

RESEARCH ARTICLE

Deep Learning Approach for Sensor Data Prediction and Sensor Fault Diagnosis in Wind Turbine Blade

WAI-XI LIU¹, (Member, IEEE), RUI-PENG YIN¹, AND PING-YU ZHU²¹School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China²School of Mechanical and Electric Engineering, Guangzhou University, Guangzhou 510006, China

Corresponding author: Wai-Xi Liu (liuwaixi@sina.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62272113, Grant 61872102, and Grant 61972104; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515012306; and in part by the Guangzhou Key Laboratory of Software-Defined Low Latency Network, China, under Grant 202102100006.

ABSTRACT Monitoring the state of wind turbine blades in real-time using sensors is crucial for early fault diagnosis. Several studies have been conducted to predict the failure of wind turbine blades based on data measured by sensors. These methods rely on accuracy of the sensor-monitoring data; even minor abnormalities can lead to misjudgment of the blade condition and cause serious consequences in service. Nevertheless, self-diagnosing schemes for sensor faults are less researched. The data measured by all sensors on the same wind turbine blade constitutes a spatiotemporal joint distribution dataset, which forms a data correlation pattern. Therefore, this paper proposes a sensor fault self-diagnosing scheme that does not depend on any labeled fault data. First, a sensor data prediction model based on deep learning is built by mining the inherent relevance between sensors. Second, a sensor fault is detected when the residual between the measured sensor value and the predicted value exceeds the control limit. The experimental results for a real-world wind turbine blade show that the model has good prediction and fault diagnosis performance.

INDEX TERMS Deep learning, fault diagnosis, prediction, spatiotemporal, wind turbine.

I. INTRODUCTION

With the increasing size of wind turbine blades, detecting technical faults that require unscheduled maintenance is becoming highly important [1]. An accurate and early detection of faults in wind turbine blades plays a significant role in modern wind farm maintenance programs [2]. Thus, many sensors are used to monitor the state of wind turbine blades in real-time. Recently, fiber Bragg-grating (FBG) sensors have been introduced to monitor wind-turbine blades. Furthermore, data measured by sensors is used in some methods, such as physics-based, artificial intelligence (AI)-based, stochastic-based, and hybrid prognostics methods, to predict the failure of a wind turbine blade [3], [4].

Machine learning has been previously applied in wind turbine fault-detection contexts. However, this has traditionally

been restricted to low-dimensional supervisory control and data acquisition (SCADA)-based data [1], [2], [5], [6], [7]. Generally, they depend on sufficient labeled fault data to achieve fully supervised learning. However, these labeled data are typically difficult to obtain from operational data, such as simulated data [5] and expert-labeled vibration data [2].

The earlier works [8], [9], [10], [11], [12], [13], [14], [15] focused on detecting wind turbine blade faults using sensors; however, the sensor itself could fail. Any slight abnormalities of the sensor can lead to error in monitoring data causing misjudgment of the rotor blade condition and even serious consequences in service. However, the self-diagnosis of sensor faults is still less researched.

In general, many sensors are deployed on wind turbine blades. These sensors periodically measure data, and the data measured by all sensors on the same blade constitute a spatiotemporal joint distribution dataset [16]. In addition,

The associate editor coordinating the review of this manuscript and approving it for publication was Yu Wang¹.

the position of the data measured by each sensor among the whole spatiotemporal joint distribution dataset of a single wind turbine blade is relatively fixed. In other words, there is a fixed relationship between the data of each sensor, which forms a data correlation pattern.

Recently, deep learning [17], [18], [19] has attracted significant academic and industrial interest. In deep learning, large amounts of unlabeled data are initially utilized to extract inherent features by pre-training a multilayer neural network. Then, labeled data is used to slightly modify the learned features for supervised fine-tuning. Thus, amounts of structure in the data can be determined.

Therefore, by mining the data correlation pattern mentioned above using deep learning, this paper proposes a scheme to predict sensor data and diagnose sensor faults in wind turbine blades, referred to as DL-P. Unlike other works [10], [12], DL-P uses the inherent relevance between sensors to predict sensor data without depending on sufficient labeled fault data. Full high-dimensional multi-sensors spatiotemporal data can be directly fed to the deep learning model without requiring detailed feature engineering.

This study aims to determine whether deep learning embedded in a large-scale monitoring system can learn from sensor data and potentially provide a fully automatized scalable fault diagnostic solution. The main contributions of this study are as follows:

- (a) This study proposes a fault self-diagnosing scheme that does not depend on any labeled fault data for the sensor fault as opposed to the wind turbine blade fault. A sensor data prediction model based on deep learning was built by mining the inherent relevance between the sensors. A sensor fault can be detected by observing the residual between the measured sensor value and the predicted value.
- (b) Under the real-world wind turbine blade dataset, DL-P showed good prediction performance and fault diagnosis performance.

The remainder of this paper is organized as follows: related work is presented in Section II; Section III introduces the background; Section IV presents the details of DL-P; in Section V, we exhibit the experimental results; Section VI presents the discussion for practical issues; and Section VII concludes the paper.

II. RELATED WORK

A notable number of methods aim to forecast the remaining useful life (RUL) of wind farm assets. However, they have several advantages and disadvantages. Physics-based prognostics [9], [13] aim to build mathematical models, such as spall progression and crack growth, to describe the physics of failure modes. If the physics of the models remain consistent across components, then such prognostics can offer the most accurate predictions with fewer data compared with data-driven techniques. However, physics-based models are defect-specific and complex to build [3], [4]. The AI-based

prognostics [10], [12] can model complex and nonlinear systems. A large amount of data over a wide range of operating conditions is required to train the prognostic AI model to achieve reasonable prediction accuracy. However, in practice, data are usually limited, especially for complex systems [2], [5], [20]. The stochastic-based prognostics [11], [14] are robust because of their capability in modeling the uncertainty inherent in the prediction horizon of wind-turbine components. The hybrid prognostic models [8], [15] use a combination of various prognosis methods and are the leading tool in prognostics because of its higher accuracy than individual prognosis methods.

Fault identification is typically used to extract fault features from a dataset [21], [22]. Du et al. [23] provided a comprehensive review of state-of-the-art damage detection techniques for wind turbine blades, including the most updated methods based on strain measurement, acoustic emission, ultrasound, vibration, thermography, and machine vision. Liu et al. [24] proposed a multidimensional kernel-domain spectrum (MD-KDS) method in which bispectrum and threshold processing is used to establish the kernel domain spectrum of the known category data. The minimum discrimination function value output from the MD-KDS recognizer reflects the data fault types to be recognized.

