Short-term wind power forecasting using Ridgelet neural network

3 authors:

Nima Amjady
Semnan University
178 PUBLICATIONS **3,453** CITATIONS

Farshid Keynia
Kerman Graduate University of Technology
29 PUBLICATIONS **887** CITATIONS

Hamidreza Zareipour
The University of Calgary
109 PUBLICATIONS **1,643** CITATIONS
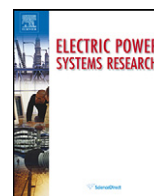
Short-term wind power forecasting using ridgelet neural network

Nima Amjady[a], Farshid Keynia[a], Hamidreza Zareipour[b],*

[a] *Electrical Engineering Department, Semnan University, Semnan, Iran*
[b] *Department of Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary, Calgary, Alberta, Canada*

ARTICLE INFO

ABSTRACT

Rapid growth of wind power generation in many countries around the world in recent years has highlighted the importance of wind power prediction. However, wind power is a complex signal for modeling and forecasting. Despite the performed research works in the area, more efficient wind power forecast methods are still demanded. In this paper, a new prediction strategy is proposed for this purpose. The forecast engine of the proposed strategy is a ridgelet neural network (RNN) owning ridge functions as the activation functions of its hidden nodes. Moreover, a new differential evolution algorithm with novel crossover operator and selection mechanism is presented to train the RNN. The efficiency of the proposed prediction strategy is shown for forecasting of both wind power output of wind farms and aggregated wind generation of power systems.

1. Introduction

In recent years, wind-based power generation has gained significant attention in many countries worldwide because of its environmental benefits. Total installed capacity of wind power in the world reached 159.2 GW at the end of 2009 [1]. In this year, wind generation was 340 TWh, about 2% of worldwide electricity usage. It is estimated that by 2020, about 12% of the world's electricity will be supplied by wind generation [2]. In fact, wind-based generation is the fastest growing source of renewable energy [3]. However, despite significant environmental benefits, wind power could be highly variable because of the variable nature of earth's atmosphere [3]. This variability can put at risk power system reliability, which in turn requires more backup conventional generation in the form of reserve and regulation services. The uncertainty of wind power also poses economical risks for wind farm owners, especially in competitive electricity markets. For instance, in the first week under the New Electricity Trading Arrangements (NETA) in the UK, the penalties incurred by inaccurate wind power forecasts resulted in net negative values for wind power generators [4]. A potential solution to these problems is to improve wind generation forecast accuracy. For this reason, different wind power forecast methods have been proposed in recent years. A review of these methods can be found in [5,6]. As some examples, in [7], artificial neural networks in combination with wavelet transform have been presented for short-term wind power prediction in Portugal. Obtained results for a real world case study are illustrated in [7]. Artificial

neural network with adaptive Bayesian learning and Gaussian process approximation has been proposed for very short-term wind power prediction in [8]. It has been shown that the neural predictor outperforms the persistent technique for the test cases considered in [8]. A two step technique based on Bayesian combination algorithm, and three neural network models, namely, adaptive linear element network (ADALINE), backpropagation (BP) network, and radial basis function (RBF) network has been presented in [9] for wind speed forecasting.

Despite the performed research works in the area, wind power is a complex process to model and forecast, and more accurate wind power prediction methods are still required. The new contribution of this paper can be summarized as follows:

1) A new prediction strategy for short-term wind power forecasting is proposed based on ridgelet neural network (RNN) with a high function approximation capability.
2) A new differential evolution algorithm with new crossover operator and selection mechanism is proposed to train the RNN. Benefiting from the proposed training algorithm, the RNN can efficiently extract the input/output mapping function of the wind power forecast process.

2. Proposed wind power prediction strategy

Structure of the proposed wind power prediction strategy is shown in Fig. 1(a), composed of a feature selection component and a forecast engine. Wind power is a nonlinear mapping function of many candidate inputs including its past values and forecast and past values of the exogenous variables, e.g., wind speed and wind

* Corresponding author. Tel.: +1 403 210 9516; fax: +1 403 282 6855.
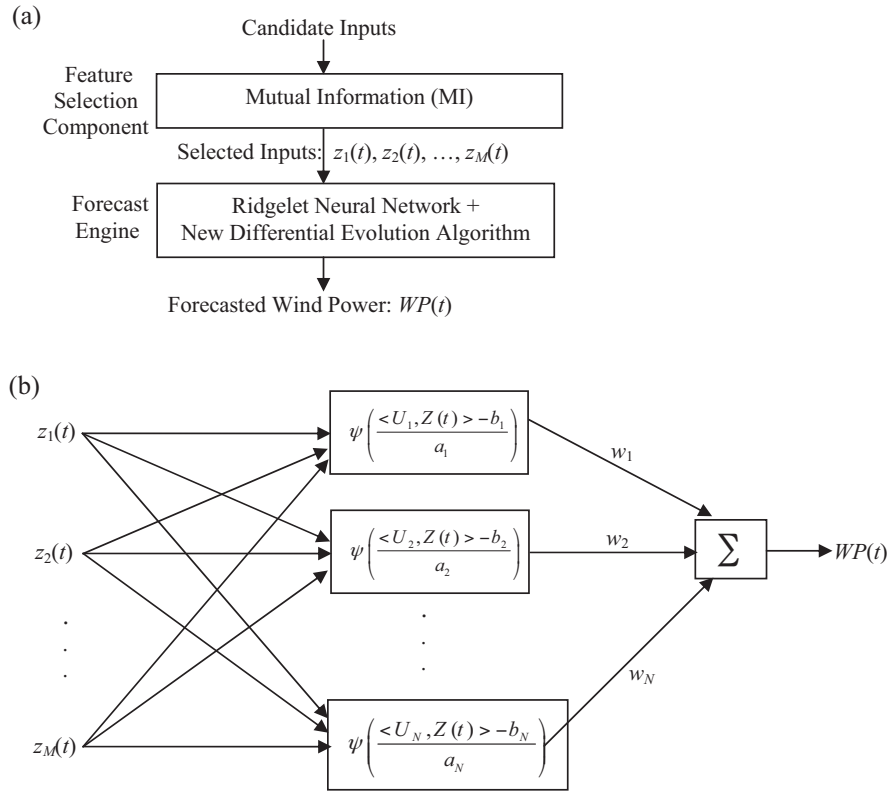*E-mail address:* h.zareipour@ucalgary.ca (H. Zareipour).

