Generative Wind Power Curve Modeling Via Machine Vision: A Self-learning Deep Convolutional Network Based Method

Luoxiao Yang, Long Wang, *Member, IEEE*, and Zijun Zhang, *Senior Member, IEEE*

***Abstract*—This paper develops a novel self-training U-net (STU-net) based method for the automated WPC model generation without requiring data pre-processing. The self-training (ST) process of STU-net has two steps. First, different from traditional studies regarding the WPC modeling as a curve fitting problem, in this paper, we renovate the WPC modeling formulation from a machine vision aspect. To develop sufficiently diversified training samples, we synthesize supervisory control and data acquisition (SCADA) data based on a set of S-shape functions depicting WPCs. These synthesized SCADA data and WPC functions are visualized as images and paired as training samples (I_x, I_{wpc}). A U-net is then developed to approximate the model recovering I_{wpc} from I_x. The developed U-net is applied into observed SCADA data and can successfully generate the I_{wpc}. Moreover, we develop a pixel mapping and correction process to derive a mathematical form f_{wpc} representing I_{wpc} generated previously. The proposed STU-net only needs to train once and does not require any data preprocessing in applications. Numerical experiments based on 76 WTs are conducted to validate the superiority of the proposed method by benchmarking against classical WPC modeling methods. To demonstrate the repeatability of the presented research, we release our code at https://github.com/IkeYang/STU-net.**

***Index Terms*—wind power curve, wind turbine, neural networks, wind energy, data-driven model**

I. Introduction

To realize carbon neutrality by the mid-page of this century, wind energy will contribute more to the future energy portfolio. Meanwhile, as the wind industry has experienced a rapid growth over past decades, a significant proportion of wind power assets installed worldwide are aging, which poses emerging demand of advanced technologies on operations and maintenance (O&M) management side. Therefore, emerging technologies which benefit wind farm O&M have become a new sweet spot in wind energy domain. A wind power curve (WPC) directly pictures the wind speed and power output relationship and is useful to help address many issues in the wind farm operations and power system operations, such as the wind power prediction [1, 2], load flow estimation [3], electricity market [4, 5] and wind energy assessment [6, 7]. Although WT manufacturers usually offer an ideal WPC of their WT models, it is widely known that real WPCs present quite different patterns due to numerous factors, such as the environmental changes, system degradations, control errors, etc.. Therefore, effective methods for modeling the real wind power curve via wind farm SCADA data are highly interesting and beneficial to both the wind industry and power grid.

In the literature, the wind power curve modeling (WPCM) based on wind turbine SCADA data has been vigorously discussed and various methods have been developed. Existing WPCM methods can be generally categorized into parametric models and non-parametric models [8]. The parametric model aims at approximating WPCs via explicit mathematical forms, such as the polynomial form, logistic function, and probabilistic model [9]. The linearized segmented model and the polynomial model were employed to fit the nonlinear part of WPC in [10] and [11] respectively. As several functions naturally have a similar S-shape curvature as the WPC, they have been applied to study the WPCM. In [12], the double exponential (DE) function and the adjusted double exponential function (ADE) were applied to perform WPCM. In [13] and [14], the four-parameter logistic function (4PLF) and the five-parameter logistic function (5PLF) were also applied into WPCM. In [15], a modified hyperbolic tangent model was introduced to characterize WPC. In [16], the author proposed a Gaussian cumulative distribution function (CDF) based model for studying WPCM. The probabilistic model studying WPCM treats all parameters of a model as random variables and employs the Bayesian inference to derive the corresponding posteriors. In [17], two statistical models, the heteroscedastic spline regression model (HSRM) and robust spline regression model (RSRM), were proposed for accurate WPCM. To allow a higher flexibility of modeling various of WPC shapes based on real SCADA data and capture the nonlinearity, non-parametric models, mainly ML based models, have been developed and applied. In [18], the spline regression (SR) method was introduced to conduct WPCM. In [19], the k nearest neighbor (KNN) was applied to accurately model the WPC in WT performance monitoring. In [20] and [21], the support vector regression (SVR) and shallow neural network (SNN) were utilized to perform WPCM respectively. In [22], four data-driven models for WPCM are compared and results

showed that the adaptive neuro-fuzzy-interference system (ANFIS) model performed best.

Though existing WPCM methods have well proved the feasibility and effectiveness of modeling the true WPC from a data-driven perspective, two pain points still wait for further solutions. First, due to the quality of controllers, wind power curtailment, environmental changes, etc. [23], collected WT SCADA data might include noisy points which will significantly affect the accuracy of developing the WPCM. Thus, existing data-driven methods for developing WPCMs usually require a detailed SCADA data preprocessing. Applied SCADA data preprocessing methods include the empirical approaches [17, 22], statistics-based models [24] and learning-based models [23, 25, 26]. As data pre-processing methods either require strong domain expertise or involve the complicated tuning of hyperparameters, such practice is a time-consuming trial-and-error process and is usually ad-hoc. Moreover, most of the existing WPCM methods are dataset-specific, which means that the built model is usually more effective for a specific WT during a specific period while applying such model into another WT or another time period might require a model renewal and even lead to a degraded performance.

To develop a more general and adaptive paradigm for accurate WPCM free of any data pre-processing based on various WT datasets, in this study, we re-visit the WPCM problem from the image processing perspective, propose a new WPCM problem formulation considering image data derived from SCADA data, and develop an applicable image based generative WPCM method, a Self-training U-net (STU-net). The proposed STU-net includes two components, the self-training (ST) strategy as well as the pixel mapping and correction. The ST strategy is of two-fold, the real WPC image synthesis and WPC generative model development. To synthesize a real WPC image, we first regard a set of considered "S-shape" functions, f_{wpc} , as the WPC ground truths and next randomly distribute data points around the curve of f_{wpc} to simulate the WPC image derived from SCADA data (SCADA WPC), I_{wpc} . The synthesized SCADA WPC images and the ground truth WPC, f_{wpc}, are paired as training samples (I_x, I_{wpc}). A deep convolutional neural network, U-net [27], is then adapted to develop a U-net based model for extracting I_{wpc} from I_x based on synthesized training samples. In the implementation, the developed U-net based model is fed with the WPC images derived from real SCADA data and produce the corresponding I_{wpc} with both a graphical visualization and a mathematical expression. To derive the mathematical form of the I_{wpc} based on the mapping relationship between pixel and SCADA data points, we employ a polynomial fitting and correct it by the domain knowledge. To verify the effectiveness of our proposed STU-net based WPCM method, four parametric models, DE, ADE, 4PLF and 5PLF, and two ML models, SNN and SR, are considered as benchmarks of the WPCM task. A comprehensive computational study has been conducted to demonstrate the effectiveness of the proposed STU-net based WPCM method based on SCADA data collected from 76 commercial WTs under three application scenarios, all methods applied with raw observed SCADA data, with minor pre-processed observed SCADA data and with carefully pre-processed observed SCADA data, as well as two patterns, normal pattern (NP) and insufficient data pattern (IDP). Computational results show that our proposed ST-U-net method, without any retain, can achieve a comparable accuracy using raw data to the best benchmark using carefully pre-processed data and offer more accurate results than SNN with insufficient data.