Data-driven fault classification has also been a major research topic in recent years [25]. Simani et al. [26] proposed a data-driven fault indicator to solve the problem of early fault detection and isolation. Li et al. [27] suggested a new data-driven methodology for classifying and predicting turbine faults based on Gaussian process classifiers, which eliminates any assumptions about the structural relationship between inputs and output.

The introduction of intelligent fault judgment and health-monitoring technology can effectively reduce operating costs while improving maintenance management. Xu et al. [28] developed a novel method by integrating the convolutional neural networks (CNNs) with the variational mode decomposition algorithms. This method directly processes raw vibration signals without artificial experience or manual intervention to realize the fault diagnosis of rolling bearings in an end-to-end manner. However, the feasibility of using other components requires further evaluation. Pang et al. [29] proposed a new automatic fault diagnosis method for wind turbines—a fault diagnosis framework was constructed, and data on the vibration status of the collected wind turbines were processed and used for fault diagnosis. The signal is analyzed in the time-frequency domain, and then the internal dynamic information characteristics of the signal are extracted. Lei et al. [30] presented a novel LSTM-based method for fault diagnosis of time-series signals. Their method eliminates the dependence on signal-processing technology by taking time-domain raw signals as input. The performance of the proposed framework was achieved by applying a CNN as the preparation function to extract local features. Milad et al. [31] used an algorithm based on a generalized, normalized neural network integration. The hybrid

fault diagnosis method combining principal component analysis (PCA) and a wavelet-based probability density function can accurately detect early faults in the blade, reduce maintenance costs, and improve system availability.

Wang et al. [32] predicted the RUL of a wind turbine bearing by employing a combination of physical knowledge and a statistical model. First, they developed an empirical model for spalling evolution based on the Paris formula. Subsequently, they developed a particle filter using a recursive numerical approach based on the sequential Monte Carlo sampling technique to estimate the posterior PDF of the state.

Wu et al. [33] used wavelet packet energy spectrum analysis and operational modal analysis to detect damage to wind turbine blades. Zhang et al. [34] proposed a data-driven model-based condition monitoring method to detect faults in the gearbox of a wind turbine using a hierarchical extreme-learning machine algorithm.

III. BACKGROUND

CNNs [17], [18] have exhibited significant learning ability in image understanding owing to their unique method of extracting critical features from images. In general, a CNN comprises three types of layers: convolutional, pooling, and fully connected layers; this is excluding the input and output layers.

The **convolutional layer** aims to learn input features and comprises convolution kernels that are used to compute different feature maps. The new feature map can be obtained by first convolving the input with a learned kernel, and then an element-wise nonlinear activation function is applied to the convolved results. A complete feature map can be obtained using different kernels. Let $z_{i,j,k}^l$ denote the feature value at location (i, j) in the k -th feature map of the l -th layer, which is computed as follows:

$$z_{i,j,k}^l = w_k^{lT} x_{i,j}^l + b_k^l \quad (1)$$

where w_k^l and b_k^l are the weight value and bias term of the k -th filter of the l -th layer, respectively, and $x_{i,j}^l$ is the input patch centered at location (i, j) of the l -th layer.

The activation function is used to introduce nonlinearities into the CNN, which are useful for detecting nonlinear features. Sigmoid and rectified linear unit (ReLU) [30] are typical activation functions. We let $h(\cdot)$ denote the activation function and $a_{i,j,k}^l$ denote the activation value of the convolutional feature $z_{i,j,k}^l$, which can be computed as:

$$a_{i,j,k}^l = h(z_{i,j,k}^l) \quad (2)$$

The **pooling layer**, which is usually deployed between two convolutional layers, aims to achieve shift invariance by reducing the resolution of feature maps. Each feature map of the pooling layer is connected to the corresponding feature map of the preceding convolutional layer. We let $\text{pool}(\cdot)$ denote the pooling function, and for each feature map $a_{i,j,k}^l$ that we have:

$$y_{i,j,k}^l = \text{pool}\left(a_{m,n,k}^l\right), \quad \forall(m, n) \in R_{i,j} \quad (3)$$

where $R_{i,j}$ is a local neighbourhood around location (i, j) . Higher-level feature representations can be gradually extracted by stacking the convolutional and pooling layers.

The **fully connected** layer aims to perform high-level reasoning. It takes all neurons in the previous layer and connects them to every neuron of the current layer [35] to generate global semantic information.

The **output layer** is the last layer of the CNN. Different tasks employ different output layers; for example, *Softmax* is commonly employed for classification tasks [26].

We can obtain the best-fitting set of parameters of the CNN by minimizing the loss function for a specific task; the stochastic gradient descent is a common solution to complete such a global optimization.

If there are M desired input-output relations $\{(x^{(n)}, y^{(n)}), n \in [1, \dots, M]\}$, where $x^{(n)}$ is the n th input data, $y^{(n)}$ is its corresponding target label, and $o^{(n)}$ is the output of the CNN. Let $L(\cdot)$ denote the loss function. The loss of the CNN (denoted as \mathcal{L}) can then be computed as follows:

$$\mathcal{L} = \frac{1}{M} \sum_{n=1}^M L(\theta; y^{(n)}, o^{(n)}) \quad (4)$$

where θ represents all the parameters of a CNN.

IV. SYSTEM DESIGN

Typically, many sensors are deployed on the same wind turbine blade. There is a fixed relationship between the data of each sensor forming a data correlation pattern. DL-P uses machine learning to investigate the relationships between the coexisting measurements to reveal such a potentially helpful correlation and synergistic effect, and then accurately predicts the sensor data. As the key, we built a spatiotemporal joint mapping function between sensors to identify this hidden information using a data-driven method. It is assumed that N sensors on the wind turbine blade monitor it continuously for T days at r resolution, referred to as (N, T, r) .

A. WORKFLOW OF DL-P

As shown in Figure 1, DL-P includes the following five main steps:

1) DATA PREPROCESSING

In general, deep learning algorithms benefit from dataset standardization. Thus, data preprocessing, including checking for missing and invalid values, and then deleting the data from duplicated or non-working sensors, should be performed. In addition, we also performed data transformation using min-max normalization.

In the article, the dataset was obtained from an actual fatigue test of a wind turbine blade captured by 29 FBG sensors. This dataset is referred to as dataset-29FBG, and more details can be found in Section V-A. By observing the data of each sensor node, we found that the data collected are sinusoidal and approximately symmetrical about the x-axis, but the data amplitude will change after a period of time.

