

# Wind Power Curve Modeling and Wind Power Forecasting With Inconsistent Data

Yun Wang, Qinghua Hu , Senior Member, IEEE, Dipti Srinivasan, Senior Member, IEEE, and Zheng Wang

**Abstract**—Wind power curve modeling is a challenging task due to the existence of inconsistent data, in which the recorded wind power is far away from the theoretical wind power at a given wind speed. In this case, confronted with these samples, the estimated errors of wind power will become large. Thus, the estimated errors will present two properties: heteroscedasticity and error distribution with a long tail. In this paper, according to the above-mentioned error characteristics, the heteroscedastic spline regression model (HSRM) and robust spline regression model (RSRM) are proposed to obtain more accurate power curves even in the presence of the inconsistent samples. The results of power curve modeling on the real-world data show the effectiveness of HSRM and RSRM in different seasons. As HSRM and RSRM are optimized by variational Bayesian, except the deterministic power curves, probabilistic power curves, which can be used to detect the inconsistent samples, can also be obtained. Additionally, with the data processed by replacing the wind power in the detected inconsistent samples with the wind power on the estimated power curve, the forecasting results show that more accurate wind power forecasts can be obtained using the above-mentioned data processing method.

**Index Terms**—Power curve modeling, wind power forecasting, heteroscedasticity, robustness, inconsistent samples.

## I. INTRODUCTION

CURRENTLY, wind power as a clean and renewable energy has been receiving more and more attention [1]. Large-scale wind power integration will ease energy stress to a certain extent, and bring economic and environmental benefits. However, owing to the intermittency and randomness of wind power, it will be a challenge to the stability of power system operations

Manuscript received November 11, 2017; revised February 24, 2018; accepted March 25, 2018. Date of publication March 28, 2018; date of current version December 14, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61732011, and in part by the Open Fund of State Key Laboratory of Operation and Control of Renewable Energy & Storage Systems (China Electric Power Research Institute) and in part by the China Scholarship Council. Paper no. TSTE-01028-2017. (Corresponding author: Qinghua Hu.)

Y. Wang is with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: wangyun15@tju.edu.cn).

Q. Hu is with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China, and also with the State Key Laboratory of Engines, Tianjin University, Tianjin 300072, China (e-mail: huqinghua@tju.edu.cn).

D. Srinivasan is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: dipti@nus.edu.sg).

Z. Wang is with the State Key Laboratory of Operation and Control of Renewable Energy & Storage Systems, China Electric Power Research Institute, Beijing 10092, China (e-mail: wangzheng1618@yeah.net).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSTE.2018.2820198

as the wind power penetration increases [2]. Therefore, more accurate wind power forecasting models are necessary for integration purpose, which are also helpful to reduce the operating costs [3], [4].

In literature, wind power forecasting models can be grouped into two categories: direct forecasting approaches and indirect forecasting approaches [3]. The former mainly utilize the historical wind power data to forecast future wind power. However, the latter are two-step methods, which generate the wind speed forecasts and then provide the wind power forecasts by power curve, which reflects the nonlinear relationship between wind speed and wind power [5]. Therefore, the wind power forecasting error is mainly caused by the forecasting error of wind speed for indirect models.

In general, the theoretical power curve is provided by manufactures without considering the environment factors (e.g. temperature and humidity) [6]. In practice, it will change depending on the terrain and climate of different areas, etc. Thus, using the theoretical power curve to conduct wind power forecasting will lead to additional errors [3], [7]. The practical power curve should be estimated. In addition to helping to forecast wind power when wind speed forecasts are available, an accurate power curve facilitates online monitoring and reduces operation and maintenance costs of wind turbines [8].

Generally, power curve modeling techniques can be divided into two groups: parametric models and non-parametric models. Usually, parametric models, including warping function [9], linearized segmented model [10], polynomial power curve, ideal power curve [11], probabilistic model [12], dynamical power curve [13], four and five-parameter logistic models (4-PLM and 5-PLM) [10], [14], are constructed by mathematic expressions with several parameters to describe power curves [7], [15], as well as the modified hyperbolic tangent model [8]. For polynomial model, cubic power curve [6], quadratic power curve [16], sixth and ninth degree polynomials [3], [17] are often used. The main drawback of parametric techniques is their limited performances in describing the dynamic nature of power curve [18]. Unlike parametric models, non-parametric models are not made up of mathematical expressions and don't need any prior knowledge about the shape of power curve. They can model various power curves only using the real wind speed and wind power data. Examples of non-parametric models contain spline regression model (SRM) [15], artificial neural networks (ANNs) [19], fuzzy methods [20] and data mining methods (e.g. support vector machine (SVM) [18], random forest, boosting algorithm and  $K$ -nearest neighbor (KNN) [21]). Although

non-parametric techniques are more flexible than parametric techniques, they suffer from higher computation costs [18].

Apart from the power curve modeling approaches, another factor that affects the acquisition of accurate power curves is the data quality [22]. In reality, there are many inconsistent data in our databases due to several reasons such as sensor errors, blade damage, maintenance issues, wind curtailment and environment factors (e.g., ice) [8], [22]. The main character of these data is that the real wind power is far away from the theoretical wind power on the power curve at a given wind speed. In order to get accurate power curves, some researchers use flitter methods to process these inconsistent data in advance [3], [8], [18], [22]. For instance, Taslimi-Renani *et al.* firstly grouped the wind speed into many small intervals, then the mean  $\mu_p$  and the deviation  $\sigma_p$  of the corresponding wind power data in each interval were computed, the samples in which wind power data were located out of the boundary  $[\mu_p - \sigma_p, \mu_p + \sigma_p]$  were seen as the inconsistent data and were flitted [8]. Marčiukaitis *et al.* employed a statistic model named Tukeys method to detect the inconsistent samples [3]. Kusiak *et al.* used a KNN model to construct a power curve, then used residual approach and control charts to detect the inconsistent data [23]. Zhao *et al.* proposed a data-driven method combining a quartile method and a density-based clustering method to eliminate the inconsistent samples [22].

However, for the current detection models, we cannot ensure that all inconsistent samples can be identified in any conditions, there may still be some inconsistent samples hidden in our processed data. In such case, the estimated errors of those power curve modeling techniques may present two new characters. Owing to the presence of inconsistent samples, the estimated errors when encountering these samples will be larger than that of normal samples. First, the variances of the estimated errors will vary with the samples, which shows that the error is heteroscedastic [24]. Second, the larger errors will be far away from zero, so the distribution of the estimated errors will present a long tail. However, the estimated error in the current power curve modeling techniques is often assumed to be Gaussian distributed [15], that cannot deal with the heteroscedasticity and long-tail error distribution well. In this paper, within the Bayesian framework, heteroscedastic spline regression model (HSRM) and robust spline regression model (RSRM) are proposed to deal with the above problem. The main contributions of this work are summarized as follows:

- 1) HSRM, in which the regression errors of training samples are assumed to be Gaussian distributed with different variances, is first developed to model power curves with the data contaminated by some inconsistent samples.
- 2) RSRM, which uses an infinite mixture of Gaussians (IMoG) to model the error distribution with long tail, is also first designed to model power curves with the data corrupted by some inconsistent samples.
- 3) HSRM and RSRM are optimized via variational Bayesian. Thus, they can generate deterministic and probabilistic power curves simultaneously. And, the latter can also be used to detect inconsistent samples in power data.
- 4) We develop a new strategy for wind power forecasting. First, inconsistent samples are detected by HSRM and

RSRM, and replaced by the data on the estimated power curves. Second, with the processed data, ANN based forecasting model is constructed to obtain power forecasts.

The rest of this paper is constructed as follows. Section II introduces some power curve models. Two proposed models based on spline regression are presented in Section III. Section IV shows the applications of the proposed models, including power curve modeling, inconsistent samples detecting and wind power forecasting. Section V concludes the paper and shows the future work.

## II. WIND POWER CURVE MODELS

Currently, many approaches are employed to model power curves. Here, some representative parametric models and non-parametric models are introduced.

### A. Parametric Models

1) *Four-Parameter Logistic Model (4-PLM)*: Owing to the shape of 4-PLM is similar to the real power curves, it is widely used to model them. The expression of 4-PLM is

$$y_P = \alpha(1 + \beta e^{x_S/\gamma}) / (1 + \delta e^{x_S/\gamma}) \quad (1)$$

where  $y_P, x_S$  are the wind power and wind speed, respectively. Four parameters  $\alpha, \beta, \gamma, \delta$  control the shape of the power curve. Usually, intelligent optimization methods are employed to get the four optimal parameters based on the objective function [8],

$$\min_{\alpha, \beta, \gamma, \delta} \sum_{i=1}^N (\hat{y}_P^i - y_P^i)^2 \quad (2)$$

where  $\hat{y}_P^i$  is the estimated wind power while  $y_P^i$  is the real wind power.