Fig. 1. Structure of the proposed wind power prediction strategy (a), structure of its forecasting engine (b).

direction. The feature selection component, based on the information theoretic criterion of mutual information (MI), refines this large set of candidate inputs. For this purpose, the MI based feature selector evaluates the mutual information between each candidate input and forecast feature (here, wind power of the next time interval) using the data mining techniques. More mutual information between a candidate input and forecast feature leads to higher information value of that candidate for the forecast process. Thus, the feature selector can select a minimum subset of the most informative features and filters out unimportant candidate inputs with low information value for the wind power prediction process. This feature selection technique has been presented for price forecast of electricity markets in [10], wherein it has been shown that this method outperforms the well-known linear feature selection technique of correlation analysis, since the MI based feature selector can also evaluate nonlinear dependencies of the forecast feature on the candidate inputs. Moreover, an efficient approach, based on the entropy concept and binomial distribution, to compute MI between two random variables with low computation burden has been presented in [10]. Due to its ability to process nonlinear dependencies and low computation burden, the MI-based feature selector has been employed as the feature selection component of the wind power prediction strategy in this research work (Fig. 1(a)). However, since this feature selector is not a contribution of this paper, it is not further discussed here. The interested reader can refer to [10] for details of the MI based feature selection component. The selected relevant candidate inputs for the wind power forecast process, denoted by $z_1(t), z_2(t), \ldots, z_M(t)$ in Fig. 1(a), are applied as inputs to the forecast engine.

In the following, we focus on the new part, i.e., the forecast engine, of the proposed wind power prediction strategy. Its structure and training mechanism are introduced in Sections 2.1 and 2.2, respectively. Then, the performance of the proposed wind power prediction strategy is described in Section 2.3.

2.1. Structure of the proposed forecast engine

Structure of the RNN applied as the forecast engine is shown in Fig. 1(b). $Z(t)$ is a vector as follows:

$$Z(t) = [z_1(t), z_2(t), \ldots, z_M(t)] \tag{1}$$

Also the function $\langle \cdot, \cdot \rangle$ in Fig. 1(b) represents inner product of its two vector arguments. The activation functions of the hidden units of the RNN, denoted by $\psi(\cdot)$, are called ridgelets. Ridgelets are a set of basis functions, developed by Candes [11], to represent multivariate functions in an efficient and stable way. To define ridgelets, consider a smooth function $\psi: R \to R$ satisfying the following condition:

$$\int \frac{|\hat{\psi}(\omega)|^2}{|\omega|^M} \, d\omega < \infty \tag{2}$$

where $\hat{\psi}(\cdot)$ is the Fourier transform of $\psi(\cdot)$ and M indicates dimension of the space (here, for wind power forecast process, M is equal to the number of inputs of the RNN). Then, $\psi(\cdot)$ is called an admissible neural activation function. Based on $\psi(\cdot)$, the functions

$$\psi_\gamma(Z(t)) = \frac{1}{\sqrt{a}} \psi \left(\frac{\langle U, Z(t) \rangle - b}{a} \right) \tag{3}$$

are referred to as ridge functions or ridgelets [12–14], used as the activation functions of the RNN. Also, the subscript γ represents the parameters of the ridgelet as follows:

$$\gamma = (a, b, U), \quad a, b \in R, \quad a > 0, \quad U \in US^M, \quad \|U\| = 1 \tag{4}$$

where US^M indicates unit sphere in the M dimensional space. In other words, a, b and U represent scale, location and direction of
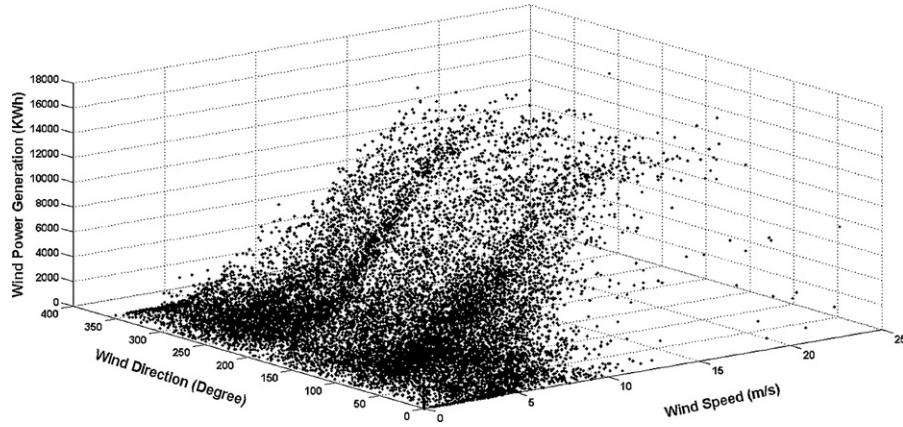
Fig. 2. Wind power with respect to wind speed and direction for Sotavento wind farm in Spain (hourly data from October 24, 2008 to October 24, 2010).

the ridgelet, respectively. Finally, the output of the RNN is obtained as follows:

$$WP(t) = \sum_{i=1}^{N} w_i \cdot \psi \left(\frac{\langle U_i, Z(t) \rangle - b_i}{a_i} \right) \qquad (5)$$

where N indicates the number of the hidden layer nodes as shown in Fig. 1(b). At this stage, the underlying idea behind using the RNN as the wind power forecast engine can be described. Typical three dimensional graph of wind power $WP(t)$ versus wind speed $WS(t)$ and wind direction $WD(t)$ is shown in Fig. 2. Although the number of selected inputs, i.e., M, may be more than two, variation of wind power with respect to more than two inputs cannot be illustrated on a single graph. Since wind speed and wind direction are important drivers for wind power [2], these two inputs are considered for this figure. Spatial inhomogeneity of wind power as a multivariate function can be seen from Fig. 2 such that most of the points are concentrated in a small subset of the space. Moreover, plane singularities of the wind power multivariate function can also be observed from this figure. In general for a hypersurface $F(x, y, z, \ldots) = 0$, the singular points are those at which all the partial derivatives simultaneously vanish. As seen from Fig. 2, the partial derivatives of wind power with respect to its input variables approach zero and the surface becomes nearly flat at the both sides of the surface. This is largely related to the characteristic curve of wind turbines [15]. In the higher dimensional spaces including more inputs (such as lagged values of wind power, wind speed and wind direction in addition to $WS(t)$ and $WD(t)$), this behavior of wind power leads to hyperplane singularities. Now, observe from (5) that the output of the forecast engine is obtained by a linear combination of the ridgelets, where the weights w_i of the RNN (Fig. 1(b)) can be considered as the coefficients of the combination. Ridgelets have proved to be an appropriate basis set for constructing multivariate functions with certain kinds of spatial inhomogeneities [16]. Moreover, in [12], it has been discussed that the RNN can deal with a wide range of functions especially those with hyperplane singularities. Approximating these functions by ridgelets converges more rapidly than traditional Legendre, Fourier and wavelet representations [17]. Thus, the RNN is an appropriate choice for wind power forecast engine considering wind power characteristics.