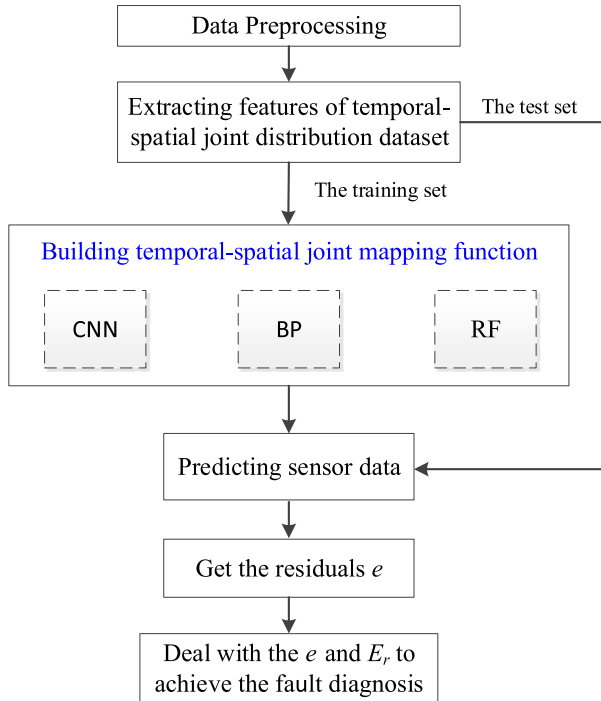


FIGURE 1. System workflow.

As a typical example, the range of data collected from sensor node A2 on October 7, 2018 was $[-600, 700]$, but its range changed to $[-400, 900]$ after a few hours; in other words, there was an overall upward trend. Therefore, we uniformly add 1000 to all values, rather than taking the absolute value, to make all data positive.

We divided the originaldataset-29FBG(detailed in Section V-A) into three classes: data items that are normal constitute the dataset S1 and data items that are faulty constitute the dataset S2, where S1 and S2 comprise 500000 samples and 12000 samples, respectively; data that cannot be determined as normal or faulty form the dataset S3.

To train the prediction model, 50% of S1 is used and named trainingset-0; to test the prediction performance of the prediction model, 30% of S1 is used and named trainingset-1; S2 and the remaining 20% of S1 form testset-1 to test the fault diagnosis performance; that is, the data items in testset-1 have faulty and normal labels.

2) EXTRACTING FEATURES OF TEMPORAL-SPATIAL JOINT DISTRIBUTION DATASET

We set the sensor data to build a temporal-spatial joint distribution dataset. At the i -th time slot, the data of N_1 sensors among N sensors within $[i + 1 - T_1, i]$ time slots are represented by the temporal-spatial joint matrix F_i , as shown in Table 1. Thus, many matrices from many time slots build a temporal-spatial joint distribution dataset. Larger the value of N_1 and T_1 , more the information provided by the temporal-spatial joint mapping function (referred to as the mapping function), and better the performance of the prediction model.

TABLE 1. Temporal-spatial joint matrix F_i at the i -th time slot.

	Number of sensor	1	2	...	N
Time slot	Sensor data				
	$i+1-T_1$				
	$i+2-T_1$				
	...				
	i				

3) BUILDING THE TEMPORAL-SPATIAL JOINT MAPPING FUNCTION

In the second step, the temporal-spatial joint distribution dataset was fed to train the mapping function. The mapping function must be capable of modeling nonlinear relationships. Because different techniques might show different performances in predicting different types of signals, we should select a suitable regression technique according to the properties of the object signal.

The physical understanding of the system should not limit the selection of the input-output relations for the mapping function to find all possible correlations. In contrast, each available signal from sensors $(x_1, x_2, x_3, \dots, x_N)$ should be used as an input for modeling one of the other sensors (x_i) . The mapping functions were evaluated in terms of prediction performance. The prediction performance can be assessed to determine whether a signal is independent if all possible inputs are used.

Thus, our key idea for building the temporal-spatial joint mapping function is as follows: we use the temporal-spatial joint distribution matrix F_i that includes the data of N_1 sensors within T_1 time slots as the input of the mapping function, and the data of another sensor that maps to N_1 sensors, as the output of the mapping function. We call this mapping function as the $N_1 - 1$ function.

Mathematically, for sensor x_i among the N sensors, we select N_1 sensors among the $(N - 1)$ sensors that map to sensor x_i , to build the mapping function between sensor x_i and N_1 sensors. In other words, when each signal acts once as the output, we can build N multiple-input and single-output mapping functions for N sensors. For sensor x_i , the mapping function with an error ε_i can be defined as:

$$x_i = f_i(x \in X \setminus x_i) + \varepsilon_i, \quad X = \{x_1, x_2, \dots, x_N\}, \quad i = 1, 2, \dots, N \quad (5)$$

Thus, the trained function can accurately predict the sensor data based on learned features because the input of the mapping function inherently contains a temporal-spatial joint correlation of the sensor data.

We built three types of temporal-spatial joint mapping functions between sensors, based on random forests (RF), backpropagation neural networks (BP), and CNN, to find

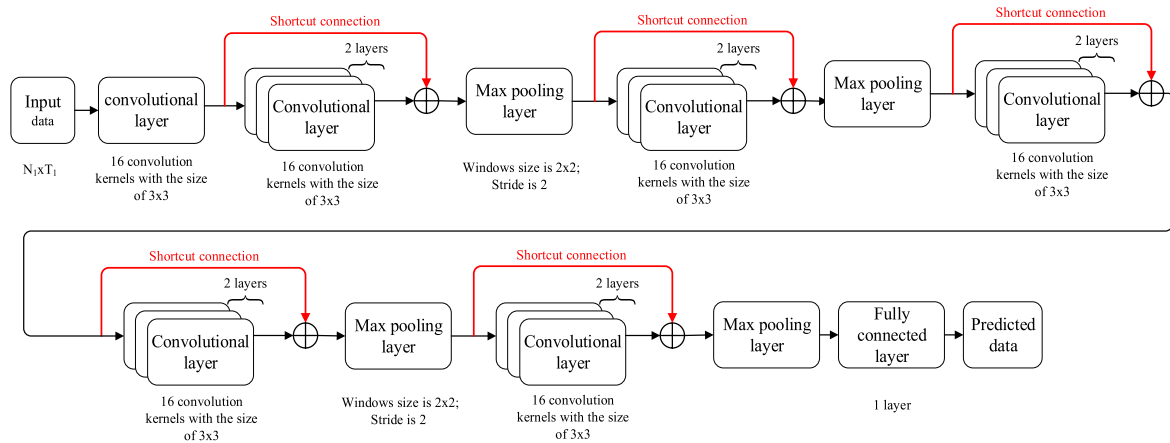


FIGURE 2. DL-P-CNN structure.

correlation data patterns. We refer to them as DL-P-RF, DL-P-BP, and DL-P-CNN, respectively.

