IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 2, 2018, accepted March 8, 2018, date of publication April 2, 2018, date of current version May 2, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2818678

A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost

DAHAI ZHANG, LIYANG QIAN, BAIJIN MAO, CAN HUANG, BIN HUANG, AND YULIN SI, (Member, IEEE)

Ocean College, Institute of Ocean Engineering and Technology, Zhejiang University, Zhoushan 316000, China

Corresponding author: Yulin Si (yulinsi@zju.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51705453 and Grant 51579222, in part by the international collaboration and exchange program from the NSFC-RCUK/EPSRC under Grant 51761135011, and in part by the Fundamental Research Funds for the Central Universities under Grant 2017XZZX001-02A, Grant 2017FZA4028, and Grant 2017QNA4040.

ABSTRACT Wind energy has seen great development during the past decade. However, wind turbine availability and reliability, especially for offshore sites, still need to be improved, which strongly affect the cost of wind energy. Wind turbine operational cost is closely depending on component failure and repair rate, while fault detection and isolation will be very helpful to improve the availability and reliability factors. In this paper, an efficient machine learning method, random forests (RFs) in combination with extreme gradient boosting (XGBoost), is used to establish the data-driven wind turbine fault detection framework. In the proposed design, RF is used to rank the features by importance, which are either direct sensor signals or constructed variables from prior knowledge. Then, based on the top-ranking features, XGBoost trains the ensemble classifier for each specific fault. In order to verify the effectiveness of the proposed approach, numerical simulations using the state-of-the-art wind turbine simulator FAST are conducted for three different types of wind turbines in both the below and above rated conditions. It is shown that the proposed approach is robust to various wind turbine models including offshore ones in different working conditions. Besides, the proposed ensemble classifier is able to protect against overfitting, and it achieves better wind turbine fault detection results than the support vector machine method when dealing with multidimensional data.

INDEX TERMS Wind turbines, fault detection, data-driven, random forests, extreme gradient boosting.

I. INTRODUCTION

Wind energy in the world has been developed rapidly in the past decade. Global wind energy new capacity additions in 2016 totalled nearly 54GW, bringing the cumulative installed capacity to 486GW [1]. Particularly, offshore wind has an exciting performance, the new capacity of which in 2016 totalled 2.3GW, and the global number has reached 14GW [2]–[4]. Although offshore wind farm has better wind resource and wide operating space for large turbines, the cost of offshore wind is still much higher than onshore [5]. The high cost of machinery and infrastructure demands that the availability and reliability need to be concerned. If maintenance or fault tolerant control is not implemented timely as a minor fault happens, the collateral damage to other components will bring a huge loss [6]. Besides, compared to onshore wind turbines, offshore ones have higher failure rate due to harsh and complex wind and wave environment. The expensive offshore operation and maintenance is also a barrier to low cost of energy (CoE) [7], [8]. In order to improve system reliability and reduce CoE, advanced fault detection (FD) and fault tolerant control (FTC) schemes are required [9], [10]. The FD process evaluates the measured data and rises a warning at an early stage when a fault has occurred. Then the controller can apply a suitable approach to extend the turbine availability and also protect the turbine from further damage [11].

FD methods have been widely applied in industry, and they can usually be divided into two categories, model-based [12]–[15] and data-driven approaches [16]–[18]. Model-based FD methods rely on accurate system modelling, which are able to perform efficient and accurate FD if the system dynamics can be well described. However, model based methods have difficulty in practice if the application cannot fit sufficiently well with the assumptions about the models and objectives. On the contrary, system modelling is not needed for data-driven methods [18]–[20], with only

requiring data that can be stored and transmitted easily with the condition monitoring process using appropriate sensors. Since no model parameters are used to establish the classifier, data-driven methods are supposed to be more robust [17]. This feature makes the methods practical in industrial applications, especially on the occasions where accurate system models are not available, such as wind turbines. However, one problem for data-driven methods is their statistical instability, so there are also other FD strategies combining both model-based and data-driven approaches [21], [22].

Among data-driven methods, support vector machines (SVM) is a popular supervised learning algorithm which many researchers use to train a classifier for FD [23], [24]. SVM has its own merit of efficiency and robustness that it usually has a good performance when dealing with low dimensional data. However, SVM is vulnerable to overfitting in front of high dimensional inputs, which will lead to deceptive diagnostic results [25]. In contrast, an ensemble learning method does not overfit easily in multi-dimensional classifier design. Besides, it will aggregate multiple weighted base learning models to obtain a combined model, which outperforms each single classification or regression model in it [26]. In this work, two ensemble learning models, random forests (RF) and extreme gradient boosting(XGBoost), are combined to improve the performance of wind turbine fault classifier. These two ensemble models have already been employed in various classification or regression applications, which have shown that RF and XGBoost are effective and efficient classifier design algorithms [27].

This paper is organized as follows. Section 2 describes the wind turbine FD benchmark model, where the state-of-the-art wind turbine simulator FAST is used [28], and ten different kinds of assumed typical fault scenarios are described. Section 3 introduces the proposed data-driven ensemble classifier design process, where RF and XGBoost are combined to improve the FD robustness and accuracy. Numerical simulation results are given in Section 4, including both performance comparison studies and robustness tests, which are used to verify the effectiveness of the proposed method. Conclusions are drawn in the last section.

II. BENCHMARK MODEL

This section introduces the wind turbine FD framework benchmark and ten different kinds of fault scenarios. Reference wind turbine models used in this work are introduced here as well.

A. WIND TURBINE FAULT DETECTION FRAMEWORK

The wind turbine FD framework, as shown in Fig. 1, proposed by Odgaard and Johnson [29] is used as the simulation benchmark. The simulations are based on the state-of-the-art wind turbine simulator FAST version 8 [28] and implemented in Matlab/Simulink environment. In this benchmark, it is assumed that fifteen signals from the wind turbine can be monitored by various kinds of sensors, including wind velocity at hub height V_{hub}, rotor angular velocity ω_r,

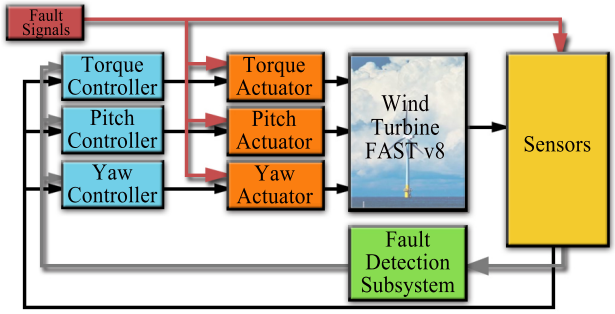


FIGURE 1. Simulation Framework.

generator angular velocity ω_g, generator torque τ_g, generator power P_g, blade 1-3 pitch angle $\beta_i (i = 1, 2, 3)$, low speed shaft azimuth φ, torque at blade root of blade 1-3 $M_{\text{B},i}$ $(i = 1, 2, 3)$, nacelle fore-art and side-side accelerations $\ddot{x}_X, \ddot{x}_Y$, and nacelle yaw error estimate Ξ_e. These measurement data are then used as the inputs of the wind turbine FD subsystem and wind turbine fault tolerant controller. It can also be seen from Fig. 1 that out from the wind turbine controller blocks, three different control signals, i.e. generator torque, blade pitch angle, and yaw position/rate, are sent into corresponding actuators, respectively.

