Applied Energy 98 (2012) 574–583

Contents lists available at SciVerse ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Monitoring of wind farms' power curves using machine learning techniques

Antonino Marvuglia [a,*], Antonio Messineo [b]

[a] CRP Henri Tudor/CRTE, 66 rue de Luxembourg, L-4002 Esch/Alzette, Luxembourg
[b] Faculty of Engineering & Architecture, Kore University of Enna, Italy

HIGHLIGHTS

- We compare different non-parametric models of a wind farm power curve.
- The models can be used as a reference profile for on-line monitoring of the power generation process.
- The approach followed is data-driven, based on machine learning techniques (more specifically neural networks).
- The statistical quality control chart approach was used to detect anomalies in the power generation.
- The control chart method was applied to detect anomalies in the wind farm power output.

ARTICLE INFO

Article history:
Received 11 November 2011
Received in revised form 11 March 2012
Accepted 24 April 2012
Available online 22 May 2012

Keywords:
Wind farm
Power curve
Data-driven
Neural network
Machine learning

ABSTRACT

The estimation of a wind farm's power curve, which links the wind speed to the power that is produced by the whole wind farm, is a challenging task because this relationship is nonlinear and bounded, in addition to being non-stationary due for example to changes in the site environment and seasonality. Even for a single wind turbine the measured power at different wind speeds is generally different than the rated power, since the operating conditions on site are generally different than the conditions under which the turbine was calibrated (the wind speed on site is not uniform horizontally across the face of the turbine; the vertical wind profile and the air density are different than during the calibration; the wind data available on site are not always measured at the height of the turbine's hub).

The paper presents a data-driven approach for building an equivalent steady state model of a wind farm under normal operating conditions and shows its utilization for the creation of quality control charts at the aim of detecting anomalous functioning conditions of the wind farm. We use and compare three different machine learning models – viz. a self-supervised neural network called GMR (Generalized Mapping Regressor), a feed-forward Multi Layer Perceptron (MLP) and a General Regression Neural Network (GRNN) – to estimate the relationship between the wind speed and the generated power in a wind farm. GMR is a novel incremental self-supervised neural network which can approximate every multidimensional function or relation presenting any kind of discontinuity; MLPs are the most widely used state-of-the-art neural network models and GRNNs belong to the family of kernel neural networks.

The methodology allows the creation of a non-parametric model of the power curve that can be used as a reference profile for on-line monitoring of the power generation process, as well as for power forecasts. The results obtained show that the non-parametric approach provides fair performances, provided that a suitable pre-processing of the input data is accomplished.

1. Introduction and state of the art

Diminishing fossil fuel reserves coupled with steadily rising energy demand has created new challenges for the energy sector, resulting in this latter experiencing a transition phase worldwide. The problem is compounded with rising input costs and environmental externalities that exacerbate the woes. In this context, along with the opportunities offered by the renewable energy market, new solutions have also been sought to enhance energy security, diversification, reliability and affordability of gas supply [1,2], together with actions oriented towards energy saving and recovery in the industrial field [3].

Among the renewable energy sources, wind is one of the most promising, due to its competitive economic and environmental payback times. However, wind energy generation is difficult to manage, due to the irregular nature of the wind flows. Thus it is obvious that a reliable wind power forecasting model, capable of giving rapid answers, is important for energy management purposes.

* Corresponding author. Tel.: +352 4259914652; fax: +352 425991555.
E-mail address: antonino.marvuglia@tudor.lu (A. Marvuglia).

0306-2619/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.apenergy.2012.04.037

Nomenclature

Latin

w	neural network's synaptic weight
n_o	number of neural network's output units
N_p	number of input samples in the training set
N_t	number of data samples in the test set
$\widehat{Y}_{i,p}$	response of the i-th neural network's output unit calculated using the p-th input sample vector
$Y_{i,p}$	i-th target value corresponding to the p-th input sample vector
$\widehat{Y}_i$	network's prediction corresponding to the i-th data sample
$\overline{\widehat{Y}}$	mean value of the network's predictions for the test set
Y_i	target value corresponding to the i-th data sample
$\overline{Y}$	mean value of the test set data
$\mathbf{X}_p$	p-th input sample vector
$\boldsymbol{w}$	vector of the whole set of network's synaptic weights

Greek

λ	penalty term for regularization
σ	kernel width
σ_{pred}	standard deviation of the prediction test set data
σ_{test}	standard deviation of the test set data
$\chi^2_{\alpha/2,N_{Test}-1}$	right $\alpha/2$ percentage points of the chi-square distribution

Data mining and machine learning (ML) techniques have been successfully applied in many different domains. The environmental sciences have greatly benefited from the use of such tools in different fields [4–10].

Numerous applications of data mining in the wind energy field have also proven to be effective for decision making support [11–17]. A review of current methods in forecasting of wind power generation is presented by Foley et al. [18].

An interesting aspect of current research on wind power generation is the definition and implementation of accurate models to predict the energy output of a wind farm, rather than single wind turbines. Accurate wind power forecasting and prediction reduces the risk of uncertainty and allows for better grid planning and integration of wind into power systems.

Wind power forecasting and prediction tools are therefore invaluable because they enable better dispatch, scheduling and unit commitment of thermal generators, hydro and energy storage plant. Overall they reduce the financial and technical risk of uncertainty of wind power production for all electricity market participants [18].

Researchers have applied different methodologies in studying wind farms. In [19] the author built a model to predict the power produced by a wind farm using the data from the weather prediction model (HIRLAM) and the local weather model (WASP). In [20] a regression and a neural network (NN) model are compared in order to estimate a turbine's power curve. A novel approach for the analysis and modeling of wind vector fields was introduced by Goh et al. [21] and developed by Mandic et al. [22]. In these papers the wind vector is represented as a complex-valued quantity and wind speed and direction are modeled simultaneously. In [23] a new algorithm based on fuzzy logic was applied to estimate a wind turbine power curve. In [24] a variety of different approaches have been used to build prediction models that characterize the power curve of a wind farm. In particular, a nonlinear parametric model of a power curve of the wind farm was built using an evolutionary strategy algorithm, which was then applied to monitor the performance of a wind farm, as well as to improve the quality of the data by filtering abnormal data (i.e. states where the wind speed is too low or high, turbines undergo maintenance or are affected by environmental issues, and low power output due to control issues). [25] apply and compare parametric and non-parametric models to solve the same problem. An extensive review of the existing wind speed and related generated power forecasting approaches can be found in [26].

Pousinho et al. [27] propose a new hybrid approach for short-term wind power prediction, combining particle swarm optimization (PSO) and adaptive-network-based fuzzy inference system, for short-term wind power prediction. The mean absolute percentage error (MAPE) obtained in the study has an average value of 5.41%, with an average computation time lower than 5 s.

Hong et al. [28] propose a new method of wind power and speed forecasting using a multi-layer feed-forward neural network (MFNN) to develop forecasting in time-scales that can vary from a few minutes to an hour. Inputs for the MFNN are modeled by fuzzy numbers followed by a simultaneous perturbation stochastic approximation (SPSA) algorithm to train the MFNN.

