learning machine (ELM), long short-term memory (LSTM), radial basis function neural network (RBFNN), or Elman neural network (ENN) are usually reported, generally with a preprocess of decomposition.

Regarding the horizon in multistep wind energy forecasting of hybrid models, according to the objective, these can be separated into (Yang and Wang 2018) very-long-term, long-term, mid-term, short-term, and very-short-term forecasting methods. Very long-term forecasting methods are most useful to the viability of wind energy farm projects and evaluate wind farms' annual power capacity. Long-term forecasting methods (1 day to 1 week or more) help create schedules to improve operating costs. Mid-term forecasting models (6 h to 1 day ahead) offer a foundation for the management and maintenance scheduling of wind energy farms and power plants' load distribution planning. Short-term forecasting methods (30 min to 6 h ahead) are useful for planning the transmission of the power grid and guaranteeing the energy supply rationally. Very short-term methods (a few seconds to 30 min ahead) are implemented to control wind turbine regulation and improve power generation.

Usually, recent publications about multistep wind energy forecasting like those in the study by Jiang and Liu (2019) where a variable weight combined model is used, Luo et al. (2021) that presents a system based on decomposition-ensemble and multi-objective optimization, Wu and Lin (2019) that proposes forecasting based on Variational Mode Decomposition and Least-Squares Support Vector Machine, or Jiang et al. (2020) with a method based on ANNs and deep learning methods and also in the study by Tian, Ren, and Wang (2018), with an improved particle swarm optimization algorithm for wind speed prediction is presented, all these studies and those detailed before (all since 2018) are focused on the horizon of very-short or short-term forecasting models. We found some long-term wind speed forecasts in the study by Bilgili and Sahin (2013) of daily, weekly, and monthly wind speed predictions, but this research is from 2013. The principal reason for this focus is the chaotic

characteristics of the wind speed time series, and a detailed study on this matter is found in Tian (2019). However, some of the longer forecast horizons are necessary to address the power plants' load distribution planning.

A review of previous literature shows that studies in multistep wind energy forecasting have some inherent drawbacks. The disadvantages of such studies are summarized as follows:

(1) Usually, decomposition methods are applied in the first stage of the modeling process. When applied as a middle stage, methods are always introduced after using a previous decomposition method.

(2) Despite its extensive application, GARCH-based models are applied without an in-depth specification of the conditional distributions and optimization methods. Key elements for the correct parametrization of these models, in the same way, do not usually report the construction of ensemble GARCH-based models.

(3) A double ensemble approach is not usually presented. With this approach, the benefits of GARCH models can be combined with the oscillations of neural networks and the opportunities offered by current decomposition methods such as VMD.

(4) Multistep wind energy forecasting research has focused increasingly on short-term forecasting, leaving aside the challenging horizons of medium- and long-term predictions; only a few of the publications reviewed perform more than six-step predictions. Mid- and long-term horizons represent scenarios where a single model could hardly make a better prediction than when considering an ensemble-based approach.

We propose a novel double ensemble system for multistep wind energy forecasting with the analysis above, grounded on autoregressive conditional heteroskedasticity and neural network models with variational mode decomposition. The system creates the first GARCH-ensemble forecast (E-I), on which an autoregressive model is adjusted with logarithmic autoregressive conditional heteroskedasticity variance AR(1)-log-ARCH(1), considering or not the inclusion of intercept in the mean specification. Additionally, 90 ARMA(1,1)-GARCH(1,1) models are taken, which come from the combination of nine conditional distributions for the innovations, multiplied by five optimization algorithms and 45 models by two versions with and without the inclusion of the mean (9 conditional distributions*5 optimization algorithms*2 versions = 90 models). With these 92 predictions, an initial forecast from the E-I will be made under the following two schemes: taking the ensemble average (EAV) and the mean between the prediction with higher and lower values, or in short, the ensemble mid-range (EMR).

In the second ensemble (E-II), based on neural network models, it is considered the original series plus the two forecasts (EAV or EMR) of the first ensemble to train three kinds of neural network models including the following: Jordan and Elman neural networks and a multilayer perceptron, all with three different learning functions that produce a new average forecast. Finally, the system introduces the variational mode decomposition (VMD) before or after the first ensemble forecast and the average and mid-range of the ensemble GARCH (EAV and EMR). If VMD is applied before E-I, the ensemble is constructed based on the original series after the VMD application, which is the traditional method. A novelty of this study arises when we apply VMD after E-I since it is applied to the forecasting of the previous GARCH ensemble.

The primary contributions and novelties of this study are described below:

(1) ***An extensive forecasting ensemble is proposed with two AR-log-ARCH and ninety ARMA-GARCH different models.***

We constructed a first ensemble (E-I) based on an autoregressive model with logarithmic autoregressive conditional heteroskedasticity variance AR(1)-log-ARCH(1), considering the inclusion of intercept in the mean specification. Including 90 ARMA(1,1)-GARCH(1,1) models: combining nine conditional distributions for the innovations, multiplied by five optimization algorithms, and those 45 models by two versions (with and without the mean's inclusion). The research contributes to the

existing literature filling the current gap in studying a wide variety of innovation distribution and optimization methods that can be implemented with GARCH-type models.

(2) ***A novel double ensemble is proposed based on the first ensemble of ninety-two AR-log-ARCH and ARMA-GARCH models with the second ensemble of three different neural networks and learning functions.***

The first ensemble (E-I) result is used to train the following three kinds of neural network models: Jordan and Elman neural networks and multilayer perceptron, all with three different learning functions, setting up the second ensemble (E-II). The experimental result shows an improvement in the results of short-term multistep horizons of forecast (three to five steps) obtained from the first ensemble (E-I) when the second ensemble (E-II) is included.

(3) ***New uses of variational mode decomposition (VMD) is introduced when applied between the double ensemble.***

Taking into consideration variational mode decomposition (VMD), another kind of model is introduced. The application of this technique is proposed to make it in the following two different ways: before the first ensemble (E-I) to denoise the original series, before the application of GARCH-type models and between the ensembles to improve the learning process of the neural network models. The experimental result shows improvement with VMD application between the ensembles in mid- and long-term forecast horizons (6, 12, 24, and 48 steps).

(4) ***Extensive test of the ensemble accuracy with rolling forecasting in sixty-three scenarios, considering seven multistep horizons of the forecast, three weather stations, and three different months.***

The three very short and short-term multistep horizons of the forecast are selected from 3 to 5 steps or 30 min to 50 min (the time series comes in 10 min frequency), with four from short- to long-term horizons of 6, 12, 24, and 48 steps or 1, 2, 4 and 8 h of prediction. The method applied in all the experiments is a rolling forecast, where the period to train the models will always be 7 days or 1008 points, and it is proposed to take the first 15 days or 2160 points of March, June, and September over three different stations to train and verify the models, giving a total of 63 different scenarios

