IET Renewable Power Generation

Research Article

Wind turbine power curve estimation based on earth mover distance and artificial neural networks

ISSN 1752-1416
Received on 30th April 2019
Revised 31st July 2019
Accepted on 9th September 2019
E-First on 29th October 2019
doi: 10.1049/iet-rpg.2019.0530
www.ietdl.org

Li Bai[1] ✉, *Emanuele Crisostomi*[1], *Marco Raugi*[1], *Mauro Tucci*[1]
[1] *Department of Energy, Systems, Territory and Constructions Engineering, University of Pisa, Largo Lucio Lazzarino, 56122, Pisa, Italy*
✉ *E-mail: baili123246530@gmail.com*

Abstract: A data-based estimation of the wind–power curve in wind turbines may be a challenging task due to the presence of anomalous data, possibly due to wrong sensor reads, operation halts, malfunctions or other. In this study, the authors describe a data-based procedure to build a robust and accurate estimate of the wind–power curve. In particular, they combine a joint clustering procedure, where both the wind speeds and the power data are clustered, with an Earth Mover Distance-based Extreme Learning Machine algorithm to filter out data that poorly contribute to explain the unknown curve. After estimating the cut-in and the rated speed, they use a radial basis function neural network to fit the filtered data and obtain the curve estimate. They extensively compared the proposed procedure against other conventional methodologies over measured data of nine turbines, to assess and discuss its performance.

1 Introduction

1.1 Motivation

Accurate identification of wind turbine power curve plays a fundamental role in different applications, such as wind power assessment and forecasting [1, 2], wind turbine selection, condition monitoring [3–5], troubleshooting, and predictive control and optimisation [6, 7]. In practice, the theoretical power captured by a wind turbine depends, among other things, on the air density, which is itself a function of temperature, pressure and humidity, which clearly depend on where the turbine is installed [1]. In addition, the power curve is basically nonlinear and non-stationary because of the fluctuating and stochastic nature of the wind resource, and it also comprises a noise component which represents all the unavailable microscopic interactions [8]. Finally, the power curve is affected by the conditions of the turbine and its associated equipment. For this reason, aging, wear and tear of turbine, anomalies and faults, blade condition, yaw and pitch misalignments, controller settings, and so forth, cause the power curve to depart from actual values [1]. Accordingly, it has been observed that in a wind farm, power produced by turbines with identical specifications may significantly differ, even if the wind speed is the same. This is partly due to the fact that the shadowing effect of turbines causes this difference as the turbines which operate in wake of other turbines may get reduced wind speeds. However, it is also due to factors such as wear, tear, aging, or dirt and ice deposition on blades, which may vary from turbine to turbine.

As a matter of fact, the International Electrotechnical Commission (IEC) technical committee has prepared the International Standard IEC 61400-12-1 to specify the standard methodology for measuring the power performance characteristics of a single wind turbine [9]. This procedure requires simultaneous measurement of wind speed and power output (for a sufficiently long duration to create a significant database under varying atmospheric conditions) [6]. Then, given such a sequence of pairs of measured wind speeds and corresponding generated power, the problem of the estimation of the wind turbine power curve, which is investigated in this paper, becomes that of accurately identifying the wind–power function.

1.2 State-of-the-art

Generally, the process of the identification of the wind turbine power curve consists of three steps: (i) a pre-processing step is required to clean the data of the obvious outliers; we shall denote this step as *filtering*; (ii) an intermediate step is required to accurately identify the cut-in and the rated wind speeds. In fact, the shape of the power curve heavily depends on such quantities [10]. If the cut-in and the rated wind speeds are exactly known, then the procedure described in Section 3.1 is not necessary; (iii) finally the filtered data are fitted to obtain a final curve in each wind interval. The whole or part of the three steps is included in the following methods.

Various methods have been widely employed in power curve modeling and can mainly be divided into two categories as parametric and non-parametric methods [6, 11–14]. The former category mainly contains linearised segmented model [15], polynomial regression [8, 16], maximum principle method, dynamic power curve, logistic regression [17] and probabilistic model. On the other hand, the latter one contains copula [18], cubic spline interpolation, neural network [19], fuzzy logic and data mining algorithms [20, 21].

In practice, these algorithms are applied with one or both techniques of filtering and clustering. A fuzzy logic algorithm with clustering centre is shown to give better performance than a classic least square method [22]. A Gaussian Process (GP)-based data filtering technique and artificial neural network are proposed to model the power curve with independent wind turbines datasets in which each turbine wind speed is measured by the nacelle anemometers [23]. A hybrid approach, including parametric methods with clustering and filtering steps, is proposed to give better performance than an only parametric method, in which *k*-means and *k*-medoids are both used [24]. When using clustering techniques, only wind speed datasets are clustered [22] or only wind power datasets are clustered [24].

1.3 Contribution

The various methods mentioned above are applied together with one or both techniques of filtering and clustering, while only the input data of wind speeds or only the output data of wind power output are used in those techniques. Instead of exploring information on only input or output data, we explore the information from both by using the Earth Mover Distance (EMD) method.

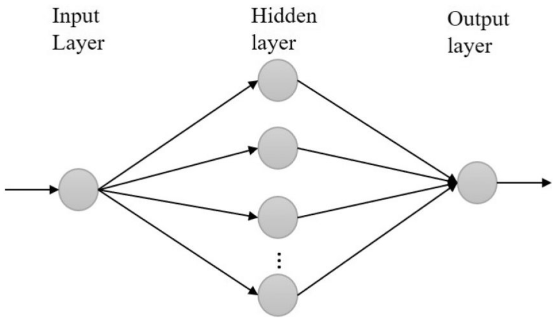

Fig. 1 *ELM structure*

In this paper, we illustrate a methodology that combines the three phases of filtering, identification and fitting of the data to obtain an improved estimate of the power curve. As for the filtering stage, differently from the literature, we jointly cluster both the historical data-sets of wind speeds and also their corresponding generated power. This allows us to retain the most frequent pairs of wind speeds and power and to discard the unlikely ones (e.g. occurrences of high generated power when the wind is weak or absent). For this purpose, we evaluate the distance between any two clusters as the average l_2 error between actual power measurements and their predictions (by using an Extreme Learning Machine (ELM)). Then, an EMD filter is designed as the outcome of a joint optimisation problem where the constraints are given by the relative frequency of occurrences of wind and power data. In an iteration process, the probabilities of cluster pairs related to outliers converge to zero, ensuring the outliers samples make no contribution to the estimate of the wind turbine power curve. After the samples are filtered, the cut-in and the rated wind speeds are identified via a segmented linear regression for each wind cluster. Finally, a radial basis function neural network (RBFNN) is used to fit the remaining data within the identified cut-in and rated wind speeds. Extensive results obtained on real data measured from nine wind turbines with rated power of 2 MW are provided to evaluate the performance of the proposed procedure against other well-established methodologies in wind power forecasting. Also, the impact of single choices, as for instance the filtering stage, or the choice of the weights of the ELM, is evaluated by comparing the results obtained with and without single steps.