Following several research works on the RNN, such as [13,17,18], we adopt Meyer wavelet as the $\psi(\cdot)$ function to construct the ridge functions of the RNN based on (3). Meyer wavelet is shown in Fig. 3. Its smoothness, sufficient decay, vanishing mean and oscillatory behavior with diverse oscillations can be seen from the figure leading to high capability of Meyer wavelet to be chosen as the $\psi(\cdot)$ function (used for constructing the basis functions). A detailed discussion about mathematical details of Meyer wavelet and its

advantages for constructing ridgelets can be found in [17,18]. It is noted that although the ridge functions are built from Meyer wavelet, the proposed RNN is different from wavelet neural network. Wavelet lacks description of direction in high dimension, while direction is an important attribute of data [12]. On the other hand, as seen from (3), ridgelet defines direction in addition to scale and location in its expression.

2.2. Training mechanism of the proposed forecast engine

For using the RNN as the forecast engine, it should be first trained. In other words, its free parameters, denoted by vector X,

$$X = [w_1, w_2, \ldots, w_N, a_1, a_2, \ldots, a_N, b_1, b_2, \ldots,$$
$$b_N, U_1, U_2, \ldots, U_N] \qquad (6)$$

should be determined. The first three parts of X, i.e., $w_1, w_2, \ldots, w_N$, $a_1, a_2, \ldots, a_N$ and $b_1, b_2, \ldots, b_N$ contain scalar variables. However, the last part, i.e., $U_1, U_2, \ldots, U_N$ (directions of the ridgelets), includes M dimensional vectors as follows:

$$U_1 = [u_{11}, u_{12}, \ldots, u_{1M}], \quad U_2 = [u_{21}, u_{22}, \ldots, u_{2M}], \quad \ldots,$$
$$U_N = [u_{N1}, u_{N2}, \ldots, u_{NM}] \qquad (7)$$

Thus, the RNN has $N + N + N + M.N = (M + 3).N = NP$ free parameters. The goal of the training mechanism is optimization of these free parameters such that the training error of the RNN to learn the input/output mapping function of the wind power forecast process is minimized.
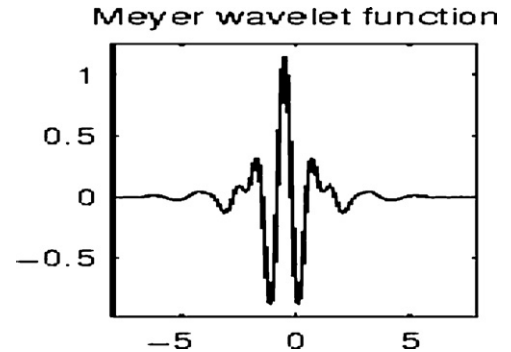

Fig. 3. Meyer wavelet function. (a) Classical DE crossover. (b) The first operator of the proposed NDE crossover shown in (11). (c) The second operator of the proposed NDE crossover shown in (12). (d) The third operator of the proposed NDE crossover shown in (13).

Training of the RNN by traditional neural network learning algorithms (such as gradient search, Newton and quasi-Newton methods) is not an easy task since computation of derivative of the Meyer wavelet (activation function), required in these learning algorithms, is computationally expensive. Moreover, these learning algorithms usually search the solution space in a specific direction (such as the error gradient direction [19]) and thus for learning the nonlinear input/output mapping function of wind power may be trapped in a local minimum (wherein the error gradient becomes zero). To remedy these problems, a new differential evolution algorithm is proposed in this paper for training the RNN. The proposed differential evolution algorithm (called hereafter new DE or NDE) does not require computation of derivative of the Meyer wavelet. Moreover, the NDE can widely search the solution space in various directions and hence, the training process is more likely to escape out of the local minimum. In the following, at first, the classical differential evolution (DE) algorithm is briefly described. Then, the proposed NDE is introduced. Its application for training of the RNN will be presented in the next subsection.

DE is a population based stochastic search technique developed by Storn and Price [20]. As an optimization technique, DE finds increasing interest in recent years due to its search capabilities and adaptability to different solution spaces. Performance of classical DE can be summarized as the following step by step algorithm:

1) Initialization of DE population. Randomly initialize the entire individuals of DE population within the given upper and lower limits. For training the RNN, each individual of the DE population has the structure shown in (6) including $NP = (M + 3).N$ parameters.

2) Compute the value of the objective function, denoted by $OF(\cdot)$, for each individual of DE population. The training error of the RNN is considered as $OF(\cdot)$ of DE that should be minimized.

3) Mutation operation. Consider ith individual of DE population in iteration k denoted by X_i^k. DE mutation operation generates a trial vector V_i^k for it as follows:

$$V_i^k = X_{i1}^k + \beta \cdot (X_{i2}^k - X_{i3}^k), \quad i = 1, 2, \ldots, NI \tag{8}$$

where X_{i1}^k, X_{i2}^k and X_{i3}^k are three randomly selected individuals from DE population in iteration k such that $i \neq i1 \neq i2 \neq i3$; NI indicates number of individuals of DE population; β is the scaling factor of DE, controlling the amplification of the differential variation [20]. Theoretically, $\beta \in (0, \infty)$, but it is usually taken from the range [0.1,1].

4) Crossover operator. Suppose that $V_i^k = [v_{i,1}^k, v_{i,2}^k, \ldots, v_{i,NP}^k]$ is the trial vector produced for the individual $X_i^k = [x_{i,1}^k, x_{i,2}^k, \ldots, x_{i,NP}^k]$. DE crossover operator generates the offspring $Y_i^k = [y_{i,1}^k, y_{i,2}^k, \ldots, y_{i,NP}^k]$ corresponding to the parent X_i^k as follows:

$$y_{i,j}^k = \begin{cases} v_{i,j}^k & \text{if } Rand \leq CR \\ x_{i,j}^k & \text{if } Rand > CR \end{cases}, \quad 1 \leq j \leq NP, \ 1 \leq i \leq NI \tag{9}$$

where CR indicates the crossover rate of DE in the range [0,1] and $Rand$ function generates a random number uniformly distributed in the interval [0,1]. For each $y_{i,j}^k$, a separate random number is generated by this function. After producing the offspring Y_i^k, its $OF(\cdot)$ value or $OF(Y_i^k)$ is computed.