Framework of the double ensemble system

Autoregressive moving average

This is a standard baseline forecasting model used widely in time-series prediction, proposed in 1976 by Box and Jenkins (Box and Jenkins 1976). The ARMA process comprises an autoregressive (AR) model and a movil average (MA) model. The AR process and MA process orders are represented as p and q, respectively, and the ARMA process with the AR process order p and MA process order q is denoted as ARMA(p,q). The autoregressive representation (AR) of order p, denoted by AR (p), can be expressed as follows:

$$X(t) = a_0 + a_1X(t-1) + a_2X(t-2) + \cdots + a_pX(t-p) + e(t) \tag{1}$$

where $X(t)$ is the value at time t, t is the time index, $\{a_0, a_1, a_2, \ldots, a_p\}$ are the parameters of the AR (p) model, and $e(t)$ is the stochastic disturbance term at time t. At any point, the random shocks term is supposed to have identical characteristics: a zero mean with the identical $X(t)$ variance σ^2 and an iid (independently and identically distributed) form. On the other hand, MA representation of order q, denoted by MA (q), can be expressed as follows:

$$X(t) = \mu + e(t) - \theta_1 e(t-1) - \theta_2 e(t-2) - \cdots - \theta_p e(t-q) \tag{2}$$

where μ is the expectation value of $X(t)$; $\{\theta_0, \theta_1, \theta_2, \ldots, \theta_p\}$ are the parameters of the MA (q) model; $\{e(t), e(t-1), e(t-2), \ldots, e(t-q)\}$ are the stochastic disturbance terms as iid variables. With the aggregation of these two processes, we can express the observation $X(t)$ of the ARMA(p,q) model as follows:

$$X(t) = a_0 + a_1 X(t-1) + \cdots + a_p X(t-p) + e(t) - \theta_1 e(t-1) - \cdots - \theta_p e(t-q) \tag{3}$$

Generalized autoregressive conditional heteroskedasticity

Autoregressive conditional heteroskedasticity (ARCH) (Engle 1982) is a statistical method for time-series data that defines the variance of the present error or innovation as a function of the current magnitude of the preceding periods' error terms; typically, the variance is associated with the squares of the preceding innovations. The ARCH model is suitable when the error variance in a time series can be described as an autoregressive (AR) model. When an ARMA model is presumed for the error variance, the model is a GARCH model (Bollerslev 1986). The autoregressive conditional heteroskedasticity has the following form:

$$y_t = \varepsilon_t h_t^{1/2} \tag{4}$$

$$h_t = \alpha_0 + \alpha_1 y_{t-1}^2 \tag{5}$$

where ε is white noise with $V(\varepsilon_t) = 1$. Including normality's supposition, it can be straightforwardly stated in terms of ψ_t the data set accessible at time t. Through conditional densities,

$$y_t|\psi_t \sim N(0, h_t), \tag{6}$$

$$h_t = \alpha_0 + \alpha_1 y_{t-1}^2 \tag{7}$$

the variance function can be stated generally as

$$h_t = h(y_{t-1}, y_{t-2}, \ldots, y_{t-p}, \alpha) \tag{8}$$

where p is the order of the ARCH process, and α is a vector of unknown parameters.

The ARCH model is obtained by supposing that the mean $y_t$is specified as $x_t\beta$, a linear combination of lagged endogenous and exogenous variables involved in the data set ψ_{t-1} with β a vector of unknown parameters. Formally,

$$y_t|\psi_t \sim N(x_t\beta, h_t), \tag{9}$$

$$h_t = h(\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-p}, \alpha) \tag{10}$$

$$\varepsilon_t = y_t - x_t\beta \tag{11}$$

The extension of the ARCH process to the GARCH process with orders (p, q) is given by

$$\varepsilon_t|\psi_{t-1} \sim N(0, h_t), \tag{12}$$

$$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i h_{t-i} = \alpha_0 + A(L)\varepsilon_t^2 + B(L)h_{t,} \tag{13}$$

where $p \geq 0$, $q > 0$, $\alpha_0 > 0$, $\alpha_i \geq 0$, $i = 1, \ldots, q$ $\beta_i \geq 0$, $i = 1, \ldots, p$.

In GARCH models, the density function is typically expressed in terms of the location and scale parameters, normalized to give zero mean and unit variance, so we can also write it in the following form:

$$\alpha_t = (\mu_t, \sigma_t, \omega) \tag{14}$$

Where the conditional mean is given by

$$\mu_t = \mu(\theta, x_t) = E(y_t|x_t), \tag{15}$$

The conditional variance is,

$$\sigma_t^2 = \sigma^2(\theta, x_t) = E\big((y_t - \mu_t)^2|x_t\big), \tag{16}$$

where $\omega = \omega(\theta, x_t)$ represents the other parameters of the distribution, such as the shape and skew. The conditional mean and variance are implemented to scale the innovations,

$$z_t(\theta) = \frac{y_t - \mu(\theta, x_t)}{\sigma(\theta, x_t)}, \tag{17}$$

having a conditional density, which may be specified as

$$g(z|\omega) = \frac{d}{dz} P(z_t < z|\omega), \tag{18}$$

And connected to $f(y|\alpha)$ by,

$$f\big(y_t|\mu_t, \sigma_t^2, \omega\big) = \frac{1}{\sigma_t} g(z_t|\omega). \tag{19}$$

Conditional distributions

The conditional distribution in the GARCH model must be self-decomposable. While having the linear conversion characteristic is essential to center $(x_t - \mu_t)$ and scale (ε_t/σ_t) the innovations, afterward, the modeling is performed with the zero-mean, unit variance, distribution of the standardized variable, z_t is a scaled version of the equivalent conditional distribution of x_t, as described in Equations 17, 18 and 19. Now, some of the distributions can be used in the GARCH models. Some of the distributions that can be used in the GARCH models are as follows: The Normal Distribution (N), Student Distribution (T) (Bollerslev 1986), Generalized Error Distribution (GE), Normal Inverse Gaussian Distribution (NIG), Johnson's SU Distribution (JSU) (Johnson 1949), and the Generalized Hyperbolic Distribution (GH) (DeMarco and Basu 2018). We can also include skewness into unimodal and symmetric distributions by inserting inverse scale coefficients in the positive and negative real half-lines, which was proposed by Fernández and Steel (1998), the Normal, Student, and Generalized Error distributions have skewed modifications that have been standardized to zero mean unit variance.

Artificial neural network

An ANN computes the inputs' function by propagating the estimated observation from the input neurons to the output neurons and the weights as middle limitations. The learning procedure occurs by varying the weights linking the neurons with the training data, which comprise samples of input-output couples of the situation to be learned (Acikgoz, Yildiz, and Sekkeli 2020). A neuron is a data-processing unit that is central to the action of a neural network. The three essentials of the neural model are (i) a group of synapses, or connecting links, each one with weight or strength of its own.

Recurrent neural networks (RNN)

This neural network comprises hidden states spread over time; this permits the network to stock many data about the historical competently. In this kind of system, the input signal $x_j(n)$, internal signal $x'_j(n)$, and output signal $y_k(n)$ are functions of the discrete-time variable n; here, a linear system involving a forward path and a feedback path is characterized by "operators" A and B, respectively, so we readily note the input-output relationships as follows:

$$y_k(n) = \mathrm{A}\left[x'_j(n)\right] \tag{20}$$

$$x'_j(n) = x_j(n) + \mathrm{B}[y_k(n)] \tag{21}$$

where the square brackets emphasize that A and B act as operators. For example, in single-loop feedback where A is a fixed weight $\boldsymbol{w}$ and B is a delay operator $z^{-t}\left[x_j(n)\right] = x_j(n-t)$, rewriting the previous formulas and using the binomial expansion for $(1 - wz^{-1})^{-1}$, we have,

$$y_k(n) = \frac{\mathrm{A}}{1 - \mathrm{AB}}\left[x_j(n)\right] = \frac{w}{1 - wz^{-1}}\left[x_j(n)\right] = w\sum_{t=0}^{\infty} w^t z^{-t}\left[x_j(n)\right] \tag{22}$$

Finally, we can rewrite the exit signal $y_k(n)$ as an infinite weighted summation of current and past samples of the input signal $x_j(n)$, as shown by

$$y_k(n) = \sum_{t=0}^{\infty} w^{t+1} x_j(n-t) \tag{23}$$

Two variations of RNN are listed below.