4) PREDICTING SENSOR DATA BY USING THE MAPPING FUNCTION

We let the data of N_1 sensors within $[i + 1 - T_1, i]$ time slots be the input of its corresponding trained $N_1 - 1$ function at the i -th time slot for sensor x_i ; this function's output is the prediction of sensor x_i .

5) FAULT DIAGNOSIS

We used the residuals for continuous (online) condition monitoring in the fault diagnosis stage. We let the actual measured sensor value be compared with the predicted value to obtain the residuals e , and the fault is eventually detected when the residual e goes beyond the control limit E_r . It is imperative to determine a reasonable control limit E_r , that is, the lower and upper control limits [36]. Section IV-C provides more details regarding fault diagnosis.

B. DL-P's PREDICTING MODEL BASED ON CNN

As shown in Figure 1, we built three types of temporal-spatial joint mapping functions: DL-P-RF, DL-P-BP, and DL-P-CNN, where DL-P-RF employs 15 C4.5 decision trees and DL-P-BP employs BP neural networks with four hidden layers (the number of hidden layer units is [300 300 50 100]). Figure 2 presents the structure of DL-P-CNN, whose input is the temporal-spatial joint matrix F_i ($N_1 T_1$ matrix) as shown in Table 1. Where each convolutional layer requires batch normalization, and its activation function is a ReLU. Furthermore, we add a shortcut connection [35] to address the well-known problem of vanishing/exploding gradients when we stack more layers to allow DL-P to obtain a better learning ability. Convolution with a kernel size of 1×1 is performed on the shortcut connection to ensure the dimensional consistency of the two data when they are added together.

C. DL-P's FAULT DIAGNOSIS METHOD

As shown in Section V-C, the DL-P can achieve an accurate prediction. Thus, we regarded the predicted value (P) of the model as correct and compared it with the collected value (C) of the sensor. The sensor is considered faulty if the deviation between P and C exceeds the control limit $[d_2, d_1]$. We need to increase the deviation to achieve a better fault diagnosis. Therefore, the residual square sum $(P - C)^2$ was selected to calculate the deviation.

1) DETECTING THE CONTROL LIMIT OF THE RESIDUAL

For trainingset-1, we refer to P_{ij} and C_{ij} as the predicted value and sensing data of node i in the j -th time slot, respectively, $i = 1, 2, \dots, m; j = 1, 2, \dots, n$, and compute γ_{ij} as shown in Equation (6). We call matrix γ_{ij} the fluctuation matrix of the residual value. We can refer to the maximum value and minimum value of matrix γ_{ij} as d_2 and d_1 , respectively. In other words, the control limit E_r of residual e is $[d_2, d_1]$.

$$\gamma_{ij} = (P_{ij} - C_{ij})^2 \quad (6)$$

2) FAULT DIAGNOSIS PROCESS

The fault diagnosis process is illustrated in Figure 3. For testset-1, we obtained the $\gamma_{ij} = (P_{ij} - C_{ij})^2$ through the predicted value P_{ij} and sensing data C_{ij} . If γ_{ij} exceeds $[d_2, d_1]$, node i is judged to be faulty; otherwise, it is judged to be normal.

The judgment result of node i in the j -th time slot is recorded as F_{ij} . When it is faulty, $F_{ij} = 0$, and when it is normal, $F_{ij} = 1$. According to the actual situation, when one of the following two situations occurs, report to the administrator that the node is faulty:

- (1) When F_{ij} is 0 for w_1 times in a row, $w_1 \geq 1$, report that node i has failed.
- (2) In the most recent N_2 collections, when the ratio of the number of $F_{ij} = 0$ to N_2 is greater than w_2 , node i has failed.

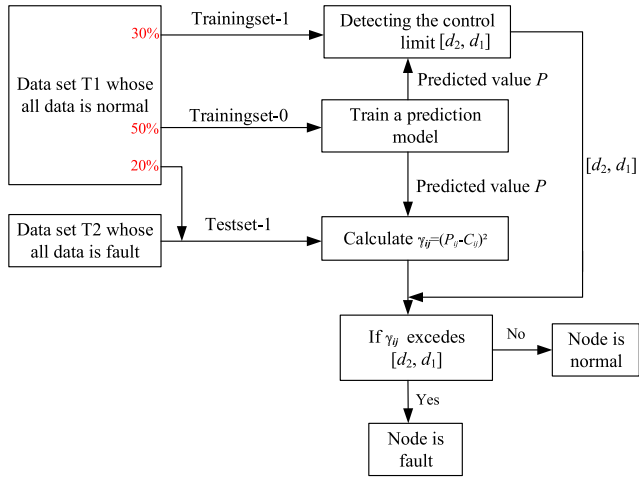


FIGURE 3. Fault diagnosis process.

V. EXPERIMENTS

A. EXPERIMENT SCENARIOS AND SETUP

Experimental data set: to obtain the dataset-29FBG, FBG sensors were implanted into the wind turbine blades and 29 monitoring points were deployed. The locations of the measurement points are shown in Figure 4.

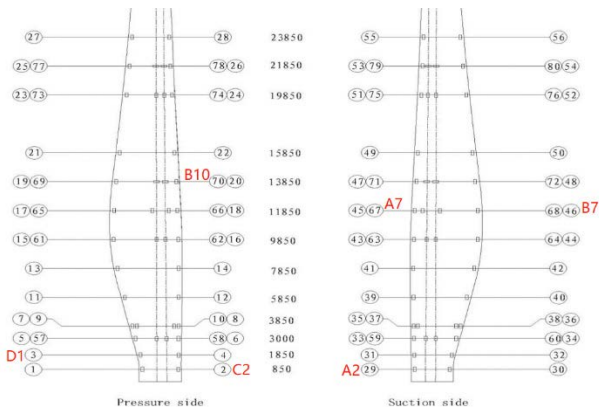


FIGURE 4. Specific location of each monitoring point on the blade.

The wind blade parameters of the fatigue test are as follows: the strain changes at each point when the 56.85 m carbon fiber blade swings and shimmies, the load is 8 tons, and the shimmy frequency is 43.0-43.5 HZ. Continuous real-time monitoring of the various monitoring points of the blade was performed every 1 s for over a month. The monitoring values of all nodes form a dataset that is jointly distributed in time and space, thereby forming a dataset.

In the dataset-29FBG, possible sensor failures include:

- ① Chirp phenomenon appears;
- ② The sensor is continuously interrupted;
- ③ The sensor is intermittently interrupted and normal;
- ④ The fixed block bounce gradually loosens as the blade swings, resulting in a smaller measured value;
- ⑤ The sensor is separated from the blade.