2) *Five-Parameter Logistic Model (5-PLM)*: 5-PLM is also employed to model power curves, its expression is given by

$$y_P = \alpha + (\beta - \alpha) / (1 + (x_S/\gamma)^\delta)^\epsilon \quad (3)$$

where  $\alpha, \beta, \gamma, \delta, \epsilon$  are the corresponding five parameters, and  $\gamma, \epsilon \geq 0$ . Similar to optimize 4-PLM, intelligent optimization methods are also employed to tune 5-PLM by the same objective function as expressed in (2) [8].

3) *Polynomial Regression*: Generally, a  $p$ th degree polynomial regression can be expressed as

$$y_P = a_0 + a_1 x_S + a_2 x_S^2 + \dots + a_p x_S^p \quad (4)$$

where  $a_0, \dots, a_p$  denote the corresponding parameters. In literature, sixth and ninth degree polynomial regression models (6-PRM, 9-PRM) have been tested to have good performances in modeling power curves. And least squares method is usually used to estimate the model parameters  $a_0, \dots, a_p$  [15].

### B. Non-Parametric Models

1) *Artificial Neural Network (ANN)*: ANNs are powerful tools to model nonlinear characteristics. The most popular one in ANNs is BP neural network, which can approximate any nonlinear continuous functions with proper structure and weights in theoretically [25]. A general BP has three layers: input layer, hidden layer and output layer. Given a BP neural network with

$D$  input neurons,  $H$  hidden neurons and one output neuron, the output of  $j$ th neuron in the hidden layer ( $y_j^h$ ) with input vector  $\{x_1^{in}, \dots, x_D^{in}\}$  is

$$y_j^h = f\left(\sum_d w_{dj} x_d^{in} + b_j\right), (j = 1, \dots, H) \quad (5)$$

where  $f(\cdot)$  is a Sigmoid activation function,  $w_{dj}$  is the connection weight from the  $d$ th input node to the  $j$ th hidden node,  $b_j$  is the bias of hidden node  $j$ . The output of the network  $y_j^o$  can be obtained via an activation function (e.g. a linear function)  $f^o(\cdot)$ , the calculation is

$$y_j^o = f^o\left(\sum_j w_j^o y_j^h + b^o\right) \quad (6)$$

where  $w_j^o$  is the connection weight between the  $j$ th hidden node and the output node,  $b^o$  is a bias. BP neural network uses error gradient descent algorithm to minimize the total error  $E$ , which is defined as  $E = \sum_{i=1}^N (y_i^o - y_i^r)^2$ , where  $N$  is the length of the training data,  $y_i^o$  and  $y_i^r$  are the output of network and the real data of the  $i$ th sample, respectively [25].

2) *Support Vector Machine (SVM)*: The basic idea of support vector machine is using a nonlinear function  $\phi(\cdot)$  to map the input into a high-dimension feature space, in which a linear regression is constructed. The most popular one is the  $\varepsilon$ -SVM with  $\varepsilon$ -nonsensitive loss function. Given the training data  $\{x_i, y_i\}_{i=1}^N$ , based on the minimum structure risk criteria, the objective function is

$$\begin{aligned} \min_{w, b, \zeta_i, \zeta_i^*} & \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^N (\zeta_i + \zeta_i^*) \\ \text{s.t.} & \begin{cases} \langle \mathbf{w} \cdot \phi(x_i) \rangle + b - y_i \leq \varepsilon + \zeta_i \\ y_i - \langle \mathbf{w} \cdot \phi(x_i) \rangle - b \leq \varepsilon + \zeta_i^* \\ \zeta_i, \zeta_i^* \geq 0, i = 1, 2, \dots, N. \end{cases} \end{aligned} \quad (7)$$

where  $\zeta_i, \zeta_i^*$  are the two slack variables,  $\langle \mathbf{w} \cdot \phi(x_i) \rangle$  denotes the inner produce of  $\mathbf{w}$  and  $\phi(x)$ ,  $\varepsilon$  is a positive value,  $C$  is the penalty parameter,  $\mathbf{w}$  and  $b$  are the regression coefficient vector and the bias, respectively.

After taking the Lagrangian, the estimated function in dual representation is given by

$$g(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) K(x_i, x) + b \quad (8)$$

where  $\alpha_i, \alpha_i^*$  denote the solutions of the dual problem,  $K(x_i, x) = \langle \phi(x_i), \phi(x) \rangle$  is the kernel function. Usually, radial basis function (RBF) kernel function is used.

3) *Spline Regression Model (SRM)*: Generally, spline regression model can be expressed as  $y_i = f(x_i) + e_i$ , where  $e_i$  is the error term. Owing to there are many types of spline basis functions, the expressions of  $f(x_i)$  are also different. Here, we introduce two popular spline bases: truncated power basis and B-spline basis.

Given  $x_i \in [a, b]$ , and divide the interval  $[a, b]$  by  $K$  distinct points,  $(\xi_0 =) a < \xi_1 < \xi_2, \dots, \xi_K < b (= \xi_{K+1})$ ,  $\xi_1, \xi_2, \dots, \xi_K$  are called knots. When using the truncated power basis

to form a spline regression model,  $f(x_i)$  is expressed as

$$f(x_i) = \beta_0 + \sum_{r=1}^p \beta_r x_i^r + \sum_{k=1}^K \beta_{k+p} (x_i - \xi_k)_+^p \quad (9)$$

where  $p$  means the order of spline,  $\beta_0, \beta_1, \dots, \beta_{K+p}$  are the regression coefficients, and  $(x_i - \xi_k)_+^p$  is defined as [15]

$$(x_i - \xi_k)_+^p = \begin{cases} 0 & x_i < \xi_k \\ (x_i - \xi_k)^p & x_i \geq \xi_k \end{cases} \quad (10)$$

Before constructing the B-spline basis based spline regression model, the augmented knot sequence  $\vartheta$  is defined as [26]

$$\begin{cases} \vartheta_0 \leq \vartheta_1 \leq \dots \leq \vartheta_p \leq \xi_0 \\ \vartheta_{p+k} = \xi_k, k = 1, \dots, K \\ \xi_{K+1} \leq \vartheta_{K+p+1} \leq \dots \leq \vartheta_{K+2p} \end{cases} \quad (11)$$

Then,  $f(x_i)$  of the B-spline regression is

$$f(x_i) = \sum_{j=0}^{K+p} \beta_j B_j^p(x_i) \quad (12)$$

where

$$\begin{aligned} B_j^1(x_i) &= \begin{cases} 1, & x_i \in [\vartheta_j, \vartheta_{j+1}) \\ 0, & \text{otherwise} \end{cases} \\ B_j^p(x_i) &= \frac{x_i - \vartheta_j}{\vartheta_{j+p-1} - \vartheta_j} B_j^{p-1}(x_i) + \frac{\vartheta_{j+p} - x_i}{\vartheta_{j+p} - \vartheta_{j+1}} B_{j+1}^{p-1}(x_i) \end{aligned} \quad (13)$$

Given the training samples of size  $N$ , the above two spline regression models can be rewritten as the same matrix form,

$$\mathbf{Y} = \mathbf{Z}\boldsymbol{\beta} + \mathbf{e} \quad (14)$$

where  $\mathbf{Y}, \mathbf{e} \in R^N$ ,  $\boldsymbol{\beta} \in R^{K+p+1}$  and  $\mathbf{Z} \in R^{N \times (K+p+1)}$ . The estimated parameter vector  $\boldsymbol{\beta}$  by least squares method is  $\hat{\boldsymbol{\beta}} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{Y}$  [15].

### III. PROPOSED SPLINE REGRESSION MODELS

In this section, the proposed models (HSRM and RSRM) optimized by variational Bayesian are introduced in detail.

#### A. Heteroscedastic Spline Regression Model

An underlying assumption of the least squares method is that  $e_i$  obeys a normal distribution with zero mean and a constant variance for all samples. However, the variance of the estimated error in case of inconsistent samples will be larger than that of normal sample. Thus, the variance is not a constant but varies with the samples. A heteroscedastic spline regression model is proposed here to solve this problem.