Elman and Jordan neural networks (ENN, JNN)

ENN was proposed by Elman in 1990 (Elman 1990). They are made of an input layer, a delay layer, a hidden layer, and an output layer. In the input layer and the relating layer, if $x_{it}(i = 1, 2, 3, \ldots, n)$ represents the i-th input values of neurons at time t in the input layer, the relating layer neurons $u_{jk}(j = 1, 2, 3, \ldots, m)$ and $n_{jk}(j = 1, 2, 3, \ldots, m)$ at time t can be modeled as follows:

$$u_{jk}(k) = Z_{jt}(k-1), i = 1, 2, 3, \ldots, n, j = 1, 2, 3, \ldots, m \tag{24}$$

$$n_{jt}(k) = \sum_{i=1}^{n} w_{ij}\mathbf{x}_{it}(k-1) + \sum_{j=1}^{m} c_j u_{jt}(k) \tag{25}$$

In the last two formulas, y_{t+1} denotes the output value in the recurrent layer at time $t+1$. In the hidden layer, the neurons $Z_{jt}(j = 1, \ldots, m)$ are

$$V_{jt}(k) = \frac{1}{(1 + e^{-x})}\left(\sum_{i=1}^{n} w_{ij}\mathbf{x}_{it}(k) + \sum_{j=1}^{m} c_j u_{jt}(k)\right) \tag{26}$$

where w_{ij} and c_{ij} are the connected weights between the input layer and hidden layer, the hidden layer, and the relating layer, respectively. Finally, in the output layer,

$$Z_{t+1}(k) = f_T\left(\sum_{j=1}^{m} v_j V_{jt}(k)\right) \tag{27}$$

where v_j are the linked weights among the hidden layer and output layer and $f_T(\cdot)$ is the activation function that has an identity plan. The JNN was proposed in 1997 by Jordan (Jordan 1997). Similar to the ENN, the delay neurons are connected from the output layer as a substitute for the hidden layer. The plan and state units were collected to establish the input units for the network.

Neural network learning functions

The objective of these functions is to discover a synaptic changing rule that will permit a randomly linked neural network to construct an appropriate internal architecture for a specific task (Rumelhart, Hinton, and Williams 1986). Among the basic methods used for this function, we have the backpropagation (B) proposed in 1986 by Rumelhart et al. (Elman 1990) and the following two variants: the *backpropagation with momentum term (M)* created in 1993 by Yu et al. (Yu, Loh, and Miller 1993) and the *resilient backpropagation (R)* proposed by Riedmiller and Braun in 1993 (Riedmiller and Braun 1993).

Optimization method

Optimization of a function is a broad term and refers to either minimization or maximization. The following two comprehensive types of algorithms exist: (1) those that take evidence regarding the slope or shape of the output of the objective function and (2) those whose search guidelines are autonomous of the shape of the response surface.

Nelder–Mead simplex (N-M S)

The Nelder-Mead simplex is a simplex-based method and can be classified as autonomous of the shape of the output form. A *simplex*$S\ \mathbb{R}^n$is defined as the convex hull of $n+1$vertices$x_0, \cdots, x_n \in \mathbb{R}^n$. The technique starts with a group of $n+1$ values $x_0, \cdots, x_n \in \mathbb{R}^n$established as the vertices of a working simplex S and the equivalent group of function values at the vertices $f_j := f(x_j),\ \text{for} j = 0, \ldots, n$. The primary working simplex S has to be nondegenerate, i.e., the values $x_0, \cdots, x_n$ must not be in the same hyperplane. The technique then makes a series of changes to the working simplex S to reduce the function values at its vertices. The changes are resolved at each stage by calculating one or more test points, organized with their function values, and contrasting the resultant values with those at the vertices. This procedure ended when the working simplex S converted satisfactorily insignificant in some sense or when the function values f_j were near sufficient in some sense (provided f is continuous), as proposed by Nelder and Mead (1965).

Constrained by linear approximations (CLA)

CLA is an algorithm for derivative-free optimization with nonlinear inequality and equality constraints (Powell 1994). The algorithm builds consecutive linear estimates of the objective function and restraints via a simplex of $n+1$ points (in n dimensions) and improves these estimates in a trust region at each stage.

Principal axis (PA)

The PA is a gradient-free local optimization via the "principal-axis method" proposed by Brent (1973). For an n-variable task, take a group of exploration directions $u_1, u_2, \ldots, u_n$ and a value x_0. x_i is taken as the value that minimizes f over the direction u_i from x_i (i.e., perform a "line search" fromx_{i-1}), then u_i is substituted with u_{i+1}. At the termination, u_n is substituted with$x_n - x_0$. Preferably, the novel u_i must be linearly independent so that a new repetition can be undertaken, but in practice, they are not. Brent's algorithm applies singular value decomposition (SVD) on the matrix $U = (u_1, u_2, \ldots, u_n)$ to readjust them to the local quadratic model's principal directions. With the new group of u_i obtained, additional repetition can be done.

Bound by quadratic approximation (BQA)

BQA was proposed by Powell (2009) as an iterative algorithm for finding the minimum of a function $F(\bar{x}), \bar{x} \in \mathbb{R}^n$, subject to bounds $\bar{a} \leq \bar{x} \leq \bar{b}$on the variables. F is specified by a "black box" that returns the value $F(\bar{x})$ for any feasible $\bar{x}$. Each repetition uses a quadratic approximation Q to F that satisfies $Q(\underline{y_j}) = F(\underline{y_j}), j = 1,\ 2, \ldots, m$, the interpolation values $\underline{y_j}$ being selected and changed automatically, but m is arranged constant, the value $m = 2n + 1$ being typical.

Subspace-searching simplex (Sub S)

Sub S is a subspace-searching simplex technique for the unconstrained optimization of broad multivariate functions developed by Thomas Harvey Rowan (1990). The subplex technique is appropriate for optimizing noisy objective functions. The number of function valuations demands for convergence typically only grows linearly with the problem magnitude, so the subplex technique is more effective than the simplex technique.