5) Selection mechanism. DE has a deterministic selection mechanism to select X_i^{k+1} of the next generation between X_i^k and its offspring Y_i^k as follows:
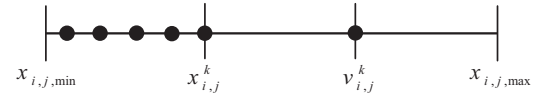
$$X_i^{k+1} = \begin{cases} Y_i^k & \text{if } OF(Y_i^k) < OF(X_i^k) \\ X_i^k & \text{otherwise} \end{cases}, \quad 1 \leq i \leq NI \tag{10}$$
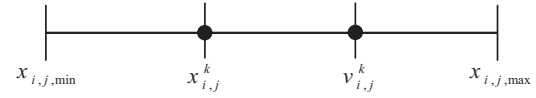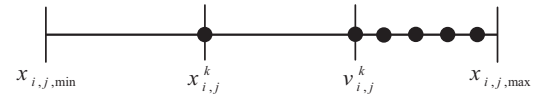

(a) Classical DE crossover
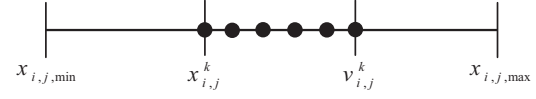
(b) The first operator of the proposed NDE crossover shown in (11)

(c) The second operator of the proposed NDE crossover shown in (12)

(d) The third operator of the proposed NDE crossover shown in (13)

Fig. 4. Comparison of the classical DE crossover and proposed NDE crossover.

6) Stopping condition. If the iteration number $k < k_{\max}$ (maximum number of iterations), increment k and go back to step 3; otherwise return the best individual with the lowest value of $OF(\cdot)$ as the final solution.

In the above algorithm, β, CR, NI, and $k_{\max}$ are user defined settings of DE. More details about classical DE algorithm can be found in [20].

To convert classical DE to the proposed NDE, new crossover operator and selection mechanism (for steps 4 and 5 of the above algorithm) are introduced. As seen from (9), the crossover operator of the classical DE produces a single offspring Y_i^k, while the crossover operator of the proposed NDE generates three children, denoted by $Y_i^{k,1} = [y_{i,1}^{k,1}, y_{i,2}^{k,1}, \ldots, y_{i,NP}^{k,1}]$, $Y_i^{k,2} = [y_{i,1}^{k,2}, y_{i,2}^{k,2}, \ldots, y_{i,NP}^{k,2}]$ and $Y_i^{k,3} = [y_{i,1}^{k,3}, y_{i,2}^{k,3}, \ldots, y_{i,NP}^{k,3}]$, from each parent X_i^k as follows:

$$y_{i,j}^{k,1} = x_{i,j}^k + Rand \times (v_{i,j}^k - x_{i,j}^k), \quad 1 \leq j \leq NP, \ 1 \leq i \leq NI \tag{11}$$

$$y_{i,j}^{k,2} = x_{i,j}^k + Rand \times (x_{i,j}^k - v_{i,j}^k), \quad 1 \leq j \leq NP, \ 1 \leq i \leq NI \tag{12}$$

$$y_{i,j}^{k,3} = v_{i,j}^k + Rand \times (v_{i,j}^k - x_{i,j}^k), \quad 1 \leq j \leq NP, \ 1 \leq i \leq NI \tag{13}$$

To better illustrate the efficiency of the proposed crossover operator, the performance of the classical DE crossover and the above operators (11)–(13) is compared in Fig. 4(a)–(d). In this figure, $x_{i,j,\min}$ and $x_{i,j,\max}$ represent minimum and maximum allowable limits of the jth parameter (i.e., $x_{i,j}$), respectively. Observe from Fig. 4(a) that the classical DE crossover only searches the two points $\{x_{i,j}^k, v_{i,j}^k\}$ of the feasible region of the jth dimension of the solution space for each individual. On the other hand, based on (11)–(13), Fig. 4(b)–(d) shows that the three operators of the proposed crossover searches the area between $x_{i,j}^k$ and $v_{i,j}^k$ and outside
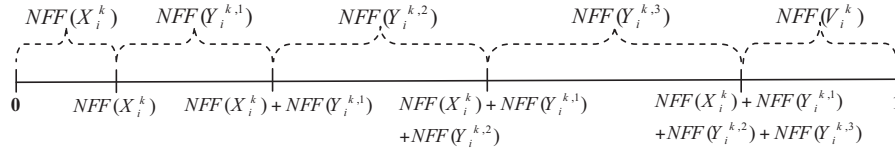
Fig. 5. Partitioning of the range [0,1].

the area from both the sides. In other words, these three operators can much better cover the feasible region of each dimension of the solution space compared with the classical DE crossover leading to significant enhancement of the search capability of the proposed NDE. Thus, the NDE will have a high chance for finding the global optimum of the optimization problem (here, training of the RNN).

The other new aspect of the proposed NDE, compared to the classical DE, is its selection mechanism. Instead of deterministic selection between X_i^k and Y_i^k shown in (10), the proposed NDE selects X_i^{k+1} among X_i^k, $Y_i^{k,1}$, $Y_i^{k,2}$, $Y_i^{k,3}$ and V_i^k in a stochastic manner. For this purpose, at first, a normalized fitness function, denoted by $NFF(\cdot)$, is defined for each of these five individuals as follows:

$$NFF(X_i^k) = \frac{(1/OF(X_i^k))}{(1/OF(X_i^k)) + (1/OF(Y_i^{k,1})) + (1/OF(Y_i^{k,2})) + (1/OF(Y_i^{k,3})) + (1/OF(V_i^k))} \tag{14}$$

$$NFF(Y_i^{k,1}) = \frac{(1/OF(Y_i^{k,1}))}{(1/OF(X_i^k)) + (1/OF(Y_i^{k,1})) + (1/OF(Y_i^{k,2})) + (1/OF(Y_i^{k,3})) + (1/OF(V_i^k))} \tag{15}$$

$$NFF(Y_i^{k,2}) = \frac{(1/OF(Y_i^{k,2}))}{(1/OF(X_i^k)) + (1/OF(Y_i^{k,1})) + (1/OF(Y_i^{k,2})) + (1/OF(Y_i^{k,3})) + (1/OF(V_i^k))} \tag{16}$$

$$NFF(Y_i^{k,3}) = \frac{(1/OF(Y_i^{k,3}))}{(1/OF(X_i^k)) + (1/OF(Y_i^{k,1})) + (1/OF(Y_i^{k,2})) + (1/OF(Y_i^{k,3})) + (1/OF(V_i^k))} \tag{17}$$

$$NFF(V_i^k) = \frac{(1/OF(V_i^k))}{(1/OF(X_i^k)) + (1/OF(Y_i^{k,1})) + (1/OF(Y_i^{k,2})) + (1/OF(Y_i^{k,3})) + (1/OF(V_i^k))} \tag{18}$$

The training error of the RNN or $OF(\cdot) \in \mathbb{R}^+$ and thus each of the five normalized fitness functions in (14)–(18) is in the range of [0,1]. Moreover, we have:

$$NFF(X_i^k) + NFF(Y_i^{k,1}) + NFF(Y_i^{k,2}) + NFF(Y_i^{k,3}) + NFF(V_i^k) = 1 \tag{19}$$