As stated above, the key contributions are threefold:

- An EMD-based filter is designed as an optimisation problem whose objective function is the sum of weighted squared error instead of the commonly used mean square error model, where the weights are included in the solution of the optimisation problem with constraints determined by both distributions of input and output data.
- An ELM-based wind turbine power curve model included in the filter design is applied as a coarse model to make the filter be more easily implemented and robust.
- A comprehensive comparison and analysis of the proposed wind–power curve is done by using other conventional methods as benchmarks and taking into account of different factors such as the data size and the initial conditions of ELM + EMD-based filters.

2 Data clustering and filtering

Given the availability of a dataset of pairs of wind and its corresponding generated power by a wind turbine, wind–power curve estimation problem corresponds to the task of finding the mapping function from wind (input variable) to power (output variable). In this paper, we shall denote such a mapping function as f.

In the filtering phase, we use the outcome of a plain ELM identification on the input–output data as an initial condition for such an estimate of f. ELM is an efficient technique for training a single hidden-layer feed-forward NN, in that it combines the advantages of a simple formulation and an extremely fast training speed [25]. The general structure of an ELM is given in Fig. 1. The number of the neurons in the hidden layer is denoted as s and the infinitely differential activation function in the hidden layer is denoted as ϕ.

Let us further denote K independent observations by (x_k, y_k), $k = 1, \dots, K$ in the dataset, and x_k is the explanatory variable (wind speed), and y_k is the response variable (wind power output). The ELM-based power output y_k regarding the wind x_k can be expressed as

$$y_k = f(a_\alpha, b_\alpha, z_\alpha, x_k) = \sum_{\alpha = 1}^{s} z_\alpha \phi(a_\alpha x_k + b_\alpha) \tag{1}$$

where $\alpha = 1, \dots, s$, a_α is the input weight connecting the input neuron and the α'th hidden neuron, z_α is output weight connecting the α'th hidden neuron and the output neuron, and b_α is the bias of the α'th hidden neuron. The hidden layer parameters a_α and b_α are randomly generated and constant. We use the hyperbolic tangent function as the activation function. For simplicity, since a_α and b_α are constant, we shall express (1) as

$$y_k = f(z_\alpha, x_k) = \sum_{\alpha = 1}^{s} z_\alpha g_\alpha(x_k) \tag{2}$$

where only the output weights $\{z_\alpha\}$, $\alpha = 1, \dots, s$, need to be estimated.

2.1 Earth mover distance

EMD was firstly proposed to solve transportation problems, and then it became widely used as a distance measure in image retrieval [26, 27]. The original problem can be formalised as a linear programming problem: Let $P = \{(p_1, w_{p_1}), \dots, (p_m, w_{p_m})\}$ be the first signature with m clusters, where p_i is the cluster representative and w_{p_i} is the weight of the cluster; $Q = \{(q_1, w_{q_1}), \dots, (q_n, w_{q_n})\}$ be the second signature with n clusters, where q_j is the cluster representative and w_{q_j} is the weight of the cluster; and $\boldsymbol{D} = \{d_{ij}\}$ be the ground distance matrix, where d_{ij} is the distance between clusters p_i and q_j and denotes the cost of moving a unit from cluster p_i to q_j. Specifically, two signatures P and Q can be viewed as two datasets, and they are grouped into m and n clusters, respectively.

A transportation flow $\boldsymbol{T} = \{t_{ij}\}$ is to be found, with the flow t_{ij} between p_i and q_j, that minimises the overall cost [26, 27], as

$$\min \sum_{i = 1}^{m} \sum_{j = 1}^{n} t_{ij} d_{ij}, \tag{3}$$

subject to the constraints

$$t_{ij} \ge 0, \ 1 \le i \le m, \ 1 \le j \le n, \tag{4}$$

$$\sum_{j = 1}^{n} t_{ij} \le w_{p_i}, \ 1 \le i \le m, \tag{5}$$

$$\sum_{i = 1}^{m} t_{ij} \le w_{q_j}, \ 1 \le j \le n, \tag{6}$$

$$\sum_{i = 1}^{m} \sum_{j = 1}^{n} t_{ij} = \min\left(\sum_{i = 1}^{m} w_{p_i}, \sum_{j = 1}^{n} w_{q_j}\right). \tag{7}$$

2.2 EMD-based filter

Now we show how the EMD-based theory previously recalled can be applied in practice in our case. In the EMD formulation, two signatures of P and Q are two sets of clusters. In our optimisation problem, the input set $X = \{x_i\}$ is treated as the clustered set of