Variational mode decomposition

VMD separates a signal into meaningful modes according to their frequency information based on discrete Fourier transform. Through VMD, the original signal is separated into k mode series, where each mode u_k is asked to be densest over a central pulsation ω_k calculated together with the decomposition (Dragomiretskiy and Zosso 2014). For a one-dimensional signal s, the stages to evaluate a mode's bandwidth $u_k$can be summarized in the following three steps:

(1) Calculate the signal related to u_k through the Hilbert transform to find a one-sided frequency spectrum.

$$\left(\delta(t) + \frac{j}{\pi t}\right) * u_k(t) \tag{28}$$

(1) Change the mode's frequency spectrum to the baseband by combining it with an exponential tuned to the corresponding projected core frequency.

$$\left[\left(\delta(t) + \frac{j}{\pi t}\right) * u_k(t)\right] e^{-j\omega_k t} \tag{29}$$

(3) Evaluate the bandwidth with the H1 Gaussian smoothness of the demodulated signal, such as the squared L2-norm of the gradient. Then, the constrained variational task of finding all the modes is defined by the following:

$$\min_{u_k, \omega_k} = \left\{ \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t}\right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\}, s.t. \sum_k u_k = s \tag{30}$$

where $\{u_k\} = \{u_1, u_2, \cdots, u_k\}$ are the nodes, and $\{\omega_k\} = \{\omega_1, \omega_2, \cdots, \omega_k\}$ are the center frequencies of the nodes. $\delta(t)$ is the Dirac function, and * is the convolution operation. s is the original series. The alternative direction method of multipliers (ADMM) is used to solve formula's optimization problem (Rehman, Naveed, and Aftab 2019).

Experiments proposed

To prove the forecasting effectiveness of each of the models previously discussed, we present some experiments based on wind speed data from March, June, and September from three different weather stations. Furthermore, the performance metrics and a detailed explanation of the proposed ensemble system are presented.

Dataset

The information comes from three different weather stations, but it cannot inform the location due to the confidentiality of the information. Since the data is only available for the first 9 months of the year, we select the last month of each of the three quarters, which are March, June, and September. The frequency at which the wind speed was collected was 10 min (144 points per day). In some cases, the measurements are not counted consecutively during all months. The period during which it was possible to obtain the data without omissions for the selected months was the first 15 days each month. With this limitation, we choose the first fifteen days of March, June, and September.

Performance metrics

For model evaluation, four metric rules were applied, as shown in Table 1.

Different experiments established

In the present research, we proposed seven experiments, which are conducted to calculate each method's forecasting precision. The experimental parameters of the experiments are in Table 2. The method applied in all the experiments is a rolling forecast, where the period to train the models will always be 7 days or 144 × 7 = 1008 points. It is proposed to take the first 15 days or 15 × 144 = 2160 points of March, June, and March over three different stations to train and verify the models. These seven horizons of the forecast, multiplied by three stations and multiplied by 3 months, giving 63 different scenarios.

The repetition of each experiment in Table 2 is calculated by dividing the rest of the 8 days or 144 × 8 = 1152 points (the total of 15 days minus the 7 days to train) by the forecast horizon (H). In this way,

Table 1. Statistical indicators of the data for each year and day of the week.

Metric	Definition	Equation
MAE	The average absolute forecast error of *n* times forecast	$\mathrm{MAE} = \frac{1}{N}\sum_{n=1}^{N}\left\lvert y_n - \hat{y}_n\right\rvert$
MSE	The average of the prediction error squares	$\mathrm{MSE} = \frac{1}{N}\sum_{n=1}^{N}\left(y_n - \hat{y}_n\right)^2$
MAPE	The mean of absolute percentage error	$\mathrm{MAPE} = \frac{1}{N}\sum_{n=1}^{N}\left\lvert\left(y_n - \hat{y}_n\right)/y_n\right\rvert \times 100\%$
sMAPE	The symmetric MAPE	$\mathrm{sMAPE} = \frac{2}{N}\sum_{n=1}^{N}\left\lvert y_n - \hat{y}_n\right\rvert/\left(\lvert y_n\rvert + \left\lvert\hat{y}_n\right\rvert\right) \times 100\%$

Notes: due to the small values that the series takes in some points, the result of the **MAPE** metric can be affected. This happens when the percent error in small values is more considerable than in larger values with the same absolute error. i.e., **Case 1**: real wind speed = 10, forecast = 9 which gives MAPE $= \lvert(10-9)/10\rvert \times 100\% = 10\%$, **Case 2**: real wind speed = 1, forecast = 2 which gives MAPE $= \lvert(1-2)/1\rvert \times 100\% = 100\%$, in both cases, the absolute error is 1, but the percent error increase from 10% to 100% because we are in the presence of values closer to zero in the last scenario.

Table 2. Parameters of the seven experiments.

Dimension	Interval for verifying the model		Repetition of the experiment	Train to verify ratio
Experiment	Time	Points	Times	Ratio
3-step	30 min	3	384	336:1
4-step	40 min	4	288	252:1
5-step	50 min	5	230	1008:5
6-step	1 h	6	192	168:1
12-step	2 h	12	96	84:1
24-step	4 h	24	48	42:1
48-step	8 h	48	24	21:1

Notes: In all the experiments, it is used 7 days of information to train the models, which means 144 × 7 = 1008 points.

the window to train the models (1008 points) advances H points each time, i.e., in the case of the forecast horizon of H = 48 or 8 h, the times we repeat the experiment are calculated as 1152/48 = 24, so the time window of 7 days to train the models advances 48 points 24 times over the next 8 days, wherein each of these 24 repetitions a forecast of H = 48 points is made.

Proposed ensemble system

The base of the proposed system comes from a first ensemble, denoted by "E-I," where it is taken as an autoregressive model with logarithmic autoregressive conditional heteroskedasticity variance AR(1)-log-ARCH(1), considering the inclusion of intercept in the mean specification. With 90 ARMA(1,1)-GARCH(1,1) models, from the combination of nine conditional distributions for the innovations, multiplied by five optimizations algorithms, and those 45 models by two versions, with or without the inclusion of the mean. Table 3 shows a summary of all the ARMA-GARCH models.

According to these models, we have a wide range of wind speed predictions with a total of 92 estimations. Then, an initial forecast from this first ensemble will be made under the following two schemes: taking the ensemble average (EAV) of the predictions and the mean between the prediction with higher and lower values or ensemble mid-range (EMR). The original series plus these two forecasts (EAV or EMR) is used to train three kinds of neural network models: Jordan and Elman neural networks and multilayer perceptrons, all with three different learning functions. We can consider the combination of estimations from the different neural networks and learning functions as the second ensemble or Ensemble-II (E-II) and produce a new average forecast. For example, the mean can be determined by each neural network and the three learning functions. A summary of the steps followed in this prototype of models from the system is shown in Table 4

The stages from Table 4 produce different combinations of models; a summary of the notation used is in Table 5.

Table 3. ARMA-GARCH Models of the Ensemble I (E-I).

Base models (2)	Innovation distribution (9)	Optimization method (5)
ARMA (1,1) - GARCH (1,1)[a]	Normal[b]	Constrained by Linear Approximations
	T-student's[b]	Bound by Quadratic Approximation
	Generalized error[b]	Principal Axis
	Normal Inverse Gaussian	Nelder-Mead Simplex
	Generalized Hyperbolic	Subspace-searching Simplex
	Johnson's SU	

aTwo versions: ARMA part of the model with and without zero mean.
bTwo versions: Original and skew-version of the distribution.

Table 4. Summary of stages when the VMD is not applied.