1) *Prior Distributions*: To construct a HSRM,  $e_i$  is assumed to obey a Gaussian distribution with mean 0 and variance  $\sigma_i^{-1}$ , which is given a Gamma prior distribution to complete the Bayesian model,

$$p(e_i | \sigma_i) = \mathcal{N}(e_i | 0, \sigma_i^{-1}), p(\sigma_i) = \mathcal{G}(\sigma_i | a_0, b_0) \quad (15)$$

where  $\mathcal{N}(\cdot)$  and  $\mathcal{G}(\cdot)$  denote the Gaussian and Gamma distributions, respectively,  $a_0, b_0$  are two hyperparameters.

So, the likelihood function of training samples  $\{x_i, y_i\}_{i=1}^N$  is computed as

$$p(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\beta}, \boldsymbol{\sigma}) = \prod_{i=1}^N \mathcal{N}(y_i|\mathbf{Z}_i\boldsymbol{\beta}, \sigma_i^{-1}) \quad (16)$$

The prior distribution on regression coefficient  $\beta_j$  is usually assumed to be Gaussian distributed with mean 0 and variance  $\lambda_j^{-1}$ , and  $\lambda_j$  is assumed to obey a Gamma distribution parameterized by  $c_0, d_0$ ,

$$p(\beta_j|\lambda_j) = \mathcal{N}(\beta_j|0, \lambda_j^{-1}), \quad p(\lambda_j) = \mathcal{G}(\lambda_j|c_0, d_0) \quad (17)$$

Then, our goal turns to infer the posterior distributions of all involved variables in the following joint probability density function (PDF),

$$p(\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\lambda}|\mathbf{Y}, \mathbf{Z}) = p(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\beta}, \boldsymbol{\sigma})p(\boldsymbol{\beta}|\boldsymbol{\lambda})p(\boldsymbol{\sigma})p(\boldsymbol{\lambda}) \quad (18)$$

where  $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_N]$ ,  $\boldsymbol{\sigma} = [\sigma_1, \dots, \sigma_N]$ .

2) *Posterior Distributions*: According to the theory of variational Bayesian (VB) [27], taking the expectation of the logarithmic PDF in (18),  $\log p(\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\lambda}|\mathbf{Y}, \mathbf{Z})$ , with respect to all parameters except  $\boldsymbol{\beta}$ , the logarithmic posterior distribution of  $\boldsymbol{\beta}$  can be inferred and expressed as

$$\begin{aligned} \log Q(\boldsymbol{\beta}) &= \boldsymbol{\beta}^T \boldsymbol{\Sigma}_\beta^{-1} \boldsymbol{\beta} - \boldsymbol{\beta}^T \mathbf{Z}^T \text{diag}(\langle \boldsymbol{\sigma} \rangle) \mathbf{Y} \\ &\quad - \mathbf{Y}^T \text{diag}(\langle \boldsymbol{\sigma} \rangle) \mathbf{Z} \boldsymbol{\beta} + \text{const.} \end{aligned} \quad (19)$$

where  $\boldsymbol{\Sigma}_\beta^{-1} = [\text{diag}(\langle \boldsymbol{\lambda} \rangle) + \mathbf{Z}^T \text{diag}(\langle \boldsymbol{\sigma} \rangle) \mathbf{Z}]$ ,  $\langle \cdot \rangle$  means the expectation of a random variable, function  $\text{diag}(\cdot)$  will produce a diagonal matrix, *const.* denotes a constant. Therefore, the posterior distribution of  $\boldsymbol{\beta}$  is a Gaussian distribution,

$$Q(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta) \quad (20)$$

parameterized with mean  $\boldsymbol{\mu}_\beta$  and covariance  $\boldsymbol{\Sigma}_\beta$ ,

$$\begin{aligned} \boldsymbol{\mu}_\beta &= \boldsymbol{\Sigma}_\beta \mathbf{Z}^T \text{diag}(\langle \boldsymbol{\sigma} \rangle) \mathbf{Y} \\ \boldsymbol{\Sigma}_\beta &= [\text{diag}(\langle \boldsymbol{\lambda} \rangle) + \mathbf{Z}^T \text{diag}(\langle \boldsymbol{\sigma} \rangle) \mathbf{Z}]^{-1} \end{aligned} \quad (21)$$

Using the same method to obtain the posterior distribution of  $\boldsymbol{\beta}$ , the posterior distribution of  $\sigma_i$  is a Gamma distribution parameterized with two parameters  $a_i, b_i$ ,

$$\begin{aligned} Q(\sigma_i) &= \mathcal{G}(\sigma_i|a_i, b_i) \\ a_i &= \frac{1}{2} + a_0, \quad b_i = \frac{1}{2} \langle (y_i - \mathbf{Z}_i \boldsymbol{\beta})^2 \rangle + b_0 \end{aligned} \quad (22)$$

Similarly, the posterior distribution of parameter  $\lambda_j$  is also a Gamma distribution,

$$Q(\lambda_j) = \mathcal{G}(\lambda_j|c_j, d_j) \quad (23)$$

where

$$c_i = \frac{1}{2} + c_0, \quad d_i = \frac{1}{2} \langle \beta_j^2 \rangle + d_0 \quad (24)$$

3) *Parameter Estimations and Forecasting*: Based on the principle of VB [28], the HSRM algorithm can be summarized in Algorithm 1.

According to the outputs of Algorithm 1, given  $\mathbf{Z}_i$  computed by (9) and (12), the distribution of  $y_i$  is a Gaussian distribution

---

**Algorithm 1:** The HSRM algorithm.

---

**Input:** *maxiter*,  $a_0, b_0, c_0, d_0$ .

**Output:**  $\boldsymbol{\mu}_\beta, a_i, b_i$ .

Initialize:  $\boldsymbol{\sigma}, \boldsymbol{\lambda}$

**for** *iter* = 1, *iter* < *maxiter* **do**

$\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta$  optimized according to Eq.(21) ;

$a_i$  and  $b_i$  optimized according to Eq.(22) ;

$c_i$  and  $d_i$  optimized according to Eq.(24) ;

**end**

return  $\boldsymbol{\mu}_\beta, a_i, b_i$  ;

---

with mean  $\mathbf{Z}_i \boldsymbol{\mu}_\beta$  and variance  $\langle \sigma_i \rangle^{-1} + \mathbf{Z}_i \boldsymbol{\Sigma}_\beta \mathbf{Z}_i^T$ , which can be inferred from

$$\begin{aligned} \int \mathcal{N}(y_i|\mathbf{Z}_i\boldsymbol{\beta}, \langle \sigma_i \rangle^{-1}) \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta) d\boldsymbol{\beta} \\ = \mathcal{N}(y_i|\mathbf{Z}_i\boldsymbol{\mu}_\beta, \langle \sigma_i \rangle^{-1} + \mathbf{Z}_i \boldsymbol{\Sigma}_\beta \mathbf{Z}_i^T) \end{aligned} \quad (25)$$

According to the posterior of  $\sigma_i$ ,  $\langle \sigma_i \rangle^{-1} = b_i/a_i$ . For a new wind speed  $x^*$ , based on the spline basis, the corresponding  $\mathbf{Z}^*$  can be obtained. However, the estimated variance is related with the wind power  $y^*$ , which is unknown. So,  $b_i/a_i$  can not be computed directly. In this paper, it is suggested to be replaced by the mean of all estimated  $b_i/a_i$  by HSRM with the training samples, namely  $(\sum_{i=1}^N b_i/a_i)/N$ . Therefore, the distribution of  $y^*$  is

$$p(y^*) = \mathcal{N}\left(y^*|\mathbf{Z}^* \boldsymbol{\mu}_\beta, \frac{1}{N} \sum_{i=1}^N \frac{b_i}{a_i} + \mathbf{Z}^* \boldsymbol{\Sigma}_\beta \mathbf{Z}^{*T}\right) \quad (26)$$

Then, a probabilistic power curve can be obtained according to (26).

## B. Robust Spline Regression Model (RSRM)

To deal with the error distribution with long tail is to enhance the robustness of SRM. Single Gaussian distribution cannot fit the long-tail error distribution. Considering the superior fitting ability of IMoG, a robust spline regression model is constructed using IMoG to model the complex error distribution.