Employing the dataset-29FBG, we selected six sensors (A2, A7, B7, B10, C2, and D1) to evaluate the prediction and fault diagnosis performance of the DL-P. Among them, A7, B7, B10, C2, and D1 work normally, whereas A2 has a chirp phenomenon. We apply a 5-fold cross-validation approach to reduce the error caused by dataset partitioning, and the following experimental results are the average of five experiments.

B. PERFORMANCE METRICS

We used the root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2) to evaluate prediction performance.

Subsequently, we used recall, precision, accuracy, and F1 to evaluate the fault diagnosis performance. We regard faults as positive and normal as negative because we want to focus on detecting the fault. Therefore, we expect a high recall if as many faults as possible are missed. According to the confusion matrix shown in Table 2, we can define recall, precision, accuracy, and F1 as follows:

- Recall (i.e., TPR (True Positive Rate))

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

- Precision

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

- Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

- F1

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2TP}{2TP + FP + FN} \tag{10}$$

TABLE 2. Confusion matrix.

		Predicted	
		Fault	Normal
Actual	Fault	True Positive (TP)	False Negative (FN)
	Normal	False Positive (FP)	True Negative (TN)

C. PREDICTION EXPERIMENT RESULTS

1) PREDICTION PERFORMANCE

As a comparison, Table 3 shows the best results for DL-P-RF, DL-P-BP, and DL-P-CNN when $N_1 = 28$, $T_1 = 3$, and

TABLE 3. The prediction performance of four methods.

	RMSE	MAE	R^2
DL-P-CNN	0.00116	0.000218	0.999989
DL-P-BP	0.003718	0.00069	0.999926
DL-P-RF	0.003122	0.000579	0.999948
Hybrid [8]	0.00155	0.000287	0.999968

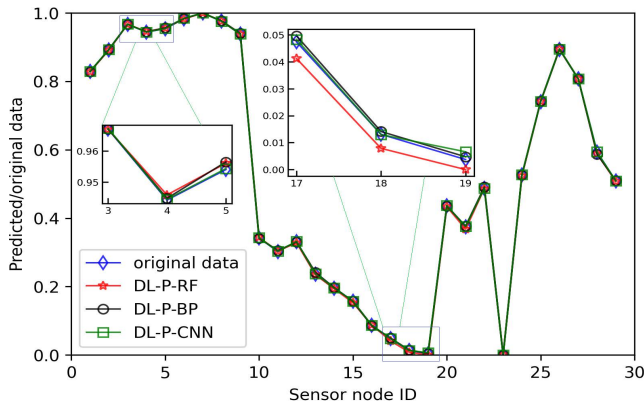


FIGURE 5. A demonstration example of three methods predicting sensor data.

Hybrid [8]. All three methods show excellent performance owing to the proper construction of the temporal-spatial joint mapping function. In particular, DL-P-CNN is the best one because it can achieve lower RMSE and MAE, and higher R^2 than DL-P-RF and DL-P-BP. Furthermore, DL-P-CNN can reduce RMSE and MAE by 25.11% and 24.04% than that of Hybrid [8].

As a demonstration example, Figure 5 shows the result of predicting the sensor data at a one time slot, where the horizontal axis is the sensor node ID, and the vertical axis is the predicted/original data at a one time slot. It is clear that the predicted data almost completely coincided with the original data. Our experiments show that the predicted result at any time slot is similar to that shown in Figure 5. These results further confirm the results presented in Table 3.

2) PREDICTION PERFORMANCE OF DL-P-CNN UNDER DIFFERENT SPATIOTEMPORAL SCALE

For the CNN, the section evaluates the effects of N_1 and T_1 on the prediction performance, and the results are shown in Table 4. From Table 4, we can observe that, as previously analyzed, the larger N_1 and T_1 are, the better the prediction performance is. On the hand, RMSE, MAE, and R^2 are quickly improved when N_1 increase from 9 to 16 to 28. On the other hand, RMSE, MAE, and R^2 are improved when T_1 increase from 1 to 3 to 5. However, too large N_1 and T_1 lead to the model suffering from high complexity and low efficiency. At the same time, T_1 should not be too large because previous traffic patterns make no difference for the current analysis. The experiment results also confirm this inference. The performance improvement increasing from 3 to 5 is far less than from 1 to 3. Thus, the tradeoff between performance and complexity, $T_1 = 3$ is reasonable.

3) PREDICTION PERFORMANCE OF THREE METHODS UNDER DIFFERENT SPACE SCALE

This section evaluates the effect of N_1 on the prediction performance of the three methods when $T_1 = 3$; the results are shown in Figure 6. We can observe that the larger N_1 is,

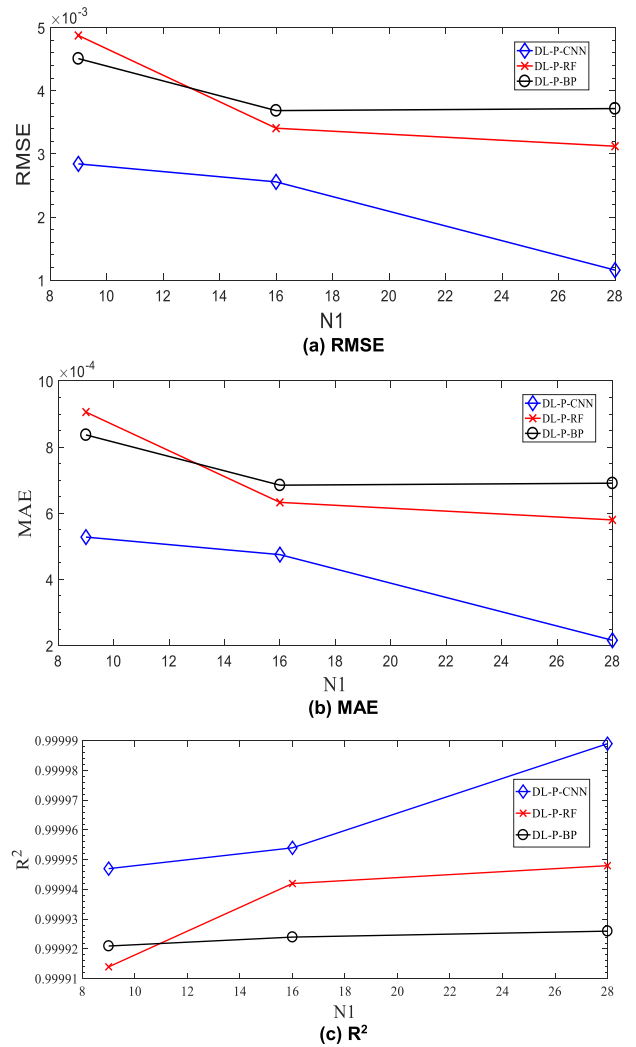


FIGURE 6. Performance of three methods under different space scale.

the better the prediction performance. The RMSE, MAE, and R^2 rapidly improved when N_1 increased from 9 to 16 to 28. However, DL-P-CNN always has an obvious advantage over DL-P-RF and DL-P-BP. In contrast, DL-P-RF outperformed DL-P-BP because $N_1 = 16$.