Stage	Description of stage	Models/Algorithm/Formula
1	Application of Ensemble I (E-I) to the original series	***E-I*** = AR(1)-log-ARCH(1) (2 models) + ARMA(1,1)-GARCH(1,1) (90 models)
2	Calculation of Ensemble I Average (EAV)	***EAV*** = mean (E-I)
3	Calculation of Ensemble I Mid-Range (EMR)	***EMR*** = (max (E-I) – min(E-I))/2
4	Application of NN models to the original series + EAV or original series + EMR and calculation of average over these Ensemble-II (E-II) with each NN model	***NN: ENN, JNN***, and ***MLP*** all with three learning functions ***NN-(B, M, R)*** ***EAV-NN*** = mean (E-II-NN-(B, M, R)) ***EMR-NN*** = (max (E-II-NN-(B, M, R)) – min (E-II-NN-(B, M, R)))/2
5	Computing of performance metrics between the forecasts and real wind speed values	***MAE, MSE, MAPE, sMAPE*** (Table 1)

Notes: The Notation of the NN's learning function are ***B***: Backpropagation, ***M***: Backpropagation with Momentum, and ***R***: Resilient backpropagation.

Table 5. Summary of basic notation when the VMD is not applied.

Type of ensemble	By type of NN	By NN and learning function
EAV: Ensemble I Average	***EAV-(ENN, JNN, MLP)***	***EAV-NN-(B, M, R)***
EMR: Ensemble I Mid-Range	***EMR -(ENN, JNN, MLP)***	***EMR-NN-(B, M, R)***

Notes: when the notation only expresses the type of **NN** (i.e., ***EAV-ENN***), the average among the forecast of the different learning functions ***(B, M, R)*** was calculated.

Table 6. Summary of stages when the VMD is applied before the Ensemble-I.

Stage	Description of stage	Models/Algorithm/Formula
1	Application of VMD to the original series	VMD
2–5	Application of stages 1–4 from Table 4 but to the series after applied VMD	see Table 4

Notes: the notation of the models, in this case, is the same as the Table 5, but with ***VMD*** at the beginning (i.e., ***VMD-EAV-ENN-B***, which means that first was applied the ***VMD*** to the original series, with this result, the ***EAV*** it is calculated and finally, to the series (after ***VMD***) plus the forecast it was applied an ***ENN*** with ***B***ackpropagation).

Table 7. Summary of stages when the VMD is applied after the Ensemble-I.

Stage	Description of stage	Models/Algorithm/Formula
1–3	Application of stages 1–3 from Table 4	E-I, EAV, EMR
4	After the Ensemble-I it is applied the VMD to the original series + EAV and the original series + EMR	VMD
5–6	Application of stages 4–5 from Table 4	see Table 4

Notes: the notation of the models, in this case, is the same as the Table 5, but with VMD just before the NN models (i.e., ***EAV-VMD-ENN-B*** means that was calculated the Ensemble-I to the original series, the ***VMD*** was applied to the original series plus the ***EAV***, and finally, to series plus the forecast with ***VMD*** it was applied an ***ENN*** with Backpropagation).

Another type of model is introduced when variational mode decomposition (VMD) is introduced. The application of this technique is made before or after Ensemble-I. If the VMD is applied before the Ensemble-I means, the ensemble is made based on the original time series of wind speed after VMD application, but the rest of the stages will be the same in Table 4. A summary of the stages followed in this second prototype of models with VMD before Ensemble-I is shown in Table 6.

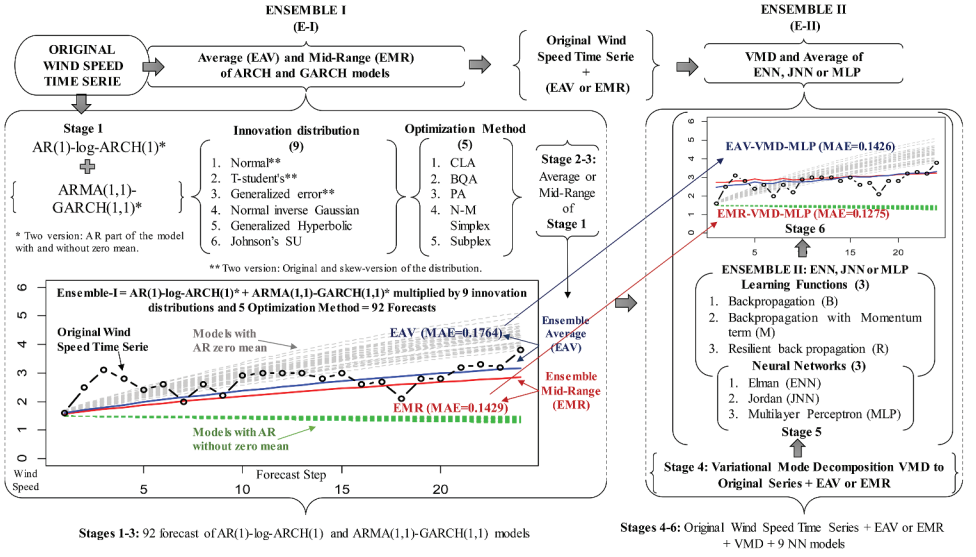

Figure 1. Proposed ensemble system with the application of VMD after the Ensemble-I, following the stages from .Table 7

In the third type of model, VMD is applied after Ensemble-I or just before Ensemble-II. A summary of the steps followed in this third prototype of the models, with VMD after the ensemble, is shown in Table 7. A graphical explanation of this third kind of model is shown in Figure 1, in which a real forecast of 12-steps is shown.

Experimental results

To show the model's performance constructed by the proposed ensemble system, Table 8 shows the MAE metric results from Table 1 per horizon of forecast specified in Table 2 and month. The results with non-VMD arise when applying stages of Table 4, the values with the VMD before Ensemble-I following stages of Table 6, and the VMD after Ensemble-I, according to Table 7. The model notation can be seen in Table 5. The inclusion of the means of EAV-None and EMR-None models was not applied to any NN model after Ensemble-I. The forecast is made based only on two estimations (EAV or EMR), excluding the VMD method or considered before and after the ensemble.

Experimental and metaheuristic analysis

The following section discusses the best models for each horizon of the forecast, analysis of the metaheuristic, and the learning function results in each neural network model and innovation distribution by the optimization method.

Best models per horizon of the forecast

Summarizing the results indicated in Table 8, which followed Figure 1, but with all the metrics and each horizon of the forecast, we have the values in Table 9. This summary shows some exciting findings; in general, the ensemble mid-range is positioned as the best model in five of the seven prediction horizons, leading from 1 h or six steps. This outcome also leads us to divide the prediction horizons into two broad groups, those where VMD's inclusion did not improve the ensemble results, which are the forecasts of 3-, 4-, and 5-step horizons that could be classified as short-term.

In these cases, in the short-term forecast, the second ensemble (E-II) improved the results of Ensemble-I, specifically with the Elman neural network (ENN). It should be noted that the results of the first ensemble do not contain oscillations, which means that the inclusion of the neural networks adds noise to the predictions that the ARCH and GARCH models do not express.

On the other hand, in the forecast with 6, 12, 24, and 48 steps, which we can classify as mid- and long-term forecast horizons, the inclusion of the oscillations or noises coming from the neural network models does not benefit the predictions; it is enough that the application of VMD allows the improvement in the results.

