A PCC-Ensemble-TCN model for wind turbine icing detection using class-imbalanced and label-missing SCADA data

Shenyi Ding, Zhijie Wang, Jue Zhang, Fang Han, Xiaochun Gu and Guangxiao Song

Abstract
Blade icing problems are ubiquitous for wind turbines located in cold climate zones. Data-driven indirect icing detection methods based on supervisory control and data acquisition system have shown strong potential recently. However, the supervisory control and data acquisition data is annotated through manual observation, which will cause the data between normal condition and icing condition to be unlabeled. In addition, the amount of normal data is far more than icing data. The above two issues restrict the performance of most current data-driven models. In order to solve the label missing problem, this article proposes a Pearson correlation coefficient–based algorithm for measuring the degree of blade icing, which calculates the similarity between the unlabeled data and the icing data as its label. Aiming at the class-imbalance problem, this article constructs multiple class-balanced subsets from the original dataset by under-sampling the normal data. Temporal convolutional networks are trained to extract features and make predictions on each subset. The final prediction result is obtained by ensembling the prediction results of all temporal convolutional network models. The proposed model is validated using the actual supervisory control and data acquisition data collected from a wind farm in northern China, and the results indicate that ensuring the consecutiveness and class-balance of the data are quite advantageous for improving the detection accuracy.

Keywords
Wind turbine blade icing detection, Pearson correlation coefficient, ensemble learning, temporal convolutional network, time-series modeling

Date received: 3 January 2021; accepted: 27 September 2021

Handling Editor: Francesc Pozo

Introduction
With the depletion of global fossil energy and the trend of global warming, wind power, as one of the clean and renewable energy sources, has attracted much attention from countries around the world. Wind farms, where wind turbines (WTs) are installed to collect wind energy, are distributed across a wide range of climates, especially cold climates. Cold climate areas are usually characterized by high altitude, low temperature, and high humidity, where WT blades are prone to icing.
Blade icing will severely affect the power output of the WT as well as the life span of equipment, and even endanger personal safety. Therefore, it is of vital significance to realize WT blades icing detection in early stage and activate the de-icing system\(^2,^3\) to remove ice. Recent icing detection methods can be divided into two categories: direct detection methods and indirect detection methods. Direct detection methods rely on additional sensors\(^4\) to detect physical properties changes in blades (such as emissivity, conductivity, and mass) to determine whether there is icing. However, most of the WTs in service do not have those direct detection sensors, and the installation of those sensors is complicated and expensive, so direct ice detection method is only feasible on a few of newly installed WTs.

On the contrary, with the popularization of supervisory control and data acquisition (SCADA) system in wind farms, data-driven icing detection methods have gradually become mainstream.\(^7\) Indirect detection methods usually use machine learning techniques to reveal the inherent connection between data provided by the SCADA system and icing condition. These data include WT state data, environmental data, and WT motion data. Some researchers have established icing prediction models based on traditional machine learning methods.\(^8\)–\(^13\) However, traditional machine learning models rely on feature engineering, which is time-consuming and labor-intensive. Besides, due to the restriction of model size, they often fail to make good use of temporal relationships between data. As an emerging branch of machine learning, deep learning has been widely used in fault diagnosis\(^14\)–\(^18\) and achieves great success because of its outstanding ability to automatically extract effective features from big data. Among them, there is no lack of research in the field of WT icing detection. Liu et al.\(^19\) found that the representative features automatically extracted by a deep autoencoder laid a foundation for detecting icing conditions. The authors also assembled features from different hidden layers of the deep autoencoder model to improve detection accuracy. Yeh et al.\(^20\) combined convolutional neural networks (CNN) and support vector machine to predict long cycle maintenance time of WTs. Yun et al.\(^21\) established a well-behaved icing detection model using the SCADA data of one WT and applied the idea of transfer learning to make the model applicable to more WTs. As a specialist for time-series modeling, recurrent neural networks (RNN) have also been explored in WT fault detection.\(^22\)\(^,^23\) Benefiting from their strong feature extraction and big data processing capabilities, deep learning models have a prominent improvement in detection accuracy over traditional machine learning models.

Nevertheless, two key characteristics of the WT SCADA data that will affect the prediction accuracy are not taken into account by the above deep learning models. The first characteristic is that there exists some unlabeled data in the dataset. At present, the annotation of the dataset mainly relies on human labor. Staff in the wind farm observes the state of a WT at certain moments and record whether it is icing. The disadvantage of this approach is that when the staff first discovers the icing condition, he cannot determine whether the state of the WT between that moment and the last observed non-icing moment is icing or not. Therefore, some data is unlabeled. Since supervised learning requires each data to have a label, almost all present data-driven models only keep labeled data and ignore unlabeled data. There are some potential issues that arise from this practice. First, the dataset will be inconsiderately divided. Usually, the unlabeled data is between normal data and icing data. The performance of a time-series model will be largely affected by the consecutiveness of the dataset. Second, icing data is very precious in SCADA data. The unlabeled data contains important information related to icing conditions. Effective use of unlabeled data helps to better mine the features under icing conditions. The second characteristic of SCADA data is that it is class-imbalanced, which indicates that the amount of normal data is far more than that of icing data. The methods to tackle class-imbalance are divided into two categories: data preprocessing and algorithm optimization. Data preprocessing methods include under-sampling major class data,\(^24\) oversampling minor class data,\(^25\) and generating minor class data.\(^26\) Nevertheless, down-sampling will cause information loss. Over-sampling may give rise to over-fitting problems, and generating minor class data will inevitably introduce noise. Algorithm optimization methods attempt to modify the evaluation metrics and loss function, forcing them to pay more attention to minor class data. Chen et al.\(^27\) noticed the class-imbalance problem in WT SCADA data and proposed a deep neural network based on triplet loss. However, the authors did not make full use of the unlabeled data. We summarize the merits and drawbacks of above-mentioned WT icing detection models based on deep learning techniques in Table 1.