Thus, the range [0,1] can be partitioned into the normalized fitness functions as shown in Fig. 5 and the proposed selection mechanism selects X_i^{k+1} as follows:

$$X_i^{k+1} = \begin{cases} X_i^k & \text{if } 0 \le Rand \le NFF(X_i^k) \\ Y_i^{k,1} & \text{if } NFF(X_i^k) < Rand \le NFF(X_i^k) + NFF(Y_i^{k,1}) \\ Y_i^{k,2} & \text{if } NFF(X_i^k) + NFF(Y_i^{k,1}) < Rand \le NFF(X_i^k) + NFF(Y_i^{k,1}) + NFF(Y_i^{k,2}) \\ Y_i^{k,3} & \text{if } NFF(X_i^k) + NFF(Y_i^{k,1}) + NFF(Y_i^{k,2}) < Rand \le NFF(X_i^k) + NFF(Y_i^{k,1}) + NFF(Y_i^{k,2}) + NFF(Y_i^{k,3}) \\ V_i^k & \text{if } NFF(X_i^k) + NFF(Y_i^{k,1}) + NFF(Y_i^{k,2}) + NFF(Y_i^{k,3}) < Rand \le 1 \end{cases} \tag{20}$$

In other words, the individual among $\{X_i^k, Y_i^{k,1}, Y_i^{k,2}, Y_i^{k,3}, V_i^k\}$ is selected that the generated random number by the *Rand* function falls in the segment of its $NFF(\cdot)$. It is noted that an individual with a lower value of $OF(\cdot)$ has a higher value of $NFF(\cdot)$ based on (14)–(18). In other words, this individual is considered as a fitter individual and occupies a larger portion of the interval [0,1], shown in Fig. 5, leading to its more chance to be selected as X_i^{k+1}. Therefore, the proposed selection mechanism selects the individuals of the next iteration in a stochastic manner giving more chance to fitter individuals for the selection. This selection mechanism is more consistent with the stochastic search nature of a DE algorithm than the deterministic selection method of (10) and thus the population of the NDE better evolves along its iterations producing more effective final solution.

2.3. Performance of the whole proposed prediction strategy

Following many previous research works in the area, such as [2,4,5,15,21], and without losing generality, this paper focuses on hourly forecasts. Also, the forecasting horizon of 24 h ahead is adopted in this paper.

As described in Section 2.2, the training mechanism of the NDE proceeds in the successive iterations to minimize the training error of the RNN, which is $OF(\cdot)$ of the NDE. However, only minimizing the training error of a neural network (NN) in the training phase may lead to overfitting problem in which the NN begins to memorize the training samples instead of learning them. When overfitting

occurs in a NN, the training error continues to decrease and it seems that the training process progresses, while in fact the generalization capability of the NN degrades and it loses its prediction ability for unseen forecast samples (generalization is a measure of how well the NN performs on the actual problem once training is complete [19]). To remedy this problem, the generalization performance of the NN should also be monitored along its training phase. However, since forecast error is not available in the training phase, validation error is used as an approximation of it. For NN learning, validation samples are a subset of training period that are not used for the adjustment of the free parameters of the NN (such as its weights) and retained unseen for the NN. Thus, error of validation samples or validation error can give an estimate of the NN error for unseen forecast samples (here, wind powers of 24 h ahead). Compared with the training error, validation error can better measure generalization capability of a NN and avoid overfitting problem [19]. However, validation samples should be as similar as possible to forecast samples so that the validation error can give a true estimate

Table 1
RMSE and MMAPE results for prediction of aggregated wind power of Irish power system in September and October 2010.

Test month	Persistence method		Irish EirGrid company		Proposed strategy	
	RMSE (MW)	MMAPE (%)	RMSE (MW)	MMAPE (%)	RMSE (MW)	MMAPE (%)
September 2010	125.33	24.11	118.21	23.93	77.21	15.87
October 2010	164.12	25.91	160.56	23.71	73.28	14.69
Average	144.73	25.01	139.38	23.82	75.24	15.28

of the prediction error. Considering short run trend characteristic of wind speed/power signals (dependency on the previous neighboring hours' values), 24 hourly wind powers related to the day before the forecast day (the closest samples to the forecast samples) are taken into account as the validation samples in this paper.

Considering the above explanations, application of the proposed prediction strategy, depicted in Fig. 1(a), for wind power forecast can be summarized as the following *execution procedure*:

1) Feature selection phase. The inputs of the forecast engine are selected by the MI based feature selection component among the set of candidate inputs.
2) Preparation of training and validation samples. In the numerical experiments of this paper, 50 days prior to each forecast day are considered as its corresponding training period. From which, the first 49 days including $49 \times 24 = 1176$ hourly samples are taken into account as the training samples and 24 hourly samples of the last day (the day before the forecast day) are considered as the validation samples. These 1176 hourly training samples and 24 hourly validation samples are constructed by the selected inputs based on the historical data.
3) Training phase. The proposed NDE trains the RNN (the forecast engine). Error of the constructed training samples or training error is taken into account as its $OF(\cdot)$. The NDE is implemented according to the step by step algorithm of the classical DE considering that the proposed crossover operator and the proposed selection mechanism should be applied in the steps 4 and 5 instead of (9) and (10), respectively. Moreover, step 6 (stopping condition) of the step by step algorithm is also modified. Instead of setting a maximum number of iterations k_{max}, the validation error of the RNN (the validation error of the best individual of the NDE owning the lowest validation error) is also monitored along the iterations. Whenever the validation error starts to increase, the generalization performance of the RNN begins to degrade, indicating the occurrence of overfitting problem, and so the training process of the RNN by the NDE should be terminated at this iteration (also known as the early stopping technique). The NDE iteration with the minimum validation error brings the final results of the RNN training, wherein it is expected that the generalization capability of the RNN is maximized. The best individual of the NDE in this iteration determines all free parameters of the RNN shown in (6). Now, the forecast engine is trained and ready for the wind power forecast process. In addition to avoiding from overfitting problem, the proposed stopping condition does not require setting k_{max} (the maximum number of NDE iterations is automatically determined based on the RNN validation error), which is its another advantage.
4) Prediction phase. Wind power forecast of the RNN is obtained from (5). However, as seen from Fig. 1(b), the proposed forecast engine has only one output. Multi-period forecast, e.g., prediction of hourly wind power for the next 24 h, is reached via recursion, i.e., by feeding input variables with the outputs of the forecast engine. For instance, predicted wind power for the first hour is used as $WP(t-1)$ for the wind power forecast of the second hour provided that $WP(t-1)$ is among the selected inputs.