Finally, the application of the VMD before Ensemble-I did not result in any improvement; only in the case of March in the prediction horizon of 48 steps (see Table 8) was a notable improvement obtained, telling us that in general, the ARCH and GARCH models of Ensemble-I lose their predictive capacity with this previous decomposition.

Metaheuristics

Although the results are the average overall learning functions, it is crucial to analyze each result at each time horizon, which could be considered in future investigations. Figure 2 shows for each prediction horizon (each line) the mean of the absolute error (MAE) for the mid-range ensemble (EMR). This metric and model are selected, and the results are given in Table 8.

In general, the values of the three learning functions are relatively homogeneous. The only scenario where this is not fulfilled is in Jordan's neural network with resilient backpropagation. The **MAE** was increased in all experiments with less than 48 forecasting steps and was most evident when **VMD** is

Table 8. Best model per horizon of forecast (H).

	Application of VMD/Horizont of Forecast		Non-VMD							VMD Before Ensemble							VMD After Ensemble						
Month	Ensemble	NN	48	24	12	6	5	4	3	48	24	12	6	5	4	3	48	24	12	6	5	4	3
March	EAV	ENN	2.03	1.52	1.14	0.90	0.85	**0.77**	0.72	2.04	1.54	1.15	0.92	0.89	0.80	0.75	2.04	1.51	1.14	0.90	0.86	0.78	0.73
		JNN	2.07	1.61	1.20	1.05	1.04	1.00	0.89	2.13	1.59	1.25	1.13	1.07	1.02	1.02	2.30	1.97	1.70	1.81	1.55	1.50	1.08
		MLP	2.04	1.50	1.14	0.90	0.85	0.78	0.73	2.04	1.53	1.14	0.92	0.89	0.80	0.75	2.06	1.51	**1.14**	0.91	0.86	0.79	0.74
		None	2.04	**1.50**	1.14	0.90	0.85	0.78	0.73	2.03	**1.52**	1.15	0.93	0.90	0.82	0.78	2.04	1.51	1.14	0.90	0.85	0.78	0.73
	EMR	ENN	**2.00**	1.52	1.14	0.90	**0.85**	0.77	**0.72**	2.01	1.56	**1.14**	**0.90**	0.87	**0.78**	**0.73**	2.00	1.51	1.14	0.90	0.86	0.78	0.73
		JNN	2.04	1.65	1.19	1.08	1.10	0.96	0.90	2.07	1.56	1.30	1.02	1.09	0.95	0.87	2.26	1.85	2.02	1.68	1.57	1.33	1.10
		MLP	2.01	1.50	**1.14**	0.90	0.85	0.77	0.73	2.00	1.52	1.14	0.91	**0.87**	0.78	0.74	2.01	1.51	1.14	0.90	0.86	0.79	0.74
		None	2.00	1.51	1.14	**0.90**	0.85	0.77	0.72	**1.99**	1.53	1.14	0.91	0.88	0.79	0.74	**2.00**	**1.50**	1.14	**0.89**	**0.85**	**0.77**	**0.72**
June	EAV	ENN	1.78	1.43	1.12	0.92	0.81	0.77	**0.71**	1.82	1.49	1.14	0.92	0.85	0.81	0.73	1.77	1.43	1.12	0.91	0.82	0.77	0.72
		JNN	1.80	1.42	1.12	0.97	0.87	0.84	0.83	1.84	1.54	1.19	0.95	0.94	0.89	0.79	1.81	1.50	1.43	1.13	1.04	1.09	0.88
		MLP	1.77	1.42	1.11	0.91	0.82	0.77	0.72	1.81	1.47	1.14	0.93	0.85	0.81	0.73	1.78	1.44	1.12	0.92	0.82	0.78	0.72
		None	1.78	1.42	1.11	0.91	0.82	0.77	0.72	1.82	1.47	1.15	0.93	0.87	0.83	0.76	1.77	**1.37**	1.10	0.91	0.82	0.77	0.73
	EMR	ENN	**1.65**	1.40	1.10	0.91	0.81	**0.77**	0.71	1.67	1.41	1.14	**0.91**	**0.83**	0.79	**0.72**	1.68	1.38	1.10	0.91	0.81	0.77	**0.72**
		JNN	1.69	**1.38**	1.13	0.98	0.87	0.84	0.79	**1.67**	1.41	1.16	0.93	0.87	0.84	0.79	1.73	1.49	1.34	1.16	1.07	0.97	0.90
		MLP	1.70	1.39	1.10	0.91	0.81	0.77	0.72	1.68	1.42	1.13	0.92	0.83	**0.79**	0.72	1.70	1.39	1.10	0.92	0.82	0.78	0.72
		None	1.67	1.38	**1.09**	**0.91**	**0.81**	0.77	0.71	1.67	**1.40**	**1.13**	0.92	0.84	0.80	0.73	**1.67**	1.40	**1.09**	**0.91**	**0.81**	**0.77**	0.73
Sept	EAV	ENN	1.40	1.20	0.89	0.72	**0.67**	0.63	0.59	1.42	1.19	**0.88**	0.73	0.70	0.64	0.60	1.40	1.18	0.88	0.72	0.67	0.64	0.59
		JNN	1.42	1.19	0.90	0.77	0.71	0.72	0.64	1.44	1.19	0.88	0.73	0.76	0.71	0.67	1.44	1.29	0.96	0.92	0.85	0.82	0.68
		MLP	1.38	1.20	0.89	0.73	0.68	0.65	0.60	1.38	1.20	0.88	0.74	0.71	0.65	0.62	1.38	1.21	0.89	0.74	0.69	0.66	0.60
		None	**1.36**	1.19	0.88	0.72	0.67	0.63	**0.58**	**1.38**	**1.18**	0.88	0.74	0.71	0.65	0.63	1.36	1.17	0.88	0.72	**0.67**	0.63	**0.58**
	EMR	ENN	1.43	1.19	0.88	**0.72**	0.67	0.63	0.59	1.46	1.18	0.88	**0.72**	**0.69**	**0.63**	**0.59**	1.43	**1.17**	**0.88**	**0.72**	0.67	0.63	0.59
		JNN	1.44	1.17	0.89	0.80	0.78	0.70	0.67	1.47	1.22	0.94	0.74	0.76	0.67	0.66	1.53	1.36	1.09	0.97	0.86	0.80	0.67
		MLP	1.39	1.18	0.88	0.73	0.69	0.65	0.60	1.42	1.20	0.89	0.74	0.70	0.65	0.61	**1.36**	1.18	0.89	0.73	0.69	0.66	0.61
		None	1.38	**1.17**	**0.88**	0.72	0.67	**0.63**	0.58	1.41	1.19	0.89	0.73	0.70	0.64	0.61	1.38	1.17	0.88	0.72	0.67	**0.63**	0.59

Notes: The bold font indicates the lowest value for the corresponding horizon of prediction, month, and VMD application.

Table 9. Best model per horizon of forecast (H).

H	Application of VMD	E-I	E-II	MAE	MSE	MAPE	sMAPE
3	None	EAV	ENN	**0.6722**	0.8593	13.957%	**13.231%**
4		EMR		**0.7235**	**0.9951**	**15.192%**	**14.180%**
5		EAV		**0.7790**	1.1360	16.421%	**15.175%**
6	After Ensemble-I	EMR	None	**0.8389**	**1.3117**	17.567%	**16.365%**
12				**1.0386**	**1.9666**	22.461%	**19.977%**
24				**1.3495**	3.2686	29.016%	**25.159%**
48				**1.6829**	**4.7951**	37.357%	30.775%

Notes: The values are the average over March, June, and September. The bold font indicates the lowest values for the corresponding horizon of prediction.