Handling the above two key characteristics of WT SCADA data is essential to establish an accurate icing detection model. To deal with unlabeled data, we present a Pearson correlation coefficient (PCC)-based algorithm for measuring the degree of blade icing. The degree of blade icing is regarded as the label of the unlabeled data. The SCADA data covers a long period of time, and some of the parameters in it are sensitive to time due to factors such as weather and control strategies. Thus, when measuring the degree of icing of the unlabeled data in a certain period of time (i.e. calculating the PCC), comparisons should be made locally instead of globally. Exponential moving average (EMA) is introduced to process normal data and icing
data within a period of time before and after unlabeled data as the comparison standard. We call these data labeled by this algorithm soft-labeled data. Aiming at the class-imbalanced problem, we put forward an ensemble learning method that trains multiple models on multiple subsets and averages the prediction results of all models as the final prediction result. Our ensemble method is a bit different from the classic ensemble learning method-Bagging. Bagging obtains multiple subsets on the original dataset by means of sampling with replacement. The subsets may still suffer from class-imbalance problem. Our method just samples the normal data of the original dataset without replacement, and the soft-labeled data along with the icing data are all allocated to each subset, so each subset is class-balanced. The prediction model chosen in this article is temporal convolutional network (TCN). The advantages of RNN are that all historical inputs will be considered when calculating the output at the current moment, and there is a strict causal relationship between inputs and outputs. TCN takes the advantages of RNN to improve conventional CNN. Thus, TCN has a larger receptive field than conventional CNN under the same convolution kernel. In addition, TCN also guarantees the causality between inputs and outputs. We also use a mixed loss function that combines focal loss and mean square error (MSE) as the loss function. Since focal loss can only be used for classification problems, we use MSE to calculate the prediction error of those soft-labeled data. The final prediction result is obtained by averaging the prediction results of all TCN models trained on each subset. The contributions of this article are listed below:

- Aiming at the label-missing problem in WT SCADA data, we propose a novel PCC-based algorithm. The algorithm measures the similarity between unlabeled data and adjacent labeled data to annotate unlabeled data. Thereby, the integrity of the dataset is guaranteed.
- Aiming at the class-imbalance problem, we put forward an ensemble learning method. By evenly distributing the normal data in the original dataset to each subset and adding all icing data to each subset, multiple class-balanced subsets are constructed. The deep learning models are trained on these subsets, and the final prediction result is determined by averaging the results of all models.
- Deep TCN and a mixed loss function are integrated as the icing detection model, in which the temporal and spatial features of the data are fully considered. The mixed loss function is composed of focal loss and MSE. Our model works directly on raw data and thus no feature extraction stage or extra domain knowledge is required.

The rest of this article is organized as follows. Section “Proposed model” describes the proposed model in detail. Section “Experiment preparation” introduces the method of data preprocessing and model evaluation metrics. Section “Results analysis” analyzes the experiment results and section “Conclusion” summarizes the work in this article.

### Proposed model

Figure 1 describes the whole structure of our WT icing detection model. We first divide the original SCADA data into a training set and a testing set. In the training set, the PCC-based algorithm for measuring the degree of WT blade icing is used to annotate the unlabeled data. Originally, label 0 indicates that the WT is in normal condition, and label 1 indicates that the WT is in icing condition. This is a typical two-class classification problem. Considering that WT blade icing is a gradual process, the condition of the WT corresponding to the unlabeled data is between the normal condition and the icing condition, which can be regarded as a transition state. Specifically, we quantify this condition and use PCC between the unlabeled data and the icing data to measure the degree of blade icing. As mentioned above, some parameters in the SCADA data change greatly over time, so the comparison is performed in a local period of time. For a certain segment of unlabeled data, we selected 60 data points before it and 60 data points after it as the comparison standard. EMA is used to calculate the moving average of normal data and icing data, respectively. After that, we calculate the PCC

| Presented model       | Advantages                                                      | Challenges                                      |
|-----------------------|----------------------------------------------------------------|-----------------------------------------------|
| Ensemble autoencoder  | Utilized hierarchical features                                  | Unlabeled data was not fully utilized          |
| CNN                   | High training speed                                              | Needs consideration on class-imbalanced data  |
| Transfer learning     | Strengthened generalization ability                              | Features of data were not fully exploited      |
| LSTM                  | Utilized temporal relationship between data                      | Needs consideration on class-imbalanced data  |
| TL-DNN                | Dealt with class-imbalanced data                                | Unlabeled data was not fully utilized         |