1) SENSOR MODEL

In this benchmark model, signals monitored by sensors are modelled by combining FAST outputs and band limited white noises with different noise power levels [29]. Without introducing other sensor characteristics, e.g. calibration error, drift and delay, this simple structure of sensor models is used here since noises are the main uncertainties for sensor measurements.

2) ACTUATOR MODEL

Dynamics for wind turbine actuators, including blade pitch actuator, generator torque actuator, and nacelle yaw actuator, are not characterized within the FAST block, and they have to be modelled, respectively.

The actual blade pitch angle β and the pitch reference β_r are modelled as a second order closed loop transfer function

$$\frac{\beta(s)}{\beta_r(s)} = \frac{\omega_n^2}{s^2 + 2 \cdot \zeta \omega_n \cdot s + \omega_n^2}, \tag{1}$$

where the ζ is the damping factor and the ω_n denotes the natural frequency. For the no-fault case, ζ is set to be 0.6, and the ω_n is set to be 11.11. The pitch angle is limited from -2 deg to 90 deg. The pitch rate is limited from -8 deg/s to 8 deg/s. The generator and convertor torque are modelled by a first order transfer function

$$\frac{\tau_g(s)}{\tau_{g,\tau}(s)} = \frac{\alpha_{gc}}{s + \alpha_{gc}}, \tag{2}$$

where the coefficient α_{gc} depends on the generator and convertor, and it is set as 50 in this study. The power produced

TABLE 1. Description of the faults.

No.	Fault location	Fault type	Describtion	Active period
1	Blade root bending moment sensor	Scaling	Flapwise, blade 2, scaled by 0.95	20-45s
2	Tower top accelerometer	Offset	-0.5 m/s^2 offset on $\ddot{x}_X, \ddot{x}_Y$	75-100s
3	Generator speed sensor	Scaling	$\omega_{g,m}$ scaled by 0.95	130-155s
4	Pitch angle sensor	Stuck	$\beta_{1,m}$ holds to 1 deg	185-210s
5	Generator power sensor	Scaling	$P_{g,m}$ scaled by 1.1	240-265s
6	Low speed shaft position encorder	Offset	Random offset on φ	295-320s
7	Pitch actuator	Slow dynamics change	ω_n=3.42, ζ=0.9	350-410s
8	Pitch actuator	Abrupt dynamics change	ω_n=5.73, ζ=0.45	440-465s
9	Torque actuator	Offset	1000 N·m offset on $\tau_{g,m}$	495-520s
10	Yaw drive	Stuck	Yaw angular velocity set to 0 rad/s	550-575s

by the generator is given by

$$P_g = \eta_g \omega_g \tau_g, \tag{3}$$

where η_g is the generator efficiency which is defined as 0.98 in the tests. Although the pitch and generator models are implemented in Simulink, the yaw dynamics are not required, where only the yaw angular velocity $\omega_{y,r}$ and the yaw angular position reference $\psi_{y,r}$ are needed for the FAST block.

B. FAULT DESCRIPTIONS

Sensor and actuator faults are both considered in the wind turbine FD simulation framework. Ten different fault scenarios are defined, including six sensor faults and four actuator faults. Sensor faults cause measurement stuck, scaled or offset from the true values, while actuator faults lead to parameter changes in the actuators. Details for all the ten types of faults in a ten minute simulation are described in Table 1.

C. REFERENCE WIND TURBINE MODELS

The National Renewable Energy Laboratory (NREL)'s 5MW baseline wind turbine models [30], [31], including onshore, offshore fixed-bottom and floating types, are seen as the reference wind turbine models in this work. Properties of these three models are shown in table 2. Detailed properties of the semisubmersible floating wind turbine are not given in this table, which can be found in [30].

TABLE 2. Properties of wind models.

Wind Turbine Type	Onshore	Monoplie	Semi-submersible
Rating	5MW		
Rotor Configuration	Upwind, 3 blades		
Hub Height	87.6m		
Rotor, Hub Diameter	126m, 3m		
Cut-in, Rated, Cut-out Wind Speed	3m/s, 11.4m/s, 25m/s		
Cut-in, Rated Rotor Speed	4.9rpm, 12.1rpm		
Rated Tip Speed	80m/s		
Rotor, Nacelle Mass	110,000kg, 240,000kg		
Tower mass	347,460kg		249,718kg
Platform mass	0		3,852,180kg
Number of Mooring Lines	0		3
Depth to Anchors Below SWL	0		200m

III. DATA-DRIVEN ENSEMBLE CLASSIFIER DESIGN

The proposed data-driven ensemble classifier design process for wind turbine FD is introduced in this section.

Before designing the classifier, new features for each specific fault are constructed with feature analysis process. In fact, feature analysis is an essential step to make sure the classifier works well, since learning algorithms only rely on given data signals without building new features, which are sometimes combination of different sensor measurements instead of signals from individual sensor. Besides, in order to deal with the sensor noises, proper filtering process should also be performed before the ensemble model training.

After the preprocessing, the data driven fault classifier design, which is a combination of RF and XGBoost, is proposed. In the classifier design, RF model is applied to evaluate and sort the importance of features for all the faults, which are either direct sensor signals or constructed data with different sensor measurements. Then, the fault classifier is trained based on XGBoost model with three top-ranking features selected by the RF model.

The base learning model of RF and XGBoost is the decision tree. Each internal node in the decision tree is labeled with an input feature. The arcs are labeled with different possible output values. Trees grow deeper with splitting at a node, which is determined by a certain generation algorithm. A single decision tree model has to grow very deep to learn highly irregular patterns, which will result in overfitting and produce low bias but high variance classification results. In contrast, against overfitting, RF and XGBoost utilize the ensemble strategies bagging and gradient boosting, which are adopted in this work for the wind turbine FD classifier design.

A. FEATURE ANALYSIS

Before training the classifier, feature analysis should be performed to include all possible indicators from sensor measurements. It should be noted that although wind turbine system knowledge has been used to find new features, this is still a data-driven FD approach that no wind turbine related models are built. Feature analysis is conducted for all the ten faults, which are described as follows.

1) Fault 1 results that the sensor output of blade root flapwise bending moment $M_{\text{flap},m}$ for blade 2 scaled by 0.95. This is a minor fault for sensor measurement, which is not easy to be detected from the statistics point of view. In fact,

no strong correlation between this fault and the given 15 sensor measurements was found, so it is difficult to establish a new feature for this fault . Besides, this sensor fault has no influence on wind turbine normal operation, so it cannot be reflected from the wind turbine dynamics. Therefore, no proper feature was found fault 1 in this work.

2) Fault 2 causes a -0.5m/s^2 offset on the tower top accelerometer output in both fore-aft and side-side directions. These two accelerometers are within the 15 sensors, so that it is unnecessary to build a new feature.