The main contributions of this work are summarized below:

1) Novel problem formulation: A novel formulation of WPCM problem is presented, which integrates the machine vision into the inverse data synthesis function.
2) Methodology framework: A novel methodology framework is presented to produce the neat WPC as well as WPC model expression from SCADA WPC image based on ST.
3) Model generalization: The proposed STU-net can generalize to different SCADA WPC data without retaining.
4) Model performance: The proposed STU-net can offer comparable accuracy using raw data to the best benchmark using carefully pre-processed data and produce more accurate result in IDP.

II. WIND POWER CURVE MODELING PROBLEM

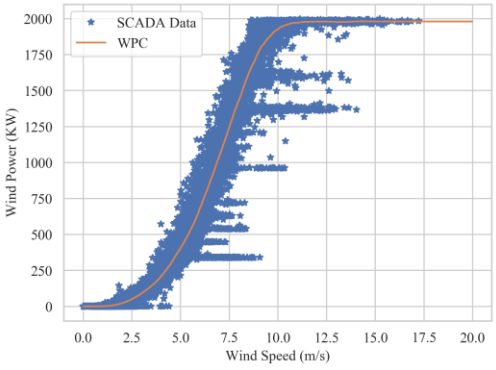


Fig. 1. A visualization of a neat WPC and the real WPC formed by SCADA data.

WPC is commonly regarded as a function describing the relationship between the wind speed and the generated wind power. Typically, the WPC adopts a piece-wise form as described in (1):

$$f_{wpc} = \begin{cases} 0, & x < x_{CutIn} \\ f(x), & x_{CutIn} \le x < x_{Rated} \\ P_{Rated}, & x_{Rated} \le x \end{cases} \quad (1)$$

where the $f(x)$ satisfies two boundary conditions as expressed in (2):

$$\begin{aligned} f(x_{CutIn}) &= 0 \qquad \frac{df(x)}{dx}|_{x=x_{CutIn}} = 0 \\ f(x_{Rated}) &= P_{Rated} \quad \frac{df(x)}{dx}|_{x=x_{Rated}} = 0 \end{aligned} \tag{2}$$

Fig. 1 visualizes a neat WPC in our expectation and the real WPC formed by WT SCADA data. It is observable that, due to various factors, such as the measurement errors, WT faults, wind curtailment, extreme weather conditions, etc., the raw WT data offers a rough curve formed by a swarm of data points. A common practice of modeling such a WPC via using SCADA data in the literature is to formulate the WPCM as a curve fitting/regression problem. Next, we will brief the traditional WPCM, discuss its limitations, and introduce the machine vision assisted WPCM formulation proposed in this work.

A. Traditional wind power curve modeling

In the literature, the WPCM problem has been widely considered as a curve fitting problem (parametric model) or a regression problem (ML model). From this aspect, the observed SCADA data mainly serve as the input information for optimizing the parameters of a "S-shape" function or a ML model as described in (3):

$$p = \underset{p}{\operatorname{argmax}}\, \mathcal{L}\big(f_p(x), y\big) \tag{3}$$

where f_p denotes a model for depicting a WPC and p denotes its parameters or weights, $\mathcal{L}$ denotes the loss function, which is usually a L_2 norm, as well as (x, y) denotes pairs of the collected wind speed and wind power output in SCADA data. Based on such problem formulation, vigorous discussions have been made on various model forms and machine learning techniques for obtaining WPCs with a higher quality [8, 9]. However, "noises" contained in the swarm of SCADA data can greatly affect the shape of curves obtained via fitting models into (x, y) pairs, and thus a careful data preprocessing is usually required. In addition, the extracted WPC model f_p is specific to the target WT only or maybe WTs sharing the same SCADA data distribution if exist. In other words, a WPC model developed via such traditional WPCM formulation is specific to data of the targeted WT and its form might no longer be effective to describe the WPC of another WT. To develop WPC models of a population of WTs using SCADA data via the traditional practice, it requires the turbine-wise repeated effort on data pre-processing and model fitting. It is interesting to revisit the WPCM problem and investigate the feasibility of developing a novel WPC modeling method which can be generalized to various WT SCADA data.

B. A New Machine Vision Assisted Wind Power Curve Modeling Problem

We propose a new formulation of the WPCM problem, which first analyses raw WT SCADA data from the machine vision perspective and next returns a parametric form accurately reflecting the SCADA WPC. As the ground truth of a neat WPC representing the SCADA WPC is unavailable, we study our proposed WPCM method based on synthesized data to allow a rigorous evaluation and further verify the method based on real SCADA data. To synthesize a WPC visually like a SCADA WPC, we first identify a parametric form offering a "S-shape" curvature and randomly adding data points around such curve as described in (4):

$$x, y = f_\Lambda(f_{WPC}) \tag{4}$$

where the f_{WPC} denotes the ground truth function picturing a neat WPC, f_Λ denotes the described data synthesis function, and (x, y) denotes pairs of the synthesized wind speed and wind power. We are able to synthesize a large volume of WPC images based on different pre-determined f_{WPC} and the data synthesis process. Based on synthesized WPC images, our task is to obtain a data-driven model representing an inverse function of f_Λ, which can derive the ground truth WPC function based on WPC images and the synthesized SCADA data as described in (5) as well as can be generalized to the observed real SCADA data:

$$f_{WPC} = f_\Lambda^{-1}(x, y) \tag{5}$$

Based on (5), the objective function of the considered data-driven modeling is then described in (6):

$$w = \underset{w}{\operatorname{argmax}}\, \mathcal{L}\big(f_{\Lambda_w}^{-1}(x, y), f_{WPC}\big) \tag{6}$$

where $f_{\Lambda_w}^{-1}$ denotes the inverse function with the weight of w and $\mathcal{L}$ denotes the loss function. The (6) directly deals with values of (x, y) while the local spatial relationship among (x, y) pairs in a graphical visualization is not well incorporated. Let I_x denote the image visualizing all pairs of (x, y) and I_{wpc} denote the visualization of f_{WPC} respectively. We are able to extend the objective in (6) to a machine vision version as described in (7):

$$w = \underset{w}{\operatorname{argmax}}\, \mathcal{L}\big(f_{\Lambda_w}^{-1}(I_x), I_{wpc}\big) \tag{7}$$

We hereafter need to develop an effective method to address the modeling task stated in (7), learning a non-parametric form of $f_{\Lambda_w}^{-1}$ that can extract I_{wpc} from I_x. Given a sufficiently large volume of training samples, we expect that the developed $f_{\Lambda_w}^{-1}$ can be applied into WPCM considering any SCADA data and can return high quality neat WPCs without any data pre-processing and model retraining.