3. Numerical results

The proposed prediction strategy is tested for both wind power forecast of Sotavento wind farm in Spain and prediction of aggregated wind power of Irish power system using their real data obtained from [22,23], respectively. Moreover, some wind speed forecast results of the proposed strategy are presented to also illustrate its effectiveness for this forecast process as well.

3.1. Wind power forecast results for the Irish power system

Since the first wind farm project was realized in 1992, 110 wind farms with a total capacity of 1379 MW have been installed at the end of June, 2010 in Ireland. More details about this test case can be found in [23]. In the proposed prediction strategy, the set of candidate inputs (Fig. 1) for this test case includes 50 lagged wind power values. The exogenous variables of the weather parameters are not used for the Irish test case, since its wind farms are spread around the country. Thus, the measured weather parameters of a single weather station may not be informative for the prediction of aggregated wind power of the entire system and the weather parameters of all of these wind farms are not available for us. However, wind generation of a power system with many wind farms (such as the Irish power system) is a more smooth signal with less sudden changes compared with the wind power of a single wind farm [24]. This is due to the aggregation of wind farms' generations leading to higher inertia for the system wind generation. A similar situation can be seen comparing system loads with bus loads [25]. Thus, only considering lagged wind power values is reasonable for wind power forecast of such a power system. The MI based feature selection component refines the set of candidate inputs and selects the most informative features for the forecast engine as described in the previous section.

Obtained forecast errors from the proposed strategy for prediction of aggregated wind power of Irish power system in two test months of September 2010 and October 2010 are shown in Table 1 and compared with the forecast errors of the Irish EirGrid company and persistence method. EirGrid company holds licenses as independent electricity Transmission System Operator (TSO) and Market Operator (MO) in the wholesale trading system in Ireland. The wind power forecast results of EirGrid, reported in Table 1, have been taken from their website [23]. Persistence is the benchmark method most frequently used for comparison in the wind power forecast research works such as [2,15,21]. In this method, the forecast for all future time intervals in the forecast horizon is set to the last measured value.

Two error criteria are adopted for the numerical experiments of this paper. The first one is root mean square error (RMSE), frequently used as error metric in wind power/speed forecast research works, defined as follows:

$$RMSE = \left[\frac{1}{NH} \sum_{t=1}^{NH} (S_{ACT(t)} - S_{FOR(t)})^2 \right]^{1/2} \quad (21)$$

where $S_{ACT(t)}$ and $S_{FOR(t)}$ indicate the actual and forecast values of the signal (wind power or wind speed) for hour t. Here, NH (number of hours) is 720 and 744 for the September and October test

Table 2
Comparison of the proposed NDE with SA, GA, PSO and DE for training of the forecast engine (RNN) on the test case of Table 1.

Test month	SA		GA		PSO		DE		Proposed	
	RMSE (MW)	MMAPE (%)	RMSE (MW)	MMAPE (%)	RMSE (MW)	MMAPE (%)	RMSE (MW)	MMAPE (%)	RMSE (MW)	MMAPE (%)
September 2010	127.91	24.22	118.41	21.54	111.84	19.83	102.42	17.19	77.21	15.87
October 2010	139.64	25.61	119.71	22.14	110.48	19.76	100.24	16.09	73.28	14.69
Average	133.77	24.91	119.06	21.84	111.16	19.79	101.33	16.64	75.24	15.28

months, respectively. It is noted that in the day ahead forecast process, the historical data is updated at the end of each day. Then, the feature selection phase, preparation of training and validation samples and training phase (steps 1–3 of the *execution procedure* mentioned at the end of the previous section) are performed by the updated data followed by the prediction of hourly wind power values of the next day (prediction phase or step 4 of the *execution procedure*). However, RMSE in (21) is computed for one month to give a better evaluation of the forecast error over on a longer period.

Mean absolute percentage error (MAPE) is one of most commonly used error measures for forecast processes, which is defined as follows:

$$MAPE = \frac{1}{NH}\sum_{t=1}^{NH}\frac{|S_{ACT(t)} - S_{FOR(t)}|}{S_{ACT(t)}} \times 100 \qquad (22)$$

where $S_{ACT(t)}$, $S_{FOR(t)}$ and NH are as defined for (21). However, MAPE cannot be directly used to measure prediction error of wind power/speed forecast, since $S_{ACT(t)}$ may be very small or even zero for these forecast processes. Thus, MAPE could be very large or even reach infinity if the forecast is not zero. To remedy this problem, a modified version of MAPE, denoted by MMAPE, is defined as follows:

$$MMAPE = \frac{1}{NH}\sum_{t=1}^{NH}\frac{|S_{ACT(t)} - S_{FOR(t)}|}{S_{AVE\text{-}ACT}} \times 100,$$

$$S_{AVE\text{-}ACT} = \frac{1}{NH}\sum_{t=1}^{NH}S_{ACT(t)} \qquad (23)$$

In other words, the average value of the signal over the evaluation period (here, one month) is used in the denominator to avoid the problem caused by very small or zero values of wind power/speed.

Observe from Table 1 that both the RMSE and MMAPE of the wind power forecast of the proposed strategy, only employing the past values of wind power, are considerably lower than the RMSE and MMAPE of the forecast of the persistence method and EirGrid Company in both the test months. Moreover, the average RMSE and average MMAPE of the proposed strategy, reported in the last row of Table 1, are $(144.73 − 75.24)/144.73 = 48.0\%$ and $(25.01 − 15.28)/25.01 = 38.9\%$ lower than those of the persistence method and $(139.38 − 75.24)/139.38 = 46.0\%$ and $(23.82 − 15.28)/23.82 = 35.9\%$ lower than those of the EirGrid Company. To better illustrate wind power prediction accuracy of the proposed strategy, its forecast is compared with the forecast of the EirGrid Company (owning better results than the other benchmark method of Table 1) in Fig. 6(a) and (b). Observe from these figures that the predictions of the proposed strategy are more accurate with lower deviations from the real curves compared with predictions of the EirGrid company in both the test months. The comparisons of Table 1 and Fig. 6(a) and (b) reveal the wind power forecast capability of the proposed strategy.

To evaluate the effectiveness of the proposed NDE for training of the RNN, it is replaced by simulated annealing (SA), genetic algorithm (GA), particle swarm optimization (PSO) and classical DE (which are four well-known stochastic search techniques) in the proposed prediction strategy and their obtained results are

compared in Table 2. GA in [12] and PSO in [14,16] have been used for training of RNN. For the sake of a fair comparison, all training methods of Table 2 have the same set of candidate inputs, the same feature selection technique and the same training, validation and forecast samples. As seen, the proposed NDE results in the lowest RMSE and lowest MMAPE among all methods of Table 2 in both the test months illustrating the effectiveness of the NDE for training of the forecast engine.