3) Fault 3 causes the generator speed sensor to be scaled by a factor of 0.95. The gear ratio for the drivetrain will be abruptly changed when this fault happens, which can be seen as a new feature. Based on the rotor angular velocity ω_r and the generator angular velocity ω_g, the gear ratio $\eta_{r,g}$ can be given by

$$\eta_{r,g} = \frac{\omega_r}{\omega_g}. \tag{4}$$

4) Fault 4 leads to that the pitch angle sensor of blade 1 has a stuck value of 1 degree. For this fault, sensor outputs at previous time steps can be stored to calculate the slope K of blade pitch angle,

$$K = \frac{\beta_t - \beta_{t-1}}{\Delta t}, \tag{5}$$

where the sample time $\Delta t = 0.0125$s.

5) Fault 5 causes the generator power sensor scaled by a factor of 1.1. Expected generator power $P_{g,e}$ can be calculated using other sensor signals, and the difference $\Delta P = P_g - P_{g,e}$ can be given by

$$\Delta P_1 = P_g - \eta_g \omega_g \tau_g, \tag{6}$$

$$\Delta P_2 = P_g - \eta_g \frac{\omega_r}{\eta_{r,g}} \tau_g, \tag{7}$$

where P_g refers to the measurements of generator power. Based on the measured generator angular velocity ω_g and rotor generator angular velocity ω_r, these two features can be respectively constructed.

6) Fault 6 causes a random offset on the low speed shaft encoder. When this fault happens, the first order derivative of the encoder output will change from a constant value to an oscillating signal. Two storage modules are used to obtain the second derivative of φ, and $\ddot{\varphi}$ is seen as the new constructed feature. Besides, once the low speed shaft rotates every 360 degrees, the sensor value will be reset to 0, so that drastic changes for $\dot{\varphi}$ have to be deleted in order to keep a stable feature signal.

7) Fault 7 is an actuator fault happened in the pitch actuator of blade 2. Due to the increasing air content, ω_n and ζ in the pitch transfer function will change slowly after the fault is activated. In fact, according to the baseline collective blade control strategy, the pitch angle for all the three blades should be the same. Therefore, the difference between β_1 and β_2 can be seen as a new feature.

8) Fault 8 is similar to fault 7. Due to the hydraulic power drop, ω_n and ζ in the actuator transfer function will be abruptly changed when the fault is activated. Difference between β_1 and β_3 will be abruptly changed as well.

9) Fault 9 causes an offset on the measured generator torque. Control signal of generator torque is used to compare the real torque and the expected torque τ_e.

Difference between them is given by

$$\Delta\tau = (\frac{P_g}{\eta_g \omega_g} - \tau_e). \tag{8}$$

10) Fault 10 results in a stuck of yaw actuator. Similar to fault 1, no satisfactory new feature is found with the given 15 sensor signals. Model based methods will probably be suitable to deal with this fault.

B. SENSOR DATA FILTERING

Different types of filters have been tested in the numerical tests, while the first order filter is chosen in the end due to its comprehensive behaviour. The discrete form of the first order filter can be given by

$$Y(n) = \alpha X(n) + (1 - \alpha) Y(n - 1), \tag{9}$$

where α is a parameter controlling the filter cutoff frequency.

Filtered signals always have time delay within the real-time filtering process. The lower the corner frequency is, the longer the time delay is, which will affect the FD accuracy. On the contrary, higher cutoff frequency will reserve more noises, so that signals may not be clear enough to train a classifier. Since the ensemble classifier is able to deal with multi-dimensional dataset, the filtered signals using different α are all included in the input in order to balance the delay-sensitivity trade-off. Actually, RF is able to choose the best α candidate by ranking the given features, the principle of which will be introduced in next part.

It is worth noting that time delay caused by filters is a serious problem for FD accuracy. In other word, the training process will be influenced by the filtering delay. For instance, when the fault has already been activated, the sensor signals might be still at a non-fault level due to the time delay, thus the error labeled data will decrease the classifier training accuracy. Therefore, it is necessary to correct these wrongly labeled data in the training process.

C. RANDOM FORESTS FEATURE RANKING

Random forests is an ensemble machine learning method, which is operated by constructing a multitude of decision trees at training time and outputting the class that is averaged or voted by every individual tree [32]. RF was proposed by Breiman in 2001, who added an additional layer of randomness to bagging method. RF is not only applicable in regression and classification, but also has excellent behavior in variable selection [33]. A sketch map for RF is given in Fig. 2.

Bagging, which refers to bootstrap aggregating, is an ensemble algorithm designed to improve the stability and accuracy of individual predictive model such as trees [34]. Bagging helps decision trees to reduce their variance and the

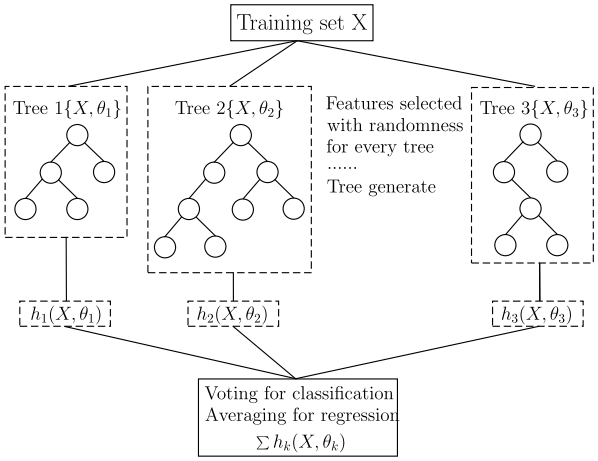


FIGURE 2. Flow chart of random forests.

influence of overfitting. Assumed that a training set is given by $X = x_{1,2...n}$ with response $Y = y_{1,2...n}$, bagging will repeat K times to select a random sample with replacement of the training set and fits trees to these samples. A tree $h_k, (k = 1, 2...K)$ will be trained every time. After training, the produced prediction model can be established by averaging the predictions from K regression trees or by taking the majority vote from K decision trees. Note that samples are selected with replacement, and the probability that a certain sample is not selected after K times selection can be given by

$$P = (1 - \frac{1}{n})^K. \tag{10}$$

In the bagging process of RF, K usually equals to n. When n is big enough, about 36.8% of the training samples will not be selected, and they are called out-of-bag samples.

Besides, RF improves the general tree growing scheme, where at each candidate split in the tree model, a random subset of the features are used instead of selecting a certain feature from all the candidates. Whereas in a traditional tree ensembling scheme, if a few features are very strong predictors for the response, these features will be selected in many of the base estimators. Then, these trees will be much correlated, thus weakening the prediction abilities.

The theoretical background of RF can be basically divided into two parts: RF convergence theorem and generalization error bound. All the proof procedure can be found in [32], while a brief description is given as follows.