III. The Self-training U-net Based Wind Power Curve Modeling Method

To address the proposed WPCM formulation, we develop a STU-net based methodological framework and describe its

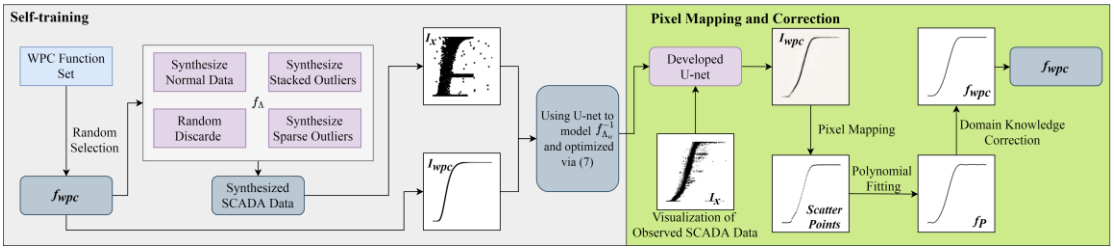


Fig. 2. The schematic diagram of the ST-U-net.

$$S_{WPC} = S_{DE} \cup S_{ADE}$$
$$S_{DE} = \{f(\cdot)|f(x) = e^{-t_1 e^{t_2 x}}, t_1 \sim U[10,50], t_2 \sim U[-15,-8]\}$$
$$S_{ADE} = \left\{f(\cdot)\middle|f(x) = e^{-e^{a_0 - a_1 x - a_2 x^2 - a_3 x^3}}, a_0 = 5, a_1 \sim U[a_2, b_2], a_2 \sim U[-15,10], a_3 = 15\right\} \quad (8)$$

principle next. Please note that the U-net is chosen as a base model in this study because it is a classical convolutional autoencoder, which offers a symmetric architecture for inverse engineering and has widely demonstrated its successes in medical imaging analytics, especially in feature engineering. Theoretically, any generative deep network capable to learn the inverse form of a convolutional process can fit into the proposed framework. Fig. 2 provides a schematic diagram of the proposed STU-net based framework, which contains two phases, the self-training phase and the pixel mapping and correction phase. In the self-training phase, based on the setup of the proposed WPC formulation in Section II, a WPC function generator randomly selects a WPC function f_{WPC} from a pre-determined function set. Considering various types of SCADA data swarms displayed in SCADA WPC as mentioned in [25], a f_Λ is designed and output the synthesized SCADA data points based on a selected f_{WPC}. The synthesized SCADA data and selected f_{WPC} are then visualized as image pairs and noted as I_x and I_{wpc} respectively. Due to the powerful feature engineering capability of the U-net, it is applied to model $f_{\Lambda_w}^{-1}$ and optimized via considering (7) as the loss function. In the pixel mapping and correction phase, the observed SCADA data are first applied to produce an image of the SCADA WPC, I_x, which is then fed into the developed U-net to generate a neat WPC, I_{wpc}. Based on the mapping relationship between the pixel and the real-world SCADA data points as well as the domain knowledge, the I_{wpc} is finally converted into a parametric representation, f_{wpc}.

We next sequentially introduce details in the ST strategy, in the pixel mapping and correction, as well as in the training and testing scheme of the proposed STU-net.

A. *Self-training strategy*

To develop a data-driven model approximating $f_{\Lambda_w}^{-1}$ via (7), training pairs, I_x and I_{wpc}, are needed. Yet, we need to address two challenges relating to data. On one hand, it is difficult to obtain a sufficient diversity of pairs of I_x and I_{wpc} via real SCADA dataset. On the other hand, the ground truth of the neat WPC I_{wpc} based on a SCADA WPC is unavailable. Thus, in this research, we develop a self-training data synthesis process, f_Λ, to generate sufficient pairs of training samples via simulating the SCADA WPC and then utilize synthesized data to develop a U-net based model serving as the $f_{\Lambda_w}^{-1}$.

1) *Self-training data synthesis*

A ST data synthesis process, f_Λ, is developed to simulate a SCADA WPC via randomly adding swarms of data points around a "S-shape" function representing a neat WPC. Based on the developed ST data synthesis process, pairs of WPC images I_{wpc} and the simulated SCADA data images I_x are generated. For simplicity, in this research, we choose the double exponential (DE) function and the adjusted double exponential (ADE) function [12] as illustrative examples and assume that they are two functions describing ground truths of neat WPCs. Thus, the DE and ADE form a WPC function set S_{WPC} as expressed in (8), where x denotes the normalized wind speed and U denotes a uniform distribution. In (8), the upper and lower bound settings of each parameter are given based on our preliminary trials, which achieve a tradeoff between the stability and diversity of generated f_{WPC}.

In [25], the observed SCADA data are categorized into three patterns, the normal data, the stacked outliers, and the sparse outliers. Based on the generated f_{WPC}, three patterns of SCADA data are sequentially simulated via (9):

$$Normal\ Data = (x, \epsilon * \phi\left(derivative\left(f_{WPC}(x)\right)\right) + f_{WPC}(x))$$
$$Stacked\ Outliers = (x, \epsilon * std + f_{WPC}(x))$$
$$Sparse\ Outliers = (a, b)$$
$$derivative = \frac{df_{WPC}(x)}{dx}$$
$$\epsilon \sim N(0,1), a, b \sim U[0,1] \quad (9)$$

where ϕ denotes the variance projection (VP) function. Here, we assume that the variance of $Normal\ Data$ is proportional to the first-order derivative of the $f_{WPC}(x)$. However, directly utilizing the derivative of $f_{WPC}(x)$ could result in a scenario that normal data are unavailable. Thus, an empirical design of the ϕ is proposed in (10) to avoid this scenario:

$$\phi(d) = \begin{cases} \eta(d), & Normaliz(d) < 0.7 \\ (1 - (\eta(d) - 1)^4)^{0.5}, & Normaliz(d) \geq 0.7 \end{cases} \quad (10)$$