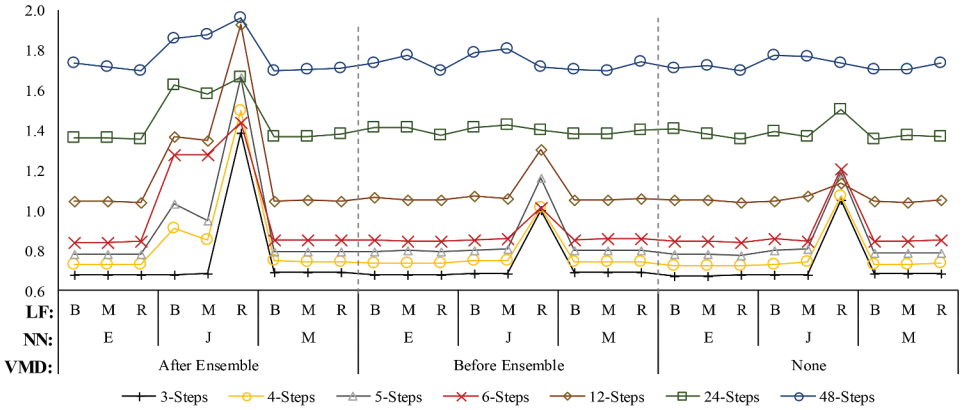

Figure 2. The figure's notation is defined as follows: **LF**: Learning Function; **B**: Backpropagation, **M**: Backpropagation with Momentum, and **R**: Resilient backpropagation. **NN**: Neural Network model, where **E**: Elman NN, **J**: Jordan NN, and **M**: Multilayer perceptron. **VMD**: Variational Mode Decomposition; for **After ensemble**, see .Table 7, **Before ensemble**, see Table 6, and for **None**, see Table 4.

Table 10. MAE model of the seven time-horizons of the forecast by type of learning function and neural network model (horizontal combinations), and by type of ensemble and application of VMD (vertical combination).

		Ensemble Average (EAV)			Ensemble Mid-Range (EMR)		
Learning Function	NN	VMD After Ensemble	VMD Before Ensemble	None VMD	VMD After Ensemble	VMD Before Ensemble	None VMD
Backpropagation	E	1.03139*	1.05543	1.03253	1.02461*	1.04008	1.02735
	J	1.25843	1.07563	1.04644*	1.24829	1.05187	1.04107*
	M	1.03677	1.04794	1.0309*	1.02785	1.03177	1.02108*
Backpropagation with Momentum	E	1.03419*	1.06436	1.04015	1.02245*	1.04422	1.02616
	J	1.26148	1.07154	1.0499*	1.22411	1.05594	1.03979*
	M	1.03873	1.04894	1.03152*	1.02889	1.03115	1.02402*
Resilient backpropagation	E	**1.02752**	**1.04382**	**1.02601***	**1.01821**	**1.02552**	**1.01552***
	J	1.59799	1.30438	1.25174*	1.64766	1.22997*	1.26907
	M	1.04449	1.05932	1.04116*	1.03039*	1.04264	1.03043

Notes: bold font indicates the lowest MAE by type of ensemble, and the asterisk (*) the lowest values MAE by learning function.

applied after Ensemble-I. Based on these results, it is not recommended using this type of learning function for these neural networks.

Table 10 shows how the Elman (E) neural network shows the best result in all ensemble scenarios and VMD applications among the learning functions and neural network models. The lowest value occurs with the ensemble mid-range (EMR) with an MAE of 1.01552, and the lowest value of all the models is shown in Table 10. This result confirms the analysis results shown in Table 9 by the horizon of the forecast. Over the ensemble and VMD application variation, the ensemble average (EAV) without VMD application shows the best result in 12 of the 18 scenarios, followed by VMD application

after the ensemble in only five cases. In general, the ensemble mid-range (EMR) without VMD application can be considered the overall best model.

Conclusion

Accurately forecasting wind speed is a crucial stage in the scheduling and management of wind energy generation in power systems, and its relevance increases with the growing incorporation of wind power into the electricity market. This research proposes an innovative double ensemble hybrid approach for wind energy prediction. The approach includes a first ensemble with the average and mid-range of a wide variety of 92 ARCH and GARCH models, as a result of combining nine conditional distributions for the innovations, five optimization algorithms and two versions of the models considering the inclusion of intercept in the mean specification. Taking the original wind speed time series and the two forecasts from the first ensemble (average and mid-range), we construct the second ensemble with a multilayer perceptron, a JNN, and an ENN, all with three different learning functions. This second process gives nine new forecasts for each prediction from the first ensemble.

The inclusion of the variational mode decomposition (VMD) is done in the following two ways: in the first step of the modeling process, as in most previous research, and in a new stage of the ensemble system, which is between the two ensembles. We could summarize three of the essential aspects seen with the results of the experiment as follows:

- The experimental outcomes lead us to divide the prediction horizons into two groups including those where VMD inclusion did not improve the ensemble results; these horizons could be classified as short-term (3, 4, and 5 steps), and mid- and long-term forecast horizons (6, 12, 24, and 48 steps), where the best performance arises with the VMD application after the first ensemble.
- Among the neural network models applied, the Elman neural network shows the best result in all the scenarios of ensemble type and application of VMD, without noticeable differences among the learning functions.
- Over the GARCH models, the principal-axis optimization method can be considered the best overall optimization method for all distributions. We found the normal inverse Gaussian as the most suitable for the experiment proposed regarding the innovation distributions.

The research contributes to the existing literature filling the current gap in studying a wide variety of innovation distribution and optimization methods that can be implemented with GARCH-type models; the present study serves as a guide for selecting the best combinations for forecasting wind speed. Similarly, a variational mode decomposition (VMD) application is proposed in a novel way not seen in the literature: by applying it to the predictions already made by other models, in this case, in ensembles of GARCH-type models.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant number 71671029).

Funding

This work was supported by the National Natural Science Foundation of China [71671029].

Notes on contributors

Angel Colmenares is a Ph.D. candidate at the School of Statistics, Dongbei University of Finance and Economics, China. He holds MSc in Random Models and a BSc in Actuarial Science from the Central University of Venezuela, where he is a

Professor. His main research interests include risk analysis, demography, machine learning, statistical learning, and forecast theory.

Jianzhou Wang is a professor in the School of Statistics at Dongbei University of Finance and Economics, China. He holds a BSc from Northwest Normal University, China, an MSc, and a Ph.D. degree from Lanzhou University, China. He has published over 150 refereed journal papers. His research interests are wind energy forecasting, data mining, machine learning, statistical learning, and forecast theory.

ORCID

Angel Colmenares http://orcid.org/0000-0002-8866-1377
Jianzhou Wang http://orcid.org/0000-0001-9078-7617

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

Acikgoz, H., C. Yildiz, and M. Sekkeli. 2020. An extreme learning machine based very short-term wind power forecasting method for complex terrain. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* 42 (22):2715–30. doi:10.1080/15567036.2020.1755390.