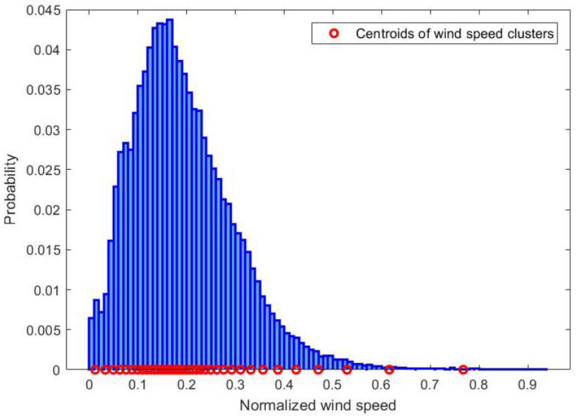

Fig. 2 *Wind histogram for the dataset of Turbine 1: the red circles refer to the centroids of wind speed clusters obtained by the one-dimensional SOM method*

wind speeds and the output set $Y = \{y_j\}$ is treated as that of wind power outputs of each wind turbine. A Self-Organising Map (SOM) is an unsupervised learning algorithm introduced by Kohonen, and it can project high-dimensional patterns onto a low-dimensional topology map for data visualisation and data properties exploration [28]. SOMs learn to cluster data based on similarity, topology, with a preference of assigning the same number of instances to each class [29]. A one-dimensional SOM is applied to cluster the wind speed dataset X and the wind power output dataset Y into m and n clusters, respectively, and the probability histograms of clusters x_i and y_i are denoted as $\{w_{x_i}\}$ and $\{w_{y_i}\}$. So they satisfy the following condition that:

$$\sum_{i=1}^{m} w_{x_i} = 1, \quad \sum_{j=1}^{n} w_{y_j} = 1. \tag{8}$$

As two clustering procedures are conducted independently for the wind speeds and the power outputs, any data pair (x, y) has clustering class labels (i, j) representing wind speed x with a class label i and power output y with a class label j, with $m \times n$ possibly different clusters. Let us denote by K_{ij} the number of the data samples (x_k^i, y_k^j) in each cluster pair (x_i, y_j), $1 \le i \le m$, $1 \le j \le n$ and $1 \le k \le K_{ij}$. d_{ij} is defined as distance for pair cluster (i, j) as follows:

$$d_{ij} = \frac{1}{K_{ij}} \sum_{k=1}^{K_{ij}} \| y_k^j - f(x_k^i) \|^2, \tag{9}$$

where function f refers to the ELM modelled mapping function in (1). Thus, substituting (2) and (9) into (3), the EMD-based optimised problem can be expressed as

$$\min_{z_\alpha, t_{ij}} \sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij} \frac{1}{K_{ij}} \sum_{k=1}^{K_{ij}} \| y_k^j - \sum_{\alpha=1}^{s} z_\alpha g_\alpha(x_k^i) \|^2, \tag{10}$$

subject to the constraints

$$0 \le t_{ij} \le 1, \; 1 \le i \le m, \; 1 \le j \le n, \tag{11}$$

$$\sum_{j=1}^{n} t_{ij} \le w_{x_i}, \; 1 \le i \le m, \tag{12}$$

$$\sum_{i=1}^{m} t_{ij} \le w_{y_j}, \; 1 \le j \le n, \tag{13}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij} = \min\left(\sum_{i=1}^{m} w_{x_i}, \sum_{j=1}^{n} w_{y_j} \right) = 1. \tag{14}$$

It can be noticed that the optimisation problem is a non-linear programming problem with constraints, and sequential quadratic programming (SQP) is applied to solve it.

For the sake of comparison, we now formulate the same problem as it is usually solved in the literature, where the mean squared error (MSE) is used as the optimisation objective function with certain constraints or regularisers in neural network (NN) and support vector regression (SVR). MSE can be expressed in the general form as

$$\text{MSE} = \frac{1}{K} \sum_{k=1}^{K} \| y_k - f(x_k) \|^2, \tag{15}$$

where K is the total number of sample pairs. Considering our notation, it can be rewritten as

$$\begin{aligned} \text{MSE} &= \frac{1}{K} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{K_{ij}} \| y_k^j - f(x_k^i) \|^2 \\ &= \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{K_{ij}}{K} \frac{1}{K_{ij}} \sum_{k=1}^{K_{ij}} \| y_k^j - f(x_k^i) \|^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{K_{ij}}{K} d_{ij} \end{aligned} \tag{16}$$

in which $\sum_{i=1}^{m} \sum_{j=1}^{n} K_{ij} = K$. Comparing (16) to (10), we can notice that now

$$t_{ij} = \frac{K_{ij}}{K}. \tag{17}$$

Thus, it can be noted that in the conventional approaches where the MSE is minimised, the values of t_{ij} are fixed and equal to the relative frequency of occurrences of wind/power pairs in the dataset. This implies that also outliers, though with low weight, contribute to the estimate of the power curve. On the other hand, in our case, the EMD-based filter can display all the clusters of datasets with their specific importance index and, most importantly, it filters out the outliers. In fact, in the training process, the t_{ij} corresponding to the outlier clusters diminish to zero.

Remark: We used here a one-dimensional SOM to separate wind speeds and power data into m and n clusters, respectively. The one-dimensional weights of the SOM network can be interpreted as the centroids. However, in principle, other clustering methods may be used as well, as the K-means method. The selection of the optimal values of m and n is a critical step. Here, we used the popular Calinski–Harabasz index [30] for this purpose. According to this index, we obtained $m = n = 40$ for both wind speeds and power outputs in our datasets.

For the sake of clarity, Fig. 2 shows the weights of the SOM clustering method for a wind turbine when $m = 40$. The wind data shown in Fig. 2 correspond to the first wind turbine of the case study that will be described in more detail in Section 4.1. It can be noted that the centroids are distributed more densely in the wind speed interval of higher probability.

3 Power curve modelling

In the previous section, we described how the dataset could be cleaned from candidate outliers. In this section, we now describe how the filtered dataset can be fitted to obtain a power curve for each wind turbine. In particular, we are interested in modelling the power curve between the cut-in and the rated speeds. Roughly speaking, this is the most interesting part of the power curve, as the generated power is expected to be practically zero for wind speeds smaller than the cut-in speed, and constant as the rated power capacity for wind speeds greater than the rated speed (and smaller than the cut-out speed).

3.1 Identification of the cut-in and the rated speeds

Piecewise linear functions are used to approximate the wind–power relationship for each wind cluster, similarly as a linearised segmented model [11]. In practice, one would expect the slope of such linear functions to be close to zero when the wind speed is lower than the cut-in speed, and then again when it is higher than the rated speed. Each piecewise linear function can be expressed as

$$p_k = \beta_k w_k + e_k, \tag{18}$$

where p_k represents the wind power corresponding to the wind speed w_k in the k'th wind speed cluster. As m clusters are acquired by the one-dimensional SOM described in the last section, we have m affine functions overall. Parameters β_k and e_k are computed by simple linear regression for each wind cluster.

