Multiscale Wavelet-Driven Graph Convolutional Network for Blade Icing Detection of Wind Turbines

Lai, Zhichen; Cheng, Xu; Liu, Xiufeng; Huang, Lizhen; Liu, Yongping

Published in:
IEEE Sensors Journal

Link to article, DOI:
10.1109/JSEN.2022.3211079

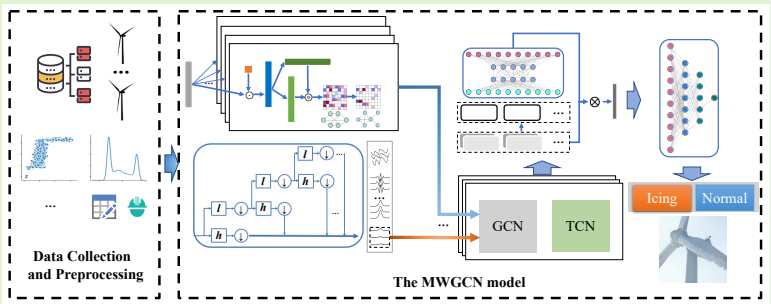
Publication date:
2022

Document Version
Peer reviewed version

Multiscale Wavelet-Driven Graph Convolutional Network for Blade Icing Detection of Wind Turbines

Zhichen Lai, *Graduate Student Member, IEEE*, Xu Cheng, *Member, IEEE*, Xiufeng Liu, *Member, IEEE*, Lizhen Huang, Yongping Liu

***Abstract*— Blade icing detection is critical to maintaining the health of wind turbines, especially in cold climates. Rapid and accurate icing detection allows proper control of wind turbines, including shutting down and clearing the ice, thus ensuring turbine safety. This paper presents a wavelet-driven multiscale graph convolutional network (MWGCN), which is a supervised deep learning model for blade icing detection. The proposed model first uses wavelet decomposition to capture multivariate information in the time and frequency domains, then employs a temporal graph convolutional network to model the intervariable correlations of the decomposed multiscale wavelets, as well as their temporal dynamics. In addition, this paper introduces scale attention to the MWGCN for a further improvement of the model and proposes the method to address the class imbalance problem of the training data sets. Finally, the paper conducts comprehensive experiments to evaluate the proposed model, and the results demonstrate the effectiveness of the model in blade icing detection and its better performance over eight state-of-the-art algorithms, with 17.2% and 11.3% higher F1 scores over the best state-of-the-art baseline on the labeled datasets.**



The proposed blade icing detection framework of wind turbine

***Index Terms*— Graph Convolutional Network, Time Series Classification, Wavelet Transform, Wind Turbine**

I. Introduction

Recently, the need of adopting renewable and green energy grows rapidly as the concerns about environmental pollution and global warming increase year by year. As a result, as one of the cleanest energy sources, wind energy attracts ever-increasing attention and investment. The U.S. Department of Energy, for example, has set an ambitious goal of generating 20% of the country's total electricity from wind turbines by 2030 and 35% by 2050 [1], while the European Commission has also presented its "Roadmap 2050" project, claiming that wind energy would meet half of Europe's electricity needs by 2050 [2]. To achieve this goal, more turbines are expected to be installed in areas with high wind speeds and dense cold air, such as high-altitude mountain areas, where the wind resource is conducive to the increase in wind power output. According to the International Energy Agency, the cumulative wind installations in cold locations account for more than 30% of the total installed wind turbine capacity globally [3]. However, blade icing is common in these places due to the cold environment, especially during the winter months. During the icing process, supercooled water droplets in the air collide with the incoming flow and impinge on the blade surface. Water droplets freeze instantly when the temperature and liquid water content (LWC) are low. However, when the temperature and LWC are high, only a portion of the droplets freeze on collision, while the rest form a water film on the surface of the blade and eventually freeze [4], resulting in problems in safety and efficiency for wind turbines [1].

The majority of research on wind turbine icing detection focuses on blades. Traditionally, based on the physical properties of ice, applied physics researchers resolve this challenge by designing and installing new physical detectors, such as damping of ultrasonic [5], thermal infrared radiometry [6], ultrasonic guided waves [7], and so on. These methods directly

Zhichen Lai, Xu Cheng, Lizhen Huang and Yongping Liu are with the Department of Manufacturing and Civil Engineering, Norwegian University of Science and Technology (NTNU), 2815 Gjøvik, Norway (zhla@cs.aau.dk, xu.cheng@ieee.org, lizhen.huang, yongping.liu@ntnu.no).