Kusiak and Zheng [29] applied data-mining algorithms to identify dynamic process models for power output, the power factor, and the rotor speed of a wind turbine from the actual wind farm data. Then an evolutionary strategy algorithm solved the bi-objective optimization model (in which active power and power factor are both optimized) with constraints to find the optimal settings of the pitch angle and generator torque.

Blonbou [30] presents an adaptive very short-term wind power prediction scheme that uses an artificial neural network as predictor along with adaptive Bayesian learning and Gaussian process approximation. The author evaluated two test cases: the first with prediction horizons of 5, 10 and 15 min and the second with prediction horizons of 10, 20 and 30 min. Although the improvement of the model with respect to the persistent model was good, at some time the discrepancy between the forecasts and the actual wind power could be significant (with a Normalized Mean Absolute Error of 6.3% for the test case n.1, with a forecast horizon of 15 min, and 14.3% for the test case n.2, with a forecast horizon of 30 min).

The paper by Catalão et al. [31] proposes artificial neural networks in combination with wavelet transform for short-term wind power forecasting. The obtained MAPE has an average value of 6.97% (which is however computed only taking into consideration four test days randomly picked up over the whole data set available, selecting 1 day for each season of the year), outperforming persistence, autoregressive integrated moving average (ARIMA) and NN approaches; the average computation time is less than 10 s.

Tar and Szegedi [32] present the sliding average model (SLIDAV) for estimating the amount of wind energy available on a given day. Using the statistics and hourly wind speed data measured in the vicinity of the wind turbines/on the wind turbines themselves; the model provides an hourly estimate of the potential wind energy for the whole day in a particular season or circulation type group. This makes it possible to forecast average daily wind power 6-9 h before the end of the day with an error of 20%. The magni-

tude of the estimation error depends on the given season and/or synoptic type group.

The key issue addressed in this paper is the estimation of the relationship between the wind speed and a wind farm power output. This relationship is expressed as a power curve (the nominal power curve), which has a logistic function shape. However, as discussed in [24], the experimental power curve of a wind turbine (and of an entire wind farm as well) is not an ideal logistic function; the measured data points are scattered around the nominal power curve. The data points on the left hand side of the nominal power curve are regarded as over-performance (power gains), whereas the data points on its right hand side are regarded as under-performance (power losses). The data points with null wind speed but non-null power typically indicate that the anemometer is faulty, as it should provide a non-zero wind speed measurement while the turbine is producing power. The data points with zero power but large wind speeds are typical of a situation in which the turbine did not work.

The presence of outliers and abnormal values might be due to several reasons: the presence of values of wind speed close to the cut-in or the cut-out speed of the installed wind turbines; environmental issues (blades affected by dirt, bugs and ice); shut-down due to maintenance or energy curtailment; control system issues; sensors malfunctions; pitch control malfunctions; unsuitable blade pitch angle setting; blade damage [24]. Finally, the measured powers at different wind speeds are generally different than the rated power also because the operating conditions on site are generally different than the conditions under which the turbine was calibrated (the wind speed on site is not uniform horizontally across the face of the turbine; the vertical wind profile and the air density are different during the operation phase than during the calibration; the wind data on site are not always measured at the height of the turbine's hub).

If the model of a power curve reflecting a normal status is available, it can be used as a visual tool to check the performance of the turbine; the abnormal status of the turbine itself can then be monitored and detected by this model. However, there is no systematic approach to use power curve for turbine diagnostics yet. The main motivation of this paper lies in the detection of the abnormalities of the wind turbines (and as a consequence also wind farms as a whole) power curve. The recent developments in data mining and evolutionary computation (EC) offer promising approaches to modeling turbines power curves. The majority of the literature has focused on parametric power curve profile monitoring. A comparison between parametric and some non-parametric models of wind farm power curve has been showed in [25]. In this paper we investigate the use of two non-parametric (data driven) models not explored in [25]. The first one is the self-supervised neural network called GMR (Generalized Mapping Regressor) [33,34]. GMR is an incremental self-supervised neural network which can approximate every multi-dimensional function or relation with any kind of discontinuity. The methodology allows the creation of a non-parametric model of the power curve that can be used as a reference profile for on-line monitoring of the power generation process, as well as for power forecasts. The second is a General Regression Neural Network (GRNN). We then compare the results obtained using GMR and GRNN with those obtained with a "classical" feed-forward Multi Layer Perceptron (MLP), which is probably the most utilized neural network model worldwide in a number of different applications.

The rest of the paper is organized as follows: Section 2 describes the models applied in the case study (providing more details especially about GMR); Section 3 describes the case study and presents the results; Section 4 draws some conclusions and Section 5 describes a future research plan.

2. Materials and methods

All the simulations performed with the GMR neural network have been run using the software Matlab 7.11.0; the simulations performed with the MLP and the GRNN model have been run using the software "Machine Learning Office", distributed with Ref. [7]. All the calculations have been performed using a 2.53 GHz Intel Core 2 Duo laptop running MS Windows XP.

2.1. The Generalized Mapping Regressor

GMR is an incremental self-organizing neural network with chains (second layer weights) among neurons, able to approximate every kind of mapping (function or relation) in both senses, i.e. $M(x,y) : x \in \Re^m \leftrightarrow y \in \Re^n$. In the field of wind energy it has already been applied for the estimation of wind speed over a complex terrain [14]. The basic idea of GMR is to transform the function approximation problem into a pattern recognition problem under an unsupervised framework. Hence, a coarse-to-fine covering strategy of the mapping is used. Suppose one wants to model the mapping relationship between a set of inputs $\mathbf{x}$ and a set of outputs $\mathbf{y}$. GMR algorithm works transforming the data mapping problem f: $\mathbf{x} \rightarrow \mathbf{y}$ into a pattern recognition problem in the augmented space Z represented by vectors $\mathbf{z} = [\mathbf{x}^T\ \mathbf{y}^T]^T$, which are the inputs of GMR. In this space, the branches of the mapping become clusters which have to be identified.

The algorithm comprises four phases (training, linking, merging, and recalling) which are succinctly described in the following.

The ***training phase*** concerns the *vector quantization* of the Z space. The aim of vector quantization is, in practice, the reduction of the original data set to a small representative set of prototype vectors that are easier to manage [35]. During training, the augmented space is recovered by either creating neurons or adapting their weights according to the novelty of the input data. At the presentation of each input belonging to the training data set (TS), there are two possibilities: either creation of a new neuron (whose weight vector is equal to the input vector) or adaptation of the weight vector of the closest neuron (in input/weight space). Given a threshold ρ (*vigilance threshold*), a new neuron is created if the hyperspheres of radius ρ, centred in the weight vectors of the neurons created previously, do not contain the input. The parameter ρ is very important: it determines the resolution of the training. Learning can be divided into two sub-phases: *coarse quantization* and *fine quantization*. The vigilance threshold used in the coarse quantization phase (ρ_1) is higher than the one used in the fine quantization, ρ_2. In general, the first epoch (i.e. presentation of the entire TS) defines the number of neurons needed for mapping and the others adapt their weights for a better approximation. The neurons thus obtained identify the *objects*, which are compact sets of data in Z. The resulting neurons are called *object neurons*. In the second sub-phase, at first a pre-processing is required for labeling each neuron with the list of the input data which had the neuron as winner (i.e. whose weight vector is the closest to the input vector); it can be accomplished by presenting all data (*production phase*) to GMR and recording the corresponding winning neurons. At the end, for each neuron a list of the inputs for which it won is stored. This list represents the *domain* of the object neuron. Every list is considered as the TS for a subsequent training which takes place separately (and in parallel) for each object domain. At the end, the neural network is made up of the neurons generated by the secondary learning phases (*final neurons*), labeled as belonging to an object by the corresponding object neuron.