4) PREDICTION PERFORMANCE OF THREE METHODS UNDER DIFFERENT TIME SCALE

This section evaluates the effect of T_1 on the prediction performance of the three methods when $N_1 = 28$. The results are shown in Figure 7. We can observe that the larger the T_1 value, the better the prediction performance. The RMSE, MAE, and R^2 improved when T_1 increased from 1 to 3 to 5. The prediction performance showed almost no improvement when T_1 was increased from 3 to 5. However, the training and inference times almost doubled when T_1 increased from 3 to 5. Thus, the tradeoff between performance and efficiency, $T_1 = 3$, is reasonable when the DL-P is applied to real sensor monitoring.

TABLE 4. Prediction Performance of DL-P-CNN under different spatiotemporal scale.

T_1	N_1 RMSE, MAE, R^2			
		9	16	28
1		0.004127, 0.000766, 0.999923	0.003371, 0.000626, 0.999935	0.002257, 0.000419, 0.999971
2		0.003532, 0.000656, 0.999932	0.003004, 0.000558, 0.999939	0.002255, 0.000418, 0.999975
3		0.002842, 0.000528, 0.999943	0.002557, 0.000475, 0.999953	0.00116, 0.000218, 0.999989
5		0.002811, 0.000522, 0.999958	0.001936, 0.000359, 0.999980	0.001154, 0.000214, 0.999993

TABLE 5. DL-P-CNN 's performance under different model parameter.

Pooling layers	Convolutional layers	Connected layers	Activation function	RMSE	MAE	R^2
0	3	1	ReLU	0.00372	0.00071	0.99992
0	7	1	ReLU	0.00249	0.00046	0.99996
2	7	1	ReLU	0.00228	0.00042	0.99997
2	7	1	GeLU	0.00304	0.00056	0.99995

TABLE 6. Fault diagnosis performance of node A2.

Recall	82.05%
Precision	75.16%
Accuracy	83.02%
F1	0.7845
* $[d_2 d_1]$: $d_2 = 0.00595$; $d_1 = 121.8186$	

5) PREDICTION PERFORMANCE OF DL-P-CNN UNDER DIFFERENT MODEL PARAMETER

As shown in Table 5, this section evaluates the effects of the CNN parameter on the prediction performance of DL-P-CNN when $N_1 = 28$ and $T_1 = 1$. We can observe that the RMSE, MAE, and R^2 are improved when the polling and convolutional layers increase. In addition, when the activation function is set as a Gaussian error linear unit (GeLU), the prediction performance of the DL-P-CNN slightly deteriorates. The performance is the best when polling layers are 2, convolutional layers are 7, connected layers are 1, and the activation function is ReLU, whose structure is shown in Figure 3, and this structure is also the default model structure of DL-P-CNN in this paper.

D. FAULT DIAGNOSIS EXPERIMENT RESULTS

According to the method described in Section IV-C, Table 6 lists the fault diagnosis results for node A2. We can see that the recall and accuracy are not very high. Through analysis, we found that there is a drift phenomenon in dataset-29FBG, which causes the control limit range to be too large, resulting in a low recall, that is, the false negative rate of failure is high.

As shown in Figure 8, the interval between the values collected by the sensor is inconsistent, and there is data drift phenomenon because the precision of the system clock crystal used in the fatigue test is low. Where the blue line represents the data obtained when A2 is running stably, and the orange line represents the data obtained by A2 after a period of time. It can be observed that the period of the data has changed. Therefore, we performed the experiment

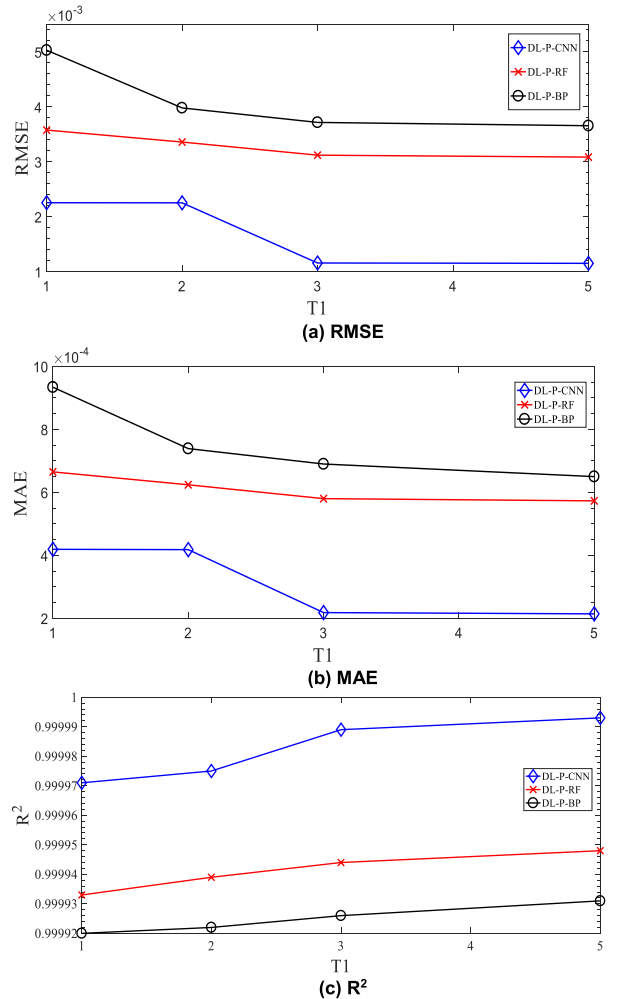


FIGURE 7. Performance of three methods under different time scale.

again after removing the drift phenomenon of the dataset-29FBG. The new control limit and experimental results are listed in Table 7. It can be observed that the control limit range is reduced, and the performance is improved. The recall increased to 89.62%, precision increased to 99.98%, and accuracy increased to 92.86%.