Zhichen Lai is also with the Department of Computer Science, Aalborg University, 9100 Aalborg, Denmark.

Xiufeng Liu is with the Department of Technology, Management and Economics, Technical University of Denmark, 2800 Kongens Lyngby, Denmark (e-mail: xiuli@dtu.dk).

Zhichen Lai and Xu Cheng are equal contribution. The work of Z. Lai was supported by the Chinese Scholarship Council. *(Corresponding authors: Lizhen Huang and Yongping Liu.)*

measure the physical properties of ice and are quite reliable. However, they may have negative effects on the structure of the blade, resulting in costly human and material resources, and they are insensitive to small amounts of ice accretion [8]. As a result, engineers adopt indirect methods to estimate blade icing conditions in practice, such as monitoring the power curve [9], anemometers [10], vibration time wave forms [11], dew point and temperature [12], to determine whether to trigger the deicing procedure. However, due to internal sensor instability, these methods may yield many erroneous estimations [13]. Furthermore, these methods are ineffective in detecting blade icing and are also insensitive to small levels of ice accretion [14].

Methods based on wind turbine monitoring signals have received a lot of attention to solve the deficiencies of traditional approaches, which can be separated into model-driven and data-driven methods due to the cheaper cost and fewer mechanical modifications. Model-based methods rely on high-level domain knowledge to develop mathematical models from various perspectives that reflect the correlations between signals and blade icing conditions, such as rotor angular speed [15], [16], power curve [17] and thermodynamics [18]. The limitations of model-based methods include an overreliance on prior domain knowledge and a lack of universality. Furthermore, expensive and specialized tools such as wind tunnels are necessary to fine-tune the model parameters. Furthermore, when the operating environment changes, the accuracy of the model decreases significantly [19].

Recently, data-driven machine learning methods, including shallow machine learning and deep learning-based approaches, seem to be a more powerful means of blade icing detection with the plentiful availability of historical data collected by the Supervisory Control and Data Acquisition system (SCADA). Specifically, SCADA is a powerful data acquisition and monitoring system, which is the most commonly used in fault diagnosis of wind turbine components [20], [21]. It is worth noting that the SCADA system collects not only wind turbine operating parameters and signals but also environmental parameters such as temperature, wind speed, and wind direction, providing an interface to connect various sensors across wind turbines for various monitoring and controlling operations, and providing a complete picture of the status of wind turbines. Since the use of SCADA data for failure detection is a potentially low-cost option that does not require additional sensors, a number of approaches based on this have been developed in recent years [20], [22]. Thus, the SCADA system can manage the installed general-purpose sensors that monitor wind turbine operation and provide multivariate time series data of the real-time operating environment to data-driven machine learning models, such as linear discriminant [23], KNN [24], SVM [19] and some hybrid methods [25]–[27]. However, shallow machine learning methods rely heavily on hand-crafted feature designs, which are time consuming and costly, while deep learning methods do not require as much manual feature extraction, which receive a great deal of interest from researchers [8], [28]–[35].

Although many deep learning algorithms have been developed for detecting wind turbine blade icing, they still face the following critical challenges: First, blade icing can have a variety of dynamics depending on weather conditions, while the resulting signals are highly dependent on the frequency, duration, severity and intensity of the icing conditions, implying that anomalous changes in the patterns embedded in both time and frequency domains must be considered and exploited. Second, although several attempts have been undertaken to extract multiscale features from the time and frequency domains using wavelet decomposition, the scale-specific intervariable correlations among different scales have not yet been exploited. Instead, existing models typically learn the prominent and shared correlations for all scales. Last, due to the imbalance problem of the training data, the results of the blade icing detection classification task may be biased.

To address the above challenges, this paper proposes a wavelet-driven multiscale graph convolutional network (MWGCN) model for wind turbine blade icing detection. First, this model uses a multiscale discrete wavelet decomposition method to extract multiscale features in the time and frequency domains and then applies scale-specific correlation learning to automatically infer scale-specific graphs, allowing the model to exploit implicit information about inter-variate relationships in the time and frequency domains. More specifically, this model employs a temporal graph convolutional network to obtain multiscale information, rather than using traditional RNN methods. That is, the network structure combines a graph convolutional network (GCN) and a temporal convolutional network (TCN). Second, to further improve the model, we implement a scale attention module, which is able to weight the importance of temporal patterns at different scales and capture the information of inter-scale correlations. Finally, most of the data from wind farm SCADA systems are unlabeled, which can lead to bias in the classification tasks. To address the data imbalance problem, this paper further proposes a class rebalancing classifier with focal loss to handle the dataset.