A RF model can be assumed as a collection of tree-structured classifiers $h(\mathbf{x}, \Theta_k)(k = 1, 2, 3\ldots)$, where the Θ_k are independent and identically distributed random vectors. Also, a performance index is needed to describe the confidence level of the RF model, which is defined as a margin function mg(·),

$$\mathrm{mg}(\mathbf{x}, \mathbf{y}) = \mathrm{av}_k I(h_k(\mathbf{x}, \Theta_k) = \mathbf{y}) - \max_{j \neq \mathbf{y}} \mathrm{av}_k I(h_k(\mathbf{x}, \Theta_k) = j), \tag{11}$$

where $I(\cdot)$ is the indicator function, and av(·) denotes an average value. First half of this index is the average number of votes at $(\mathbf{x}, \mathbf{y})$ for the right class, while the second half refers to the average vote for the most class except the right class. Confidence level will be higher as the margin is larger. Then, the generalization error PE^* is given by

$$PE^* = P_{\mathbf{x},\mathbf{y}}(\mathrm{mg}(\mathbf{x}, \mathbf{y}) < 0), \tag{12}$$

where $P(\cdot)$ represents probability. As the number of trees increases, for almost surely all sequences Θ_k, PE^* converges to

$$P_{\mathbf{x},\mathbf{y}}(P_\Theta(h(\mathbf{x}, \Theta) = \mathbf{y}) - \max_{j \neq \mathbf{y}} P_\Theta(h(\mathbf{x}, \Theta) = j) < 0). \tag{13}$$

The convergence of generalization error shows that RF can produce a limiting value of the generalization error and do not overfit as more trees are added. The upper bound for the PE^* is given by

$$PE^* \leq \frac{\bar{\rho}(1 - s^2)}{s^2}, \tag{14}$$

where $\bar{\rho}$ is the mean value of the correlation, s is the strength of an individual tree in the RF model. It means with increasing the strength of individual tree and reducing the correlation between trees, the RF model will achieve more accurate prediction results.

Besides, as described above, in order to increase the individual tree strength in the RF model, feature analysis has to be firstly performed to determine the dominant signals for each fault. In other words, proper features, either sensor measurements or combination of sensor signals, closely related to each specific fault have to be designed before ranking the features. Then, based on the out-of-bag samples, all the features can be sorted by the prediction ability with the out-of-bag estimates. More specifically, tree-structured classifiers in RF that have important variables at nodes are supposed to be highly related to the response, so that important variables can be selected in these strong trees. In this work, three top-ranking features are chosen for further FD classifier design.

D. XGBOOST CLASSIFIER DESIGN

Based on the three features chosen by RF, XGBoost is then adopted to train the wind turbine fault classifier.

XGBoost is a scalable machine learning system for tree boosting that proposed by Chen and Guestrin [35] in 2016. Among the 29 winning solutions in the machine learning competition Kaggle in 2015, XGBoost was the most popular method that 17 solutions used XGBoost. The superior performance of XGBoost in supervised machine learning is the reason why it is chosen to train the wind turbine fault classifier in this work.

Gradient boosting is the original model of XGBoost, combining weak base learning models into a stronger learner in an iterative fashion [36]. As shown in Fig. 3, at each iteration of gradient boosting, the residual will be used to correct the previous predictor that the specified loss function can be optimized. As an improvement, regularization is added to the

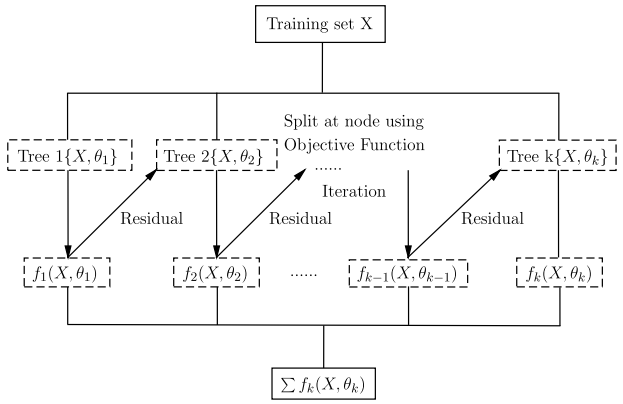


FIGURE 3. Flow chart of extreme gradient boosting.

loss function to establish the objective function in XGBoost measuring the model performance, which is given by

$$J(\Theta) = L(\Theta) + \Omega(\Theta). \tag{15}$$

The parameters trained from given data are denoted as Θ. L is the training loss function, such as square loss or logistic loss, which measures how well the model fits on training data. Ω is the regularization term, such as $L1$ norm or $L2$ norm, which measures the complexity of the model. Simpler models tends to have better performance against overfitting. Since the base model is decision tree, the output of model $\widehat{y}_i$ is voted or averaged by a collection F of k trees:

$$\widehat{y}_i = \sum_{i=1}^{k} f_k(x_i), f_k \in F. \tag{16}$$

Objective function at the t time iteration can be concrete into

$$J^{(t)} = \sum_{i=1}^{n} L(y_i, \widehat{y}_i) + \sum_{k=1}^{t} \Omega(f_k), \tag{17}$$

where the n is number of predictions. Here the $\widehat{y}_i^{(t)}$ can be given as

$$\widehat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \widehat{y}_i^{(t-1)} + f_t(x_i). \tag{18}$$

In [35], regularization term $\Omega(f_k)$ for a decision tree is defined as

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} {w_j}^2, \tag{19}$$

where the γ is the complexity of each leaf. T is the number of leaves in a decision tree. λ is a parameter to scale the penalty. w is the vector of scores on leaves. Then, second-order Taylor expansion, instead of first-order in general gradient boosting, is taken to the loss function in XGBoost. Assumed that the loss function is mean square error (MSE), the objective function can be finally derived as

$$J^{(t)} \approx \sum_{i=1}^{n} [g_i w_{q(x_i)} + \frac{1}{2}(h_i w_{q(x_i)}^2)] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} {w_j}^2, \tag{20}$$

with the constants removed. Here the $q(\cdot)$ is a function that assigns data point to the corresponding leaf. g_i and h_i is the first and second derivative of MSE loss function. In (20), the loss function is determined by sum of loss value for each data sample. Because each data sample corresponds to only one leaf node, the loss function can also be expressed by the sum of loss value for each leaf node. Thus,

$$J^{(t)} \approx \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i)\omega_j + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda)\omega_j^2)] + \gamma T. \tag{21}$$

According to (21), G_j and H_j can be defined as

$$G_j = \sum_{i \in I_j} g_i, \quad H_j = \sum_{i \in I_j} h_i, \tag{22}$$

where I_j represents all the data samples in leaf node j. Hence, the optimization of objective function can be transformed into a problem of finding the minimum of a quadratic function. In other words, after a certain node split in the decision tree, the change of model performance can be evaluated based on the objective function. If the decision tree model performance is improved after this node split, this change will be adopted, otherwise the split will be stopped. Besides, because of the regularization when optimizing the objective function, a predictive classifier can be trained against overfitting.

In this work, the machine learning models are trained on python 3.6.1 with several scientific computing libraries, such as numpy 1.12.1 and pandas 0.19.2, which provides efficient data structures and preprocessing methods. Besides, Scikit-learn 0.18.1 and Xgboost 0.6 are imported to support SVM, RF and XGBoost learning models [37].

IV. SIMULATIONS AND RESULTS

In this section, the entire simulation process of wind turbine FD is described, and then the robustness analysis and comparison study are presented to demonstrate the effectiveness of the proposed approach.