Furthermore, we improved the fluctuation matrix γ_{ij} shown in Equation (6) to Equation (11), to improve the performance. The new control limit and the experimental results are listed in Table 8. The recall has increased to 98.43%, the precision is 90.01%, and the accuracy has also increased to 98.58%. In other words, only a fault of 1.57% was missed, and this

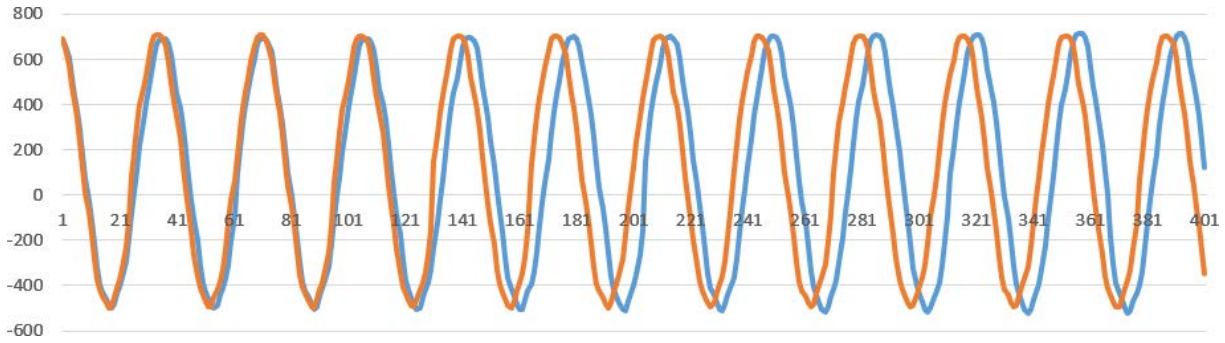


FIGURE 8. Drift phenomenon.

TABLE 7. Fault diagnosis performance of node A2 after removing the drift phenomenon.

Recall	89.63%
Precision	99.98%
Accuracy	92.86%
F1	0.9452

*[d_2 d_1]: $d_2 = 0.01835$; $d_1 = 121.8186$

TABLE 8. Fault diagnosis performance of node A2 from new γ_{ij} .

Recall	98.43%
Precision	90.01%
Accuracy	98.58%
F1	0.9403

*[d_2 d_1]: $d_2 = 0$; $d_1 = 159.9384$

performance was sufficient to satisfy the requirements of practical applications.

$$\gamma_{ij} = \sqrt{\left(P_{ij} - \frac{P_{ij} + C_{ij}}{2}\right)^2 + \left(C_{ij} - \frac{P_{ij} + C_{ij}}{2}\right)^2} \quad (11)$$

VI. DISCUSSION

As mentioned in the Section Introduction, there is a fixed relationship between the sensor data although the sensor works independently, and mining data correlation pattern is the basis of the DL-P. Such a pattern can also reflect the distribution of sensor failures. We define the probability of k sensors being faulty simultaneously to observe this pattern, which is referred to as p_k . p_k is defined as $p_k = G_k/G$ ($k = 1, 2, \dots, m$). Here, for the combined set of S1 and S2, when k sensors fail at the j -th time slot, it is counted as a failure event, and G_k is the total number of occurrences of such a failure event in the entire dataset, and G is the total number of time slots.

Considering the relationship between nodes, the probability of k nodes failing at the same time slot is low; hence, the larger the k , the smaller the p_k . The experimental results shown in Table 9 confirm this inference. It can be observed that the larger k is, the smaller p_k is, and p_k decreases significantly from $k = 0$ to $k = 1$, where $k = 0$ indicates that no fault event occurs.

TABLE 9. Probability of multiple nodes failing simultaneously.

k	Number of sensor nodes failing simultaneously	p_k
0	23807	93.596%
1	1292	5.079%
2	337	1.325%
3	0	0
4	0	0
5	0	0
6	0	0

In short, judging sensor to be faulty according to either Equation (6) or Equation (11) is a supervised learning method. Where we need determine a control limit E_r (i.e., [d_2 , d_1]) by observing lots of training data with labels. However, such a training data with labels is generally rare in real world. Inspired by the observation of Table 9, in the future, we aims to find an unsupervised learning method which does not depend on lots of training data with labels. Our initial idea is to probabilistically determine the sensor faulty according to p_k observed by Table 9, since the probability of k nodes failing simultaneously is low, and this is our future works.

VII. CONCLUSION

To detect errors in the monitoring data caused by faults in the sensor instead of faults in the wind turbine blade, this study proposes a self-diagnosing scheme that does not depend on any labeled fault data. By mining the inherent relevance between sensors, this study builds a deep learning-based prediction model for sensor data prediction and fault diagnosis. For the dataset from real-world wind turbine blades, the experimental results show that the model has a good prediction performance (RMSE = 0.001154, MAE = 0.000214, and $R^2 = 0.999993$) and fault diagnosis performance (a recall of up to 98.43%, an accuracy of up to 98.58%, and a precision of up to 90.01%).

REFERENCES

- [1] G. Helbing and M. Ritter, "Deep learning for fault detection in wind turbines," *Renew. Sustain. Energy Rev.*, vol. 98, pp. 189–198, Dec. 2018.
- [2] M. Bach-Andersen, B. Rømer-Odgaard, and O. Winther, "Deep learning for automated drivetrain fault diagnosis," *Wind Energy*, vol. 21, no. 1, pp. 29–41, 2018.