WT: wind turbine; CNN: convolutional neural network. LSTM: long short-term memory; TL-DNN: triplet loss deep neural network.
between the average normal data and the unlabeled data, and the PCC between the average icing data and the unlabeled data. The two PCC values are processed with softmax function. Finally, we choose the PCC between the unlabeled data and the average icing data after softmax processing as its label. If the label is close to 1, the correlation between the two data is very high, indicating that the icing condition severe; if the label is close to 0, the correlation is low, indicating that the WT is in normal condition. The algorithm is also applied in the testing set. However, in order to ensure the correctness, the prediction results of these soft-labeled data are not included in the calculation of the evaluation metrics. Adding these data is mainly to maintain the consecutiveness of the dataset to improve the prediction results. Correspondingly, the output of the model during the training process becomes the degree of icing (a number between 0 and 1), and the output of the model during the testing process is still the classification result of whether it is icing. Afterward, we divide the normal data in the training set into eight equal parts due to the ratio of normal data to soft-labeled data together with icing data is about 8:1. Each subset contains all soft-labeled data, all icing data, and one part of normal data. Then we get eight class-balanced subsets. The icing detection model, TCN model, is trained on each subset. All eight TCN models have the same structure. Since they are trained on different subsets, their weights are different from each other. For a time-series data of length $T$ with $N$ parameters, we cut a $W \times N$ data segment and feed it into TCN, where $W$ is the window length. We move forward one step each time, so the number of data segments is $T - W + 1$. Thus, the dimension of input becomes $(T - W + 1) \times W \times N$. The dimension of the output of TCN output layer is $(T - W + 1) \times W \times N_f$, where $N_f$ is the number of filters. After the fully connected layer and the activation layer, the dimension of the output changes to $(T - W + 1) \times 1$. We can see that the predicting result at each time step is related to $W \times N$ input data. In the TCN model, a mixed loss function is applied. Since part of the labels are annotated by the PCC-based algorithm, the loss function for traditional classification problem cannot be used directly. Therefore, when calculating those data with label 0 or 1, we use focal loss function; when calculating those data annotated by the PCC-based algorithm, we use MSE loss function. The structure of a TCN model is presented in Figure 2. We get the final prediction results by rounding the average values of eight prediction results acquired from eight
Next, we will introduce the PCC-based algorithm, the ensembling theory, and the TCN model.

**PCC-based algorithm for measuring the degree of blade icing**

Figure 3 shows the data recorded by the SCADA system of a wind turbine in northern China. Each data point contains 26 parameters, such as wind speed and power. The detailed information of 26 parameters is listed in Table 2. The sampling interval of all parameters is the same, which is 10 s. The whole dataset is arranged in chronological order. The staff of the wind farm confirmed the condition of the WT every once in a while. For example, he found that the WT blades were normal at $t = 4$ and found that the WT blades were icing at $t = 9$. What he can determine is that the data before $t = 4$ is normal data and the data after $t = 9$ is icing data. It is not certain whether the data between $t = 5$ and $t = 8$ is icing or normal. This is the reason why some data is unlabeled.

Nearly all present data-driven icing detection models ignore these unlabeled data, since only labeled data can be used to train the model according to the requirement of supervised learning. For conventional modeling problems, deleting part of the data has no effect, but WT icing detection is a time-series modeling problem. In other words, the output at the current moment is not only related to the input at the current moment but also related to the inputs at previous moments. For example, if we want to predict icing condition at $t = 9$, we will feed the data from $t = 5$ to $t = 9$ into the model (assuming that the window length is 5), instead of just inputting the data at $t = 9$. However, if the data from

| Abbreviation       | Description                        | Abbreviation       | Description                        |
|--------------------|------------------------------------|--------------------|------------------------------------|
| wind_speed         | Wind speed                         | pitch_1_moto_tmp   | Motor temperature of blade 1       |
| generator_speed    | Generator speed                    | pitch_2_moto_tmp   | Motor temperature of blade 2       |
| power              | Power                              | pitch_3_moto_tmp   | Motor temperature of blade 3       |
| wind_direction     | Angle of wind direction            | acc_x              | Nacelle acceleration in X direction |
| wind_direction_mean| Mean angle of wind direction       | acc_y              | Nacelle acceleration in Y direction |
| yaw_position       | Taw position                       | environment_tmp    | Environment temperature            |
| yaw_speed          | Taw speed                          | int_tmp            | Nacelle temperature                |
| pitch_1_angle      | Angle of blade 1                   | pitch_1_ng5_tmp    | ng5 temperature of blade 1         |
| pitch_2_angle      | Angle of blade 2                   | pitch_2_ng5_tmp    | ng5 temperature of blade 2         |
| pitch_3_angle      | Angle of blade 3                   | pitch_3_ng5_tmp    | ng5 temperature of blade 3         |
| pitch_1_speed      | Speed of blade 1                   | pitch_1_ng5_DC     | ng5 direct current of blade 1      |
| pitch_2_speed      | Speed of blade 2                   | pitch_2_ng5_DC     | ng5 direct current of blade 2      |
| pitch_3_speed      | Speed of blade 3                   | pitch_3_ng5_DC     | ng5 direct current of blade 3      |
t = 5 to t = 8 is ignored, the data used to predict whether WT blades are icing at t = 9 becomes the data from t = 1 to t = 4 and t = 9, as shown in Figure 4. Obviously, this will affect the accuracy of the prediction. In order to make full use of these unlabeled data and ensure the consecutiveness of the dataset, this article proposes a PCC-based method for measuring the degree of blade icing. Since the unlabeled data is between the normal data and the icing data, it is reasonable to speculate that these data could represent a process which the blades gradually transform from the normal condition to the icing condition. We use numbers between 0 and 1 to measure the degree of blade icing and annotate these unlabeled data. The closer the number is to 1, the more severe the icing, and vice versa. Next, we elaborate the procedures of the proposed algorithm. In the first step, we find the start moment \( t_{start} \) and the end moment \( t_{end} \) of a certain period of unlabeled data. Then, 60 data points before \( t_{start} \) and 60 data points after \( t_{end} \) are selected. In the second step, the EMA is utilized to calculate the average of the normal data \( S_{normal} \) and the icing data \( S_{icing} \). The equation of EMA is shown below

\[
S_t = \begin{cases} 
Y_t, & t = 1 \\
\alpha Y_t + (1 - \alpha)S_{t-1}, & t > 1 
\end{cases}
\]

\( Y_t \) refers to the original data. \( S_t \) is the data after EMA. \( \alpha \in [0, 1) \) is the weight coefficient. The effect of EMA is to make the update of data related to the historical data within a period of time. In the calculation process of \( S_{normal} \), \( \alpha \) is set to 0.9 which means that the data close to \( t_{start} \) is more important; in the calculation process of \( S_{icing} \), the sequence is reversed and \( \alpha \) is also set to 0.9.