The contributions of this paper can be summarized as follows:

1) We propose a wavelet-driven multiscale graph convolutional network (MWGCN) model for wind turbine blade icing detection. To the best of our knowledge, this is the first attempt to apply a graph convolutional model to wind turbine blade icing detection.
2) We propose a network structure to model the relationship of multiscale wavelets in the time and frequency domains. This network design allows for better extraction of the characteristics and features of inter-variate correlations for multiscale wavelets.
3) We conduct comprehensive experiments to evaluate the proposed model using real-world SCADA data from wind farms, and the results validate the effectiveness of the proposed model, as well as the superiority and better performance compared to eight baselines.

The remainder of the paper is organized as follows. Section II presents the related work; Section III describes the proposed model in detail; Section IV conducts the experiments to evaluate the model and compare it with the baselines; and Section V concludes the paper and presents future work.

II. Related Work

A. Effects of Blade Icing of Wind Turbines

According to [36], the highest number of wind turbine incidents is attributed to blade failures, causing 20–50% power loss, which is much higher than other adverse factors [36], [37]. Furthermore, cumulative wind installations in cold regions exceed 30% of the total installed wind turbine capacity worldwide [3], where blade icing develops. It can cause vibration of wind turbines as a result of uneven ice shedding, which can alter the aerodynamic capability of blades by affecting the blade surface roughness. Subsequently, the life span and power generation rate of wind turbines are reduced [38]. Furthermore, the studies of effects of blade icing can be categorized into: full stop of the turbine, disruption of aerodynamics, overloading due to delayed stalls, decreased fatigue life, and human safety risks [37].

Jasinski et al. analyzed 1,337 full-stop records of wind turbines in Sweden between 1998 and 2003, resulting in a total downtime of 161,523 hours. Specifically, 7% of the full stop incidents were caused by blade icing, leading to 5% power loss [39]. Antikainen et al. studied the disruption of aerodynamics caused by blade icing, and the findings revealed that even in the early stages of icing, mass and aerodynamic imbalance can occur [40]. Jasinski et al. demonstrated that blade icing affects airfoil efficiency, resulting in higher maximum power and deterioration of blade mechanical health [39]. The WECO project (Wind Energy Production in Cold Climates) investigated fatigue loads caused by blade icing, finding that additional ice masses can cause higher deterministic loads, increase the excitation of edge-wise vibrations, and resonance may occur due to the changed natural frequencies of the blades [41].

The aforementioned studies demonstrate the seriousness of the potential risks and effects caused by blade icing, therefore, it is necessary to analyze the mechanisms responsible for blade icing and investigate advanced intelligent algorithms to detect blade icing of wind turbines.

B. Data-driven Blade Icing Detection

There are two main types of data-driven blade icing detection methods: shallow machine learning-based and deep learning-based data-driven approaches. Shallow machine learning methods, such as linear discriminant, were adopted by [23] to explore the features of the power and wind speed distribution and adopted random forest classifiers for ice detection. Besides, the K-Nearest Neighbors algorithm (KNN) was employed with lateral vibration data from the nacelle and power performances reduction to detect ice accretion [24]. Xu et al. [19] adopted a Support Vector Machine (SVM) for wind turbine icing detection, with Particle Swarm Optimization (PSO) to optimize its parameters. The aforementioned shallow machine learning methods do not rely on precise mathematical models but on the hand-crafted feature design of these models, which is time consuming and expensive.