The next phase is the ***linking phase***. Neurons' linking is accomplished by computing the second layer weights, which are discrete and equal to zero in the absence of a link. A link is computed at the

presentation of each piece of data from the TS. The technique used in this paper to perform the linking phase exploits the direction of the principal component of the domain data (i.e. the direction corresponding to the highest eigenvalue of the autocorrelation matrix of the domain data) [33]. This direction is here referred to as domain *principal direction* (PD). For each data point, the weights are sorted according to the Euclidean distance from it, and the winning neuron is determined. It is then linked to another neuron chosen in a subset of neurons (*candidate neurons*). The subset was here determined by defining in advance a number k of nearest neighbors of the input. Then, for each candidate, the absolute value of the scalar product between its PD and the winner's PD is evaluated. The winner is linked to the candidate yielding the maximum scalar product (i.e. the candidate whose PD is closest in direction to the winner's PD). This approach is justified by the fact that clusters with similar shapes have to be connected.

In the ***merging phase***, GMR checks whether different objects are linked. If they are, the objects are merged. The ***recalling phase*** replaces the neurons in the reduced manifold with Gaussians representing the domain. Their parameters are estimated by the maximum likelihood (ML) technique. This phase is essentially a process of Gaussian labeling followed, if required, by an interpolation step. A more detailed description of GMR is contained in [33].

2.2. The Multi Layer Perceptron

MLPs are the most widely used neural network models and thus a detailed explanation does not seem appropriate in this context. A comprehensive description of MLP neural networks is contained in [36].

As it is well known in the machine learning field, one of the most important tasks of the modeler is finding a model with the optimal complexity with respect to the generalization error. A related notion depicting this situation is the *bias-variance dilemma* [37]. In order to guarantee good generalization performances to the model (i.e. the capacity of providing good estimates on unknown data samples) a tradeoff between model complexity and empirical risk (training error) needs to be found. Simple models provide high empirical risk (they cannot find a good fit to training data), whereas too complex models result in low empirical risk but high expected risk, since they rely too much on the noisy and incomplete training set used, resulting in a high generalization error.

A general approach to control the complexity of the model is the application of *regularization* techniques [36,37]. The approach to regularization applied in the training phase of the MLPs used in this paper is based on the weight decay technique. In the weight decay approach, a penalty function is added to the error function (which has to be minimized during the training process) which is proportional to the sum of the networks weights squared, thus penalizing large values of the weights:

$$\text{Error} = \text{MSE} + \frac{\lambda}{2}\sum w_k^2 \tag{1}$$

where λ is an hyper-parameter which controls the penalty term and MSE is the mean square error function:

$$\text{MSE} = \frac{1}{n_o N_p}\sum_{p=1}^{N_p}\sum_{i=1}^{n_o}(\widehat{Y}_{i,p}(\mathbf{X}_p, \boldsymbol{w}) - Y_{i,p})^2 \tag{2}$$

2.3. The General Regression Neural Network

GRNNs belong to the family of kernel neural networks. GRNN is another name for the statistical non-parametric method called Nadaraya-Watson Kernel Regression Estimator (NWKRE) from the names of the scientists who independently proposed it in 1964 [38,39]. In 1991 it was implemented as a neural network by Specht [40]. General regression obtains the conditional mean of the system's output (simulated values) $\widehat{Y}$ using non-parametric estimator of the joint probability density function, when system's input vector variable (measured value) $\mathbf{X}$ is provided.

The following equation is the basic GRNN published by Specht [40]:

$$\widehat{Y}(\mathbf{X}) = \frac{\sum_{p=1}^{N_p} Y_p \exp\left(\frac{D_p^2}{\sigma}\right)}{\sum_{p=1}^{N_p} \exp\left(\frac{D_p^2}{\sigma}\right)} \tag{3}$$

where

$$D_p^2 = (\mathbf{X} - \mathbf{X}_p)^T(\mathbf{X} - \mathbf{X}_p), \tag{4}$$

Y_p is the generic p-th sample value of the variable Y; $\mathbf{X}_p$ is the generic p-th sample value of the variable $\mathbf{X}$ and σ (the smoothing parameter or the kernel size) is the only important computing parameter to be determined.

The ordinary GRNN training procedure is a mean square error (MSE) minimization, accomplished using a cross-validation (*leave one out*) approach [36].

For a particular value of σ, the *leave one out* method removes one sample at a time and builds the corresponding network based on the remaining samples. Then, the network is used to estimate Y for the removed sample. By repeating this procedure for each sample and storing each estimate, the MSE between the actual sample values and their estimates can be calculated. The procedure is repeated for different values of σ taken from an interval $[\sigma_{low}, \sigma_{high}]$:

$$\sigma_i = \sigma_{low} + (i-1)\frac{\sigma_{high} - \sigma_{low}}{N} \quad i = 1, \ldots, N \tag{5}$$

where N is the number of steps to be performed.

The initial interval of possible σ values and the number of steps must be defined. The starting interval and the number of steps have to be consistently chosen in order to catch the expected optimal σ value, especially in automatic mapping procedures. In other cases, choosing too wide interval and/or a large number of steps only entails a longer running time.

Once the cross-validation procedure has been performed, the value of σ with the smallest error is employed in the final network.

3. Case study

The data used for the case study refer to a wind farm (whose location is not disclosed for confidentiality reasons) and the available data set comprises an hourly wind speed time series (covering one whole year) measured by an anemometer located at 50 m above the ground level (a.g.l.), as well as the power produced by each of the wind turbines, collected by a SCADA (Supervisory Control and Data Acquisition) systems at the wind farm. Since the aim of the study was modeling the power curve of the wind farm as a whole, the sum of the power produced by all the turbines was used for the application of the algorithms under investigation. A training and a test set containing respectively 90% and 10% of the entire time series available were created. The two data sets were built by uniform selection across the whole available data set. No particular sampling technique was deemed necessary, since the data did not exhibit particular cluster patterns, which are instead more common in high dimensional data vectors.