A. SIMULATION SETUP

As mentioned above, the wind turbine FD simulations in this work are based on FAST, which is packaged as an S-Function within the Matlab/Simulink framework. Fig. 1 has illustrated the schematic diagram of the simulation model. The wind turbine parameters and operation settings for simulations are described in the FAST input files, and all the intermediate data can be read from the Matlab workplace. Particularly, sensor data are read from the FAST S-function outputs and added with noises of different levels. We have uploaded all

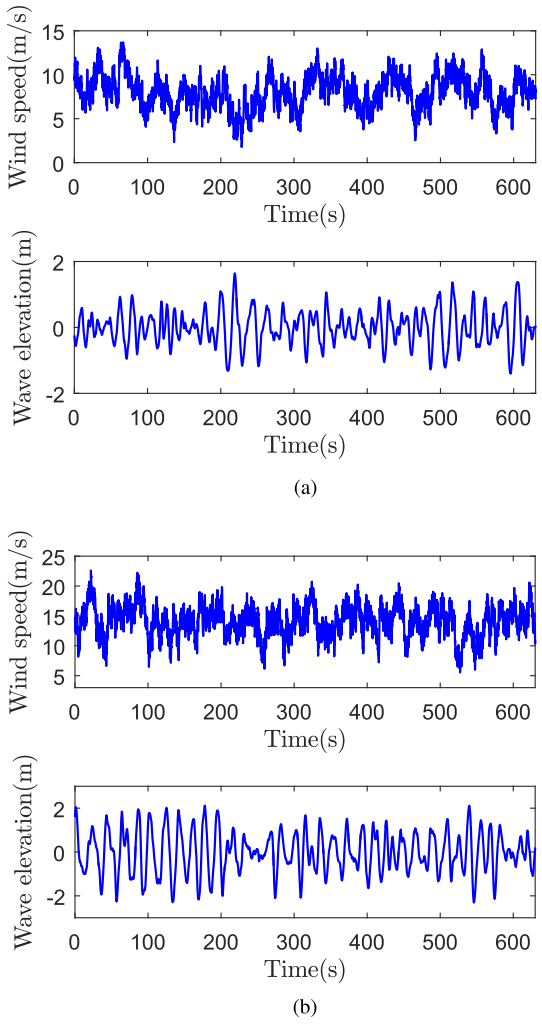


FIGURE 4. Wind and wave conditions. (a) Below rated. (b) Above rated.

the simulation files in this work onto the Matlab Central file exchange server.[1]

A fixed time step 0.0125s was used for all the simulations. In the training process, simulations for 200s long, including 16000 sets of data for each test, have been used to train the learning model for one faulty scenario, where each specific fault was both activated and deactivated for 5 times. While in the testing process, simulations last 630s long, and two types of design load cases, as shown in Fig. 4, were considered for each fault, i.e. below and above rated conditions. The wind data were generated by the NREL's Turbsim [38], where Kaimal spectra and the power law exponent of 0.14 are used. The mean wind speed at the reference height 90m is set as 8 m/s and 14 m/s, respectively, representing the below and above rated cases, while the normal turbulence intensity is set as 18% (8 m/s case) and 15%(14 m/s case). Wave data used in this work were generated by HydroDyn within FAST, where the JONSWAP spectrum is utilized to generate the stochastic wave inputs. The significant wave height is set as 2.3 m (8 m/s case) and 3.7 m (14 m/s case), and the peak spectral period is defined as 14s.

[1] https://cn.mathworks.com/matlabcentral/fileexchange/65213-data-driven-design-for-wind-turbine-fault-detection

B. SIMULATION RESULTS AND ANALYSIS

Parts of the wind turbine FD simulation results are shown in Fig. 5-6, which illustrate the fault prediction time series for three types of wind turbines as faults are activated and deactivated. Notice that FD results for fault 1 and 10 are not shown here, since no sensible results have been found for these two fault scenarios. This is because the produced learning models have poor generalization abilities due to lack of appropriate features, although they have good performance on the training datasets.

Firstly, it can be observed that classifiers have better performance with regards to sensor faults compared with actuator faults. Particularly, the proposed approach is good at detecting fault 4, since the constructed feature, i.e. the first and second derivatives of pitch angle sensor measurement, has an immediate response to the fault signal. In comparison, the performance of actuator fault classifiers are a bit worse, especially fault 7 and 8 in the above rated condition. Regarding fault 7, transfer function of the pitch actuator is gradually changed due to the slow growing air content, which will lead to a 5-20s delay in the FD process. While for fault 8, no matter the pitch actuator fault is abruptly introduced or eliminated, the constructed feature, i.e. the pitch angle difference between blade 1 and 3, will have an impulse response sounding the fault alarm.

Secondly, it can be easily found that some of the FD results have a long time delay for several seconds after the fault deactivation, such as fault 2, 5, 6 and 9. The statistical dispersion of samples is considered to be the main reason. The optimal classifier will be closer to the normal samples with a low-dispersion than those dispersed fault samples. In these fault scenarios, due to the first-order filter, fault alarm will always have a shorter delay in the fault activation transient process than that in the fault deactivation procedure.

Thirdly, wind turbine faults are relatively easier to be detected in the above rated conditions. It can be seen that unstable prediction output leads to many false detections for fault 5 in the below rated condition as shown in Fig. 5(d). The signal-to-noise ratio is considered to be the key reason for the error prediction, where the power output below rated is lower than that above rated. This explanation may also apply to fault 9 as the generator torque is closely related with power output. It is even more serious for fault 7 and 8, which totally fail in the below rated conditions when blade pitch actuators are kept still.

Fourthly, it is worth mentioning that the hit rate for fault 7 and 8 in the below rated condition is meaningless since there is no blade pitch in this region. Although their classifiers can be trained with low bias results, they do not perform well on the test data as shown in Fig. 6(b)-(c). This is because the designed classifier might wrongly fit unrelated features, so that prior knowledge is required on this kind of occasions.