1) *Prior Distributions*: The probability density function of MoG can be expressed as

$$p(e_i) = \sum_{g=1}^G \pi_g \mathcal{N}(e_i|0, \tau_g^{-1}) \quad (27)$$

where  $G$  is the number of Gaussian components in MoG,  $\tau_g$  and  $\pi_g$  denote the precision and the proportion of the  $g$ th Gaussian, respectively, and  $\sum_{g=1}^G \pi_g = 1, \pi_g \geq 0$ . In the Bayesian framework, (27) can be represented by a two-level generated model with the indicator variable  $r_{ig}$ , which obeys a multinomial distribution with parameter  $\boldsymbol{\pi}$ , and  $\boldsymbol{\pi}$  is assumed to obey a Dirichlet distribution parameterized with  $\boldsymbol{\alpha}_0$ ,

$$\begin{aligned} p(e_i) &= \prod_{g=1}^G \mathcal{N}(e_i|0, \tau_g^{-1})^{r_{ig}} \\ p(\mathbf{r}_i) &= \mathcal{M}(\mathbf{r}_i|\boldsymbol{\pi}), \quad p(\boldsymbol{\pi}) = \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0) \end{aligned} \quad (28)$$

where  $\mathbf{r}_i = [r_{i1}, r_{i2}, \dots, r_{iG}]$ ,  $\boldsymbol{\pi} = [\pi_1, \pi_2, \dots, \pi_G]$  and  $\boldsymbol{\alpha}_0 = [\alpha_{01}, \alpha_{02}, \dots, \alpha_{0G}]$ . According to (28), the likelihood function of training samples is:

$$p(\mathbf{Y}|\mathbf{R}, \mathbf{Z}, \boldsymbol{\beta}, \boldsymbol{\tau}) = \prod_{i=1}^N \prod_{g=1}^G \mathcal{N}(y_i | Z_i \boldsymbol{\beta}, \tau_g^{-1})^{r_{ig}} \quad (29)$$

However, the main drawback of the original MoG model is that the number of Gaussian components in MoG should be determined before constructing a MoG model. Therefore, the original MoG model is extended into an IMoG model by reconstructing  $\boldsymbol{\pi}$  according to the theory of stick-breaking construction. The corresponding hierarchical Bayesian prior for  $\boldsymbol{\pi}$  becomes:

$$\begin{aligned} \pi_g(\boldsymbol{\varpi}) &= \varpi_g \prod_{m=1}^{g-1} (1 - \varpi_m) \\ p(\varpi_g | v) &= \mathcal{B}(\varpi_g | 1, v), \quad p(v) = \mathcal{G}(v | e_0, f_0) \end{aligned} \quad (30)$$

where  $\mathcal{B}(\cdot)$  is a Beta distribution. Under this assumption for  $\boldsymbol{\pi}$ , given a larger value of  $G$  randomly, the limitation of the sum of component weights equals to 1, namely:

$$\lim_{G \rightarrow \infty} \sum_{g=1}^G \pi_g = \lim_{G \rightarrow \infty} \sum_{g=1}^G \varpi_g \prod_{m=1}^{g-1} (1 - \varpi_m) = 1 \quad (31)$$

For the regression coefficient  $\beta_j$ , similar to the prior distributions in the HSRM model,  $\beta_j$  is given a Gaussian prior with zero mean and precision  $\varrho_j$ , and  $\varrho_j$  is assumed to obey a Gamma distribution:

$$p(\beta_j | \varrho_j) = \mathcal{N}(\beta_j | 0, \varrho_j^{-1}), \quad p(\varrho_j) = \mathcal{G}(\varrho_j | g_0, h_0) \quad (32)$$

Usually, a Gamma prior is put on the parameter  $\tau_g$ , namely:

$$p(\tau_g) = \mathcal{G}(\tau_g | s_0, t_0) \quad (33)$$

Then, we should infer the posterior distributions of all variables involved in the following joint PDF:

$$\begin{aligned} p(\mathbf{R}, \boldsymbol{\beta}, \boldsymbol{\varpi}, \boldsymbol{\varrho}, \boldsymbol{\tau}, v | \mathbf{Y}, \mathbf{Z}) &= p(\mathbf{Y} | \mathbf{R}, \mathbf{Z}, \boldsymbol{\beta}, \boldsymbol{\tau}) p(\boldsymbol{\tau}) \\ & p(\mathbf{R} | \boldsymbol{\varpi}) p(\boldsymbol{\beta} | \boldsymbol{\varrho}) p(\boldsymbol{\varrho}) p(\boldsymbol{\varpi} | v) p(v) \end{aligned} \quad (34)$$

2) *Posterior Distributions*: Based on the theory of VB, the logarithmic posterior distribution of  $\mathbf{R}$  can be expressed as

$$\log Q(\mathbf{R}) = \sum_{i=1}^N \sum_{g=1}^G r_{ig} \log \varphi_{ig} \quad (35)$$

where  $\log \varphi_{ig} = [\frac{1}{2} \langle \log \tau_g \rangle + \frac{1}{2} \log(2\pi) + \langle \log \pi_g \rangle - \frac{1}{2} \langle \tau_g \rangle \cdot \langle (y_i - \mathbf{Z}_i \boldsymbol{\beta})^2 \rangle]$ . According to the above equation, the posterior distribution of  $\mathbf{R}$  is also a multinomial distribution with the expectation of  $r_{ig}$  expressed as

$$\langle r_{ig} \rangle = \varphi_{ig} \left( \sum_{g=1}^G \varphi_{ig} \right)^{-1} \quad (36)$$

For  $\boldsymbol{\beta}$ , its posterior distribution is also a Gaussian distribution with mean  $\boldsymbol{\mu}_\beta$  and covariance  $\boldsymbol{\Sigma}_\beta$ ,

$$\begin{aligned} Q(\boldsymbol{\beta}) &= \mathcal{N}(\boldsymbol{\beta} | \boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta) \\ \boldsymbol{\mu}_\beta &= \boldsymbol{\Sigma}_\beta \mathbf{Z}^T \text{diag}(\mathbf{A}) \mathbf{Y} \\ \boldsymbol{\Sigma}_\beta &= [\text{diag}(\langle \boldsymbol{\varrho} \rangle) + \mathbf{Z}^T \text{diag}(\mathbf{A}) \mathbf{Z}]^{-1} \end{aligned} \quad (37)$$

where  $A_i = \sum_{g=1}^G \langle r_{ig} \rangle \langle \tau_g \rangle$  and  $\mathbf{A} = [A_1, \dots, A_N]$ .

The posterior distribution of  $\varrho_j$  is a Gamma distribution

$$Q(\varrho_j) = G(\varrho_j | g_{\varrho_j}, h_{\varrho_j}) \quad (38)$$

with the parameters  $g_{\varrho_j}$  and  $h_{\varrho_j}$  defined as

$$g_{\varrho_j} = g_0 + \frac{1}{2}, \quad h_{\varrho_j} = h_0 + \frac{1}{2} \langle \beta_j^2 \rangle \quad (39)$$

As to  $\varpi_g$ , its posterior is a Beta distribution,

$$Q(\varpi_g) = \mathcal{B}(\varpi_g | l_{\varpi_g}, m_{\varpi_g}) \quad (40)$$

with

$$\begin{aligned} l_{\varpi_g} &= 1 + \sum_{i=1}^N \langle r_{ig} \rangle \\ m_{\varpi_g} &= \langle v \rangle + \sum_{i=1}^N \sum_{q=g+1}^G \langle r_{iq} \rangle \end{aligned} \quad (41)$$

The posterior distribution of parameter  $v$  is a Gamma distribution

$$Q(v) = G(v | e_v, f_v) \quad (42)$$

with the parameter  $e_v$  and  $f_v$  defined as

$$\begin{aligned} e_v &= e_0 + G - 1 \\ f_v &= f_0 - \sum_{g=1}^{G-1} \langle \log(1 - \varpi_g) \rangle \end{aligned} \quad (43)$$

Similarly, the posterior distribution of  $\tau_g$  is also a Gamma distribution,

$$Q(\tau_g) = G(\tau_g | s_{\tau_g}, t_{\tau_g}) \quad (44)$$

with the parameter  $s_{\tau_g}$  and  $t_{\tau_g}$  defined as

$$\begin{aligned} s_{\tau_g} &= s_0 + \frac{1}{2} \sum_{i=1}^N \langle r_{ig} \rangle \\ t_{\tau_g} &= t_0 + \sum_{i=1}^N \langle r_{ig} \rangle \langle (y_i - \mathbf{Z}_i \boldsymbol{\beta})^2 \rangle \end{aligned} \quad (45)$$

3) *Parameter Estimations and Forecasting*: Algorithm 2 shows the full VB-based RSRM.