In order to speed up the training, a sampling of one datum out of three was then made on the training set (TS), so to select a smaller subset (that was used as the actual TS). Before starting the training phase, the data which had been clearly recorded erroneously, due to a malfunctioning of the SCADA system, were filtered

out (namely, data with an abscissa greater than the cut-in speed of the wind turbines and with a null ordinate). The remaining data were normalized in the interval [−1; +1] and denoised using the Kernel Principal Component Analysis (KPCA) technique with a bandwidth of the Gaussian kernel $\sigma = 0.4$. KPCA is a *generically nonlinear* signal processing technique; it is often used for denoising in image applications [41]. The denoised TS used to train GMR contains 2828 observations. A test data set containing 876 observations has been used in the recalling phase. After a trial and error phase, a number of nearest neighbors $k = 4$ was used during the linking phase. In the rough phase the vigilance threshold was set to $\rho_1 = 0.5$ and in the fine tuning phase $\rho_2 = 0.025$ was used. The number of final neurons is 101 and the number of object neurons is 26. During the recalling phase, GMR was used both without interpolation and with Gaussian interpolation. Fig. 1 shows the results of the linking and merging phases of GMR. A magnification of the part of the curve contained in the dashed box is showed in the upper left and lower right corners of Fig. 1b. In the lower right corner different objects are represented with different symbols in order to improve the readability.

The same TS used for the GMR model was also used to train a feed-forward MLP and a GRNN. Two different learning algorithms have been applied to train the MLP: (1) back-propagation with momentum [42] using regularization with weight decay (with a penalization term $\lambda = 10^{-5}$) and (2) the optimization technique known as Levenberg–Marquardt (LM) algorithm [43]. Four different MLPs have been trained in order to choose the one with the best performances. Each of them has two layers of neurons, the hidden layer with a different number of neurons for each network (5, 10, 15 and 20 neurons for the MLPs respectively called MLP_5, MLP_10, MLP_15 and MLP_20) and the output layer with 1 neuron (since the output is mono-dimensional). The input layer obviously contains only one neuron, which takes as input the wind speed. The bias term [42] has been used for each neuron of both the hidden and the output layer. For all the networks a logistic sigmoid function and a linear function were respectively used as the activation function of the neurons of the first and the second layer. The choice of the number of neurons in the hidden layer was based on a combination of trial-and-error and prior experience of the authors. Increasing the number of neurons over 20 resulted in increasing training times without any improvement in the results.

During the training of the MLPs using the LM algorithm, the damping factor [43] used to approximate the Hessian matrix (the matrix of second partial derivatives of the error function which has to be minimized during the training) is varied in accordance with the results of minimization at the current training step. It is decreased after each successful step (i.e. a step producing a reduction in the value of the error function) and is increased only when a tentative step would increase the value of the error function.

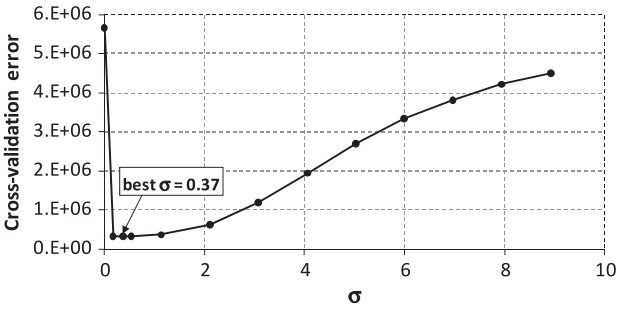

Fig. 2. Cross-validation error as a function of the kernel size σ for the GRNN.

The kernel size for the GRNN was selected using the leave-one-out method. Fig. 2 shows the cross-validation error as a function of the kernel size. The optimal value for σ was $\sigma = 0.37$.

Three goodness-of-fit indicators have been computed for the GRNN and for all the MLPs: the coefficient of determination (R^2) between network predictions and measured data; the root mean square error (RMSE), i.e. the square root of the MSE defined in Eq. (2); the coefficient of correlation between predictions and measured data, defined as:

$$R_0 = \frac{1}{N_t \sigma_{pred} \sigma_{test}} \sum_{i=1}^{N_t} (\hat{Y}_i - \overline{\hat{Y}})(Y_i - \overline{Y}) \quad (6)$$

Figs. 3–5 respectively show the values of R^2, RMSE and R_0 calculated for the MLPs and the GRNN on the test set data. From the observation of the graphs, the network with 10 neurons in the hidden layer resulted to be the best among all the MLPs. 179 training steps were necessary to train it.

Fig. 6 shows the results of the modeling obtained both on the TS and test set data, with the best MLP network (MLP_10), the GMR models and the GRNN, while Table 1 summarizes their prediction accuracy. The data showed in Fig. 6 have been previously rescaled using a scaling factor not disclosed to the reader, in order to protect data confidentiality.

Before calculating the set of performance indicators showed in Table 1, the wind power data have been scaled to the interval [0 kW, 100 kW]. This allows comparison with other state-of-the-art results [25,44]. In particular, Table 1 shows the values of the mean absolute error (MAE) and the symmetric mean absolute percentage error (sMAPE) calculated using the data of test set. The respective standard deviations (Std dev, in Table 1) are also showed in the table.

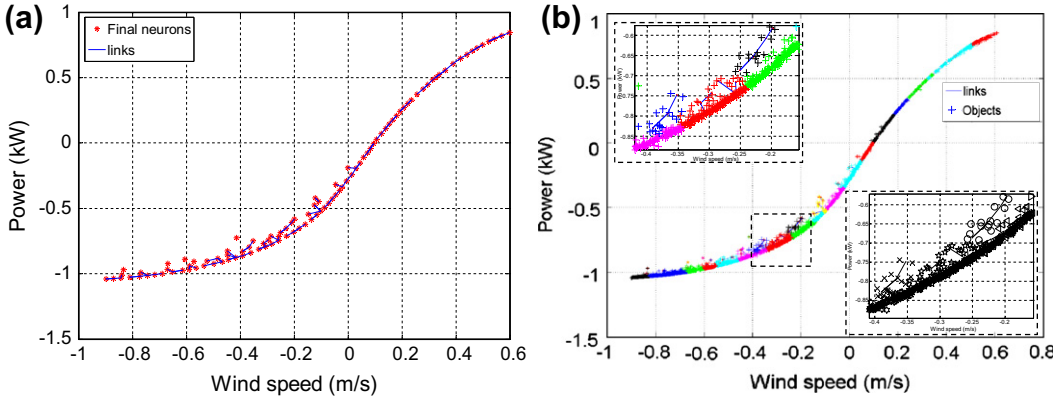

Fig. 1. Results of the linking (a) and merging and (b) phases of GMR.

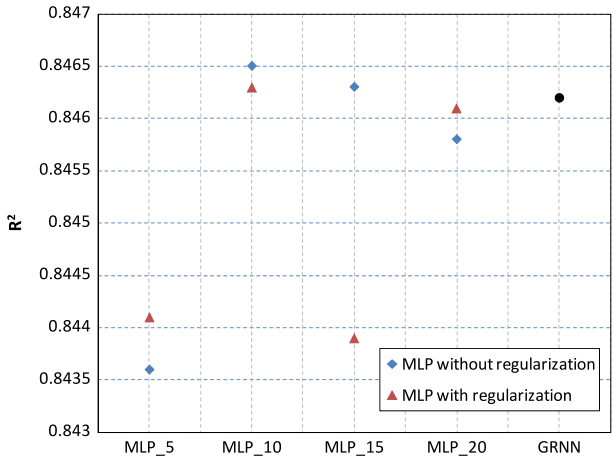

Fig. 3. Values of the coefficient of determination R^2 between model's prediction and measured data (computed using the test data set) for all the trained models.