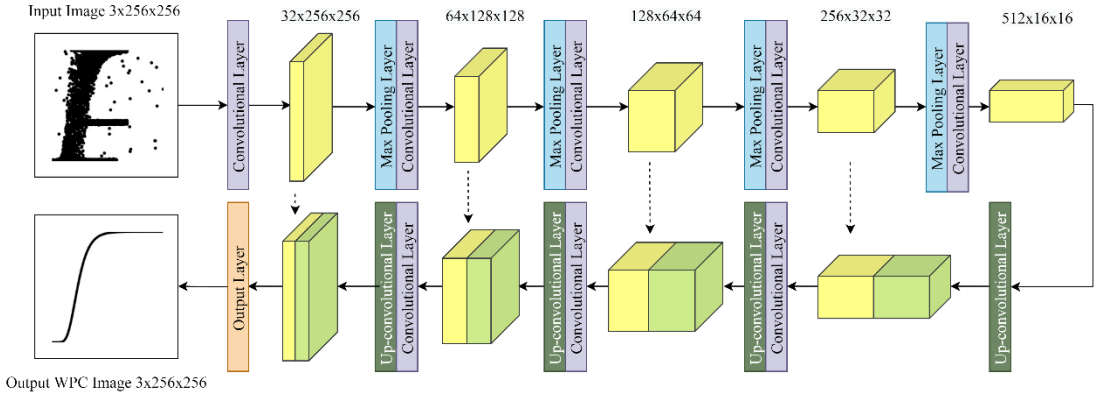


Fig. 3 The architecture of the U-net utilized in self-training

Algorithm 1: Self-training data synthesis

1: Input: The number of total synthesized samples n, the number of three types SCADA data, n_{normal}, $n_{stacked}$, n_{sparse}.
2: **for** i=1 to n
3: Generated $f_{WPC}(\cdot)$ from S_{WPC}.
4: $SSD = [\,]$
5: Synthesize n_{normal} normal data d_{norm} based on (9).
6: $SSD = [SSD, d_{norm}]$
7: Synthesize $n_{stacked}$ stacked outliers $d_{stacked}$ based on (9).
8: $SSD = [SSD, d_{stacked}]$
9: Synthesize n_{sparse} sparse outliers d_{sparse} based on (9).
10: $SSD = [SSD, d_{sparse}]$
11: Randomly discard SSD.
12: Display $f_{WPC}(\cdot)$ as I_{wpc} and SSD as I_x. Save (I_x, I_{wpc}).
13: **end for**
14: **Output**: A dataset with n training pairs.

where $\eta(\cdot)$ denotes a minmax normalization. The finally synthesized SCADA WPC is a concatenation of previously mentioned three patterns of data. To simulate scenarios of WPCs containing insufficient data, the synthesized SCADA data points exceeding a randomly set wind power or wind speed level will be discarded. Finally, the synthesized SCADA data points and the f_{WPC} are displayed graphically and saved as images for model training. The Pseudo code of the proposed ST data synthesis process is offered in **Algorithm 1**, where *SSD* is short for Synthesized SCADA Data.

2) A U-net Based Model Development

A U-net is a DNN adopting the autoencoder architecture and is firstly proposed for the cell detection and segmentation in [27]. Fig. 3 presents the architecture of the U-net utilized in this paper, where the number on the top specifies the configuration of feature maps and the dash line denotes a copy and concatenate operation. It is observable that the considered U-net is composed of four basic layers, the convolutional layer, the max pooling layer, the up-convolutional layer, and the output layer. Each convolutional layer repeats three successive operations, the 3×3 convolutions, batch normalization [28] and ReLU activation, twice. The max pooling layer utilizes a 2×2 pooling operation for down sampling and the up-convolutional layer utilizes a 2×2 transpose convolutional operation for up sampling. The final output layer employs a single convolutional

Algorithm 2: Self-training

1: Input: The hyperparameter controlling self-training data synthesis (n, n_{normal}, $n_{stacked}$, n_{sparse}), the number of U-net training iteration n_{iter}
2: Synthesize n training pairs (I_x, I_{wpc}) via **Algorithm 1**.
3: **for** epoch i=1 to n_{iter}
4: $loss_{Unet} = \left\| I_{wpc} - Unet(I_x) \right\|_2^2$
5: Update U-net via a gradient based optimizer
6: **end for**
7: **Output**: Developed U-net

operation aiming to map the feature map to the RGB image space. The encoder part of the U-net conducts four times of down sampling operations to reduce the feature size from 256×256 to 16×16. Accordingly, the decoder part conducts four times of up sampling operations to restore the original image resolution from low-dimensional high-level semantic features output by the encoder. In addition, the skip connection is utilized between the same stage of the encoder and decoder, which ensures that the final feature map could fuse both low-level features and high-level features.

The mean square error (MSE) loss described in (11) is considered as the $\mathcal{L}$ in (7):

$$\mathcal{L} = \left\| I_{wpc} - Unet(I_x) \right\|_2^2 \tag{11}$$

The pseudo code of the whole self-training process is offered in **Algorithm 2**.

B. Pixel mapping and correction

After the ST, the developed U-net is capable to perform online WPCM based on real SCADA data. The observed SCADA data of wind speed and wind power output are first applied to produce images I_x, a SCADA WPC, and then sent to the developed U-net to obtain the output I_{wpc}, a neat WPC. To derive f_{wpc} from I_{wpc}, in this section, we present a pixel mapping and correction process, which contains three steps, the pixel mapping, polynomial fitting, and domain knowledge correction (DKC). Before elaborating the proposed pixel mapping and correction process, we define the coordinate of the pixel in an image as the pixel coordinate, which is composed of two integers representing the relative position of a pixel in an image, and we define the coordinate of normalized SCADA data as the SCADA coordinate, which is composed of two floats

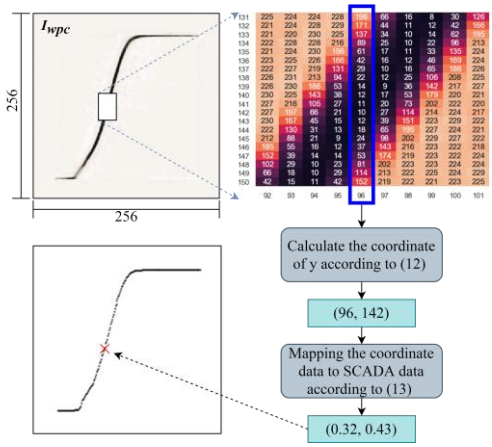


Fig. 4. The pixel mapping process.

ranging from 0 to 1 representing the normalized wind speed and the corresponding normalized wind power respectively.