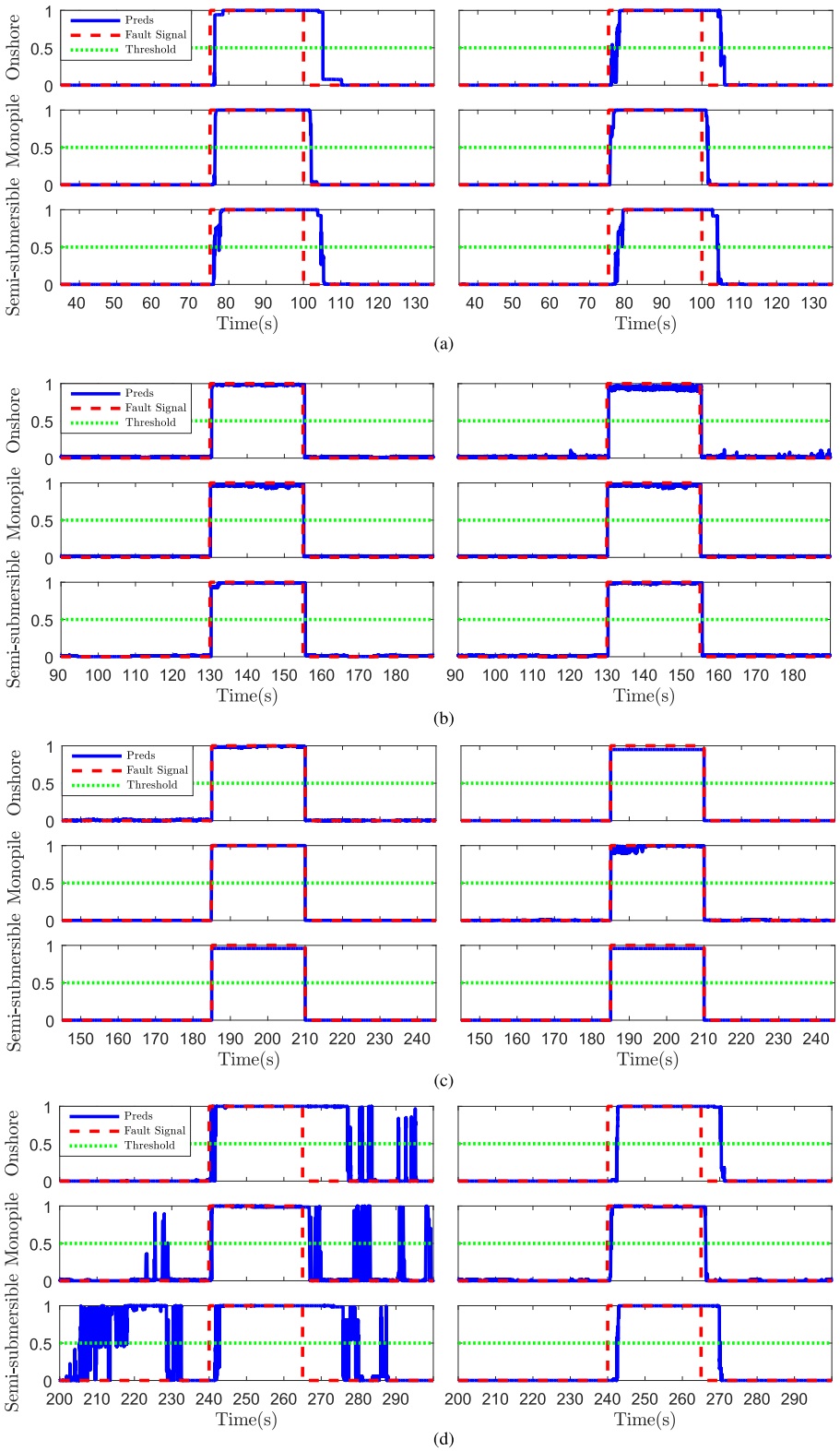
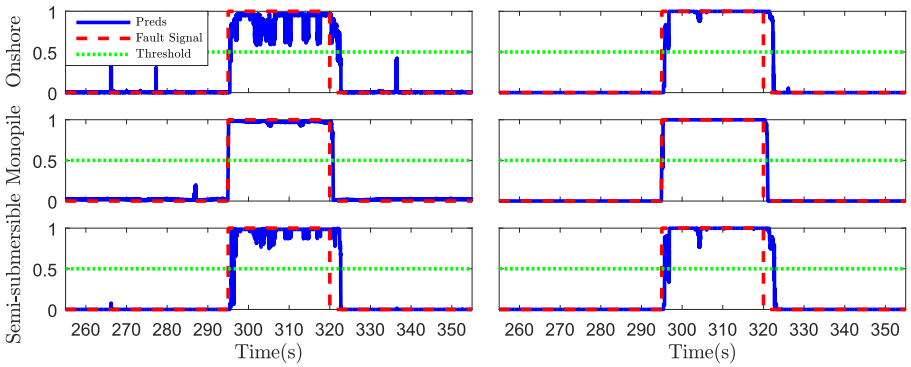


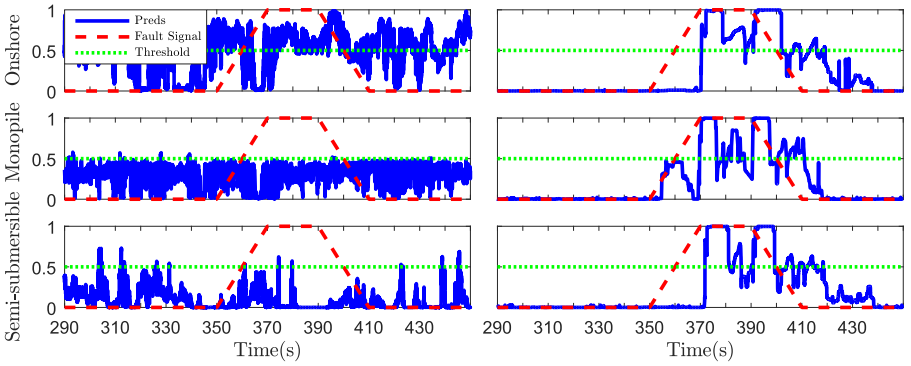
FIGURE 5. Fault detection results for fault 2-5 (left: below rated, right: above rated). (a) Fault 2. (b) Fault 3. (c) Fault 4. (d) Fault 5.

Fifthly, for different types of wind turbines, the performance of fault classifiers are very close. It kind of shows the robustness of the proposed method as it applies for both onshore, offshore, and even floating wind turbines.

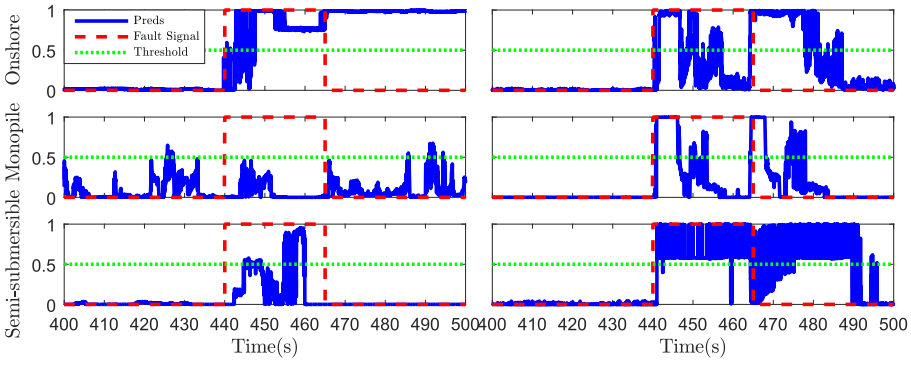
Statistics of the wind turbine FD simulation results and the comparison with different learning models are listed in Table 3. The labels "1st", "2nd" and "3rd" refer to the top three features from the importance ranking by RF model.



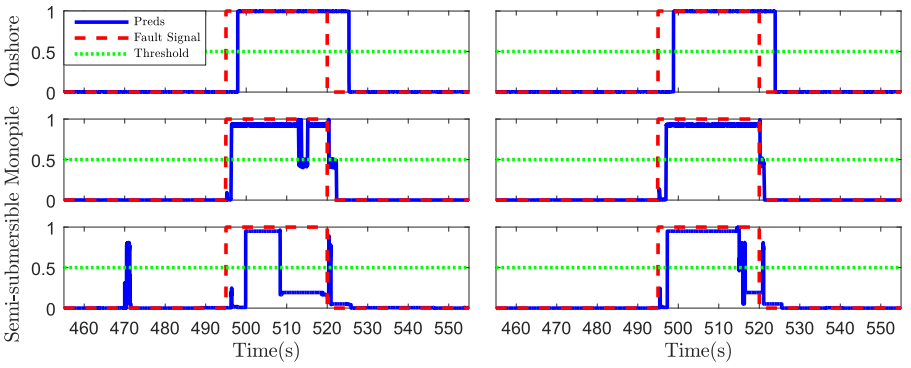
(a)



(b)



(c)



(d)

FIGURE 6. Fault detection results for fault 6-9 (left: below rated, right: above rated). (a) Fault 6. (b) Fault 7. (c) Fault 8. (d) Fault 9.