Let us consider again the data of the first turbine in our datasets. As 40 clusters are used in the one-dimensional SOM, thus 40 slope values for each cluster and 39 boundaries for any 2 adjacent clusters are obtained. All the identified slopes and actual wind speed values are displayed in blue and red curves, respectively, and the identified cut-in speed and the rated speed are highlighted in two parallel lines in Fig. 3. It can be noted that 40 clusters are used in the one-dimensional SOM and the slope starts increasing from the 6th cluster and dropping to zero from the 33th cluster. Therefore, the cut-in speed is selected as the average SOM weights between the 5th and the 6th clusters and the rated speed is selected as the average SOM weights between the 32th and the 33th clusters in a similar way. In detail, the identified cut-in speed is 2.99 m/s and the obtained rated speed is 11.23 m/s in this case study.

3.2 RBFNN-based power curve modelling

After having identified the cut-in and the rated speeds, only the wind–power pairs having wind speed in the interval of our interest are considered to model the wind–power curve. Let us again denote K independent observations by $(x_k, y_k), k = 1, \dots, K$ in the dataset, and x_k is the explanatory variable (wind speed), and y_k is the response variable (wind power output). A Gaussian non-linear regression model is used [31]

$$y_k = f(x_k) + \varepsilon_k, \quad k = 1, \dots, K, \tag{19}$$

where $f(x_k)$ can be expressed as a linear combination of s radially symmetric non-linear basis functions $\phi_\alpha(x)$:

$$f(x_k) = \sum_{\alpha = 1}^{s'} w_\alpha \phi_\alpha(x_k) + w_0 \,. \tag{20}$$

Each basis function forms a localised receptive field in the input space, and the most commonly used radial basis function is the Gaussian basis

$$\phi_\alpha(x, \mu_\alpha, \gamma) = \exp(-\gamma \| \, x - \mu_\alpha \, \|^2), \quad \alpha = 1, \dots, s', \tag{21}$$

where μ_α is the centre of the radial basis function for the unit α, $1/\gamma$ is the width parameter and $\| \cdot \|$ is the Euclidean norm. MSE is applied as the loss function in a Gaussian kernel-based NN. The number of centroids s' and the window width $1/\gamma$ is reciprocally related, because lower window width implies that each centroid has less contribution than others, and thus more centroids are needed. The hyperparameters of $\gamma = 10$ and $s' = 60$ are acquired in our dataset by grid search by means of 5-fold cross-validation [32], when RBFNN is used together with EMD to construct an RBFNN + EMD-based filter. For the consistency and fairness for comparison, we also use the same number of neurons in hidden layer in ELM + EMD-based filter, namely $s = 60$. As we are interested in the power curve modelling within the interval between the cut-in speed and the rated speed, the centroids used in RBFNN for the RBFNN + EMD-based filter falling in the interval are used as the centroids for the RBFNN in the fitting phase, and the number of these centroids is naturally the number of neurons (e.g. in our case, $s' = 48$ in the fitting phase). As the centroids are kept the same in the interval, the hyperparameter $\gamma = 10$ as well.

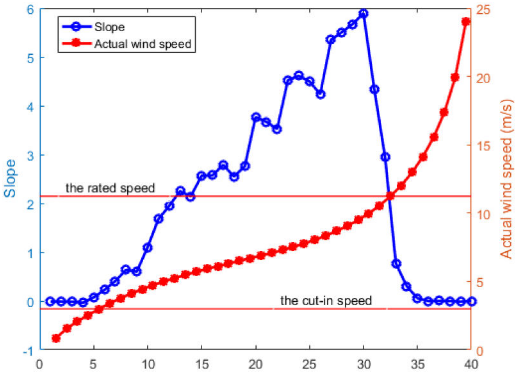

Fig. 3 *Slopes of piecewise linear functions of Turbine 1: the blue-dotted line indicates the slope curve of any two neighbouring clusters; the red-dotted line refers to the actual wind speed corresponding to each cluster*

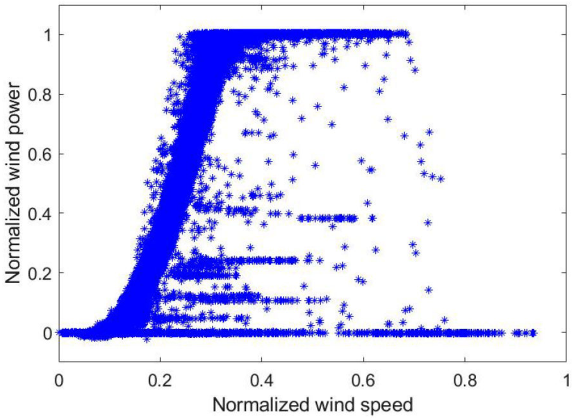

Fig. 4 *Pairs of the normalised wind speed and its corresponding normalised wind power for Turbine 1*

4 Evaluation

4.1 Data description

In our case study, we consider a wind farm that contains nine turbines, where each turbine has a nominal power of 2 MW. The dataset contains 103,309 samples with a time resolution of 10 min for each wind turbine. Each sample consists of a wind speed measured at hub height, and its corresponding measured power. We used 70% of the total samples to estimate the wind–power curve (training dataset). Then, we used the remaining 30% to compare different strategies (test dataset). In particular, note that, obviously, the test set is not clustered nor filtered. All the data samples corresponding to the first turbine are presented in Fig. 4. Note that many outliers can be seen even by just visual inspection (e.g. occurrences of large wind speeds and their corresponding very small power generation). Such samples should be carefully handled when modelling the wind turbine power curve.

4.2 Performance indices

The three most commonly used evaluation metrics for this application are the MSE, the mean absolute error (MAE) and the mean absolute percentage error (MAPE)

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^{N} \| \, p_k - p_k' \, \|^2, \tag{22}$$

$$\text{MAE} = \frac{1}{N} \sum_{k=1}^{N} |p_k - p_k'|, \tag{23}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{k=1}^{N}\left|\frac{p_k - p'_k}{p_k}\right|, \qquad (24)$$

where p_k and p'_k denote the normalised actual power output and the normalised predicted power output, respectively (both of them are normalised by dividing by the nominal power), N is the size of the test dataset. Note that the MAPE error (24) is not well defined when the actual power at the denominator has too small values. To avoid numerical issues, only for the computation of the MAPE, we have excluded samples when the normalised actual power is smaller than 10^{-4}.