- [3] M. Rezamand, M. Kordestani, R. Carriveau, D. S.-K. Ting, M. E. Orchard, and M. Saif, "Critical wind turbine components prognostics: A comprehensive review," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9306–9328, Dec. 2020.
- [4] J. Cheng, Y. Yang, X. Li, and J. Cheng, "Adaptive periodic mode decomposition and its application in rolling bearing fault diagnosis," *Mech. Syst. Signal Process.*, vol. 161, Dec. 2021, Art. no. 107943.
- [5] G. Jiang, P. Xie, H. He, and J. Yan, "Wind turbine fault detection using a denoising autoencoder with temporal information," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 89–100, Feb. 2018.
- [6] J. Tautz-Weinert and S. J. Watson, "Comparison of different modelling approaches of drive train temperature for the purposes of wind turbine failure detection," *J. Physics: Conf. Ser.*, vol. 753, Sep. 2016, Art. no. 072014.
- [7] F. Cheng, J. Wang, L. Qu, and W. Qiao, "Rotor-current-based fault diagnosis for DFIG wind turbine drivetrain gearboxes using frequency analysis and a deep classifier," *IEEE Trans. Ind. Appl.*, vol. 54, no. 2, pp. 1062–1071, Mar./Apr. 2018.
- [8] M. A. Djeziri, S. Benmoussa, and R. Sanchez, "Hybrid method for remaining useful life prediction in wind turbine systems," *Renew. Energy*, vol. 116, pp. 173–187, Feb. 2018.
- [9] D. Breteler, C. Kaidis, T. Tinga, and R. Loendersloot, "Physics based methodology for wind turbine failure detection, diagnostics & prognostics," in *Proc. EWEA Annu. Event*, 2015, pp. 1–9.
- [10] J. Deutsch and D. He, "Using deep learning-based approach to predict remaining useful life of rotating components," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 1, pp. 11–20, Jan. 2018.
- [11] J. Herp, M. H. Ramezani, M. Bach-Andersen, N. L. Pedersen, and E. S. Nadimi, "Bayesian state prediction of wind turbine bearing failure," *Renew. Energy*, vol. 116, pp. 164–172, Feb. 2018.
- [12] T. Zhang, S. Liu, Y. Wei, and H. Zhang, "A novel feature adaptive extraction method based on deep learning for bearing fault diagnosis," *Measurement*, vol. 185, Nov. 2021, Art. no. 110030.
- [13] J. Zhu, J. M. Yoon, D. He, and E. Bechhoefer, "Online particle-contaminated lubrication oil condition monitoring and remaining useful life prediction for wind turbines," *Wind Energy*, vol. 18, no. 6, pp. 1131–1149, Jun. 2015.
- [14] M. Rezamand, R. Carriveau, D. Ting, M. Davison, and J. J. Davis, "Aggregate reliability analysis of wind turbine generators," *IET Renew. Power Gener.*, vol. 13, no. 11, pp. 1902–1910, Aug. 2019.
- [15] F. Ding, Z. Tian, F. Zhao, and H. Xu, "An integrated approach for wind turbine gearbox fatigue life prediction considering instantaneously varying load conditions," *Renew. Energy*, vol. 129, pp. 260–270, Dec. 2018.
- [16] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.
- [17] Y. Bengio, I. Goodfellow, and A. Courville, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2016.
- [18] W.-X. Liu, J. Cai, Y. Wang, Q. C. Chen, and J.-Q. Zeng, "Fine-grained flow classification using deep learning for software defined data center networks," *J. Netw. Comput. Appl.*, vol. 168, Oct. 2020, Art. no. 102766.
- [19] W.-X. Liu, J. Zhang, Z.-W. Liang, L.-X. Peng, and J. Cai, "Content popularity prediction and caching for ICN: A deep learning approach with SDN," *IEEE Access*, vol. 6, pp. 5075–5089, 2017.
- [20] Z. Liu, X. Wang, and L. Zhang, "Fault diagnosis of industrial wind turbine blade bearing using acoustic emission analysis," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 9, pp. 6630–6639, Sep. 2020, doi: 10.1109/TIM.2020.2969062.
- [21] Y. Qiu, Y. Feng, and D. Infield, "Fault diagnosis of wind turbine with SCADA alarms based multidimensional information processing method," *Renew. Energy*, vol. 145, pp. 1923–1931, Jan. 2020.
- [22] M. Kordestani, M. Saif, M. E. Orchard, R. Razavi-Far, and K. Khorasani, "Failure prognosis and applications—A survey of recent literature," *IEEE Trans. Rel.*, vol. 70, no. 2, pp. 728–748, Jun. 2021, doi: 10.1109/TR.2019.2930195.
- [23] Y. Du, S. Zhou, X. Jing, Y. Peng, H. Wu, and N. Kwok, "Damage detection techniques for wind turbine blades: A review," *Mech. Syst. Signal Process.*, vol. 141, Jul. 2020, Art. no. 106445.
- [24] W. Liu, "Intelligent fault diagnosis of wind turbines using multi-dimensional kernel domain spectrum technique," *Measurement*, vol. 133, pp. 303–309, Feb. 2019.
- [25] L. Colone, M. Reder, J. Tautz-Weinert, J. J. Melero, A. Natarajan, and S. J. Watson, "Optimisation of data acquisition in wind turbines with data-driven conversion functions for sensor measurements," *Energy Proc.*, vol. 137, pp. 571–578, Oct. 2017.
- [26] S. Simani and C. Turhan, "Fault diagnosis of a wind turbine simulated model via neural networks," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 381–388, 2018.
- [27] Y. Li, S. Liu, and L. Shu, "Wind turbine fault diagnosis based on Gaussian process classifiers applied to operational data," *Renew. Energy*, vol. 134, pp. 357–366, Apr. 2019.
- [28] Z. Xu, C. Li, and Y. Yang, "Fault diagnosis of rolling bearing of wind turbines based on the variational mode decomposition and deep convolutional neural networks," *Appl. Soft Comput.*, vol. 95, Oct. 2020, Art. no. 106515.
- [29] Y. Pang, L. Jia, X. Zhang, Z. Liu, and D. Li, "Design and implementation of automatic fault diagnosis system for wind turbine," *Comput. Electr. Eng.*, vol. 87, Oct. 2020, Art. no. 106754.
- [30] J. Lei, C. Liu, and D. Jiang, "Fault diagnosis of wind turbine based on Long Short-term memory networks," *Renew. Energy*, vol. 133, pp. 422–432, Apr. 2018.
- [31] M. Rezamand, M. Kordestani, R. Carriveau, D. S.-K. Ting, and M. Saif, "A new hybrid fault detection method for wind turbine blades using recursive PCA and wavelet-based PDF," *IEEE Sensors J.*, vol. 20, no. 4, pp. 2023–2033, Feb. 2020.
- [32] J. Wang, Y. Liang, Y. Zheng, R. X. Gao, and F. Zhang, "An integrated fault diagnosis and prognosis approach for predictive maintenance of wind turbine bearing with limited samples," *Renew. Energy*, vol. 145, pp. 642–650, Jan. 2020.
- [33] F. Gao, X. Wu, Q. Liu, J. Liu, and X. Yang, "Fault simulation and online diagnosis of blade damage of large-scale wind turbines," *Energies*, vol. 12, no. 3, p. 522, Feb. 2019.
- [34] P. Qian, D. Zhang, X. Tian, Y. Si, and L. Li, "A novel wind turbine condition monitoring method based on cloud computing," *Renew. Energy*, vol. 135, pp. 390–398, May 2019.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.
- [36] Q. Xu, Z. Fan, W. Jia, and C. Jiang, "Quantile regression neural network-based fault detection scheme for wind turbines with application to monitoring a bearing," *Wind Energy*, vol. 22, pp. 1390–1401, Oct. 2019.



WAI-XI LIU (Member, IEEE) received the Ph.D. degree from Sun Yat-sen University, Guangzhou, China, in 2013. He is currently an Associate Professor with the Department of Electronic and Communication Engineering, Guangzhou University, Guangzhou. He has published more than 25 papers. His research interest includes big data.



RUI-PENG YIN received the B.E. degree from the Hunan University of Technology, in 2020. Currently, he is a Graduate Student with the Department of Electronic and Communication Engineering, Guangzhou University, China. His current research interest includes data center networks.



PING-YU ZHU received the Ph.D. degree from Central South University, China, in 2003. She is currently a Professor with Guangzhou University, China. She has published more than 30 papers. Her research interest includes big data.

...