Based on the outputs of Algorithm 2, we can infer from (46) that the distribution of  $y^*$  is also a MoG.

$$\begin{aligned} \int \sum_{g=1}^G \langle \pi_g \rangle \mathcal{N}(y^* | \mathbf{Z}^* \boldsymbol{\beta}, t_{\tau_g} / s_{\tau_g}) \mathcal{N}(\boldsymbol{\beta} | \boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta) d\boldsymbol{\beta} \\ = \sum_{g=1}^G \langle \pi_g \rangle \mathcal{N}(y^* | \mathbf{Z}^* \boldsymbol{\mu}_\beta, t_{\tau_g} / s_{\tau_g} + \mathbf{Z}^* \boldsymbol{\Sigma}_\beta \mathbf{Z}^{*T}) \end{aligned} \quad (46)$$

where  $\langle \pi_g \rangle$  is the expectation of variable  $\pi_g$ .

---

**Algorithm 2:** The RSRM algorithm.

---

**Input:**  $maxiter, e_0, f_0, g_0, h_0, s_0, t_0$ .

**Output:**  $\mu_\beta, \Sigma_\beta, l_{\varpi_g}, m_{\varpi_g}, s_{\tau_g}, t_{\tau_g}$ .

Initialize:  $\mathbf{R}, \boldsymbol{\tau}, \boldsymbol{\rho}, \nu$

**for**  $iter = 1, iter < maxiter$  **do**

$\mu_\beta, \Sigma_\beta$  optimized according to Eq.(37) ;  
 $g_{\rho_j}$  and  $h_{\rho_j}$  are optimized according to Eq.(39) ;  
 $l_{\varpi_g}$  and  $m_{\varpi_g}$  are optimized according to Eq.(41) ;  
 $e_\nu$  and  $f_\nu$  are optimized according to Eq.(43) ;  
 $s_{\tau_g}$  and  $t_{\tau_g}$  are optimized according to Eq.(45) ;  
 $\langle r_{ig} \rangle$  is optimized according to Eq.(36) ;

**end**

return  $\mu_\beta, \Sigma_\beta, l_{\varpi_g}, m_{\varpi_g}, s_{\tau_g}, t_{\tau_g}$  ;

---

TABLE I  
THE NUMBERS OF SAMPLES PAIRS IN TRAINING AND TEST SETS

WTG	Dataset	Samples	Train	Test
WTG1	A (Spring)	6000	5000	1000
	B (Winter)	6000	5000	1000
WTG2	C (Spring)	7500	6000	1500
	D (Winter)	5800	4800	1000

#### IV. WIND POWER CURVE MODELING AND WIND POWER FORECASTING

Here, four real-world datasets from two wind turbine generators (WTG1 and WTG2) are employed to test the effectiveness of our proposed models. After that, wind power forecasting is conducted with the data processed by the estimated power curves. All experiments are conducted on MATLAB.

##### A. Data Descriptions

Datasets A and B are collected from WTG1, but come from spring and winter, respectively. However, Datasets C and D are gathered from WTG2, also collected in spring and winter, respectively. Dataset A contains 6000 sample pairs (wind speed  $S$  and wind power  $P$ ). Before constructing a power curve model of a wind turbine, the dataset is divided into two parts, the first 5000 samples pairs are used to train the power curve models and the rest sample pairs are employed to test the performances of different models. For the other three datasets, the numbers of samples in training set and test set are presented in Table I. For the above four datasets, all data are recorded every 10 minutes.

##### B. Evaluation Indices

Here, two indices, the mean absolute error (MAE) and the root mean squared error (RMSE), are employed to evaluate the performances of different models, computed as [21]

$$MAE = L^{-1} \sum_{l=1}^L |y_l - \hat{y}_l|$$

$$RMSE = \sqrt{L^{-1} \sum_{l=1}^L (y_l - \hat{y}_l)^2} \quad (47)$$

where  $L$  is the length of the test set,  $y_l, \hat{y}_l$  denote the actual wind power and the estimated wind power, respectively.

##### C. Power Curve Modeling

In this section, except the two proposed models (HSRM and RSRM), four parametric power curve modeling techniques (4-PLM, 5-PLM, 6-PRM and 9-PRM), one non-parametric model (SRM), and two learning-based models (ANN and SVM) are also taken as benchmark models to demonstrate the effectiveness of the proposed models. For all power curve modeling techniques, there are only one input (wind speed) and one output (wind power).

As to 4-PLM and 5-PLM, the optimal parameters are obtained by intelligent optimization methods [8]. Follow Taslimi's work [8], backtracking search algorithm (BSA) published on the website<sup>1</sup> is employed to tune 4-PLM and 5-PLM. The number of iteration in BSA is set as 5000 in this paper. Least squares method [15] is used to get the optimal parameters in 6-PRM and 9-PRM. The structure of BP neural network (a popular ANN) affects the final fitting results. Here, for power curve modeling, ANN with one hidden layer is used. The number of neurons in hidden layer is determined by the Hecht-Nelson method: the number of hidden neurons is  $2l + 1$  when the number of neurons in input layer is  $l$  [29], [30]. Owing to there is only one input (wind speed) for power curve modeling  $l = 1$ , the number of hidden neurons is 3. Then, BP neural networks are implemented on MATLAB by newff function with default settings. As to SVM, the kernel selected here is the RBF kernel, and we use grid search to select the optimal kernel parameter  $\kappa$  and penalty parameter  $C$ . For the proposed models, several parameters should be determined. Specifically,  $a_0, b_0, c_0, d_0$  in HSRM are set as 0.001,  $e_0, f_0, g_0, h_0, s_0, t_0$  in RSRM are also set as 0.001, the maximum number of iterations is set as 50 for two proposed models.

In practice, we can not guarantee that all inconsistent samples can be detected. Before estimating power curves, the original data are processed roughly by a simple detection method (SDM). For Dataset A, samples located in the intervals  $\{S > 5, P < 20\}$  and  $\{S > 10, P < 400\}$  will be detected as inconsistent samples and will be removed. Similarly, for Dataset B, the intervals are  $\{S > 6, P < 50\}$  and  $\{S > 13, P < 700\}$ , while they are  $\{S > 6, P < 50\}$  and  $\{S > 12, P < 600\}$  for Dataset C. As to Dataset D, the intervals are  $\{S > 4.5, P < 20\}$ ,  $\{S > 9, P < 400\}$  and  $\{S > 13.5, P < 800\}$ . Tables II shows the results of different power curve modeling techniques under the condition that the inconsistent samples are not completely removed. Also, the computational time of each model except SVM is presented in Table II. Because it is time-consuming to select optimal values of  $\kappa$  and  $C$  by grid search, SVM takes more than one hour to get the estimated power curves on four datasets.

For WTG1, from Table II, RSRM performs the best among all power curve modeling techniques in spring and winter. Compared with seven benchmark models, HSRM usually generates a power curve with a lower RMSE in two seasons. However, in terms of MAE, they are SVM and SRM that perform better than HSRM in spring and winter, respectively.

As to WTG2, from the results in Table II, RSRM outperforms the other models in spring, while it is HSRM that performs

<sup>1</sup><https://www.mathworks.com/matlabcentral/fileexchange/44842-backtracking-search-optimization-algorithm>

TABLE II  
THE RESULTS OF BENCHMARK MODELS AND THE PROPOSED TECHNIQUES

Models	Dataset A (Spring)			Dataset B (Winter)			Dataset C (Spring)			Dataset D (Winter)		
	MAE(kW)	RMSE(kW)	Time(s)	MAE(kW)	RMSE(kW)	Time(s)	MAE(kW)	RMSE(kW)	Time(s)	MAE(kW)	RMSE(kW)	Time(s)
4-PLM	10.7014	16.4032	31.8281	15.2098	21.6827	14.8472	13.9392	20.1685	16.7753	17.4767	23.7912	9.5609
5-PLM	10.1842	16.5166	77.0703	14.3098	20.9462	72.2486	13.6516	20.7080	92.3389	15.5035	22.6504	50.5541
6-PRM	10.8125	17.4423	0.0057	13.1309	21.0368	<b>0.0100</b>	13.6681	21.0748	0.0080	14.4672	22.9717	0.0060
9-PRM	9.7809	16.3948	0.0078	13.2194	20.5591	0.0106	12.3629	20.7911	0.0114	15.3235	22.8945	0.0086
ANN	10.1662	16.3317	7.9521	12.5599	20.5092	0.7279	12.3683	21.0994	0.8222	15.1149	22.7527	0.8652
SVM	9.6414	16.1533	/	12.5479	20.5546	/	12.2006	21.1840	/	14.2111	23.9912	/
SRM	9.8412	16.2275	<b>0.0038</b>	12.5062	20.6213	0.0246	12.9252	20.9527	<b>0.0045</b>	14.7076	23.5212	<b>0.0045</b>
HSRM	9.8201	16.0542	38.8948	12.6907	20.2383	40.1168	12.0962	19.8303	46.7465	<b>13.2237</b>	<b>21.3484</b>	11.8032
RSRM	<b>9.5729</b>	<b>15.9769</b>	39.6117	<b>12.0040</b>	<b>20.0050</b>	41.3014	<b>11.0042</b>	<b>19.0558</b>	47.8473	13.2897	22.1377	12.0906