4.3 Performance comparison

We now use the training data to obtain different wind–power curves by adopting many different methodologies, and then compared the ability of such curves to explain the (unfiltered and uncleaned) test data. In particular, we want to assess the importance of the filtering step, and we want to compare parametric (i.e. 4-degree polynomial regression) fitting algorithms and non-parametric (i.e. RBFNN) fitting algorithms on both the full speed range (i.e. from a wind speed equal to zero up to the cut-out speed) and on the specific interval of our interest (i.e. between the cut-in speed and the rated speed). In the fitting phase, we first use RBFNN, a special artificial NN (ANN). For the sake of comparison, we also consider other shallow ANNs of three layers, i.e. a feedforward NN (FNN) and an ELM as benchmarks. The transfer function of the hidden layer of such FNN is a hyperbolic tangent sigmoid function that is mathematically equivalent to the hyperbolic tangent function used for the hidden layer in ELM in both filtering and fitting phases described in Section 2. To show the performance of the proposed filtering technique, we use GP filtering for comparison, as proposed in [23]. All methodologies, and their corresponding abbreviations, are listed in Table 1. More specifically, the first part of the abbreviation indicates whether the methodologies are compared over the full (f-) or the single interval (i-) of wind speeds. The second part indicates whether a filter is used in the pre-processing step and whether RBFNN, ELM or GP have been used for this purpose. Finally, the last part of the abbreviation regards the fitting step, and whether a polynomial regression, RBFNN, FNN or ELM is used. In addition, to be fair for the comparison among three ANNs (i.e. RBFNN, FNN, or ELM) in the fitting phase, the same number of neurons in the hidden layer is used (i.e. the number is 48 according to our datasets).

4.3.1 Importance of the filtering step: The importance of the filtering step can be also evaluated by visual inspection by comparing Fig. 5 (no filtering), and Fig. 6, where the filtered samples are shown in red. In particular, the fitting curve of f-RBFNN in Fig. 5 performs very poorly for wind speeds greater than the rated speed.

Results of all algorithms in both the training and the test set for the first turbine are reported in Table 2. From the table, it is possible to appreciate again that any time the filtering step is employed, the performance is better than the same methodology without the filtering step. On the other side, the use of RBFNN in the filtering phase provides slightly better results than those of GP and ELM (in conjunction with EMD). This can be explained by the fact that RBFNN gives higher priority to the data in the neighbourhood (selected by the centroids) than other data samples (including outliers), while GP uses all the datasets. As for ELM, the randomly generated weights of the edges connecting the first two layers can only capture coarse input features. Moreover, the performances of ELM + EMD-based filter and GP filter are similar. Considering the lower computational overhead of ELM + EMD-based filter (see in Section 5), we are interested in discussing its properties in the following.

4.3.2 ANNs outperform polynomial regression: From Table 2, it is also possible to observe that in general ANNs systematically outperform PolyReg (i.e. polynomial regression). The results of RBFNNs, FNNs and ELMs applied in the fitting phases are similar, provided that an appropriate filtering step is applied first. This result seems to suggest that non-parametric methods may be more convenient than parametric ones for this specific fitting application. Also, while the test error is in general greater than the training error (e.g. because test data are not filtered), still the errors remain very close. This seems to suggest that the models had been well trained, and no overfitting or underfitting issues have been encountered.

Table 1 Power curve modelling methods

Speed range	Filter	Fitting	Abbreviation
full	NO	PolyReg	f-PolyReg
full	NO	RBFNN	f-RBFNN
full	NO	FNN	f-FNN
full	NO	ELM	f-ELM
full	—	RBFNN + EMD	f-RBFNN + EMD
interval	NO	PolyReg	i-PolyReg
interval	NO	RBFNN	i-RBFNN
interval	NO	FNN	i-FNN
interval	NO	ELM	i-ELM
interval	ELM + EMD	PolyReg	i-ELM + EMD + PolyReg
interval	ELM + EMD	RBFNN	i-ELM + EMD + RBFNN
interval	ELM + EMD	FNN	i-ELM + EMD + FNN
interval	ELM + EMD	ELM	i-ELM + EMD + ELM
interval	RBFNN + EMD	PolyReg	i-RBFNN + EMD + PolyReg
interval	RBFNN + EMD	RBFNN	i-RBFNN + EMD + RBFNN
interval	RBFNN + EMD	FNN	i-RBFNN + EMD + FNN
interval	RBFNN + EMD	ELM	i-RBFNN + EMD + ELM
interval	GP	PolyReg	i-GP + PolyReg
interval	GP	RBFNN	i-GP + RBFNN
interval	GP	FNN	i-GP + FNN
interval	GP	ELM	i-GP + ELM

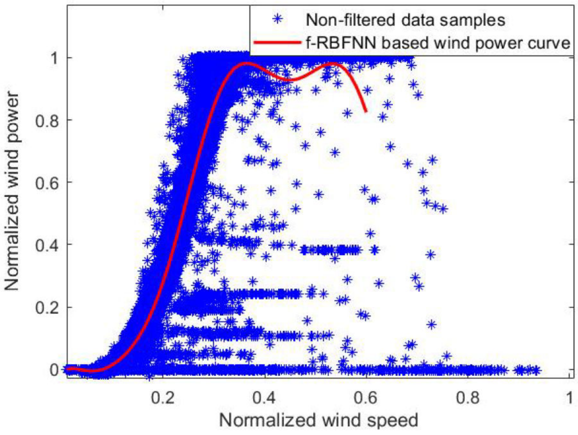

Fig. 5 *Non-filtered data of Turbine 1: the stars correspond to the original pairs of normalised wind speed and its corresponding normalised power; the red line displays the non-accurate f-RBFNN model of non-filtered data*

As the performances of all three ANNs used in the fitting phase are similar, we only use RBFNN as an example of ANN in the following analysis.