Deep learning based methods have attracted great research interest for their capability in feature extraction [29]. Jiang et al. [30] proposed the multiscale Convolutional Neural Networks (MSCNN) for the fault diagnosis of wind turbines, which incorporated multiscale learning into the traditional CNN architecture. Chen et al. [31] proposed a novel intelligent diagnosis approach for wind turbine failure detection, which addressed the problem of imbalanced distribution of normal and abnormal data. Liu et al. [32] introduced ensemble learning in a deep learning model to improve the accuracy and generalization of the model. To determine the importance of sensors among time-steps and automatically identify discriminative features from raw sensor data, Cheng et al. [34] proposed a temporal attention convolutional neural network. In particular, since the label imbalance problem is serious in the wind turbine SCADA data, Cheng et al. [35] presented a semi-supervised model for wind turbine blade icing detection. Yuan et al. [33] proposed a wavelet transformation and a fully convolutional neural network (WaveletFCNN) to automatically obtain multiscale wavelet features from the time domain and frequency domains. Tian et al. [8] improved WaveletFCNN by introducing a parallel structure consisting of an LSTM and CNN branch network structure. However, although the detail coefficients from Discrete Wavelet Transform (DWT) can be interpreted as an additive decomposition of the signal, referred as multi-resolution [33], the existing approaches still have not specifically considered the scale-specific inter-variate correlations between multiple wavelet scales sufficiently, which could lead to inaccurate representations for real-world scenarios. For example, inter-variate correlations are complicated from different perspectives and scales, including distinct time domains and frequency domains at multiple scales [42], [43]. Besides, wind turbines operate normally for most of the time, with just a small period of time when the blades are icing, resulting in a significant imbalance between these two labels, making data-driven methods difficult to train. As a result, it is necessary to take into account the data complexity when creating the blade icing models.

C. Wavelet-Driven Time Series Analysis

Various wavelet-driven models have recently been presented for time series analysis to help model the important frequency information of time series effectively [7], [33], [44]–[47]. The Discrete Wavelet Transform (DWT) is particularly well suited to noise filtering, data reduction, and singularity detection, and unlike the discrete Fourier transform, which retains only spectral information, the DWT retains information in both time and frequency domains, making it a great choice for time series data analysis [48]. DWT was used by Percival et al. to analyze the geophysical time series of Arctic sea ice [47]. To investigate trends in stream flow and precipitation in Canada, Nalley et al. used DWT with the sequential Mann-Kendall test. Norbre et al. employed PCA, DWT, and an XGBoost classifier to predict the next day's market direction [46]. Wang et al. proposed to use trainable filter parameters in the wavelet transformation under a deep neural network framework, which improves the interpretability of wavelet-driven time series analysis [44]. Combining the wavelet transformation with trainable filter parameters, XGBoost and LSTM, Liu et al. proposed an adaptive wavelet transform model for time series data prediction.

Blade icing is triggered by non-stationary and unstable natural factors such as wind speed, live wind direction, ambient temperature, which may have diverse dynamics due to the changing weather conditions. The period, severity and intensity of blade icing can all have significant impacts on the reflected signals. Thus, to detect changes in the pattern inherent in multivariate signals in both time and frequency domains, several studies have investigated wavelet-driven blade icing detection models for wind turbines. WaveletFCNN is a fully convolutional neural network integrated with the discrete wavelet transform, as proposed by Yuan et al. [33]. Furthermore, Yuan et al. also proposed WaveletAE, an unsupervised model that combines the autoencoder and the discrete wavelet transform [49]. By introducing an LSTM branch network to temporal information, Tian et al. improved WaveletFCNN and proposed MCRNN [8].

Although the wavelet transformation was used to improve time series data analysis in the approaches mentioned above, the results of the wavelet transformation of time series were simply used as the input, with no further exploration of the inter-variate correlations of the multiple wavelet components at different levels, which could help improve performance.

III. Proposed Method

This section will first present the overall framework of the proposed MWGCN, then describe all modules in greater detail, including the data preprocessing module, the multiscale wavelet decomposition module, the scale-specific correlation learning module, the Temporal Graph Convolutional (TGC) module, the scale-attention module, and the blade icing classifier.

A. Overall Framework

Figure 1 shows the overall framework for wind turbine blade icing detection, which consists of the following phases: (a) the wind turbine SCADA system collects historical raw data for further modeling; (b) data preprocessing; (c) the preprocessed data are decomposed into multiple scales by the MDWD module; (d) the scale-specific correlation learning module learns the inter-variate relationships of the input data at multiple wavelet scales; (e) the GCN submodule generates the scale-specific features as inputs to the TCN submodule at each scale and each timestep, which then extract the scale-specific features of each wavelet detail coefficient within the window size; (f) the scale-attention module further refines the scale-specific features; (g) the class rebalancing classifier conducts final classifications; (h) the trained model can be integrated with the SCADA systems for online blade icing detection. Algorithm. 1 presents the learning process of MWGCN.