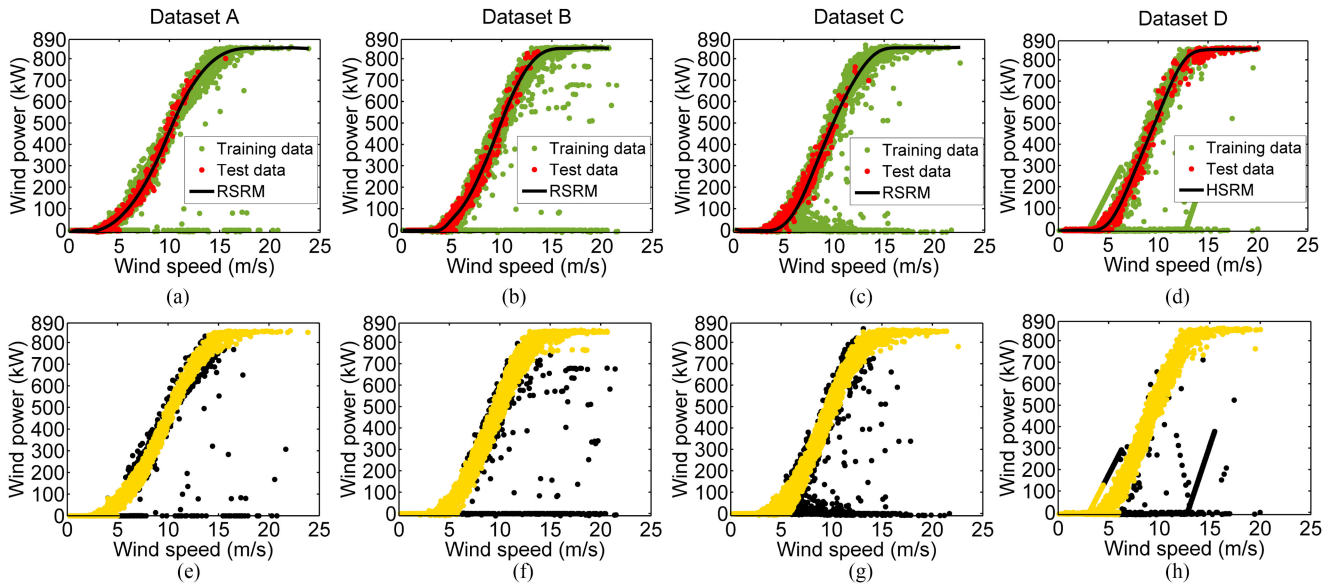


Fig. 1. The estimated power curves of the proposed models and the corresponding detected inconsistent samples. In sub-figures (e)–(h), the yellow points represent the normal samples while the black ones represent the inconsistent samples.

the best in winter. Moreover, in terms of MAE and RMSE, both HSRM and RSRM perform better than the other seven benchmark models in spring and winter, respectively.

Generally, from Table II, the performances of the two proposed models (RSRM and HSRM) are better than seven benchmark models in two seasons. Fig. 1(a)–(d) show the power curves generated by the best models on all datasets. Table III shows the optimal parameters of some models on Dataset A.

For the model to be fully trained, it is usually given a larger iteration number. Therefore, from Table II, using RSRM and HSRM will take more time to obtain the estimated power curves than the majority of benchmark models. Here, we haven't studied the effect of the iteration number on the model performance.

#### D. Inconsistent Samples Detecting

For the SDM described in Section IV-C, there are 152, 309, 931 and 2085 inconsistent samples are detected in Dataset A, B, C and D, respectively. For the proposed models, given a confidence level (95%), a confidence interval can be obtained based on the distribution of  $y^*$  in (26) and (46), the inconsistent samples will be detected when the values of wind power exceed the

computed confidence intervals. Fig. 1(e)–(h) show the detected results of the best power curve models. Table IV presents the number of the detected inconsistent samples on the training set of all datasets.

From Table IV, RSRM detects more inconsistent samples than SDM and HSRM in Dataset A, Dataset B and Dataset C. However, it is SDM that detects more inconsistent samples in Dataset D. This phenomenon shows that some normal samples are detected and removed by SDM in Dataset D. Nevertheless, the performances of the proposed models are better than the others in power curve modeling in Dataset D.

#### E. Wind Power Forecasting

Wind speed is related with wind power by a power curve. So, except the historical wind power data, wind speed data also should be considered as inputs to forecast future wind power. In reality, the real power data are sometimes far away from the theoretical power data on power curve. These inconsistent samples will degenerate the wind power forecasting accuracy. In this paper, we propose a new strategy to solve the above problem, flowchart for wind power forecasting is shown in Fig. 2. From

TABLE III  
THE ESTIMATED PARAMETERS OF POWER CURVE MODELS ON DATASET A

Models	Optimal parameters
4-PLM	$\{\alpha, \beta, \gamma, \delta\} = \{859.1575, -3.5229, -2.0718, 100.1930\}$ ;
5-PLM	$\{\alpha, \beta, \gamma, \delta, \epsilon\} = \{844.1168, -7.2802, 57.9413, 3.3437, 282.7908\}$ ;
6-PRM	$\{a_i\}_{i=0}^6 = \{-0.0144, 0.9155, -12.8034, 75.9390, -143.6777, 110.8355, -30.1304\}$ ;
9-PRM	$\{a_i\}_{i=0}^6 = \{0.0041, 0.6488, -198694, 228.4711, -1250.2500, 4133.1796, -8152.4685, 9234.5494, -5525.8426, 1352.5791\}$ ;
SVM	$\kappa = 5.2780, C = 256$ ;
HSRM	$\mu_\beta = [0.0255, -0.5618, 3.6037, 1.8169, 0.0644, 0.0101, 0.0414, -46.3312, 46.9480, 3.7663, 0.3389, 0.0695, -0.0614, -0.0240]'$ ;
RSRM	$\mu_\beta = [-0.0005, 0.6791, -12.6233, 66.5417, -70.1664, 5.6984, -0.0107, -46.6068, 47.975, -0.0518, 6.3929, -0.0101, -13.7757]'$ .

TABLE IV  
THE NUMBER OF THE DETECTED INCONSISTENT SAMPLES

Models	A	B	C	D
SDM	152	309	309	2085
HSRM	171	365	944	1461
RSRM	308	397	1102	1433

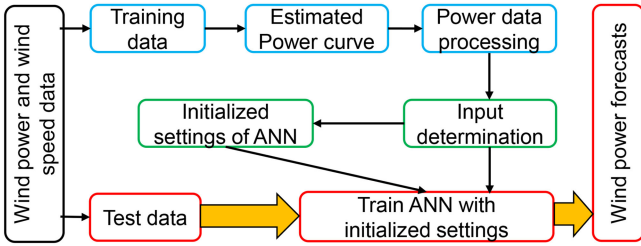


Fig. 2. Flowchart for wind power forecasting.

Fig. 2, wind power forecasts can be obtained via the following five steps.

*Step 1: Data separation.* We group the dataset into two separate parts, training set and test set. The details are described in Section IV-A.

*Step 2: Power curve modeling.* Both deterministic and probabilistic power curves are obtained by HSRM and RSRM. The process and the results are shown in Section IV-C.

*Step 3: Power data processing.* Inconsistent samples in training set can be detected by the probabilistic power curves obtained in Step 2. The detecting results are described in Section IV-D. Then, the real wind power in inconsistent samples will be replaced by the estimated wind power on the deterministic power curves obtained in Step 2.

*Step 4: ANN training.* Inputs and the structure of BP neural network play key roles in the forecasting performances. Here, both the historical wind power and wind speed are considered to make wind power forecasting. PACF (partial autocorrelation function) is employed to select the optimal historical wind power data that have the highest correlation with the target power. Historical wind speed data corresponding to the optimal historical wind power data are also considered as inputs. With the above inputs, the number of hidden neurons in BP neural network are determined by the Hecht-Nelson method [29], [30]. The other settings for a BP neural network are defaults.