4.3.3 Sensitivity on the initial conditions of the ELM + EMD-based filter: The ELM + EMD-based filter behaves in a stochastic way as the weights of the edges connecting the input layer and hidden layer are generated randomly. We tested that the randomness of the algorithm did not have an impact on final results, by computing the performance indices for ten different runs, starting from ten different (randomly generated) initial conditions. Results for the first turbine are displayed in Table 3. Both polynomial regression and RBFNN perform similarly and consistently with various weights of ELM.

4.3.4 Generalisation of the results: We have generalised the obtained results by implementing the proposed method on all nine the turbines. Given the small geographical size of the overall wind farm, all nine the turbines provided similar results. Also, the same hyperparameters were used for all turbines (e.g. 40 clusters for both the wind and the power datasets for all turbines). The results shown in Table 4, for the specific comparison between a polynomial regression fitting strategy and the RBFNN, show that the results already given for the first turbine, can be in fact extended to all the other turbines as well.

4.3.5 Sensitivity on the size of the training dataset: Finally, we also analysed the sensitivity of the size of the training dataset. Given that the original size of the training data is rather large (72,316 samples), we compared the performance for four different sizes of the training dataset, corresponding to $K = 72,316$, $K/10$, $K/100$ and $K/1000$, respectively. The results are presented in Table 5, as the mean of ten different implementations to average out stochastic effects. It can be noticed that the indices of the test errors in both methods present three levels according to the size of the training dataset: the size of K belongs to the first level where all the indices are significantly better than those of other two levels, especially MSEs; the size ranging from $K/10$ to $K/100$ falls into the second level where all the indices are practically constant (about 30% worse than before in terms of MSE); when the size is

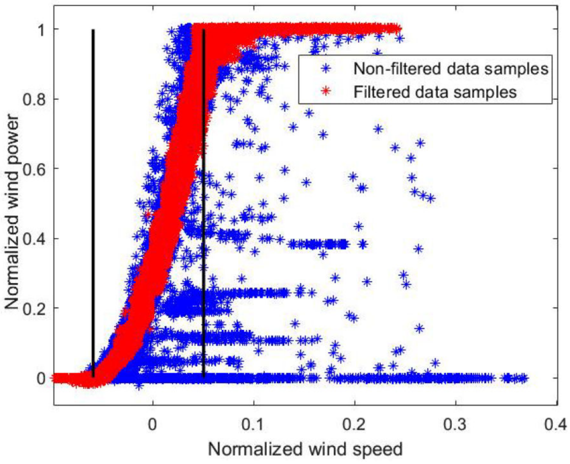

Fig. 6 *Filtered wind power samples by ELM + EMD-based filter of Turbine 1: blue stars represent the original and non-filtered samples, red stars represent the filtered samples, and the two black lines represent the identified cut-in and rated wind speeds*

Table 2 Power curve modelling comparison for the dataset of Turbine 1

Methods	Training error			Test error		
	MSE	MAE	MAPE	MSE	MAE	MAPE
f-PolyReg	0.0113	0.0562	0.9594	0.0110	0.0558	0.9260
f-RBFNN	0.0093	0.0385	0.3028	0.0092	0.0386	0.3356
f-FNN	0.0092	0.0381	0.2752	0.0093	0.0384	0.3287
f-ELM	0.0095	0.0418	0.5521	0.0094	0.0418	0.5559
f-RBFNN + EMD	0.0150	0.0713	0.9658	0.0147	0.0710	0.9675
i-PolyReg	0.0066	0.0377	0.2816	0.0069	0.0383	0.2572
i-RBFNN	0.0066	0.0377	0.2838	0.0069	0.0383	0.2583
i-FNN	0.0066	0.0378	0.2686	0.0069	0.0384	0.2435
i-ELM	0.0066	0.0377	0.2838	0.0069	0.0383	0.2582
i-ELM + EMD + PolyReg	0.0014	0.0259	0.2067	0.0071	0.0377	0.2619
i-ELM + EMD + RBFNN	0.0014	0.0259	0.1987	0.0071	0.0377	0.2553
i-ELM + EMD + FNN	0.0014	0.0257	0.1631	0.0071	0.0379	0.2351
i-ELM + EMD + ELM	0.0014	0.0259	0.1957	0.0071	0.0377	0.2531
i-RBFNN + EMD + PolyReg	0.0013	0.0251	0.2154	0.0070	0.0379	0.2572
i-RBFNN + EMD + RBFNN	0.0013	0.0251	0.1991	0.0070	0.0379	0.2470
i-RBFNN + EMD + FNN	0.0012	0.0248	0.1694	0.0070	0.0384	0.2347
i-RBFNN + EMD + ELM	0.0013	0.0251	0.1981	0.0070	0.0379	0.2464
i-GP + PolyReg	0.0015	0.0267	0.2500	0.0070	0.0346	0.2547
i-GP + RBFNN	0.0014	0.0262	0.2674	0.0070	0.0375	0.2660
i-GP + FNN	0.0014	0.0261	0.2317	0.0070	0.0375	0.2433
i-GP + ELM	0.0014	0.0264	0.2290	0.0070	0.0377	0.2415

Table 3 Sensitivity of the initial random weights of ELM + EMD-based filter on the dataset of Turbine 1

	i-ELM + EMD + PolyReg						i-ELM + EMD + RBFNN					
	Training error			Test error			Training error			Test error		
Run	MSE	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE
1	0.0012	0.0237	0.2487	0.0062	0.0355	0.2810	0.0012	0.0237	0.2372	0.0062	0.0355	0.2710
2	0.0015	0.0274	0.1823	0.0076	0.0388	0.2367	0.0014	0.0273	0.1795	0.0076	0.0387	0.2273
3	0.0015	0.0278	0.2448	0.0078	0.0385	0.2786	0.0015	0.0274	0.1654	0.0078	0.0385	0.2295
4	0.0014	0.0264	0.2534	0.0078	0.0386	0.2810	0.0013	0.0260	0.1768	0.0078	0.0384	0.2291
5	0.0014	0.0262	0.2351	0.0077	0.0384	0.2553	0.0014	0.0259	0.1864	0.0077	0.0384	0.2314
6	0.0014	0.0260	0.2039	0.0070	0.0375	0.2554	0.0014	0.0260	0.2327	0.0070	0.0375	0.2765
7	0.0013	0.0263	0.2433	0.0079	0.0387	0.2737	0.0013	0.0261	0.1812	0.0079	0.0386	0.2315
8	0.0013	0.0248	0.2323	0.0061	0.0356	0.2617	0.0013	0.0248	0.2364	0.0061	0.0357	0.2632
9	0.0013	0.0256	0.1660	0.0077	0.0391	0.2236	0.0013	0.0255	0.1672	0.0077	0.0389	0.2243
10	0.0013	0.0247	0.2318	0.0064	0.0359	0.2582	0.0013	0.0247	0.2342	0.0064	0.0360	0.2588