B. Data Preprocessing Module

In this study, multivariate time series data are collected from an SCADA system with general purpose sensors, including wind speed, generator speed, power, wind direction, and yaw speed. However, due to sensor or human errors, sensors may stop recording or output some outlier readings. As a result, to reduce the effects of outliers and missing data, a data preprocessing module is necessary, which comprises data labeling, data normalization, statistical analysis, and missing data handling.

1) Data labeling: Wind turbine engineers were tasked with labeling the raw blade icing data. Due to the inherent instabilities of wind turbine operation, which are extremely difficult for even the most experienced engineers to distinguish as normal or icing conditions, some equivocal intervals are removed. Also, within a fixed time step (window size), engineers assign two types of labels to the internals: the normal label and the blade icing label.

2) Data Normalization: The multivariate time series data from SCADA systems often contain a lot of noise, thus the raw data should be normalized to effectively reduce the effect of noise. In addition, data normalization also increases the efficiency of training and the convergence speed of the models. Thus, in this study, the Min-Max normalization method is adopted. For a given SCADA signal from the dataset $X = (x_1, x_2, ..., x_T)$, where T is the number of points in a time series. The normalization of X is calculated using the following formula:

$$x_{i,\mathrm{norm}} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

where $X_{norm} = (x_{1,\mathrm{norm}}, x_{2,\mathrm{norm}}, ..., x_{T,\mathrm{norm}})$ represents the normalized data, x_{max} and x_{min} are the maximum and minimum of X, respectively.

3) Statistics Analysis: The wind turbine SCADA system collects data from hundreds of different sensors. However, most of the data contains redundant information, which can degrade the performance of the detection model and increase the computation. To this end, statistical analysis is required to better understand SCADA data: 1) To identify and discard redundant features that contribute little or nothing to blade icing detection, in this study we adopt the Pearson correlation coefficient. 2) To avoid multicollinearity, which could lead to severe overfitting of the detection models, we replace the highly correlated features with their average values.

4) Missing Data Processing: Due to sensor or communication errors, the SCADA system sometimes fails to collect data. Thus, we present two methods for collecting missing values: 1) linear interpolation is used if the missing values last for a very short period of time, such as only a few points; 2) the history data of the same period are used to fill the gap if the missing values last for a long period, such as several hours or even several days. This is because the interpolation data can mislead the detection model, resulting in high inaccuracy.

C. Multiscale Wavelet Decomposition Module

Since wind turbine is a complex system with fluctuating operating conditions, the time series data collected by the wind turbine SCADA system are complex and varied [8]. To this end, we adopt wavelet decomposition to analyze the time series data not only in the time domain but also in the frequency domain to fully exploit the richer underlying information.