"CF" refers to "constructed feature." "g" and "r" denote the constructed features using the rotor speed and generator speed, respectively. "HR" in the table represents the "hit rate."

Note that RF is used to perform feature ranking in this work, while RF is still a complete ensemble learning method, which is also able to perform data classification. Therefore, the prediction results of RF are also listed in Table 4 as a

TABLE 3. Top-ranking features.

No.	1st	2nd	3rd
2	$\ddot{x}_Y(\alpha$=0.002)	$\ddot{x}_Y(\alpha$=0.004)	$\beta_{1,m}$
3	$\eta_{r,g}(\alpha$=0.02)	$\eta_{r,g}$	$\beta_{2,m}$
4	$\beta'_{1,m}$	$\beta_{1,m}$	$\omega_{r,m}$
5	$\Delta P_2(\alpha$=0.002)	$\Delta P_1(\alpha$=0.002)	$\Delta P_1(\alpha$=0.003)
6	$\varphi''(\alpha$=0.008)	$\varphi''(\alpha$=0.012)	$\beta_{3,m}$
7	$(\beta_{1,m}-\beta_{2,m})(\alpha$=0.001)	$(\beta_{1,m}-\beta_{2,m})(\alpha$=0.004)	$(\beta_{1,m}-\beta_{2,m})(\alpha$=0.008)
8	$(\beta_{1,m}-\beta_{3,m})(\alpha$=0.0004)	$(\beta_{1,m}-\beta_{3,m})(\alpha$=0.002)	$\beta_{3,m}$
9	$\Delta\tau(\alpha$=0.002)	$\Delta\tau(\alpha$=0.012)	$\beta_{1,m}$

TABLE 4. Wind turbine fault detection hit rate statistics.

No	Below rated			Above rated		
	XGBoost with RF	RF	SVM	XGBoost with RF	RF	SVM
2	0.9819	0.9687	0.9215	0.9823	0.9815	0.8689
3	0.9938	0.9682	0.8967	0.9964	0.9887	0.9025
4	0.9999	0.9999	0.9611	0.9999	0.9999	0.9481
5	0.9515	0.9501	0.7853	0.9891	0.9891	0.9094
6	0.9826	0.9754	0.7673	0.9864	0.9849	0.8026
7	0.4932	0.4308	0.4194	0.9097	0.9087	0.7668
8	0.5423	0.4752	0.5495	0.8709	0.8683	0.7869
9	0.9585	0.9417	0.9068	0.9620	0.9533	0.8916

comparison. Besides, as a popular data-driven classifier design approach, SVM is applied as well in the comparison study. In the SVM classifier design process, model training and testing are also based on the scikit-learn package in python 3.0, and the parameter C is set to 1. Meanwhile, the kernel function are set to 'radial based function'.

It can be seen from the table that the classifier based on RF has a much higher hit rate than SVM, which are both using all the features as training inputs. This is because the ensemble learning methods are able to mitigate overfitting, which will produce better classification results when dealing with high-dimensional inputs.

Similarly, the XGBoost with RF method also has much better performance than SVM, while it has slightly higher hit rate than RF. Particularly, XGBoost are trained only with the three top-ranking features given by RF, instead of all the features for SVM and RF. This has again demonstrated that overfitting is a heavy problem while dealing with high-dimensional features, and proper feature selecting process is beneficial for classification improvement whether using automatic feature ranking method or prior knowledge. In fact, SVM based classifier will also has good FD result if proper features are selected.

V. CONCLUSIONS

In this paper, a data-driven wind turbine FD model based on XGBoost and RF is presented. There are four main steps in designing the classifier. Firstly, feature analysis is conducted in order to find the most relevant features for each specific fault. Secondly, signals are filtered using the first-order filters with proper coefficients to reduce noise disturbances. Thirdly, in order to select the three most relevant features, RF is used to perform feature ranking with the out-of-bag estimates. At last, FD classifiers are trained using the XGBoost model with the three selected top-ranking features.

Numerical simulations have been conducted for both onshore and offshore wind turbines in different operating conditions. The results have shown that the proposed design is robust to different wind turbine models including offshore ones in various working conditions. Besides, it is observed that the designed classifier has better performance with regards to sensor faults compared with actuator faults. Moreover, the designed fault classifier in most of the faulty scenarios is more accurate as the wind turbine works in the above rated condition. Statistics of the simulation results demonstrate that, compared with SVM, the proposed ensemble learning method has strong anti-overfitting ability, which has significant improvement when dealing with multi-dimensional signals. Additionally, with the feature ranking process, XGBoost is able to give better FD results than the RF method.

In future works, more comprehensive training data in various wind turbine working conditions should be used and tested, and more faulty scenarios should be well taken into account in the wind turbine FD design, including fault 1 and 10 as well as other unconsidered faults.

REFERENCES

[1] Global Wind Energy Council. (Apr. 2017). *Global Wind Report 2016*. [Online]. Available: gwec.net/publications/global-wind-report-2/

[2] B. K. Sahu, "Wind energy developments and policies in China: A short review," *Renew. Sustain. Energy Rev.*, vol. 81, pp. 1393–1405, Jan. 2018.

[3] N. Akbari, C. A. Irawan, D. F. Jones, and D. Menachof, "A multi-criteria port suitability assessment for developments in the offshore wind industry," *Renew. Energy*, vol. 102, pp. 118–133, Mar. 2017.

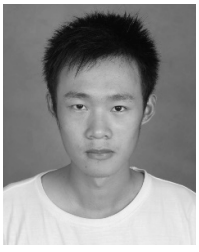
[4] R. Damiani, A. Ning, B. Maples, A. Smith, and K. Dykes, "Scenario analysis for techno-economic model development of U.S. offshore wind support structures," *Wind Energy*, vol. 20, no. 4, pp. 731–747, 2017.

[5] C. Mone *et al.*, "2015 cost of wind energy review," Nat. Renew. Energy Lab., Golden, CO, USA, Tech. Rep. NREL/TP-6A20-66861, 2015.

[6] J. M. P. Pérez, F. P. G. Márquez, A. Tobias, and M. Papaelias, "Wind turbine reliability analysis," *Renew. Sustain. Energy Rev.*, vol. 23, pp. 463–472, Jul. 2013.