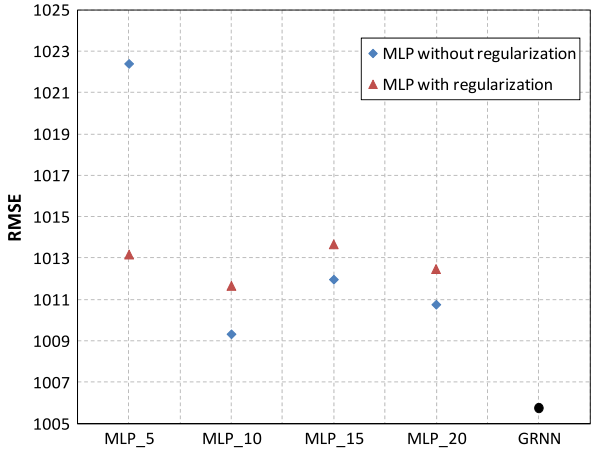

Fig. 4. Values of the RMSE calculated for all the trained models using the test data set.

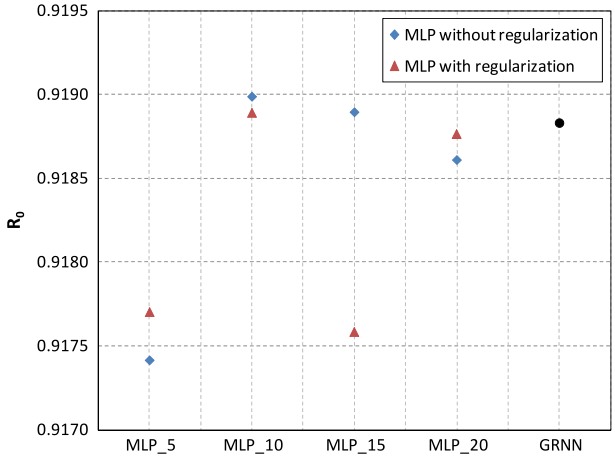

Fig. 5. Values of the coefficient of correlation R_0 between predictions and measured data (computed using the test data set) for all the trained models.

The sMAPE measure computes the absolute error in percent between the absolute value of the actual observation Y_i and the absolute value of the forecast $\widehat{Y}_i$ across all the N_t observations of the test set [45]:

$$\text{sMAPE} = \frac{1}{N_t}\sum_{i=1}^{N_t}\frac{|Y_i - \widehat{Y}_i|}{(|Y_i| + |\widehat{Y}_i|)/2} \times 100 \tag{7}$$

With the adopted scaling interval, the MAE coincides with the Normalized Mean Absolute Percentage Error (NMAPE), given by:

$$\text{NMAPE} = \frac{1}{N_t}\sum_{i=1}^{N_t}\frac{|Y_i - \widehat{Y}_i|}{\max_{i=1}^{N_t}(Y_i)} \times 100 \tag{8}$$

In [44] the authors tackle a time series forecasting problem and provide wind power forecasts with different prediction lengths. In our paper the task performed is different because the power curve models we built are specifically conceived for steady state performance assessment, i.e. estimation of annual energy production of the wind farm (we do not provide power output forecasts ahead of time). However, the values of NMAPE showed in [44] can be used as a benchmark for our study.[1]

The best model in terms of absolute error is the GRNN, whose predictions differ from those of the other models above all in the last part of the power curve. However, the GMR model without interpolation performs better than all the other models in terms of sAPE (%). The trend of the values forecasted using GMR without interpolation appears less fragmented with respect to the distribution of the values forecasted using GMR with Gaussian interpolation. However, both the models recognize some information content in the "thickness" of the data cloud around the average logistic sigmoid trend (see the links in Fig. 1a). This is the reason why both the GMR forecasts do not distribute themselves smoothly along a logistic profile.

3.1. On-line monitoring by residual approach and control charts

Once the non-parametric models mentioned above have been trained, they can be used to characterize the wind farm power in normal conditions, and therefore they can serve as an on-line wind farm power generation profile. It can be used to understand and differentiate normal from abnormal behavior of the wind farm, thus allowing troubleshooting and scheduling maintenance and repair interventions on fault conditions. Diagnostics of the potential faults will ultimately help maintaining and improving the efficiency of the wind energy conversion system.

The residual control chart techniques (statistical quality control) [46] are used to analyze residuals between model predicted power and observed power. The control chart approach allows the residuals and their variations to be monitored, thus detecting abnormal conditions of the wind farm functioning.

The means of the residuals obtained on the TS (μ_{Train}) and the test set (μ_{Test}), as well as their standard deviations (σ_{Train}; σ_{Test}) have been computed. Once μ_{Train} and σ_{Train} are known, the upper and lower control limits of the control chart can be computed and used to detect the anomalies. Control limits for the control chart can then be derived using the following expressions [46] and used to detect anomalies:

$$UCL_1 = \mu_{Train} + \eta\frac{\sigma_{Train}}{\sqrt{N_{Test}}}; \quad UCL_1 = \mu_{Train} - \eta\frac{\sigma_{Train}}{\sqrt{N_{Test}}} \tag{9}$$

N_{Test} is the number of points in the test data set, but it can be adjusted to make the control chart less sensitive to the data variability and thus reduce the risk of false alarms. In the application described in this paper, N_{Test} was set equal to 2, while for η the value was set to $\eta = 3$.

[1] In [44] NMAPE has been computed after scaling of the wind power data in the interval [0 kW, 1 kW] instead of [0 kW, 100 kW]; however these two scaling intervals obviously do not affect the values of the resulting NMAPE.

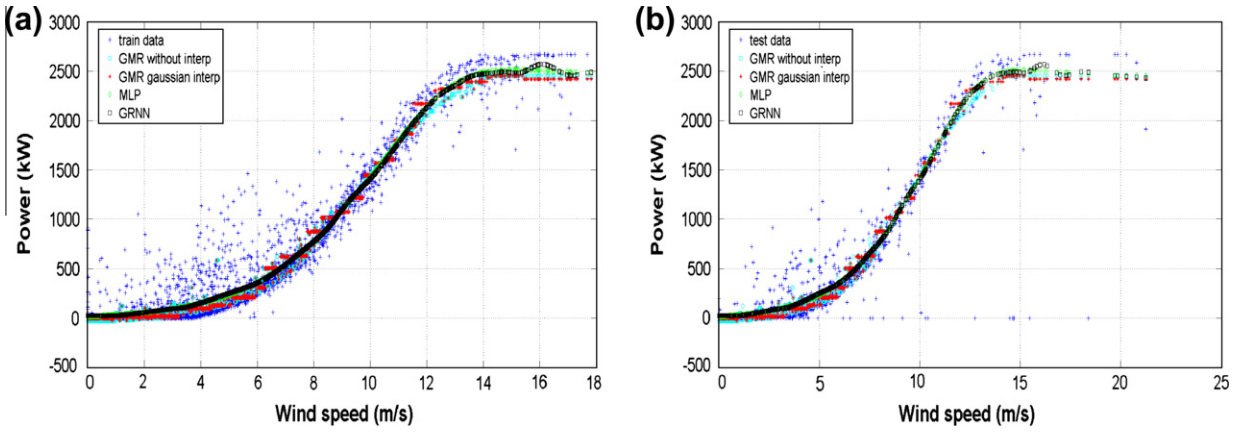

Fig. 6. Results of the GMR, MLP_10 and GRNN predictions on the training set (a) and the test set (b).