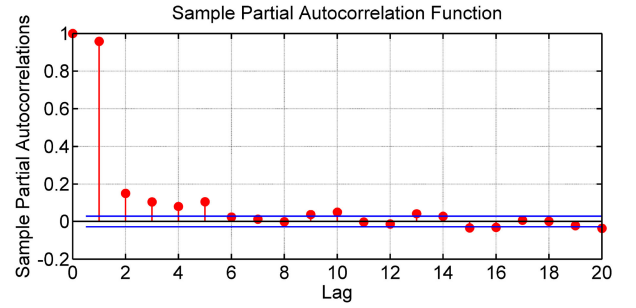


Fig. 3. Picture of PACF on the processed wind power data in Dataset A.

*Step 5: Wind power forecasting.* Wind power forecasts will be obtained by feeding the optimal inputs of test set into the trained BP neural network in Step 4.

The two proposed models, which follow the above five steps to make wind power forecasting, are named as HANN (S+P) and RANN (S+P) for easy description. HANN (S+P) uses the power curve estimated by HSRM to process the power data, while the power curve estimated by RSRM is used by RANN (S+P). Here, we take Dataset A as an example to show the details of wind speed forecasting by the proposed models. Picture of PACF for the wind power time series in the processed data in Step 3 is shown in Fig. 3.

According to Fig. 3, wind power data at time  $t-4, t-3, t-2, t-1, t$  are used to forecast wind power at time  $t+1$  and  $t+3$ , respectively. Namely, five historical wind power data are employed to realize the 10-min ahead and 30-min ahead wind power forecasting. Moreover, with the same lag as the power data, wind speed data at time  $t-4, t-3, t-2, t-1, t$  are also taken as inputs to forecast 10-min ahead and 30-min ahead wind power. So, the number of inputs is 10. The number of hidden neurons in BP neural network is  $2 \times 10 + 1 = 21$  based on the Hecht-Nelson method [29], [30]. Then, two BP neural networks with the above settings will be trained to make 10-min and 30-min ahead wind power forecasting, respectively.

In the final step, 10-min and 30-min ahead wind power forecasts are obtained by feeding the test data into the trained BP neural networks. The experimental results are presented in Table V. And Figs. 4 and 5 show the 10-min and 30-min ahead wind power forecasts of different models on Dataset A.

In Table V, comparing the performances of ANN (P) and ANN (P+S), HANN (P) and HANN (S+P), RANN (P) and RANN (S+P), we find that wind speed will be helpful to enhance the accuracy of wind power forecasting. HANN (P) and RANN (P) perform better than ANN (P). HANN (S+P) and



TABLE V  
RMSE (kW) OF DIFFERENT WIND POWER FORECASTING MODELS AND THE CORRESPONDING COMPUTATIONAL TIME (S)

Models	10-min ahead				30-min ahead				10-min ahead				30-min ahead			
	A	Time	B	Time	A	Time	B	Time	C	Time	D	Time	C	Time	D	Time
Persistence	65.7583	/	59.7222	/	97.0809	/	97.6344	/	57.1498	/	55.2758	/	88.5949	/	80.7584	/
ANN (P)	69.8397	0.8481	94.9885	0.8133	96.7525	0.7955	124.5896	2.0441	92.3139	1.6441	112.7638	0.9951	110.7524	0.9502	132.2341	0.8827
HANN (P)	65.4192	0.8601	59.5629	<b>0.7723</b>	89.8311	0.7263	95.9638	1.0181	56.2249	0.9525	54.9711	<b>0.6635</b>	82.0520	1.3744	79.3396	<b>0.6641</b>
RANN (P)	64.9271	<b>0.8185</b>	59.1738	0.7942	90.7180	<b>0.6876</b>	95.1619	<b>0.7009</b>	55.7745	<b>0.9225</b>	55.0455	0.7107	81.8475	<b>0.9002</b>	79.8959	0.7691
ANN (S+P)	66.3513	1.0154	62.7612	1.0704	92.4854	0.9671	98.4718	1.0257	58.6879	1.6999	55.8218	1.0068	83.2740	1.4890	82.2633	0.8541
HANN (S+P)	65.1493	1.0706	59.1822	0.9628	89.6216	1.0867	95.5469	1.0028	55.5367	1.5002	<b>54.3399</b>	0.8056	<b>81.5438</b>	1.5063	78.7501	0.7444
RANN (S+P)	<b>63.9624</b>	1.0880	<b>59.1267</b>	0.9596	<b>88.9369</b>	1.0840	<b>94.8358</b>	1.0480	<b>55.3352</b>	1.4026	55.0142	0.6893	81.7189	1.1337	<b>78.5600</b>	0.7588

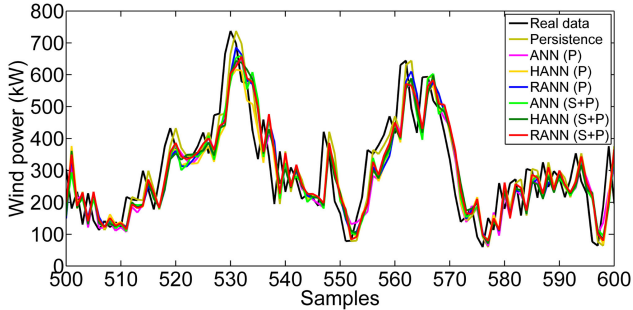


Fig. 4. 10-min ahead wind power forecasts of all models on Dataset A.

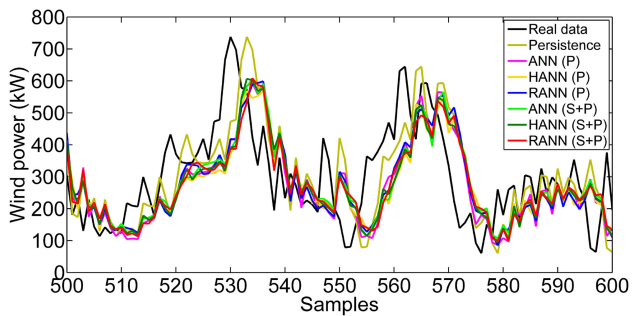


Fig. 5. 30-min ahead wind power forecasts of all models on Dataset A.

RANN (S+P) also outperform ANN (S+P). The above phenomena show that inconsistent samples will prevent us from obtaining satisfactory power forecasts, and processing those inconsistent samples by HSRM and RSRM will help to improve the forecasting accuracy.

For WTG1, in both winter and spring, RANN (S+P) always has the best forecasting results in both 10-min and 30-min ahead wind power forecasting from Table V. For WTG2, the forecasting performances vary with the seasons and the forecasting steps. In spring (Dataset C), RANN (S+P) performs better than the other models in 10-min ahead wind power forecasting while it is HANN (S+P) that performs better in winter. As to 30-min ahead wind power forecasting, HANN (S+P) is better in spring while RANN (S+P) outperforms the other models in winter.

In real applications, the computation cost of a forecasting model also should be taken into consideration. For the two proposed models, HANN (S+P) and RANN (S+P), there are two processes, power curves modeling and BP neural networks training, that take up most of the computation time. However, the total cost of the above two processes is less than

10 minutes. Therefore, considering the forecasting performance and the computational cost, two proposed models may be good choices to make wind power forecasting in real wind farms. And, the data quality can be improved by HSRM and RSRM, they are valid pre-processed methods for power data when wind speed data are available.

## V. CONCLUSION AND FUTURE WORK

In power curve modeling, inconsistent samples will prevent us from obtaining the optimal power curves. However, the current detection models cannot ensure that they can find all inconsistent samples. Thus, the estimated errors will own two properties: heteroscedasticity and long tail. In this paper, we propose two models to deal with the above problem. The first one is HSRM which assumes that the error of each sample is different. The second one is RSRM which uses an infinite mixture of Gaussians to model the long-tail error distribution. The results of power curve modeling show that HSRM and RSRM perform better than the other seven benchmark models in different seasons. But, they need more time to be trained.

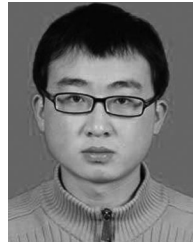
As the proposed models are optimized via variational Bayesian, except the deterministic power curves, the probabilistic power curves, which can be used to detect inconsistent samples, are obtained accordingly. Additionally, the proposed models can also be used as data processing methods for power data, which replace the wind power in the detected inconsistent samples with the wind power on the estimated power curves. Hybrid models based on an ANN and the above data processing methods are constructed to make wind power forecasting. The results show that wind power forecasting accuracy will be enhanced using the proposed data processing methods, and taking wind speed as additional inputs is also helpful to forecast future wind power.