Fig. 4 illustrates the pixel mapping process. It is observable that the output I_{wpc} is first converted into a grey scale image with a size of 256×256. The feasible pixel coordinate of x is ranged from 32 to 229 and y is ranged from 31 to 227. We note $227 - 31 + 1 = 197$ as the validated resolution R_s of I_{wpc}. The inconsistency of the range between x and y is caused by the plot package utilized in this paper. Thus, there are $229 - 32 + 1 = 198$ feasible coordinate pairs to be decided. In this research, the coordinate of y is decided based on the minimum value of the pixel in one column as expressed in (12):

$$y = \text{mean}\left(\underset{i}{\text{argmin}}\, I_{wpc}\,[x, i]\right) \tag{12}$$

To map the pixel coordinate into SCADA data coordinate, the coordinate pairs are converted via (13):

$$\begin{aligned} x &= \frac{x - 32}{229 - 32} \\ y &= 1 - \frac{y - 31}{227 - 31} \end{aligned} \tag{13}$$

After pixel mapping, there are 198 normalized SCADA data points, which are utilized to fit an n-order polynomial as described in (14):

$$\begin{aligned} &\min \sum_{i=1}^{198} \left(f_p(x_i) - y_i\right)^2 + \lambda \|\boldsymbol{a}\|_2^2 \\ &s.t. \qquad f_p(x) = \sum_{i=0}^{n} a_i x^i \end{aligned} \tag{14}$$

where the λ is the penalty term.

However, due to the high flexibility of the polynomial in fitting a curve and the error in the pixel mapping process, the fitted f_p could conflict with the domain knowledge in terms of the cut-in and rated wind speed as shown in Fig. 5. Thus, in this research, we propose a DKC process to ensure that the output f_{wpc} meets constraints mentioned in (2).

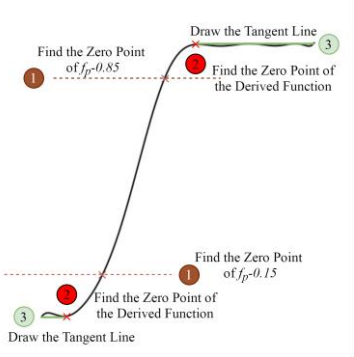


Fig.5 The domain knowledge correction process.

The main purpose of the DKC is to determine the cut-in wind speed (CWS) x_{CutIn} and the rated wind speed (RWS) x_{Rated} in the fitted curve, which are the zero points of the first-order derivative of f_p. Considering that it is convenient to drive the explicit expression of a polynomial derivative, in this paper, the Newton-Raphson method (NRM) is utilized to decide CWS and RWS. Fig. 5 presents the details of the DKC, which can be summarized into three steps. First, a bisection is utilized to decide an appropriate initial point of the NRM for calculating CWS and RWS respectively as described in (15):

$$x_{init\_c} = Bisection\left(f_p - c\right) \tag{15}$$

where c is set to 0.85 for RWS and is set to 0.15 for CWS.

Next, based on x_{init}, the corresponding RWS x_{CutIn} and CWS x_{Rated} are calculated via (16):

$$x_{CutIn\backslash Rated} = NRM\left(f_p', x_{init\_0.85\backslash 0.15}\right) \tag{16}$$

where f_p' is the first-order derivative of f_p.

Finally, a tangent line is drawn to meet the first-order continuity at point of tangency. The output f_{wpc} is then expressed in (17):

$$f_{wpc} = \begin{cases} f_p(x_{CutIn}), & x < x_{CutIn} \\ f_p(x), & x_{CutIn} \le x < x_{Rated} \\ f_p(x_{Rated}), & x_{Rated} \le x \end{cases} \tag{17}$$

Assumption 1. Assume that the order of the $f_p(x)$ is sufficient to model the curvature of SCADA data displayed in a two dimensional normalized wind speed and wind power scatter diagram, and (18) holds.

$$0 \le f_p(x) \le 1, \qquad f_p(x_{CutIn}) = min\, f_p(x), \qquad x \in [0,1] \tag{18}$$

Proposition 1. The f_{wpc} defined in (17) could meet the constrain mentioned in (2) when the validated resolution $R_s \in N^*$ approaches infinity.

Proof: Considering the $x_{CutIn\backslash Rated}$ is the zero points of the f_p', it is obvious that the derivative constraints in (2) as well as $f_{wpc}(x_{Rated}) = f_p(x_{Rated})$ are satisfied.

Note the width of the plotted line in the training set is l. Then, the y axis coordinate of SCADA data is ranged between $[0, \frac{l-1}{R_s}]$ when $x <$ CWS. Based on ***Assumption 1*** that $f_p(x)$ with an order is sufficient to model the curvature of SCADA data, we have (19):

TABLE I
DESCRIPTION OF DATASETS UTILIZED IN THIS STUDY

Dataset	Location	Number of WTs	Collected Period	No. of Data Points of each WT
QH1	Qinghai	8	Mar. 5th, 2016 - May 5th, 2016	153388
QH2	Qinghai	8	Nov. 5th, 2016 -Jan. 5th, 2017	87840
SX	Shaanxi	16	Jan. 1th, 2015 -Aug. 12th, 2014	31267
NX	Ningxia	30	Feb. 1th, 2014 - Jun. 27th, 2014	20433
LN	Liaoning	10	Apr. 1th, 2015 - Jun. 10, 2015	9895
HB	Hebei	4	Apr. 1th, 2015 - Jun. 6, 2015	9499

*The sampling interval of QH1 and QH2 is 1 min and that of other datasets is 10 min

Algorithm 3: The training and testing scheme of the proposed STU-net

1: **Training**
2: Based on Algorithm 2 train U-net.
3: **Testing**
4: Visualize the observed SCADA data noted as I_x.
5: Output $I_{wpc} = Unet\ (I_x)$.
6: Based on (12-13), calculate scatter points (x, y).
7: Fit a polynomial f_p based on (14).
8: Output f_{wpc} based on (15-17).

$$\min_{x<x_{CutIn}} f_p(x) \le \frac{l-1}{R_s} \tag{19}$$

If (19) does not hold, we can always adjust the order and the penalty term in (14) to ensure that (19) holds. Then, we can conclude that $0 \le f_p(x_{CutIn}) \le \min_{x<x_{CutIn}} f_p(x) \le \frac{l-1}{R_s}$ and (20) holds.

$$\lim_{R_s \to +\infty} f_p(x_{CutIn}) = 0 \tag{20}$$

This establishes ***Proposition* 1**.