Table 1
Prediction accuracy of the models applied (MLP_10, GMR and GRNN). Prior to the calculation of the accuracy indices, the wind power data have been scaled to the interval [0 kW, 100 kW].

	MLP_10	GMR Gaussian interp	GMR without interp	GRNN
MAE (kW)	5.02	5.72	6.10	4.63
Std dev of AE (kW)	10.42	10.54	10.91	10.11
sMAPE (%)	75.12	79.49	67.28	76.74
Std dev of sAPE (%)	74.25	74.80	68.70	75.95

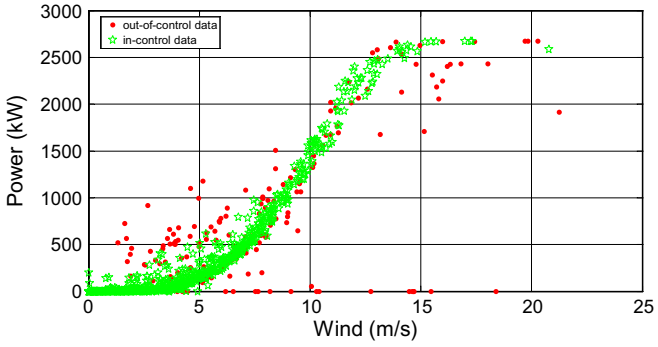

Fig. 7. Data (within the test set) detected as "in-control" and "out-of-control" using MLP_10.

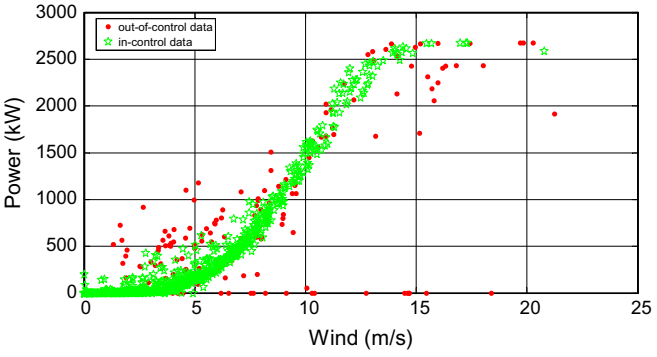

Fig. 8. Data (within the test set) detected as "in-control" and "out-of-control" using GRNN.

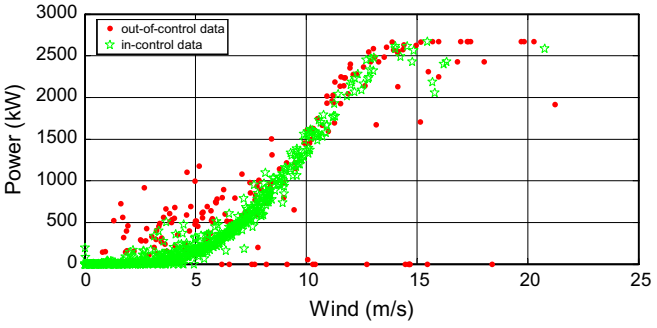

Fig. 9. Data (within the test set) detected as "in-control" and "out-of-control" using GMR without interpolation.

If μ_{Test} is above UCL_1 or below LCL_1, the power generation process at the generic sampling time $Y_{TestSet} = [Y_i, \widehat{Y}_i]$ is considered to be deficient (or "out-of-control"), otherwise it is not considered abnormal, i.e. it is considered "in-control". Increasing the value of the parameter η enhances the confidence of detecting anomalies in the generated power and tends to increase the number of points detected as "in-control", thus reducing the risk of mistaking the normal state for an abnormal one [25].

Similarly, the control limits for σ^2_{Test} can also be calculated to detect out-of-control points, using control limits defined as a function of the variance of σ^2_{Train} [24,25,46]:

$$UCL_2 = \frac{\sigma^2_{Train}}{N_{Test}-1} \times \chi^2_{\alpha/2,N_{Test}-1}; \quad LCL_2 = 0 \tag{10}$$

where $N_{Test} - 1$ are the degrees of freedom of the chi-square distribution. The parameter α needs to be adjusted to make the control chart less sensitive to the data. The value of α used in this paper is 2. If σ^2_{Test} is above UCL_2, the power status at the sampling time $Y_{TestSet}$ is considered as deficient.

LCL_2 is set to 0 to indicate that the variation of residuals in the test data is 0, so that the current power matches the reference power in the normal status.

Figs. 7–10 show the out-of-control points detected in the test set respectively by the MLP, the GRNN, GMR with no interpolation

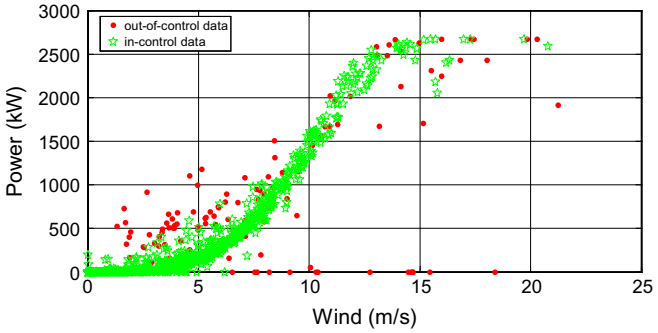

Fig. 10. Data (within the test set) detected as "in-control" and "out-of-control" using GMR with Gaussian interpolation.

and GMR with Gaussian interpolation, on the basis of the limits defined by Eqs. (9) and (10).

Out of the 876 points of the test set, MLP_10 network detected 216 out-of-control points, GRNN detected 180 of them, GMR without interpolation 228 and GMR with Gaussian interpolation 142. The GMR model without interpolation detects the highest number of out-of-control points; whereas the control chart based on the GMR model with Gaussian interpolation is the most conservative toward the risk of false alarms. The MLP and GRNN models are in an intermediate position, probably because they tend to find smoother solutions. For values of wind speeds equal to or higher than the cut off speed, the GMR models label a few points as "in-control", whereas the other two neural networks detect them as "out-of-control" points. This is due to the segmentation of the GMR forecasted curve and the fact that in its upper part it is slightly shifted toward a lower value of output power, with respect to the curves forecasted by MLP_10 and GRNN.

GMR finds some links and, as a consequence, some branches of the mapping, also in some zones of the space which are not along the average curve's trend found by interpolation models (see Fig. 1). In fact, one of the main advantages of GMR, compared with feed-forward neural networks, lays into the fact that, while for many functional (i.e. single-valued) approximation problems feed-forward network can work well by minimizing a sum-of-squares error function, they can actually give rise to high imprecision in presence of multi-valued mappings, or mappings whose structure varies for different regions of the input space. The prominent feature of GMR is its capability to output all the solutions and their corresponding mapping branches.