Bilgili, M., and B. Sahin. 2013. Wind speed prediction of target station from reference stations data. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* 35 (5):455–66. doi:10.1080/15567036.2010.512906.

Bollerslev, T. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics* 31 (3):307–27. doi:10.1016/0304-4076(86)90063-1.

Box, G. E. P., and G. M. Jenkins. 1976. *Time series analysis forecasting and control, revised edition*. Oakland, California, USA: Holden-Day.

Brent, R. P. 1973. *Algorithms for minimization without derivatives*. Englewood Cliffs, NJ. USA: Prentice-Hall.

DeMarco, A., and S. Basu. 2018. On the tails of the wind ramp distributions. Wind. *Energy* 21 (10):892–905. doi:10.1002/we.2202.

Dragomiretskiy, K., and D. Zosso. 2014. Variational mode decomposition. *IEEE Transactions on Signal Processing* 62 (3):531–44. doi:10.1109/tsp.2013.2288675.

Elman, J. L. 1990. Finding structure in time. Cognitive. *Science* 14 (2):179–211. doi:10.1207/s15516709cog1402_1.

Engle, R. F. 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica* 50 (4):1987. doi:10.2307/1912773.

Fernández, C., and M. F. J. Steel. 1998. On Bayesian modeling of fat tails and skewness. *Journal of the American Statistical Association* 93 (441):359–71. doi:10.1080/01621459.1998.10474117.

GWEC, Global wind power report; 2019.

Jiang, P., and Z. Liu. 2019. Variable weights combined model based on multi-objective optimization for short-term wind speed forecasting. *Applied Soft Computing* 82:105587. doi:10.1016/j.asoc.2019.105587.

Jiang, P., Z. Liu, X. Niu, Zhang, and A. Lifang. 2020. Combined forecasting system based on statistical method, artificial neural networks, and deep learning methods for short-term wind speed forecasting. *Energy* 119361. doi:10.1016/j.energy.2020.119361.

Johnson, N. L. 1949. Systems of frequency curves generated by methods of translation. *Biometrika* 36 (1–2):149–76. doi:10.1093/biomet/36.1-2.149.

Jordan, M. I. 1997. Serial order: A parallel distributed processing approach. *Advances in Psychology* 471–95. doi:10.1016/S0166-4115(97)80111-2.

Li, C., Z. Xiao, X. Xia, W. Zou, and C. Zhang. 2018. A hybrid model based on synchronous optimization for multistep short-term wind speed forecasting. *Applied Energy* 215:131–44. doi:10.1016/j.apenergy.2018.01.094.

Liu, H., Z. Duan, F. Han, and Y. Li. 2018. Big multistep wind speed forecasting model based on secondary decomposition, ensemble method, and error correction algorithm. *Energy Conversion and Management* 156:525–41. doi:10.1016/j.enconman.2017.11.049.

Liu, H., E. Erdem, and J. Shi. 2011. Comprehensive evaluation of ARMA–GARCH(-M) approaches for modeling the mean and volatility of wind speed. *Applied Energy* 88 (3):724–32. doi:10.1016/j.apenergy.2010.09.028.

Liu, H., X. Mi, and Y. Li. 2018. Smart multistep deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network, and ELM. *Energy Conversion and Management* 159:54–64. doi:10.1016/j.enconman.2018.01.010.

Lojowska, A., D. Kurowicka, G. Papaefthymiou, and L. Van Der Sluis (2010). Advantages of ARMA-GARCH wind speed time series modeling. 2010 IEEE 11th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), Singapore. DOI:10.1109/pmaps.2010.5528979.

Luo, L., H. Li, J. Wang, and J. Hu. 2021. Design of a combined wind speed forecasting system based on decomposition-ensemble and multi-objective optimization approach. *Applied Mathematical Modelling* 89P1:49–72. doi:10.1016/j.apm.2020.07.019.

Nelder, J. A., and R. Mead. 1965. A simplex method for function minimization. *The Computer Journal* 7 (4):308–13. doi:10.1093/comjnl/7.4.308.

Powell, M. (2009). The BOBYQA algorithm for bound constrained optimization without derivatives. Technical Report, Department of Applied Mathematics, and Theoretical Physics.

Powell, M. J. D. 1994. A direct search optimization method that models the objective and constraint functions by linear interpolation. *Advances in Optimization and Numerical Analysis* 51–67. doi:10.1007/978-94-015-8330-5_4.

Qian, Z., Y. Pei, H. Zareipour, and N. Chen. 2019. A review and discussion of decomposition-based hybrid models for wind energy forecasting applications. *Applied Energy* 235:939–53. doi:10.1016/j.apenergy.2018.10.080.

Rehman, U. Naveed, and H. Aftab. 2019. Multivariate variational mode decomposition. *IEEE Transactions on Signal Processing* 1. doi:10.1109/tsp.2019.2951223.

Riedmiller, M., and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. IEEE International Conference on Neural Networks. 1 (1993) 586–91. San Francisco, CA, USA. 10.1109/ICNN.1993.298623.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323 (6088):533–36. https://doi.org/10.1038/323533a0 .

Thomas Harvey Rowan. Functional stability analysis of numerical algorithms. Ph.D. Dissertation. The University of Texas at Austin, USA (1990).

Tian, C., Y. Hao, and J. Hu. 2018. A novel wind speed forecasting system based on hybrid data preprocessing and multi-objective optimization. *Applied Energy* 231:301–19. doi:10.1016/j.apenergy.2018.09.012.

Tian, Z. 2019. Chaotic characteristic analysis of short-term wind speed time series with different time scales. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* 1–15. doi:10.1080/15567036.2019.1649757.

Tian, Z., Y. Ren, and G. Wang. 2018. Short-term wind speed prediction based on improved PSO algorithm optimized EM-ELM. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* 1–21. doi:10.1080/15567036.2018.1495782.

Wang, J., and Y. Li. 2018. Multistep ahead wind speed prediction based on optimal feature extraction, long short-term memory neural network, and error correction strategy. *Applied Energy* 230:429–43. doi:10.1016/j.apenergy.2018.08.114.

Wu, Q., and H. Lin. 2019. Short-term wind speed forecasting based on hybrid variational mode decomposition and least squares support vector machine optimized by bat algorithm model. *Sustainability* 11 (3):652. doi:10.3390/su11030652.

Yang, Z., and J. Wang. 2018. A combination forecasting approach applied in multistep wind speed forecasting based on a data processing strategy and an optimized artificial intelligence algorithm. *Applied Energy* 230:1108–25. doi:10.1016/j.apenergy.2018.09.037.

Yu, X., N. K. Loh, and W. C. Miller. A new acceleration technique for the backpropagation algorithm. IEEE International Conference on Neural Networks. 3 (1993) 1157–61. San Francisco, CA, USA. 10.1109/ICNN.1993.298720.

Zhang, J., Y. Wei, and Z. Tan. 2019. An adaptive hybrid model for short term wind speed forecasting. *Energy*. doi:10.1016/j.energy.2019.06.132.

Zhang, Y., Y. Zhao, C. Kong, and B. Chen. 2019. A new prediction method based on VMD-PRBF-ARMA-E model considering wind speed characteristics. *Energy Conversion and Management* 112254. doi:10.1016/j.enconman.2019.112254.