The pseudo code of the overall training and testing scheme of the proposed STU-net based WPCM is offered in **Algorithm 3.**

IV. Computational Experiment

A. Dataset

In this research, six SCADA datasets collected from commercial wind farms in Mainland China are utilized to study the WPCM problem. The usage of collected SCADA data is of two-fold. First, we analyze patterns of collected SCADA data to achieve a high-quality synthesis of real WPCs. In addition, we verify the proposed method and considered benchmarks on the WPCM task using collected SCADA data. Details of the considered datasets are reported in Table I.

B. Benchmarking Models and Model Parameter Setup

In this research, six classical benchmarks, DE, ADE, 4PLF, 5PLF, SNN, and SR, are considered to verify the advantage of the proposed STU-net. The backtracking search algorithm (BSA) [29] is employed to optimize parameters of the parametric model including DE, ADE, 4PLF, and 5PLF as recommended in [17]. The number of hidden neurons of SNN is set to 50 based on a preliminary trial.

In the ST process, the number of total synthesized SCADA data n is set to 4000 with n_{normal}, $n_{stacked}$, and n_{sparse} are set to 1000, 150, and 250, respectively. To eliminate the inconsistency between the total number of data points in synthesized SCADA data and that of observed SCADA data, the marker size (MS) of I_x of the observed SCADA data should be proportional to the MS of I_x of the synthesized SCADA data as described in (21)

$$K * MS_{ST}^2 * 1400 = MS_{Test}^2 * n_{data} \tag{21}$$

where n_{data} is the number of data points in the observed SCADA data, $MS_{ST}^2 = 6$, as well as K is the proportionality coefficient and is set to 0.07 based on the grid search. Then, the detailed visualization settings are reported in Table II.

TABLE II
VISUALIZATION SETTINGS.

Setting	ST	Testing
Figure size	256*256*3	256*256*3
Marker Size	6	$\sqrt{K * 1400 * 6^2 / n_{data}}$
Color	Black	Black

C. Three application scenarios and two patterns

In this research, we consider three application scenarios to verify the proposed method and benchmarks.

Scenario I. Real time WPCM: Scenario I (S1) targets on online WPCM, in which developed WPC models directly consider observed SCADA data without any pre-processing.

Scenario II. Low latency WPCM: Scenario II (S2) targets on low latency WPCM, in which developed WPC models consider slightly pre-processed observed SCADA data.

Scenario III. Offline WPCM: Scenario III (S3) targets on offline WPCM, in which developed WPC models consider SCADA data carefully pre-processed by experts.

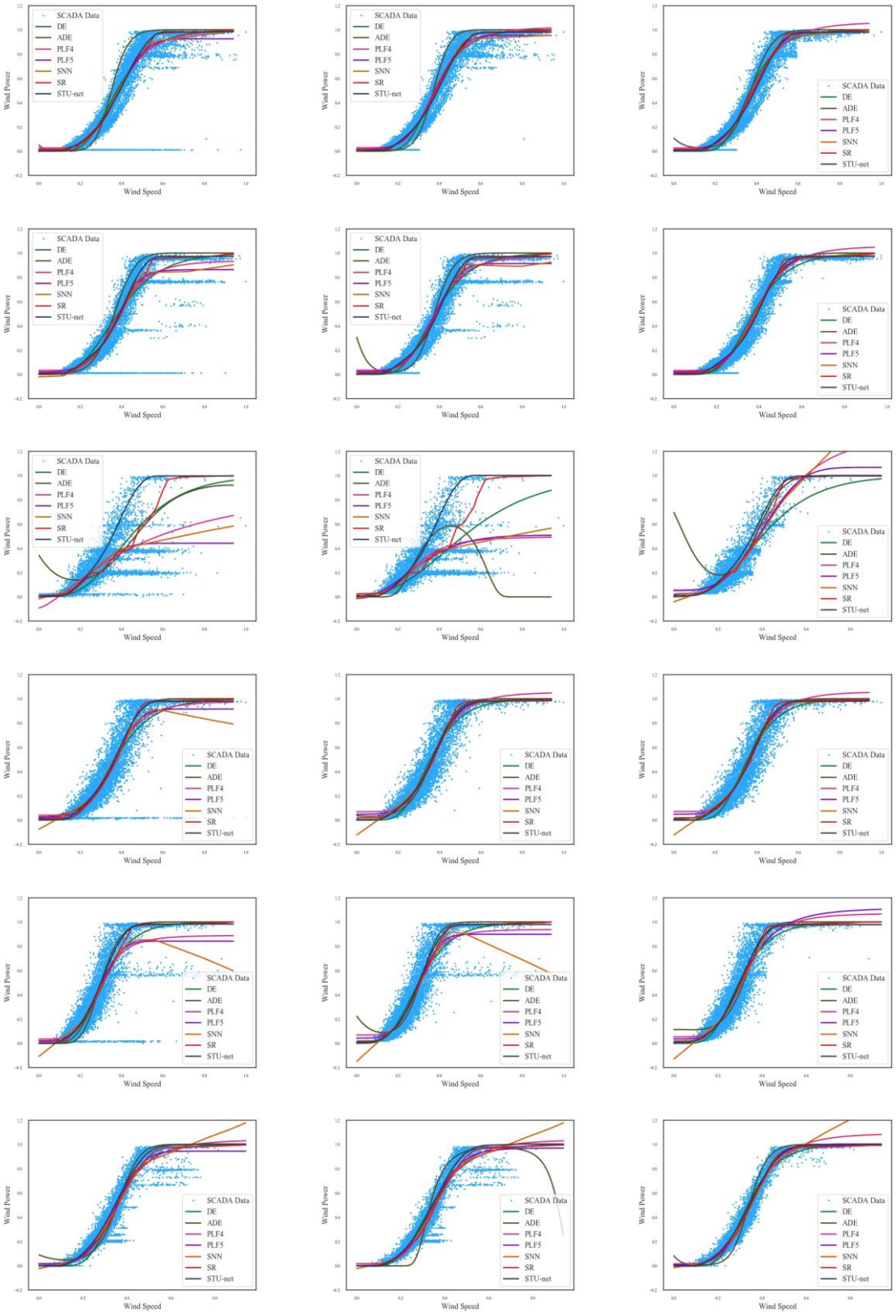
We randomly sample 20% of data points from the data utilizes in S3 as the testing dataset. We also consider two application patterns, **the normal pattern (NP)**, in which the model could access all training data, and **the insufficient data pattern (IDP)**, which is widely considered in the start-up wind farm of insufficient historical records so that the model could only access to a part of training data.