3.2. Wind power/speed forecast results for the Sotavento wind farm

Sotavento wind farm, located in Galicia, Spain, has a line of 24 wind turbines of 5 different technologies. Its nominal power is 17.56 MW and predicted annual production is 38.5 MWh [22]. The set of candidate inputs for this test case includes 50 lagged wind power, wind speed and wind direction values as well as predicted values for the wind speed and direction (totally, $50 \times 3 + 2 = 152$ candidate inputs). For this single wind farm, the weather parameters' data can be obtained from its weather station [22]. Effective inputs for the wind power forecast process are selected among this set by the feature selection technique.

In the previous case study, the effectiveness of the proposed strategy for prediction of aggregated wind power of a power system and efficiency of the NDE to train the proposed forecast engine have been shown. For the sake of conciseness and avoiding repetition,
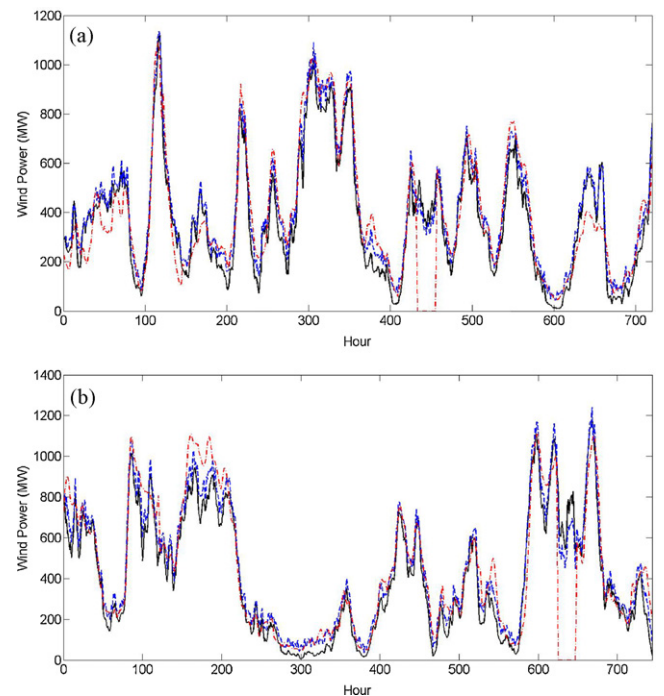

Fig. 6. Curves of the real values (black, solid line), forecast of the proposed strategy (blue, dashed line) and forecast of EirGrid Company (red, dash-dot line) for the test months of September 2010 (a) and October 2010 (b). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

Table 3
RMSE and MMAPE results for wind power forecast of Sotavento wind farm in April–July 2010.

Test month	Persistence method		Multivariate ARIMA		RBF		MLP		Proposed strategy	
	RMSE (MW)	MMAPE (%)	RMSE (MW)	MMAPE (%)	RMSE (MW)	MMAPE (%)	RMSE (MW)	MMAPE (%)	RMSE (MW)	MMAPE (%)
April 2010	1.124	35.91	0.843	28.74	0.594	25.08	0.514	22.44	0.463	7.75
May 2010	0.848	30.84	0.742	27.21	0.516	18.11	0.618	19.82	0.435	11.43
June 2010	0.784	34.33	0.702	28.91	0.593	26.45	0.521	25.24	0.437	16.06
July 2010	0.826	36.84	0.691	29.54	0.501	27.75	0.467	26.14	0.376	9.33
Average	0.895	34.48	0.744	28.60	0.551	24.35	0.530	23.41	0.428	11.14

Table 4
RMSE and MMAPE results for wind speed forecast of Sotavento wind farm in April–July 2010.

Test month	Persistence method		Multivariate ARIMA		RBF		MLP		Proposed strategy	
	RMSE (m/s)	MMAPE (%)	RMSE (m/s)	MMAPE (%)	RMSE (m/s)	MMAPE (%)	RMSE (m/s)	MMAPE (%)	RMSE (m/s)	MMAPE (%)
April 2010	9.11	99.23	7.92	90.15	6.12	69.34	6.33	70.14	2.13	23.08
May 2010	9.21	98.15	7.52	84.34	6.92	76.91	5.38	58.12	1.74	22.23
June 2010	9.34	97.69	8.02	85.59	6.48	70.29	6.51	73.39	1.50	21.63
July 2010	8.56	98.77	6.34	74.25	6.31	72.34	5.93	66.91	1.51	20.67
Average	9.05	98.46	7.45	83.58	6.46	72.22	6.04	67.14	1.72	21.90

Table 5
RMSE and MMAPE results of the proposed strategy for wind power forecast of Sotavento wind farm in April–July 2010 with different training periods.

Test month	30 days		40 days		50 days		60 days		70 days	
	RMSE (m/s)	MMAPE (%)	RMSE (m/s)	MMAPE (%)	RMSE (m/s)	MMAPE (%)	RMSE (m/s)	MMAPE (%)	RMSE (m/s)	MMAPE (%)
April 2010	1.051	22.43	0.581	13.08	0.463	7.75	0.562	12.82	0.752	18.23
May 2010	1.004	24.49	0.554	13.91	0.435	11.43	0.532	12.72	0.703	16.25
June 2010	0.841	20.42	0.573	17.41	0.437	16.06	0.501	17.02	0.686	20.07
July 2010	0.862	17.91	0.416	12.27	0.376	9.33	0.420	13.44	0.662	16.02
Average	0.939	21.31	0.531	14.17	0.428	11.14	0.504	14.00	0.701	17.64

the other aspects of the proposed strategy, including its ability for prediction of wind power and speed of a wind farm, are evaluated in this case study. Moreover, the test months are also changed to illustrate the effectiveness of the proposed strategy for different months and seasons.

Wind power and speed forecast errors of the proposed strategy for Sotavento wind farm in the four test months of April 2010, May 2010, June 2010 and July 2010 are shown in Tables 3 and 4, respectively, and compared with the results of some other well-known forecast methods. The benchmark methods of Tables 3 and 4 include persistence method, multivariate ARIMA time series, radial basis function (RBF) neural network and multi-layer perceptron (MLP) neural network trained by the efficient Levenberg–Marquardt (LM) learning algorithm. These forecast methods have been used in many other wind power forecast research works such as [2,4–6,15,21]. All forecast methods of Tables 3 and 4 have the same set of candidate inputs, feature selection technique and training period (except the persistence method that does not require feature selection and training process), since the purpose of these numerical experiments is comparison of the efficiency of different forecast engines. RMSE and MMAPE in these tables are as defined in (21) and (23), respectively. Observe that the proposed prediction strategy outperforms all other methods of Tables 3 and 4 for both wind power and wind speed forecasts. RMSE and MMAPE of the proposed strategy are significantly lower than RMSE and MMAPE of all other methods of Tables 3 and 4 in the four test months indicating efficiency of the proposed strategy for both wind power and wind speed forecasts.