We describe effective power curve modeling and wind power forecasting techniques if there are some inconsistent data. However, if the portion of inconsistent data is larger than 50%, the proposed techniques will generate a wrong model. In this case, we should introduce a prior knowledge to preprocess data and reduce inconsistent samples. We will discuss the challenge in the future.

## REFERENCES

- [1] T. Jin and Z. Tian, "Uncertainty analysis for wind energy production with dynamic power curves," in *Proc. IEEE 11th Int. Conf. Probabilistic Methods Appl. Power Syst.*, 2010, pp. 745–750.

- [2] Y. Wang, Q. Hu, D. Meng, and P. Zhu, "Deterministic and probabilistic wind power forecasting using a variational Bayesian-based adaptive robust multi-kernel regression model," *Appl. Energy*, vol. 208, pp. 1097–1112, 2017.
- [3] M. Marčūkaitis, I. Žutautaitė, L. Martišauskas, B. Jokšas, G. Gecevičius, and A. Sfetsos, "Non-linear regression model for wind turbine power curve," *Renewable Energy*, vol. 113, pp. 732–741, 2017.
- [4] J. Jung and R. P. Broadwater, "Current status and future advances for wind speed and power forecasting," *Renewable Sustain. Energy Rev.*, vol. 31, pp. 762–777, 2014.
- [5] R. Li and Y. Wang, "Short-term wind speed forecasting for wind farm based on empirical mode decomposition," in *Proc. Int. Conf. Electr. Mach. Syst.*, 2008, pp. 2521–2525.
- [6] C. Carrillo, A. O. Montaño, J. Cidrás, and E. Díaz-Dorado, "Review of power curve modelling for wind turbines," *Renewable Sustain. Energy Rev.*, vol. 21, pp. 572–581, 2013.
- [7] M. Lydia, S. S. Kumar, A. I. Selvakumar, and G. E. P. Kumar, "A comprehensive review on wind turbine power curve modeling techniques," *Renewable Sustain. Energy Rev.*, vol. 30, pp. 452–460, 2014.
- [8] E. Taslimi-Renani, M. Modiri-Delshad, M. F. M. Elias, and N. A. Rahim, "Development of an enhanced parametric model for wind turbine power curve," *Appl. Energy*, vol. 177, pp. 544–552, 2016.
- [9] P. Kou, F. Gao, and X. Guan, "Sparse online warped Gaussian process for wind power probabilistic forecasting," *Appl. Energy*, vol. 108, pp. 410–428, 2013.
- [10] M. Lydia, A. I. Selvakumar, S. S. Kumar, and G. E. P. Kumar, "Advanced algorithms for wind turbine power curve modeling," *IEEE Trans. Sustain. Energy*, vol. 4, no. 3, pp. 827–835, Jul. 2013.
- [11] F. Trivellato, L. Battisti, and G. Miori, "The ideal power curve of small wind turbines from field data," *J. Wind Eng. Ind. Aerodyn.*, vol. 107, pp. 263–273, 2012.
- [12] D. Villanueva and A. Feijóo, "Normal-based model for true power curves of wind turbines," *IEEE Trans. Sustain. Energy*, vol. 7, no. 3, pp. 1005–1011, Jul. 2016.
- [13] J. Gottschall and J. Peinke, "How to improve the estimation of power curves for wind turbines," *Environ. Res. Lett.*, vol. 3, no. 1, 2008, Art. no. 015005.
- [14] A. Feijóo and D. Villanueva, "Four parameter models for wind farm power curves and power probability density functions," *IEEE Trans. Sustain. Energy*, vol. 8, no. 4, pp. 1783–1784, Oct. 2017.
- [15] S. Shokrzadeh, M. J. Jozani, and E. Bibeau, "Wind turbine power curve modeling using advanced parametric and nonparametric methods," *IEEE Trans. Sustain. Energy*, vol. 5, no. 4, pp. 1262–1269, Oct. 2014.
- [16] M. Albadi and E. El-Saadany, "Wind turbines capacity factor modeling—a novel approach," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1637–1638, Aug. 2009.
- [17] M. M. Raj, M. Alexander, and M. Lydia, "Modeling of wind turbine power curve," in *Proc. IEEE PES Conf. Innovative Smart Grid Technologies-India*, 2011, pp. 144–148.
- [18] T. Ouyang, A. Kusiak, and Y. He, "Modeling wind-turbine power curve: A data partitioning and mining approach," *Renewable Energy*, vol. 102, pp. 1–8, 2017.
- [19] F. Pelletier, C. Masson, and A. Tahan, "Wind turbine power curve modelling using artificial neural network," *Renewable Energy*, vol. 89, pp. 207–214, 2016.
- [20] T. Üstüntaş and A. D. Şahin, "Wind turbine power curve estimation based on cluster center fuzzy logic modeling," *J. Wind Eng. Ind. Aerodyn.*, vol. 96, no. 5, pp. 611–620, 2008.
- [21] O. Janssens, N. Noppe, C. Devriendt, R. Van de Walle, and S. Van Hoecke, "Data-driven multivariate power curve modeling of offshore wind turbines," *Eng. Appl. Artif. Intell.*, vol. 55, pp. 331–338, 2016.
- [22] Y. Zhao, L. Ye, W. Wang, H. Sun, Y. Ju, and Y. Tang, "Data-driven correction approach to refine power curve of wind farm under wind curtailment," *IEEE Trans. Sustain. Energy*, vol. 9, no. 1, pp. 95–105, Jan. 2018.
- [23] A. Kusiak, H. Zheng, and Z. Song, "Models for monitoring wind farm power," *Renewable Energy*, vol. 34, no. 3, pp. 583–590, 2009.
- [24] M. K. Titsias and M. Lázaro-Gredilla, "Variational heteroscedastic gaussian process regression," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 841–848.
- [25] F. Yu and X. Xu, "A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network," *Appl. Energy*, vol. 134, pp. 102–113, 2014.
- [26] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer Series in Statistics), 2nd ed. New York, NY, USA: Springer, 2013.
- [27] X. Yang and S. Yin, "Variational Bayesian inference for FIR models with randomly missing measurements," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4217–4225, May 2017.
- [28] Y. Chen, J. Yang, C. Wang, and N. Liu, "Multimodal biometrics recognition based on local fusion visual features and variational Bayesian extreme learning machine," *Expert Syst. Appl.*, vol. 64, no. C, pp. 93–103, 2016.
- [29] R. Hecht-Nielsen, "Kolmogorovs mapping neural network existence theorem," in *Proc. Int. Conf. Neural Netw.*, 1987, pp. 11–13.
- [30] C. Ren, N. An, J. Wang, L. Li, B. Hu, and D. Shang, "Optimal parameters selection for bp neural network based on particle swarm optimization: A case study of wind speed forecasting," *Knowl.-Based Syst.*, vol. 56, pp. 226–239, 2014.



**Yun Wang** received the B.S. degree from the School of Economics, Anhui University, Hefei, China, 2012, and the M.S. degree from the School of Mathematics and Statistics, Lanzhou University, Lanzhou, China, in 2012. He is currently working toward the Ph.D. degree at the School of Computer Science and Technology, Tianjin University, Tianjin, China. His interests include wind speed/wind power forecasting, robust regression modeling, multikernel learning, functional data analysis, and Bayesian inference.



**Qinghua Hu** received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively. He was a Postdoctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. He is currently a Full Professor and the Vice Dean of the School of Computer Science and Technology, Tianjin University, Tianjin, China. He has authored more than 100 journal and conference papers in the areas of granular computing based machine learning, reasoning with uncertainty, pattern recognition, and fault diagnosis. His current research interests include rough sets, granular computing, and data mining for classification and regression. He was the Program Committee Co-Chair of the International Conference on Rough Sets and Current Trends in Computing in 2010, the Chinese Rough Set and Soft Computing Society in 2012 and 2014, and the International Conference on Rough Sets and Knowledge Technology, the International Conference on Machine Learning and Cybernetics in 2014, and the General Co-Chair of IJCRS 2015. He is currently PC-Co Chairs of CCML 2017 and CCCV 2017.



**Dipti Srinivasan** (M'89–SM'02) received the Ph.D. degree in engineering from the National University of Singapore, Singapore, in 1994. She was a Postdoctoral Researcher with the University of California, Berkeley, Berkeley, CA, USA, from 1994 to 1995, before joining the National University of Singapore, where she is currently a Professor with the Department of Electrical and Computer Engineering. Her research interest include the application of soft computing techniques to engineering optimization and control problems.



**Zheng Wang** is an Engineer with the China Electric Power Research Institute, Beijing, China. His research interests mainly include renewable generation forecasting and its uncertainty estimation.