D. Model Assessment Metrics

Let x_i and y_i denote the i^{th} ground truth wind speed and wind power in the testing set respectively. Following metrics, the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), described in (22-23) are utilized in this research to evaluate the performance of WPCM methods.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_{WPC}(x_i) - y_i)^2} \quad (22)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_{WPC}(x_i) - y_i| \quad (23)$$



(a) S1 (b) S2 (c) S3

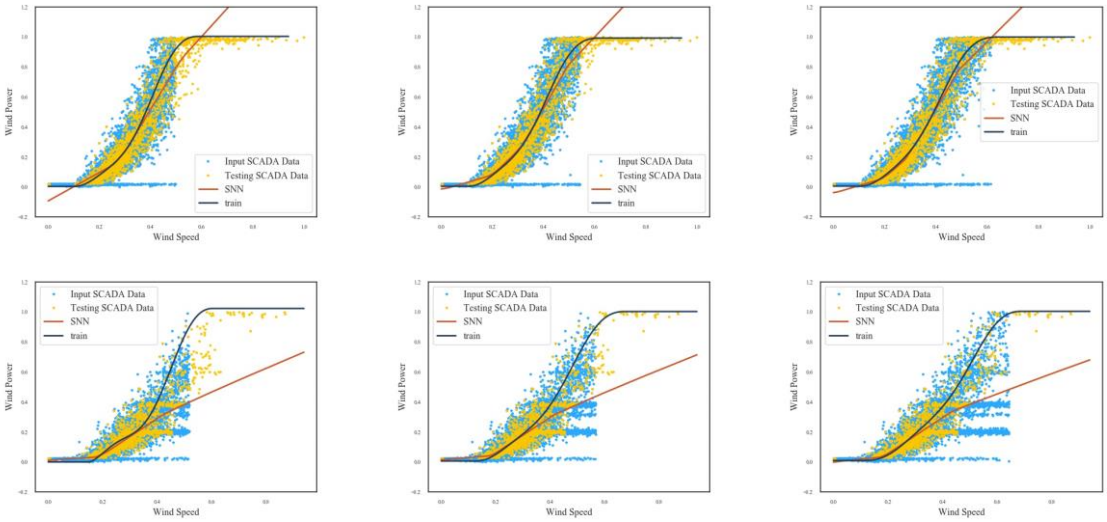
Fig. 6. Illustrations of WPCM results.

TABLE III
THE RMSE AND MAE OF THE TESTING SET.

Method	RMSE(S1)	RMSE(S2)	RMSE(S3)	MAE(S1)	MAE(S2)	MAE(S3)
DE	8.423e-2	8.194e-2	7.946e-2	6.132e-2	6.024e-2	5.893e-2
ADE	9.636e-2	9.000e-2	9.138e-2	7.258e-2	6.605e-2	6.755e-2
PLF4	7.975e-2	7.567e-2	7.102e-2	5.728e-2	5.421e-2	5.087e-2
PLF5	7.959e-2	7.640e-2	7.213e-2	5.689e-2	5.419e-2	5.049e-2
SNN	7.773e-2	7.355e-2	**6.838e-2**	5.492e-2	5.102e-2	4.626e-2
SR	7.610e-2	7.319e-2	6.861e-2	5.037e-2	4.718e-2	**4.399e-2**
STU-net	**6.96e-2**	**6.96e-2**	6.893e-2	**4.688e-2**	**4.684e-2**	4.616e-2
STU-net w/o DKC	7.062e-2	7.066e-2	6.889e-2	4.710e-2	4.709e-2	4.602e-2

TABLE IV
THE AVERAGE TIME CONSUMING FOR WPCM.

Method	STU-net	STU-net w/o DKC	DE	ADE	PLF4	PLF5	SNN	SR
Running Time (s)	1.34e-2	3.26e-3	0.441	1.73	252	252	0.907	2.76e-3



(a) 85% wind speed quantile (b) 90% wind speed quantile (c) 95% wind speed quantile

Fig. 7. The WPCM results of the proposed STU-net with insufficient data pattern

E. Computational Results

Fig. 6 visualizes the WPCM results of six randomly selected WTs under three application scenarios in NP. It is observable that the proposed STU-net can well model the WPC regardless of the observed SCADA data quantity. The other benchmarks could offer suitable WPC modeling results under S3; however, if the observed SCADA data points incorporate noisy points or the application scenario requires the low latency, they failed to model WPCs accurately.

Table III reports the testing results of considered methods under different application scenarios in NP. Considering that the testing set is sampled from data utilizes in S3, the accuracy of the proposed STU-net could not exceed ML based models, which are directly trained with data utilized in S3. Even though, the STU-net can offer a very close accuracy, which further demonstrates the effectiveness of the proposed STU-net based WPCM method.

Fig. 7 presents the WPCM results of the STU-net and SNN considering two randomly selected WTs with different quantiles of the wind speed exclusion in IDP under S1, which means the model cannot access the data of the wind speed exceeding the set quantile. It is observable that the proposed STU-net can produce more reasonable estimations of WPCs than SNN. Fig. 8 also shows the influence of the quantile settings on the performance of the proposed STU-net and SNN under S2 for a fair comparison. It is observable that the proposed STU-net can output stable WPCs in IDP; however, the performance of the SNN is diminished significantly by IDP.

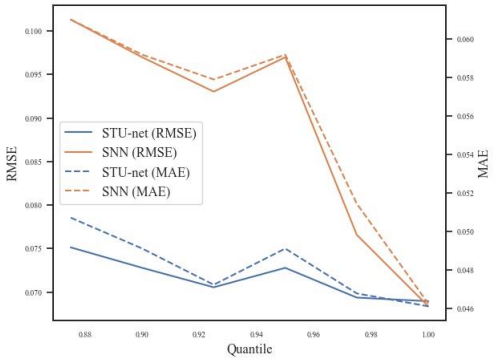


Fig. 8. Relationships between the quantile and the performance of STU-net and SNN

Table IV reports the average running time of each method. It is observable that the proposed STU is faster than most of the

considered benchmarks. Although the running time of the proposed STU-net is 5 times longer than the SR, it is already acceptable for the online implementation. A regular PC with a Core i7-8700 CPU, a NVIDIA GeForce GTX 2080Ti GPU, and a 32Gb memory is utilized to conduct the computational experiments.