Table 4 Robustness validation of ELM + EMD-based filter on the whole datasets of nine turbines

	i-ELM + EMD + PolyReg						i-ELM + EMD + RBFNN					
	Training error			Test error			Training error			Test error		
Turbine	MSE	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE
1	0.0014	0.0261	0.1957	0.0068	0.0377	0.2365	0.0014	0.0261	0.1955	0.0068	0.0377	0.2349
2	0.0014	0.0267	0.2184	0.0164	0.0549	0.3051	0.0014	0.0265	0.2169	0.0164	0.0553	0.3202
3	0.0012	0.0242	0.2185	0.0060	0.0344	0.3269	0.0012	0.0242	0.2194	0.0060	0.0344	0.3290
4	0.0009	0.0213	0.2035	0.0057	0.0316	0.2828	0.0009	0.0212	0.1686	0.0057	0.0317	0.2458
5	0.0012	0.0229	0.2337	0.0051	0.0319	0.3958	0.0012	0.0229	0.2316	0.0051	0.0319	0.3957
6	0.0011	0.0235	0.2044	0.0071	0.0353	0.2663	0.0011	0.0234	0.2057	0.0071	0.0353	0.2682
7	0.0014	0.0250	0.2164	0.0070	0.0363	0.3129	0.0014	0.0251	0.2342	0.0070	0.0363	0.3260
8	0.0009	0.0196	0.3315	0.0072	0.0371	0.4930	0.0009	0.0194	0.2921	0.0072	0.0371	0.4622
9	0.0011	0.0226	0.2815	0.0032	0.0280	0.4018	0.0011	0.0223	0.2417	0.0032	0.0280	0.3752

Table 5 Sensitivity of ELM + EMD-based filter on the size of the training datasets of Turbine 1

	i-ELM + EMD + PolyReg						i-ELM + EMD + RBFNN					
	Training error			Test error			Training error			Test error		
Size	MSE	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE
K	0.0014	0.0259	0.2242	0.0072	0.0377	0.2605	0.0014	0.0257	0.1997	0.0072	0.0376	0.2443
$K/10$	0.0013	0.0259	0.2216	0.0092	0.0410	0.2626	0.0012	0.0240	0.1698	0.0091	0.0392	0.2194
$K/100$	0.0014	0.0273	0.2477	0.0095	0.0425	0.2440	0.0012	0.0244	0.1879	0.0094	0.0403	0.2077
$K/1000$	0.0083	0.0442	0.4074	0.0089	0.0438	0.3478	0.0075	0.0415	0.6080	4.7817	0.1295	0.5799

around $K/1000$, the MSE becomes too high, meaning that the size of the dataset is too small to support a polynomial regression or a NN model.

4.3.6 Comparison of the computational cost of the filters: Both the ELM + EMD-based filter and the RBFNN + EMD-based filter have been applied in our proposed method. ELMs randomly generate the weights of the first layer in the former filter, while RBFNNs are applied with the basis centres chosen by the SOM clustering algorithm in the latter one. The computational costs of two filters are compared with the size of the training dataset $K = 72,316$ in the filtering phase by using MATLAB (R2017b) on a 2.8-GHz Intel Core processor, and the elapsed time was 1456.65 s for RBFNN + EMD-based filter and 569.35 s for ELM + EMD-based filter. In contrast, the computation cost of GP filter is 3554 s as GP filtering involves high-dimensional matrix operations. Accordingly, it can be seen that the ELM + EMD-based filter is far less computationally demanding than the RBFNN + EMD one, specifically reducing more than half computation cost of the RBFNN + EMD one.

5 Conclusion

In this paper, we have described a novel procedure consisting of three steps to build a robust wind power curve for wind turbines: first, one-dimensional SOMs were used to cluster wind and power data into different groups, respectively, and then ELM + EMD-based filter was used to filter out the data that poorly explained the wind–power curve. We then identified the cut-in and the rated speed, as the wind speeds before which, and after which, the generated power is practically constant. Finally, we used RBFNN to fit the remaining data in the wind interval of interest. We have provided extensive results to compare the outcomes of different choices for different steps. In particular, we have focused on the importance of the filtering step, and the superiority of the RBFNN (or other ANNs in general) over other polynomial fitting algorithms. Specifically, the performance of our proposed ELM + EMD-based filtering method is similar to that of the benchmark of GP filtering, while the computational cost of our method is far lower than that of GP filtering.

Typical applications of the estimated wind–power curves include wind power assessment and forecasting, and online monitoring for earlier detection of incipient faults. In principle, more accurate wind–power curves would have a beneficial impact on both applications. We are currently starting investigating such lines of research, and we are interested in quantifying the gained advantage of an improved power curve estimation for both the aforementioned applications.

6 Acknowledgment

We thank the company i-EM srl (Flyby Group, via Lampredi 45 - Livorno (Italy)) for fruitful discussions and for providing the raw data used in this paper.