In the third step, the similarity between each unlabeled data and \( S_{normal} \) along with the similarity between each unlabeled data and \( S_{icing} \) are measured by PCC. The formula of PCC is as follows

\[
PCC(A, B) = \frac{COV(A, B)}{\sigma_A \sigma_B}
\]

\( COV \) is the covariance of two data, and \( \sigma \) is the standard deviation. After getting the two PCC values, we use the softmax function to convert them into two values that sum to 1. The converted value \( PCC(S_{icing}, data) \) is the label for the unlabeled data. Algorithm 1 summarizes the above steps.

It can be seen that the label calculated by the above algorithm is a value between 0 and 1, which means that we have transformed the icing detection problem from a classification problem into a regression problem. The final output of the model is not a class (icing or normal), but a probability. Assuming that the WT is still in a normal state at \( t = 5 \) and \( t = 6 \), their labels will be relatively small. At \( t = 7 \), the blades start to freeze gradually. At \( t = 8 \), the blades are almost icing, its label should be relatively close to 1. For example, their labels calculated by our proposed algorithm are \( [0.1, 0.1, 0.6, 0.8] \). After annotating these data, there are at least two obvious benefits: The first is to ensure the consecutiveness of the dataset (see Figure 5). For time-series models, the data segments used for prediction will be consecutive, and the detection accuracy will also increase; the second is to enrich the amount of icing data. These soft-labeled data have the same features as the icing data, which can help improve the generalization ability of the model.
The ensembling theory

The main idea of ensemble learning is to complete learning tasks by constructing and combining multiple models. Sometimes a single model may not be able to learn all the features from the data. Then we can build multiple models, each of which learns a part of features from the data, and ensemble the results of all models. This article adopts this method to cope with class-imbalance problem. In the original dataset, normal data accounts for the majority. If the under-sampling method is applied, a large amount of normal data will be discarded, resulting in the loss of useful information. If over-sampling icing data or generating icing data is used, it is difficult to determine where these newly generated data is placed in order to retain the temporal relationship of the original dataset. Our solution to preserve the integrity of the dataset and the temporal relationship between data is constructing eight class-balanced subsets of the original dataset. The first step is to divide the normal data into eight subsets, through random sampling without replacement. The second step is to add all soft-labeled data and icing data to each subset. In each subset, all data is arranged in chronological order to restore the temporal relationship. In the training process, models are trained on subsets respectively, so there is no ensembling step. In the testing process, the testing data is inputted into eight models to obtain eight prediction results (a number between 0 and 1) and then these results are averaged and rounded to get the final classification result.

Temporal convolutional network

TCN is a specially designed CNN for processing time-series data. Conventional CNN usually considers the spatial relationship of data other than temporal relationship, but time-series modeling needs to consider the temporal relationship. For example, only currently observed data (i.e. not future data) can be used to judge the temporal relationship, but time-series modeling needs to consider the spatial relationship of data other than temporal relationship. For example, if long temporal dependencies are required, masked convolution is not sufficient; dilated convolution is introduced to increase the receptive field of the model. Figure 6 illustrates the increase of depth of the model. A schematic diagram of masked dilated convolution is illustrated in Figure 6. The yellow grids in the figure represent the activated neurons. The figure indicates that a four-layer masked dilated convolution network has $8 \times 8$ receptive fields, and the required parameters are only three $2 \times 2$ convolution kernel parameters.

The main advantage of deep learning is that deeper networks will lead to degradation problem. Residual block is able to ensure deeper layers contain more features than previous layers by introducing identity mapping. A residual block is calculated by the following equation

$$x_{l+1} = x_l + \mathcal{F}(x_l)$$

where $\mathcal{F}$ includes a series of transformations. $x_l$ and $x_{l+1}$ are the input and output of the residual block, respectively, as shown in Figure 7. If there are some useful information learned by $\mathcal{F}(x_l)$, the $l + 1$th layer will perform better than the $l$th layer. Residual block has been proved to be effective for deep networks by many researchers, so we adopt this approach in our model as well. Batch-normalization layer is used to alleviate the gradient vanishing problem; rectified linear unit (ReLU) is chosen as the activation function in the TCN layer; dropout layer can avoid over-fitting. The above layers constitute a residual block.
As mentioned in section “PCC-based algorithm for measuring the degree of blade icing,” the input of a time-series model is usually a data segment. We specify how the raw data is fed into TCN via sliding window method and how the output is concatenated, taking Figure 8 as an example. In the figure, the value of window length $W$ is 6. The dimension of the input is $T \times N$ ($N$ is the number of parameters and $T$ is the time). For the beginning, sliding window cuts a $6 \times N$ segment (time steps from 1 to 6) from the raw data and feeds it into TCN. TCN outputs the prediction results of time steps 6. Next, sliding window moves one step forward and cuts the second $6 \times N$ segment (time steps from 2 to 7) from the raw data. Correspondingly, TCN outputs the prediction results of time steps 7. We can figure out that the larger the value of $W$, the more historical information the model can take into account when making prediction. Finally, the output is concatenated together in chronological order to obtain the final output.