[7] X. Zhang *et al.*, "Floating offshore wind turbine reliability analysis based on system grading and dynamic," *J. Wind Eng. Ind. Aerodyn.*, vol. 154, pp. 21–33, Jul. 2016.
[8] J. Carroll, A. McDonald, and D. McMillan, "Failure rate, repair time and unscheduled O&M cost analysis of offshore wind turbines," *Wind Energy*, vol. 19, no. 6, pp. 1107–1119, 2016.
[9] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, "Fault-tolerant control of wind turbines: A benchmark model," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1168–1182, Jul. 2013.
[10] T. Wu, F. Li, C. Yang, and W. Gui, "Event-based fault detection filtering for complex networked jump systems," *IEEE/ASME Trans. Mechatronics*, pp. 1–8, 2017.
[11] Z. Hameed, Y. S. Hong, S. H. Ahn, and C. K. Song, "Condition monitoring and fault detection of wind turbines and related algorithms: A review," *Renew. Sustain. Energy Rev.*, vol. 13, no. 1, pp. 1–39, 2009.
[12] P. Odgaard, "Unknown input observer based scheme for detecting faults in a wind turbine converter," *IFAC Proc. Vol.*, vol. 42, no. 8, pp. 161–168, 2009.
[13] X. Zhang, Q. Zhang, S. Zhao, R. Ferrari, M. M. Polycarpou, and T. Parisini, "Fault detection and isolation of the wind turbine benchmark: An estimation-based approach," *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 8295–8300, 2011.
[14] H. Sanchez, T. Escobet, V. Puig, and P. F. Odgaard, "Fault diagnosis of an advanced wind turbine benchmark using interval-based ARRs and observers," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3783–3793, Jun. 2015.
[15] S. Dey, P. Pisu, and B. Ayalew, "A comparative study of three fault diagnosis schemes for wind turbines," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 1853–1868, Sep. 2015.
[16] S. Yin, H. Gao, J. Qiu, and O. Kaynak, "Fault detection for nonlinear process with deterministic disturbances: A just-in-time learning based data driven method," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3649–3657, Nov. 2016.
[17] S. Yin, G. Wang, and H. R. Karimi, "Data-driven design of robust fault detection system for wind turbines," *Mechatronics*, vol. 24, no. 4, pp. 298–306, 2014.
[18] S. Yin, G. Wang, and H. J. Gao, "Data-driven process monitoring based on modified orthogonal projections to latent structures," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 4, pp. 1480–1487, Jul. 2016.
[19] I. V. de Bessa *et al.*, "Data-driven fault detection and isolation scheme for a wind turbine benchmark," *Renew. Energy*, vol. 87, pp. 634–645, Mar. 2016.
[20] V. Pashazadeh, F. R. Salmasi, and B. N. Araabi, "Data driven sensor and actuator fault detection and isolation in wind turbine using classifier fusion," *Renew. Energy*, vol. 116, pp. 99–106, Feb. 2018.
[21] N. Sheibat-Othman, S. Othman, M. Benlahrache, and P. F. Odgaard, "Fault detection and isolation in wind turbines using support vector machines and observers," *Proc. IEEE Amer. Control Conf.*, Jun. 2013, pp. 4459–4464.
[22] J. Zeng, D. Lu, Y. Zhao, Z. Zhang, W. Qiao, and X. Gong, "Wind turbine fault detection and isolation using support vector machine and a residual-based method," in *Proc. IEEE Amer. Control Conf.*, Jun. 2013, pp. 3661–3666.
[23] B. Tang, T. Song, F. Li, and L. Deng, "Fault diagnosis for a wind turbine transmission system based on manifold learning and Shannon wavelet support vector machine," *Renew. Energy*, vol. 62, pp. 1–9, Feb. 2014.
[24] W. Liu, Z. Wang, J. Han, and G. Wang, "Wind turbine fault diagnosis method based on diagonal spectrum and clustering binary tree SVM," *Renew. Energy*, vol. 50, pp. 1–6, Feb. 2013.
[25] S. U. Jan, Y. D. Lee, J. Shin, and I. Koo, "Sensor fault classification based on support vector machine and statistical time-domain features," *IEEE Access*, vol. 5, pp. 8682–8690, May 2017.
[26] A. Dutta and P. Dasgupta, "Ensemble learning with weak classifiers for fast and reliable unknown terrain classification using mobile robots," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 11, pp. 2933–2944, Nov. 2017.
[27] C. Lindner, P. A. Bromiley, M. C. Ionita, and T. F. Cootes, "Robust and accurate shape model matching using random forest regression-voting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1862–1874, Sep. 2015.
[28] J. Jonkman and M. Buhl, Jr., "FAST user's guide," Nat. Renew. Energy Lab., Golden, CO, USA, Tech. Rep. NREL/EL-500-38230, 2005.
[29] P. F. Odgaard and K. E. Johnson, "Wind turbine fault detection and fault tolerant control—An enhanced benchmark challenge," in *Proc. IEEE Amer. Control Conf.*, Jun. 2013, pp. 4447–4452.
[30] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-MW reference wind turbine for offshore system development," Nat. Renew. Energy Lab., Golden, CO, USA, Tech. Rep. NREL/TP-500-38060, 2009.
[31] A. Robertson *et al.*, "Definition of the semisubmersible floating system for phase II of OC4," Nat. Renew. Energy Lab., Golden, CO, USA, Tech. Rep. NREL/TP-5000-60601, 2014.
[32] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[33] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognit. Lett.*, vol. 31, no. 14, pp. 2225–2236, 2010.
[34] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[35] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.
[36] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.
[37] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[38] B. J. Jonkman, "TurbSim user's guide: Version 1.50," Nat. Renew. Energy Lab., Tech. Rep. NREL/TP-500-46198, 2009.

DAHAI ZHANG was born in China in 1981. He received the Ph.D. degree from Zhejiang University in 2010. From 2011 to 2013, he was a Post-Doctoral Researcher with Zhejiang University and Lancaster University, respectively. He has been an Associate Professor with the Ocean College, Zhejiang University, since 2013. He has been the PI of various research projects in regards to modeling, design, and measurements of renewable energy electrical machines, such as wind turbines, tidal current turbines, and wave energy converters.



LIYANG QIAN received the bachelor's and master's degrees in ocean engineering and technology from Zhejiang University in 2015 and 2018, respectively. His research interests include machine learning, fault diagnosis, and fault tolerant control of offshore wind turbines.

BAIJIN MAO received the bachelor's degree in mechanical design, manufacturing, and automation from Huazhong Agricultural University in 2017. He is currently pursuing the master's degree in ocean engineering and technology with Zhejiang University. His research interests mainly focus on advanced control and load reduction of floating offshore wind turbines.

CAN HUANG received the bachelor's degree in thermal and dynamic engineering from the Agricultural University of Hebei in 2009, and the Ph.D. degree in fluid dynamics from the Beijing Institute of Technology in 2016. He holds a postdoctoral position with Zhejiang University. His current research interests include mesh-less methods, computational fluid dynamics, fluid- structure interaction, and wave and wind energy converter.

BIN HUANG received the bachelor's degree from Sichuan University in 2008 and the Ph.D. degree from Zhejiang University in 2013. He was an Assistant Professor with the Kyushu Institute of Technology from 2014 to 2016. He joined Zhejiang University and has been promoted to Associate Professor since 2018. His current research interests include theoretical design, CFD analysis, and optimization of hydraulic turbines.

YULIN SI (M'16) received the bachelor's and master's degrees in control science and engineering from the Harbin Institute of Technology in 2009 and 2011, respectively, and the Ph.D. degree in offshore wind energy from the University of Agder, Norway, in 2016. He joined Zhejiang University as an Assistant Professor. His current research interests include fault diagnosis and fault tolerant control, advanced control design, and load reduction of offshore wind turbines.