7 References

[1] Sohoni, V., Gupta, S. C., Nema, R. K.: 'A critical review on wind turbine power curve modelling techniques and their applications in wind based energy systems', *J. Energy*, 2016, **2016**, p. 8519785

[2] Renani, E. T., Elias, M. F. M., Rahim, N. A.: 'Wind power prediction using enhanced parametric wind power curve modeling'. 4th IET Clean Energy and Tech. Conf. (CEAT 2016), Kuala Lumpur, Nov. 2016, pp. 1–5

[3] Gonzalez, E., Stephen, B., Infield, D., *et al.*: 'Using high-frequency scada data for wind turbine performance monitoring: a sensitivity study', *Renew. Energy*, 2019, **131**, pp. 841–853

[4] Kusiak, A., Zheng, H., Song, Z.: 'Models for monitoring wind farm power', *Renew. Energy*, 2009, **34**, (3), pp. 583–590

[5] Chen, N., Yu, R., Chen, Y., *et al.*: 'Hierarchical method for wind turbine prognosis using scada data', *IET Renew. Power Gener.*, 2017, **11**, (7), pp. 403–410

[6] Lydia, M., Kumar, S. S., Selvakumar, A. I., *et al.*: 'A comprehensive review on wind turbine power curve modeling techniques', *Renew. Sustain. Energy Rev.*, 2014, **30**, pp. 452–460

[7] Altin, M., Hansen, A., Göksu, Ö., *et al.*: 'Wind turbine and wind power plant modelling aspects for power system stability studies'. Proc. Int. Conf. Wind Energy Grid-Adaptive Technologies, Jeju, South Korea, 2014

[8] Xu, M., Pinson, P., Lu, Z., *et al.*: 'Adaptive robust polynomial regression for power curve modeling with application to wind power forecasting', *Wind Energy*, 2016, **19**, (12), pp. 2321–2336

[9] IEC: 'Wind turbines – part 12-1: power performance measurements of electricity producing wind turbines', 2005

[10] Chang, T. P., Liu, F. J., Ko, H.H., *et al.*: 'Comparative analysis on power curve models of wind turbine generator in estimating capacity factor', *Energy*, 2014, **73**, pp. 88–95

[11] Lydia, M., Selvakumar, A. I., Kumar, S. S., *et al.*: 'Advanced algorithms for wind turbine power curve modeling', *IEEE Trans. Sustain. Energy*, 2013, **4**, (3), pp. 827–835

[12] Wadhvani, R., Shukla, S.: 'Analysis of parametric and non-parametric regression techniques to model the wind turbine power curve', *Wind Eng.*, 2018, **43**, pp. 225–232

[13] Taslimi-Renani, E., Modiri-Delshad, M., Elias, M. F. M., *et al.*: 'Development of an enhanced parametric model for wind turbine power curve', *Appl. Energy*, 2016, **177**, pp. 544–552

[14] Pandit, R., Infield, D., Kolios, A.: 'Comparison of advanced nonparametric models for wind turbine power curves', *IET Renew. Power Gener.*, 2019, **13**, pp. 1503–1510

[15] Javadi, M., Malyscheff, A. M., Wu, D*., et al.*: 'An algorithm for practical power curve estimation of wind turbines', *CSEE J. Power Energy Syst.*, 2018, **4**, (1), pp. 93–102

[16] Marčiukaitis, M., Žutautaitė, I., Martišauskas, L*., et al.*: 'Non-linear regression model for wind turbine power curve', *Renew. Energy*, 2017, **113**, pp. 732–741

[17] Villanueva, D., Feijóo, A.: 'Comparison of logistic functions for modeling wind turbine power curves', *J. Electr. Power Syst. Res.*, 2018, **155**, pp. 281–288

[18] Gill, S., Stephen, B., Galloway, S.: 'Wind turbine condition assessment through power curve copula modeling', *IEEE Trans. Sustain. Energy*, 2012, **3**, (1), pp. 94–101

[19] Pelletier, F., Masson, C., Tahan, A.: 'Wind turbine power curve modeling using artificial neural network', *Renew. Energy*, 2016, **89**, pp. 207–214

[20] Schlechtingen, M., Santos, I. F., Achiche, S.: 'Using data-mining approaches for wind turbine power curve monitoring: a comparative study', *IEEE Trans. Sustain. Energy*, 2013, **4**, (3), pp. 671–679

[21] Ouyang, T., Kusiak, A., He, Y.: 'Modeling wind-turbine power curve: a data partitioning and mining approach', *Renew. Energy*, 2017, **102**, pp. 1–8

[22] Üstüntaş, T., Şahin, A. D.: 'Wind turbine power curve estimation based on cluster center fuzzy logic modeling', *J. Wind Eng. Ind. Aerodyn.*, 2008, **96**, (5), pp. 611–620

[23] Manobel, B., Sehnke, F., Lazzús, J.A*., et al.*: 'Wind turbine power curve modeling based on Gaussian processes and artificial neural networks', *Renew. Energy*, 2018, **125**, pp. 1015–1020

[24] Yesilbudak, M.: 'Implementation of novel hybrid approaches for power curve modeling of wind turbines', *Energy Convers. Manage.*, 2018, **171**, pp. 156–169

[25] Huang, G. B., Zhou, H., Ding, X*., et al.*: 'Extreme learning machine for regression and multiclass classification', *IEEE Trans. Syst. Man Cybern. B, Cybern.*, 2012, **42**, (2), pp. 513–529

[26] Rubner, Y., Tomasi, C., Guibas, L. J.: 'The earth mover's distance as a metric for image retrieval', *Int. J. Comput. Vis.*, 2000, **40**, (2), pp. 99–121

[27] Rubner, Y., Tomasi, C.: '*The earth mover's distance*' (Springer US, Springer, Boston, Massachusetts, USA, 2001)

[28] Kohonen, T.: '*Self-organizing maps*' (Springer Science and Business Media, Berlin Heidelberg, Germany, 2012)

[29] Vesanto, J., Alhoniemi, E.: 'Clustering of the self-organizing map', *IEEE Trans. Neural Netw.*, 2000, **11**, (3), pp. 586–600

[30] Caliński, T., Harabasz, J.: 'A dendrite method for cluster analysis', *Commun. Stat.*, 1997, **3**, pp. 1–27

[31] Rasmussen, C. E., Williams, C. K. I.: '*Gaussian processes for machine learning (adaptive computation and machine learning)*' (The MIT Press, Cambridge, Massachusetts, USA, 2005)

[32] Rodriguez, J.D., Perez, A., Lozano, J. A.: 'Sensitivity analysis of *k*-fold cross validation in prediction error estimation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, **32**, pp. 569–575