For TCN, its receptive field should be larger than window length $W$ to have all historical information in each segment considered. The receptive field of TCN is calculated as follows

$$\text{receptive field} = k \times c^{m-1} \times s$$

(6)

Dilation factor $c$ is set to 2 according to empirical value. $k,m,s$ stand for kernel size, number of layer, and number of stacked residual block, respectively. Therefore, the values of $k,m,s$ should be set reasonably on the basis of $W$.

Our mixed loss function is expressed as follows

$$L_{\text{mix}} = \begin{cases} -\mu(1 - \hat{y})^\gamma \log \hat{y}, & y = 1 \\ -(1 - \mu)\hat{y}^\gamma \log (1 - \hat{y}), & y = 0 \\ (\hat{y} - y)^2, & \text{otherwise} \end{cases}$$

(7)

where $y$ is the true label and $\hat{y}$ is the predicted value. When the true label is 0 or 1, $L_{\text{mix}}$ is the same as the focal loss function; when the label is given by the PCC-based algorithm, $L_{\text{mix}}$ is equal to the MSE loss function. The reason we use mixed loss function is that although focal loss function has been proved to be able to handle classification problems well, it cannot handle regression problems directly. Therefore, for those soft-labeled data, we use the MSE loss function to calculate the error. In the focal loss function, $\mu$ is called weight parameter. For the class with small number of samples, the penalty for misclassification can be increased by increasing the weight parameter of this class. $\gamma$ is called focusing parameter ($\gamma \geq 0$). Focusing parameter can adjust the weight of easily classified and hard classified samples. For example, let’s consider the case where the true label of a sample is 0. If its predicted value is 0.1, which means that this is an easily classified sample, the value of $\hat{y}^\gamma$ will be small, and the focal loss of this sample will reduce quickly. Conversely, if its predicted value is 0.9, indicating that this is a hard classified sample, the value of $\hat{y}^\gamma$ will be close to 1. Thus, the focal loss of this sample will almost remain the same. It is
worth noting that both μ and γ are non-trainable parameters, and their values are set to 0.5 and 2 as a matter of experience.

### Experiment preparation

#### Data preprocessing

SCADA system records 26 parameters including motion parameters (such as angle of three blades and generator speed), state parameters (such as motor temperature of three blades, nacelle acceleration in X and Y direction, and nacelle temperature), and environmental parameters (such as wind speed, wind direction, and environment temperature). The abbreviation and physical meaning of all parameters are listed in Table 2. Although the importance of the 26 parameters for detecting icing conditions may be different, deep learning model can automatically learn useful features from the parameters. Therefore, this article uses all the parameters as the input of the model, instead of selecting only some of the parameters like some traditional machine learning models. 70% of the data is chosen as the training set and the rest 30% is chosen as the testing set. Researches8,12 have confirmed that analyzing wind speed-power curve is beneficial for icing detection. For example, when wind speed is greater than 2 and power is greater than 1.8, it can be determined that there are no icing conditions (see Figure 9). Thereby, these normal data can be removed from the training data.

The value of parameters differs from each other, and we have to normalize them by the equation below

$$ x = \frac{x' - x'_{\min}}{x'_{\max} - x'_{\min}} $$

where $x$ is the normalized input data and $x'$ is the original input data. $x'_{\max}$ and $x'_{\min}$ are the minimum and maximum values of each parameter.

### Model evaluation index

For class-imbalanced datasets, it is not comprehensive to evaluate a model only by accuracy, because the model can achieve high accuracy as long as all samples are classified as the major class. The confusion matrix and a series of evaluation metrics calculated by it are considered more objective to evaluate the performance of a model on class-imbalanced datasets. The confusion matrix is given in Table 3. TP represents the number of actual icing samples predicted to be icing samples; TN represents the number of actual normal samples predicted to be normal samples; FP represents the number of actual normal samples predicted to be icing samples; FN represents the number of actual icing samples predicted to be normal samples. Based on the confusion matrix, precision ($P$), recall ($R$), and F-score ($F_\beta$) are defined below

$$ P = \frac{TP}{TP + FP} \quad (9) $$

$$ R = \frac{TP}{TP + FN} \quad (10) $$

$$ F_\beta = (1 + \beta^2) \frac{P \cdot R}{\beta^2 \cdot P + R} \quad (11) $$

$$ ACC_{\text{norm}} = \frac{1}{2} \times \left( \frac{TP}{TP + FN} + \frac{TN}{FP + TN} \right) \quad (12) $$

$P$ stands for the number of TP divided by the total number of samples which are predicted to be icing samples. $R$ stands for the number of TP divided by the total number of samples which are actual icing samples. $P$ and $R$ ought to be as large as possible. Nevertheless, there is an inverse relationship between $P$ and $R$, that is, an increment of $P$ will bring a decrement of $R$. To fully consider these two metrics, $F_\beta$ is also introduced. We can adjust the value of $\beta$ according to the importance of $P$ and $R$. In this article, $\beta$ is 1 which means that $P$ and $R$ are equally important. $ACC_{\text{norm}}$ is introduced to calculate the mean of the accuracy per class. Besides,
Matthews\textsuperscript{28} correlation coefficient (MCC) is chosen as another evaluation metrics. MCC is used to evaluate the classification performance of models in binary classification problems in machine learning. It returns a value between $-1$ and $1$. The value of $1$ means perfect prediction, $0$ means no better than random prediction, and $-1$ means a complete discrepancy between predict label and true label. MCC is calculated by the following equation