4. Conclusions

The paper presents an application of three different machine learning models (more specifically neural networks) for building an equivalent steady state model of a wind farm under normal operating conditions and shows their utilization for the creation of quality control charts at the aim of detecting anomalous functioning conditions of the wind farm as a whole.

Few other applications dealing with the same topic have already been presented in the literature. The originality of this paper is twofold: on one hand its application to the modeling of the power curve of an entire wind farm, instead of a single wind turbine; on the other hand the utilization of GMR, which is a novel incremental self-organizing competitive neural network.

The models can be used to estimate the reference power curve (on-line profile) for monitoring the performances of a wind farm as a whole. The residual control chart technique has been used to analyze residuals between model predicted power and observed power, in order to monitor their variations, thus detecting abnormal conditions of the wind farm functioning. The case study demonstrated that the control chart approach produces satisfactory results in monitoring power curves, provided that a proper preprocessing of the data is performed. However, the data set available for the study was not suitable for an accurate and in-depth investigation of abnormal states.

5. Future research

A proper validation of the accuracy of detection of in-control and out-of-control situations of the algorithms presented in this paper is not possible at the moment, due to the lack of historical series of labeled data sets of normal and abnormal values of the power output. In a future experimental advancement of this research it is foreseen to couple turbine operational data (wind speed and power output) and time stamped status/fault data (not available at this time) to assess the detection accuracy of normal (in-control) and abnormal (out-of-control) conditions. Faults represent statuses entailing severe consequences to the wind turbine system, such as a major component failure, i.e. an event whose occurrence causes the shutdown of the turbine with motionless rotor (for example an emergency stop of nacelle/hub). The status/fault data will be classified in different categories (normally identified with a status code). Each status code is associated with a specific abnormality of a turbine component or a subsystem (e.g. faulty anemometer, or malfunction of cabinet heaters) and different codes represent different levels of severity of this abnormality. The statuses sorted in the different categories form a labeled data set from which a suitable training and a test data set can be extracted (in this case a specific sampling able to balance status/fault with normal operations data will be necessary in order to avoid working with biased data sets, due to the expected significantly different frequencies of occurrence of status/fault and normal turbines operations conditions). The accuracy (in-control vs out-of-control detection) of the methodology applied in this paper will then be assessed using the test data set. A schematic representation of this kind of model is showed in Fig. 11. The different strategy of working with a separate time series for each turbine (rather than working with data aggregated at the wind farm level) will be also tested when status/fault data will be available.

A further interesting application is represented by the utilization of data-driven algorithms to perform the prediction and diagnosis of wind turbine faults [47]. In this case the machine learning approach should be used not to build a steady state model of the reference power curve of the wind farm under normal operating conditions, but to predict the occurrence and the category of a fault ahead of time. The model should then be trained (off-line) with a set of 2-dimensional input vectors whose elements are the wind speed and the turbine's power output recorded at time (t–n), and a set of corresponding 2-dimensional output vectors whose elements represent: (1) the occurrence or not of a fault in the time interval between (t–n) and t; (2) the category of this fault in the case of occurrence (both of which can be expressed by a conventional code). A schematic representation of this approach is showed in Fig. 12.

However, in order to be used in an on-line application, the model should be trained using data with a much higher sampling frequency than the one available for the present study (hourly mean data). In fact, as it has been observed in the data-driven application presented in [47], even a 5 min interval is too long for this purpose since it leads "to a significant loss of the history of the fault emergence" [47].

Finally, the application tackled in this paper paves the road for higher-level decision strategies based on multiple states such as in a Bayesian network. This latter approach might be considered to

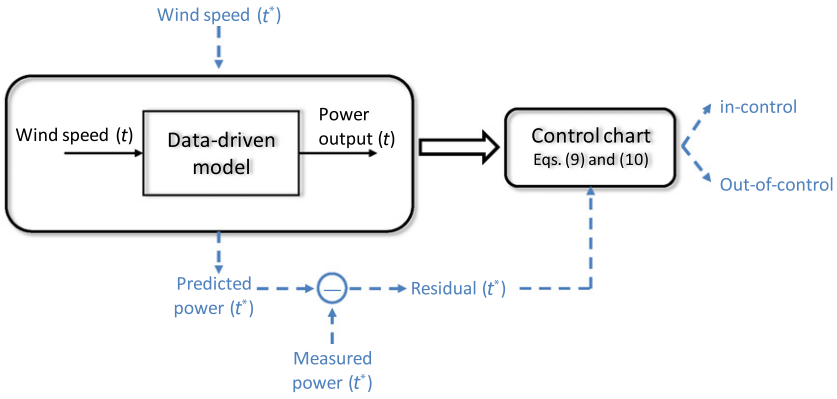

Fig. 11. Schematic representation of the data-driven model presented in the paper. The model is trained off-line (black part of the figure) with couples of wind speed (used as predictor) and wind farm power (used as dependent variable). In the on-line application (blue part of the figure), when new values of wind speed and power are available, the trained model can be used to predict the corresponding value of the power output and thus the residuals can be calculated. They are finally used to derive the values of μ_{Test} and σ^2_{Test} and the control chart is used to label the measured power as in-control or out-of-control. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

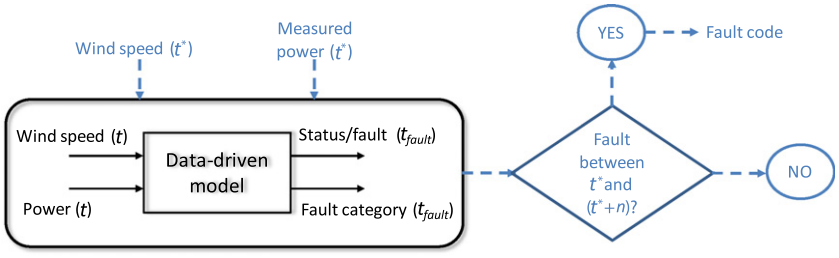

Fig. 12. Schematic representation of the data-driven model to perform the prediction and diagnosis of wind turbine faults. The model is trained off-line (black part of the figure) with 2-dimensional inputs vectors containing the wind speed and power at the generic time t and 2-dimensional output vectors containing the status/fault code and the fault category code of a fault occurred at the time t_{fault}, which is comprised between t and $t + n$. In the on-line application (blue part of the figure), when new values of wind speed and power are available at the time t^* the trained model can be used to predict if a fault will occur between t^* and $t^* + n$. In case a fault will be predicted, it will be also associated with its expected fault code. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

identify links between interrelated anomalies as well as relations between the anomalies and the specific reasons or factors causing them (which can be inferred by inspection of the turbines' maintenance logs), which is not possible simply using control charts. If coupled with an accurate wind speed forecasting model, this would also allow a proper scheduling of maintenance and fault correction works in order to operate them during periods of law wind speed, thus minimizing the loss of wind power yield.

Acknowledgment

The authors wish to thank Dr. Sameer Rege for his help in English proofreading of the manuscript.