A short training period including a small number of training samples usually causes that a neural network based forecast engine cannot correctly learn the input/output mapping function of the forecast process. On the other hand, a long training period includes far historical data. Considering time variant behavior of wind power mapping function, far training samples usually are not informative and may even be misleading for a wind power forecast engine. Thus,

to select the training period, we started with a short period and then gradually increased it. The *execution procedure* was implemented with each training period and the obtained wind power forecast results were recorded. Sample results of this analysis with training periods of 30 days, 40 days, 50 days, 60 days and 70 days prior to the forecast day are shown in Table 5. It can be observed from this table that 50 days training period results in the lowest forecast errors in terms of both RMSE and MMAPE. Thus, this training period has been selected leading to 24 validation samples (hourly samples of the day before the forecast day) and $(50 - 1) \times 24 = 49 \times 24 = 1176$ training samples.

In the performed numerical experiments, the user defined settings of the proposed strategy (defined in Section 2.2) are selected based on a few trial runs. After that, the whole computation time of the proposed prediction strategy, including steps 1–4 of the *execution procedure*, is about 1 min for the test cases of this paper. This setup time, measured on a simple hardware set of Pentium P4 3.6 GHz with 4 GB RAM, is completely acceptable within a day-ahead decision making framework. Even, it is acceptable for shorter forecast intervals such as hour ahead or 10-min ahead predictions.

4. Conclusion

Wind power is a nonlinear multivariate function owning spatial inhomogeneity and hyperplane singularities. Ridgelets constitute an efficient basis set for constructing such functions. Considering this matter, a RNN, having ridge functions as the activation functions of the hidden layer nodes, is presented for wind power forecast in this paper. For training of the suggested forecast engine (determination of its free parameters) a new stochastic search technique, named NDE, is proposed. The proposed NDE has a low computation burden and only requires a small set of training and validation samples. Moreover, it can widely search the solution space in various directions increasing the chance of finding the global optimum of the training problem. Efficiency of

the proposed prediction strategy for wind power forecast of both single wind farms and power systems and wind speed forecast is extensively illustrated.

References

[1] A report prepared by World Wind Energy Association, World Wind Energy Report 2009, March 2010, http://www.wwindea.org/home/images/stories/worldwindenergyreport2009_s.pdf.

[2] S. Fan, J.R. Liao, R. Yokoyama, L. Chen, W.-J. Lee, Forecasting the wind generation using a two-stage network based on meteorological information, IEEE Trans. Energy Convers. 24 (2) (2009) 474–482.

[3] J.W. Taylor, P.E. McSharry, R. Buizza, Wind power density forecasting using ensemble predictions and time series models, IEEE Trans. Energy Convers. 24 (3) (2009) 775–782.

[4] G. Sideratos, N. Hatziargyriou, Using radial basis neural networks to estimate wind power production, IEEE Power Engineering Society General Meeting (June 2007) doi:10.1109/PES.2007.385812.

[5] A. Costa, A. Crespo, J. Navarro, G. Lizcano, H. Madsen, E. Feitosa, A review on the young history of the wind power short-term prediction, Renew. Sustain. Energy Rev. 12 (6) (2008) 1725–1744.

[6] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, Z. Yan, A review on the forecasting of wind speed and generated power, Renew. Sustain. Energy Rev. 13 (4) (2009) 915–920.

[7] J.P.S. Catalao, H.M.I. Pousinho, V.M.F. Mendes, Short-term wind power forecasting in Portugal by neural networks and wavelet transform, Renew. Energy 36 (April (4)) (2011) 1245–1251.

[8] R. Blonbou, Very short-term wind power forecasting with neural networks and adaptive Bayesian learning, Renew. Energy 36 (March (3)) (2011) 1118–1124.

[9] G. Li, J. Shi, J.Y. Zhou, Bayesian adaptive combination of short-term wind speed forecasts from neural network models, Renew. Energy 36 (January (1)) (2011) 352–359.

[10] N. Amjady, F. Keynia, Day-ahead price forecasting of electricity markets by mutual information technique and cascaded neuro-evolutionary algorithm, IEEE Trans. Power Syst. 24 (1) (2009) 306–318.

[11] E.J. Candes, Ridgelet Theory and Applications, PhD Thesis, Department of Statistics, Stanford University, 1998.

[12] S. Yang, M. Wang, L. Jiao, A linear ridgelet network, Neurocomputing 73 (2009) 468–477.

[13] S. Yang, M. Wang, L. Jiao, Approximation of functions with spatial inhomogeneity based on true ortho-ridgelet neural network, Appl. Soft Comput. 11 (2) (2011) 2444–2451.

[14] R. Su, L. Kong, S. Song, P. Zhang, K. Zhou, J. Cheng, A new ridgelet neural network training algorithm based on improved particle swarm optimization, Third International Conference on Natural Computation (ICNC 2007) Haikou, Hainan, China, August 24–27, 2007, ISBN: 0-7695-2875-9.

[15] I.G. Damousis, M.C. Alexiadis, J.B. Theocharis, P.S. Dokopoulos, A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation, IEEE Trans. Energy Convers. 19 (2) (2004) 352–361.

[16] S. Yang, M. Wang, L. Jiao, Ridgelet kernel regression, Neurocomputing 70 (2007) 3046–3055.

[17] Ridgelets: a key to higher-dimensional intermittency, http://www-stat.stanford.edu/~donoho/Reports/1999/RoySoc.pdf.

[18] Digital Ridgelet Transform based on True Ridge Functions, http://www.famaf.unc.edu.ar/~flesia/PDF-files/DonohEetAl_ridgelets.pdf.

[19] D.R. Hush, B.G. Horne, Progress in supervised neural networks, IEEE Signal Process. Mag. 10 (1) (1993) 8–39.

[20] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.

[21] R.G. Kavasseri, K. Seetharaman, Day-ahead wind speed forecasting using f-ARIMA models, Renew. Energy 34 (5) (2009) 1388–1393.

[22] Sotavento wind farm data http://www.sotaventogalicia.com/tiempo_real/english/instantaneos.php.

[23] Ireland wind power data http://www.eirgrid.com/operations/systemperformancedata.

[24] M. Milligan, et al., Wind power myths debunked, IEEE Power Energy Mag. 7 (6) (2009) 89–99.

[25] N. Amjady, Short-term bus load forecasting of power systems by a new hybrid method, IEEE Trans. Power Syst. 22 (1) (2007) 333–341.