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(FN + TP)(TN + FP)(TN + FN)}}$$

(13)

Results analysis

Our experiments are implemented on a Dell server with Intel E5-2620 CPU, 16G memory and two NVIDIA GTX1080 graphics cards. The programming language is Python and the deep learning framework is Keras (with Tensorflow backend). In section “Ablation experiment,” we design three comparative models. For the first model, we do not use PCC-based algorithm and do not use ensemble learning method. For the second model, we use PCC-based algorithm and do not use ensemble learning method. For the third model, we do not use PCC algorithm and use ensemble learning method. Through the experiments between our proposed model and the above comparative models, it is found that both PCC-based algorithm and ensemble learning method are effective in improving the detection accuracy. Then, we compare the proposed model with a variety of existing data-driven WT icing detection models, and the results are shown in section “PCC-Ensemble-TCN model versus other data-driven models.”

Ablation experiment

As mentioned above, three comparative models are established. For the first comparison model, the original dataset is not processed by the PCC-based algorithm, and the data without label will be discarded. The solution to class-imbalance problem is to downsample the number of normal data to the number of icing data. The structure of TCN remains the same. We call this model TCN model. For the second comparison model, the original dataset is processed by the PCC-based algorithm, so the unlabeled data is annotated. The solution to class-imbalance problem is to downsample the number of normal data to the number of icing data. The structure of TCN remains the same. We call this model PCC-TCN model. For the third comparison model, the original dataset is not processed by the PCC-based algorithm, and the data without label will be discarded. The ensemble learning method is adopted, which means that the original dataset is divided into multiple class-balanced subsets, and TCN models are trained on each subset. The final prediction results are acquired by ensembling the prediction result of each TCN model. We call this model ensemble-TCN model. Figure 10 shows a schematic diagram of the three comparison models. The hyperparameters of the TCN model are optimized by grid searching. Concretely, window length $W$ is set to 60, kernel size $k$ is set to 7, number of dilation layer $m$ is set to 6, number of stacked residual block $s$ is set to 2, and number of convolutional kernels $N_f$ is set to 28.

The performance of four models is listed in Table 4. Comparing TCN model and PCC-TCN model, it can be found that the introduction of the PCC-based algorithm significantly improves the $R$ value. This is because the PCC-based algorithm augments the number of icing
data and reduces the probability of misclassifying icing data as normal data. The $P$ value increases slightly as well, so the $F_1$ score has also increased. Comparing TCN model and Ensemble-TCN model, we can find that the ensemble learning method helps to increase the $P$ value by improving model’s ability to identify normal data. The reason is that all normal data in the original dataset are sufficiently considered and learned instead of under-sampled. At the same time, the $R$ value has not increased much, which means that there are still some icing data that are not correctly identified. Comparing TCN model and PCC-Ensemble-TCN model, we can figure out that the combination of PCC-based algorithm and ensemble learning method not only improves the $P$ value but also improves the $R$ value. Correspondingly, the $F_1$ score has also increased by 0.04. In addition, the MCC value of PCC-Ensemble-TCN model is the highest among four models. Through ablation experiments, it can be concluded that the improvement of detection accuracy is due to the introduction of PCC-based algorithm and ensemble learning method.

Table 4. Ablation experiment results.

|        | $P$       | $R$       | $F_1$     | MCC       | $ACC_{\text{norm}}$ |
|--------|-----------|-----------|-----------|-----------|---------------------|
| TCN    | 0.875 ± 0.017 | 0.823 ± 0.023 | 0.848 ± 0.010 | 0.842 ± 0.010 | 0.907 ± 0.011       |
| PCC-TCN| 0.887 ± 0.019 | 0.857 ± 0.022 | 0.871 ± 0.015 | 0.866 ± 0.015 | 0.924 ± 0.011       |
| Ensemble-TCN | 0.915 ± 0.017 | 0.828 ± 0.026 | 0.869 ± 0.015 | 0.862 ± 0.016 | 0.911 ± 0.013       |
| PCC-Ensemble-TCN | 0.921 ± 0.019 | 0.866 ± 0.024 | 0.892 ± 0.017 | 0.886 ± 0.018 | 0.930 ± 0.012       |

MCC: Matthews correlation coefficient; TCN: temporal convolutional network; PCC: Pearson correlation coefficient; $ACC_{\text{norm}}$: the mean of the accuracy per class.

The average and standard deviation of the evaluation metrics are obtained after performing $n = 10$ experiments.

Figure 11 shows a segment of the original dataset, where there is an obvious label missing problem. The original dataset consists of many such segments. The data for about 2 h from 0:06 to 2:03 is unlabeled, and other data-driven models have ignored this part of data. We use the PCC-based algorithm to annotate these data, and the effect is shown by the green dots in Figure 11. It can be seen that the label slowly changes from 0 to 1, basically in line with an increasing trend. According to our definition, the WT slowly transitions from a normal condition to an icing condition. The benefits of using the PCC-based algorithm are obvious. One is to ensure the consecutiveness of the dataset, and the other is that our model can use more training data than other data-driven models.

Using an autoencoder to annotate unlabeled data is an alternative method and may also achieve good effect as our PCC-based algorithm. Autoencoder is one of deep learning methods, which requires a lot of data for training. Whether in the training phase or the inference phase, the amount of calculation of autoencoder is much greater than our PCC-based algorithm. In addition, as a black box model, autoencoder is not as interpretable as our PCC-based algorithm. Unsupervised learning is also a promising technique. The SCADA data used in this article contains plenty of labeled data, and the non-supervised learning method does not make full use of these label information. For the above considerations, we prefer the PCC-based algorithm in this icing detection problem.