V. Conclusion

This paper presented a novel machine vision assisted WPCM method, the STU-net. Firstly, we renovated the formulation of the WPCM problem from a machine vision perspective, based on which the STU-net was developed to avoid sophisticated data preprocessing and retraining. The proposed STU-net contained two steps, the ST as well as the pixel mapping and correction. The ST process was developed to approximate $f_{\Lambda_w}^{-1}$ with synthesized training image pairs, I_x and I_{wpc}, via a U-net. After optimization, the developed U-net could be fed with the WPC images constructed based on observed SCADA data and offer the corresponding neat WPC I_{wpc}.To output a parametric form of the I_{wpc}, f_{wpc}, the pixel mapping and correction was developed. Firstly, based on the pixel mapping relationship, I_{wpc} was converted to scattered SCADA data points and then fitted with a polynomial f_p. Next, the fitted f_p was corrected based on the domain knowledge to ensure that the output f_{wpc} could meet constraints of the WPC described in (2).

Six classical benchmarks, three application scenarios and two application patterns were utilized to verify the effectiveness of the proposed STU-net based on observed SCADA data collected from 76 WTs. The results demonstrated that, without any data preprocessing and retraining, the proposed STU-net could offer competitive accuracy with benchmark methods applied with careful data preprocessing. The computational experiments also illustrated the superior of the proposed STU-net for WPCM with insufficient SCADA data points.

References

[1] Y. Wang, Q. Hu, D. Srinivasan, and Z. Wang, "Wind Power Curve Modeling and Wind Power Forecasting With Inconsistent Data," *IEEE Trans. Sustain. Energy*, vol. 10, no. 1, pp. 16–25, 2019.

[2] M. Khalid and A. V. Savkin, "A method for short-term wind power prediction with multiple observation points," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 579–586, 2012.

[3] D. Villanueva, J. L. Pazos, and A. Feijóo, "Probabilistic Load Flow Including Wind Power Generation," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1659–1667, 2011.

[4] J. M. Morales, A. J. Conejo, and J. Pérez-Ruiz, "Economic valuation of reserves in power systems with high penetration of wind power," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 900–910, 2009.

[5] J. M. Morales, A. J. Conejo, and J. Pérez-Ruiz, "Short-term trading for a wind power producer," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 554–564, 2010.

[6] Z. Zhang, Q. Zhou, and A. Kusiak, "Optimization of wind power and its variability with a computational intelligence approach," *IEEE Trans. Sustain. Energy*, vol. 5, no. 1, pp. 228–236, 2014.

[7] A. Kusiak and A. Verma, "Monitoring wind farms with performance curves," *IEEE Trans. Sustain. Energy*, vol. 4, no. 1, pp. 192–199, 2013.

[8] Y. Wang, Q. Hu, D. Srinivasan, and Z. Wang, "Wind Power Curve Modeling and Wind Power Forecasting With Inconsistent Data," *IEEE Trans. Sustain. Energy*, vol. 10, no. 1, pp. 16–25, 2019.

[9] Y. Wang, Q. Hu, L. Li, A. M. Foley, and D. Srinivasan, "Approaches to wind power curve modeling: A review and discussion," *Renew. Sustain. Energy Rev.*, vol. 116, no. October, p. 109422, 2019.

[10] Q. Hu, Y. Wang, Z. Xie, P. Zhu, and D. Yu, "On estimating uncertainty of wind energy with mixture of distributions," *Energy*, vol. 112, pp. 935–962, 2016.

[11] M. Lydia, S. S. Kumar, A. I. Selvakumar, and G. E. Prem Kumar, "A comprehensive review on wind turbine power curve modeling techniques," *Renew. Sustain. Energy Rev.*, vol. 30, pp. 452–460, 2014.

[12] E. Sainz, A. Llombart, and J. J. Guerrero, "Robust filtering for the characterization of wind turbines: Improving its operation and maintenance," *Energy Convers. Manag.*, vol. 50, no. 9, pp. 2136–2147, 2009.

[13] V. Sohoni, S. Gupta, and R. Nema, "A comparative analysis of wind speed probability distributions for wind power assessment of four sites," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 24, no. 6, pp. 4724–4735, 2016.

[14] 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, 2013.

[15] 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.

[16] L. A. Osadciw, Y. Yan, X. Ye, G. Benson, and E. White, "Wind Turbine Diagnostics Based on Power Curve Using Particle Swarm Optimization," *Wind Power Syst.*, pp. 151–165, 2010.

[17] Y. Wang, Q. Hu, D. Srinivasan, and Z. Wang, "Wind Power Curve Modeling and Wind Power Forecasting With Inconsistent Data," *IEEE Trans. Sustain. Energy*, vol. 10, no. 1, pp. 16–25, 2019.

[18] S. Shokrzadeh, M. Jafari 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, 2014.

[19] A. Kusiak, H. Zheng, and Z. Song, "On-line monitoring of power curves," *Renew. Energy*, vol. 34, no. 6, pp. 1487–1493, 2009.

[20] T. Ouyang, A. Kusiak, and Y. He, "Modeling wind-turbine power curve: A data partitioning and mining approach," *Renew. Energy*, vol. 102, pp. 1–8, 2017.

[21] F. Pelletier, C. Masson, and A. Tahan, "Wind turbine power curve modelling using artificial neural network," *Renew. Energy*, vol. 89, pp. 207–214, 2016.

[22] M. Schlechtingen, I. F. Santos, and S. Achiche, "Using data-mining approaches for wind turbine power curve monitoring: A comparative study," *IEEE Trans. Sustain. Energy*, vol. 4, no. 3, pp. 671–679, 2013.

[23] H. Long, L. Sang, Z. Wu, and W. Gu, "Image-based abnormal data detection and cleaning algorithm via wind power curve," *IEEE Trans. Sustain. Energy*, vol. 11, no. 2, pp. 938–946, 2020.

[24] R. R. B. De Aquino *et al.*, "Assessment of power curves in models of wind power forecasting," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2016-October, pp. 3915–3922, 2016.

[25] 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, 2018, doi: 10.1109/TSTE.2017.2717021.

[26] Z. Wang, L. Wang, and C. Huang, "A Fast Abnormal Data Cleaning Algorithm for Performance Evaluation of Wind Turbine," vol. 9456, no. c, 2020.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9351, pp. 234–241, 2015.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456, 2015.

[29] P. Civicioglu, "Backtracking Search Optimization Algorithm for numerical optimization problems," *Appl. Math. Comput.*, vol. 219, no. 15, pp. 8121–8144, 2013