References

[1] Messineo A, Panno G. LNG cold energy use in agro-food industry: a case study in Sicily. J Nat Gas Sci Eng 2011;3:356–63.
[2] Messineo A, Panno D. Potential applications using LNG cold energy in Sicily. Int J Energy Res 2008;32:1058–64.
[3] Messineo A, Marchese F. Performance evaluation of hybrid RO/MEE systems powered by a WTE plant. Desalination 2008;229:82–93.
[4] Krasnopolsky VM, Fox-Rabinovitz MS. A new synergetic paradigm in environmental numerical modeling: hybrid models combining deterministic and machine learning components. Ecol Model 2006;191:5–18.
[5] Bhattacharya B, Solomatine DP. Machine learning in soil classification. Neural Netw 2006;19:186–95.
[6] Krasnopolsky VM, Fox-Rabinovitz MS. Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. Neural Netw 2006;19:122–34.
[7] Kanevski M, Pozdnoukhov A, Timonin V. Machine learning for spatial environmental data. theory, applications and software. Lausanne: EPFL Press; 2009.
[8] Phillips SJ, Anderson RP, Schapire RE. Maximum entropy modeling of species geographic distributions. Ecol Model 2006;190:231–59.
[9] Benetto E, Tiruta-Barna L, Perrodin Y. Combining lifecycle and risk assessments of mineral waste reuse scenarios for decision making support. Environ Impact Assess Rev 2007;27(3):266–85.
[10] Hong WC. Rainfall forecasting by technological machine learning models. Appl Math Comput 2008;200:41–57.
[11] Kusiak A, Zheng H, Song Z. Power optimization of wind turbines with data mining and evolutionary computation. Renew Energy 2010;35(3):695–702.
[12] Kusiak A, Zheng H. Optimization of wind turbine energy and power factor with an evolutionary computation algorithm. Energy 2010;35(3):1324–32.
[13] Cellura M, Cirrincione G, Marvuglia A, Miraoui A. Wind speed spatial estimation for energy planning in Sicily: a neural kriging application. Renew Energy 2008;33(6):1251–66.
[14] Beccali M, Cirrincione G, Marvuglia A, Serporta C. Estimation of wind velocity over a complex terrain using the Generalized Mapping Regressor. Appl Energy 2010;87:884–93.
[15] Whei-Min L, Chih-Ming H, Fu-Sheng C. Design of intelligent controllers for wind generation system with sensorless maximum wind energy control. Energy Convers Manag 2011;52:1086–96.
[16] Whei-Min L, Chih-Ming H, Fu-Sheng C. Fuzzy neural network output maximization control for sensorless wind energy conversion system. Energy 2010;35:592–601.
[17] Vaccaro A, Mercogliano P, Schiano P, Villacci D. An adaptive framework based on multi-model data fusion for one-day-ahead wind power forecasting. Electr Power Syst Res 2011;81:775–82.
[18] Foley AM, Leahy PG, Marvuglia A, McKeogha EJ. Current methods and advances in forecasting of wind power generation. Renew Energy 2012;37(1):1–8.
[19] Landberg L. Short-term prediction of the power production from wind farms. J Wind Eng Ind Aerodynam 1998;80(1–2):207–20.
[20] Li S, Wunsch DC, O'Hair E, Giesselmann MG. Comparative analysis of regression and artificial neural network models for wind turbine power curve estimation. J Solar Energy Eng 2001;123(4):327–32.
[21] Goh SL, Chen M, Popović DH, Aihara K, Obradovic D, Mandic DP. Complex-valued estimation of wind profile. Renew Energy 2006;31:1733–50.
[22] Mandic DP, Javidi S, Goh SL, Kuhc A, Aihara K. Complex-valued prediction of wind profile using augmented complex statistics. Renew Energy 2009;34:196–201.

[23] Üstütaş T, Şahin AD. Wind turbine power curve estimation based on cluster center fuzzy logic modelling. J Wind Eng Ind Aerodynam 2008;96:611–20.
[24] Kusiak A, Zheng H, Song Z. Models for monitoring wind farm power. Renew Energy 2009;34:583–90.
[25] Kusiak A, Zheng H, Song Z. On-line monitoring of power curves. Renew Energy 2009;34:1487–93.
[26] Lei M, Shiyan L, Chuanwen J, Hongling L, Yan Z. A review on the forecasting of wind speed and generated power. Renew Sust Energy Rev 2009;13(4):915–20.
[27] Pousinho HMI, Mendes VMF, Catalão JPS. A hybrid PSO–ANFIS approach for short-term wind power prediction in Portugal. Energy Convers Manag 2011;52:397–402.
[28] Hong YY, Chang HL, Chiu CS. Hour-ahead wind power and speed forecasting using simultaneous perturbation stochastic approximation (SPSA) algorithm and neural network with fuzzy inputs. Energy 2010;35:3870–6.
[29] Kusiak A, Zheng H. Optimization of wind turbine energy and power factor with an evolutionary computation algorithm. Energy 2010;35:1324–32.
[30] Blonbou R. Very short-term wind power forecasting with neural networks and adaptive Bayesian learning. Renew Energy 2011;36:1118–24.
[31] Catalão JPS, Pousinho HMI, Mendes VMF. Short-term wind power forecasting in Portugal by neural networks and wavelet transform. Renew Energy 2011;36:1245–51.
[32] Tar K, Szegedi S. A statistical model for estimating electricity produced by wind energy. Renew Energy 2011;36:823–8.
[33] Cirrincione G, Cirrincione M, Lu C, Van Huffel S. A novel neural approach to inverse problems with discontinuities (the GMR neural network). In: Proc Int J Conf Neural Netw (IJCNN'03), Portland, Oregon; 2003. p. 3106–11.
[34] Lu C. The Generalised Mapping Regressor (GMR) neural network for inverse discontinuous problems. MSc thesis. Katholieke Universiteit Leuven (Belgium); 2000.
[35] Gray RM. Vector quantization. IEEE ASSP Mag 1984;4(2):4–29.
[36] Bishop CM. Neural networks for pattern recognition. Oxford (UK): Oxford University Press; 1995.
[37] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning. 2nd ed. New York: Springer; 2009.
[38] Nadaraya EA. On estimating regression. Theory Probab Appl 1964;9:141–2.
[39] Watson GS. Smooth regression analysis. Sankhya Ser A 1964;26:359–72.
[40] Specht D. A general regression neural network. IEEE Trans Neural Netw 1991;2(6):568–76.
[41] Teixeira AR, Tomé AM, Stadlthanner K, Lang EW. KPCA denoising and the pre-image problem revisited. Digit Signal Process 2008;18:568–80.
[42] Hagan MT, Demuth HB, Beale MH. Neural network design. Boston (USA): PWS Publishing; 1996.
[43] Levenberg K. A method for the solution of certain non-linear problems in least squares. Q Appl Math 1944;2:164–8.
[44] De Giorgi MG, Ficarella A, Tarantino M. Error analysis of short term wind power prediction models. Appl Energy 2011;88:1298–311.
[45] Makridakis S, Wheelwright SC, Hyndman RJ. Forecasting: methods and applications. 3rd ed. New York: Wiley & Sons; 1998.
[46] Montgomery DC. Introduction to statistical quality control. 5th ed. New York (USA): John Wiley & Sons; 2005.
[47] Kusiak A, Li W. The prediction and diagnosis of wind turbine faults. Renew Energy 2011;36:16–23.