**PCC-Ensemble-TCN model versus other data-driven models**

We list some existing data-driven WT blades ice detection models and their performance in Table 5. Traditional machine learning model like particle swarm optimization-support vector machine (PSO-SVM) need to manually select features from many parameters, which is labor-intensive. In addition, it does not take full advantage of temporal relationship between data. In other words, PSO-SVM predicts whether there is icing condition only using the data at the current moment. It can be seen from the results that the $P$ value of this model is quite high, but the $R$ value is...
relatively low, which means that some icing conditions are judged as non-icing conditions. Of course, we hope that this happens as few times as possible. For ensemble autoencoder model, its ensemble method is to fully consider the features extracted by different hidden layers in an autoencoder model, which differs from our ensemble theory. The MCC value of ensemble autoencoder model is slightly higher than that of PSO-SVM model, but lower than our model, which shows that ensembling multiple models to learn the features from the data is better than a single model. Stacked autoencoder model predicts the normal condition accurately, as can be seen from its high $P$ value. In contrast, it predicts poor icing conditions since its $R$ value is rather low. As a model that specializes in processing time-series data, LSTM model performs fairly moderate, because the $P$ value and $R$ value are relatively close but not high. Thus, its $F_1$ value exceeds some other models. TL-DNN model realizes the class-imbalance problem, and specially designs triplet loss to maximize the difference between the classes and retain the characteristics within the same class. But its $R$ value is the lowest among all models, which shows that it cannot detect icing conditions well, so it is not an optimal choice in this problem. Through the above analysis, it can be seen that when the dataset is inconsecutive and class-imbalanced, neither traditional machine learning methods nor deep learning methods can achieve satisfactory results. Therefore, it is necessary to perform data preprocessing according to the characteristics of the dataset. The ensemble learning method proposed in this article can fully learn the discriminative features to distinguish icing conditions from non-icing conditions by training TCN models on multiple subsets, and the misclassification probability of an ensemble model is less than the misclassification probability of a single model. Thus, our model has a relative high $P$ value, signifying that few true normal samples are predicted to be icing samples. The PCC-based algorithm helps keep the consecutiveness of the dataset and expands the number of icing samples. Benefiting from it, we get a high $R$ value, which indicates that there are few icing samples misclassified as normal samples. Furthermore, two comprehensive evaluation metrics $F_1$ and MCC both show that the model proposed in this article is better than other data-driven models.

### Conclusion

This article proposes a PCC-based algorithm for measuring the degree of blade icing and an ensemble learning model to deal with the label missing problem and the class-imbalance problem in the wind turbine SCADA data, which are neglected in recent data-driven models. The proposed PCC-based algorithm measures the similarity between the unlabeled data and nearby icing data as its label. This not only ensures the consecutiveness of the dataset but also replenishes the information under icing conditions. Afterward, we divide the normal data in the training set into eight equal parts due to the ratio of normal data to soft-labeled data together with icing data is about 8:1. Then eight class-balanced subsets are constructed. Each subset contains all soft-labeled data, icing data, and one part of normal data. The icing detection model, TCN model, is trained on each subset. In the TCN model, the original cross-entropy loss function is replaced with a mixed loss function that combines focal loss and MSE to focus on samples with large differences between the predicted results and the actual results (difficult-to-classify samples), thereby accelerating the convergence process of the model. We get the final prediction results by rounding the average values of eight prediction results acquired from eight TCN models. The proposed model is validated using the actual SCADA data collected from a wind farm in northern China, and the results indicate that ensuring the consecutiveness and class-balance of the data are quite advantageous for improving the detection accuracy by comparing with other data-driven models.

We present a time-series prediction model for anomaly detection, and this kind of problem can be found in

---

**Table 5. Performance of our proposed model and other data-driven models.**

| Model                        | $P$   | $R$   | $F_1$ | MCC  |
|------------------------------|-------|-------|-------|------|
| PSO-SVM                      | 0.935 | 0.805 | 0.865 | 0.746|
| Ensemble autoencoder        | –     | –     | –     | 0.784|
| Stacked autoencoder         | 0.973 | 0.714 | 0.823 | –    |
| LSTM                        | 0.898 | 0.860 | 0.879 | –    |
| TL-DNN                      | 0.909 | 0.631 | 0.745 | –    |
| PCC-Ensemble-TCN            | 0.921 | 0.866 | 0.892 | 0.886|

MCC: Matthews correlation coefficient; PCC: Pearson correlation coefficient; TCN: temporal convolutional network; LSTM: long short-term memory; TL-DNN: triplet loss deep neural network; PSO-SVM: particle swarm optimization support vector machine.

The bold values represent the best values obtained by the six models of a certain evaluation metric.
many industrial scenarios.\textsuperscript{30–32} Since the model proposed in this article performs well in WT blades icing detection, it is conceivable that our proposed model should also be applicable to those problems, which needs to be further verified in the future.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship and/or publication of this article: This work was supported by the National Natural Science Foundation of China (Grants Nos. 11972115, 11572084).

ORCID iD
Shenyi Ding https://orcid.org/0000-0003-3149-2522

References
1. Karlsson T. Cold climate wind power market study 2015–2020, 2017, http://winterwind.se/wp-content/uploads/2015/08/9_1_24_Karlsson_IEA_Task_19__Cold_climate_wind_power_market_study_2015-2020_Pub_v1.pdf
2. Wang Z, Xu Y, Fei S, et al. A light lithium niobate transducer for the ultrasonic de-icing of wind turbine blades. Renew Energ 2016; 99: 1299–1305.
3. Parent O and Ilinca A. Anti-icing and de-icing techniques for wind turbines: critical review. Cold Reg Sci Technol 2011; 65(1): 88–96.
4. Muñoz CQG, Márquez FPG and Tomás JMS. Ice detection using thermal infrared radiometry on wind turbine blades. Measurement 2016; 93: 157–163.
5. Gao H and Rose JL. Ice detection and classification on an aircraft wing with ultrasonic shear horizontal guided waves. IEEE Trans Ultrason Ferroelectr Freq Control 2009; 56(2): 334–344.
6. Madi E, Pope K, Huang W, et al. A review of integrating ice detection and mitigation for wind turbine blades. Renew Sustain Energ Rev 2019; 103: 269–281.
7. Tautz-Weinert J and Watson SJ. Using SCADA data for wind turbine condition monitoring—a review. IET Renew Pow Gener 2017; 11(4): 382–394.
8. Davis NN, ByrkjedalØ, Hahmann AN, et al. Ice detection on wind turbines using the observed power curve. Wind Energy 2016; 19(6): 999–1010.
9. Song P, Yao Z and Zhao C. Section division and multi-model method for early detection of icing on wind turbine blades. In: Proceedings of the 2019 34th youth academic annual conference of Chinese association of automation (YAC), Jinzhou, China, 6–8 June 2019, pp.749–754. New York: IEEE.
10. Zhou G, Tan W and Zhang D. Ice detection for wind turbine blades based on PSO-SVM method. J Phys Conf Ser 2018; 1087: 022036.
11. Li N, Yan T, Li N, et al. Ice detection method by using SCADA data on wind turbine blades. Pow Gen Technol 2018; 39: 58–62.
12. Zhang L, Liu K, Wang Y, et al. Ice detection model of wind turbine blades based on random forest classifier. Energies 2018; 11(10): 2548.
13. Jiménez AA, Márquez FPG, Moraleda VB, et al. Linear and nonlinear features and machine learning for wind turbine blade ice detection and diagnosis. Renew Energ 2019; 132: 1034–1048.
14. Zhang Y, Li X, Gao L, et al. Intelligent fault diagnosis of rotating machinery using a new ensemble deep auto-encoder method. Measurement 2020; 151: 107232.
15. Wen L, Li X and Gao L. A new two-level hierarchical diagnosis network based on convolutional neural network. IEEE Trans Instrum Meas 2020; 69(2): 330–338.
16. Zhang Y, Li X, Gao L, et al. Ensemble deep contractive auto-encoders for intelligent fault diagnosis of machines under noisy environment. Knowl Based Syst 2020; 196: 105764.
17. Deng F, Ding H, Yang S, et al. An improved deep residual network with multiscale feature fusion for rotating machinery fault diagnosis. Meas Sci Technol 2021; 32(2): 024002.
18. Zollanvari A, Kunanbayev KM, Bitaghsir SA, et al. Transformer fault prognosis using deep recurrent neural network over vibration signals. IEEE Trans Instrum Meas 2020; 70(99): 2502011.
19. Liu Y, Cheng H, Kong X, et al. Intelligent wind turbine blade icing detection using supervisory control and data acquisition data and ensemble deep learning. Energ Sci Eng 2019; 7: 2633–2645.
20. Yeh C, Lin M, Lin C, et al. Machine learning for long cycle maintenance prediction of wind turbine. Sensors 2019; 19(7): 1671.
21. Yun H, Zhang C, Hou C, et al. An adaptive approach for ice detection in wind turbine with inductive transfer learning. IEEE Access 2019; 7: 122205–122213.
22. Li M and He X. An accurate detection method for turbine icing issues using LSTM network. IOP Conf Ser Earth Environ Sci 2019; 237: 032109.
23. Chen J, Hu W, Cao D, et al. An imbalance fault detection algorithm for variable-speed wind turbines: a deep learning approach. Energies 2019; 12(14): 2764.
24. Lin WC, Tsai CF, Hu YH, et al. Clustering-based undersampling in class-imbalanced data. Inform Sci 2017; 409–410: 17–26.
25. Xie J and Qiu Z. The effect of imbalanced data sets on LDA: a theoretical and empirical analysis. Patt Recogn 2007; 40(2): 557–562.
26. Chawla NV, Bowyer KW, Hall LO, et al. SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 2002; 16(1): 321–357.
27. Chen L, Xu G, Zhang Q, et al. Learning deep representation of imbalanced SCADA data for fault detection of wind turbines. *Measurement* 2019; 139: 370–379.

28. Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta* 1975; 405(2): 442–451.

29. Juan L, Xixia H and Xiaoli L. Icing prediction of wind turbine blade based on stacked auto-encoder network. *J Comput Appl* 2019; 39: 1547–1550.

30. Zheng X, Zhou Q, Hao Z, et al. Research on mechanism diagnosis of an idling abnormal noise of automobile engine. *Appl Acoust* 2021; 171: 107670.

31. Jan SU, Lee YD and Koo IS. A distributed sensor-fault detection and diagnosis framework using machine learning. *Inform Sci* 2021; 547: 777–796.

32. Zhang L, Cai B, Xiong G, et al. Multistage fault feature extraction of consistent optimization for rolling bearings based on correlated kurtosis. *Shock Vib* 2020; 2020: 8846156.