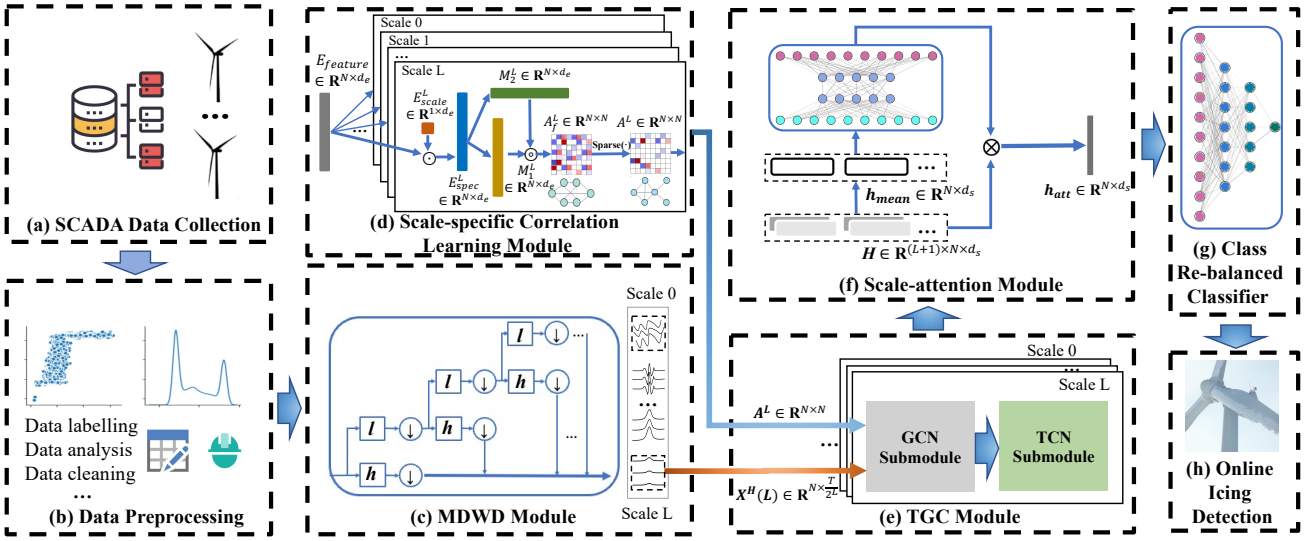


Fig. 1. The overall framework of blade icing detection model MWGCN for wind turbines.

The multiscale wavelet decomposition module is capable of extracting multiscale time-frequency features from an input time series such as low- and high-frequency subseries and a scale-by-scale basic using the Multilevel Discrete Wavelet Decomposition algorithm (MDWD) [50], which discovers the signal variance across different scales. Figure 1 (c) illustrates the structure of the MDWD module.

In the MDWD module, we denote each input signal segment in the SCADA dataset by $x_c = [x_{c,1}, x_{c,2}, ..., x_{c,T}] \in \mathbb{R}^T$, $c = 1, 2, 3, ..., N$, where N is the number of signals, and T is the length of the input segment (window size). Therefore, $X = [x_1, x_2, ..., x_N] \in \mathbb{R}^{N \times T}$ represents the input multivariate time series segment. In addition, the low and high subseries generated at the i-th scale are represented by $x^l_{(i)}$ and $x^h_{(i)}$, respectively. On the $(i+1)$-th scale, a low-pass filter $l = [l_1, l_2, ..., l_K]$ and a high-pass filter $h = [h_1, h_2, ..., h_K]$ ($K \ll T$) are adopted to convolute the approximate coefficients (low-frequency subseries) of the upper scale as:

$$\begin{aligned} a^l_n(i+1) &= \sum_{k=1}^{K} x^l_{n+k-1}(i) \cdot l_k, \\ a^h_n(i+1) &= \sum_{k=1}^{K} x^l_{n+k-1}(i) \cdot h_k, \end{aligned} \tag{2}$$

where $x^l_n(i)$ is the n-th element of the approximate coefficients (low frequency subseries) at the i-th scale, $i = 1, ..., L$, L is the number of MDWD levels. The approximate coefficients $x^l(i+1)$ and the detailed coefficients (high-frequency subseries) $x^h(i+1)$ on the $i+1$ scale are generated from the 1/2 down sampling of the intermediate variables $a^l(i+1) = [a^l_1(i+1), a^l_2(i+1), ...]$ and $a^h(i+1) = [a^h_1(i+1), a^h_2(i+1), ...]$. Then, the concatenated segment for each signal can be represented by $X_c = [x_c, x^h(1), ..., x^h(L)]$. Thus, the concatenated segment for all input data is $X_{input} = [X, X^H(1), ..., X^H(L)]$. In addition, to simplify the representations, the original multivariate time series data is denoted by the wavelet detail coefficient at the 0-th scale, which is $X^H(0)$.

D. Scale-specific Correlation Learning Module

After obtaining the wavelet detail coefficients of the input signals on multiple scales, a scale-specific correlation learning module is adopted, in order to automatically learn the scale-specific inter-variate relationships of the input segment on each scale, which was proposed in [51]. To our knowledge, existing wavelet-based models [8], [33] do not specifically consider scale-specific inter-variate correlations, which can be different patterns at different time and frequency domain scales. As a result, learning multiple scale-specific inter-variate correlations is required. The general structure of the scale-specific correlation learning module is shown in Figure 1 (d).

It is difficult to train the model since simply constructing an adjacency matrix for each scale requires high computing capacity and introduces a lot of noise. Consequently, the adjacency matrix is built using two types of embedding vectors: the shared feature embedding $E_{feature} \in \mathbb{R}^{N \times d_e}$ and the unique scale embedding for each scale $E^i_{scale} \in \mathbb{R}^{1 \times d_e}$, $i = 0, 1, ..., L$, which contain shared and scale-specific information, respectively, and d_e is the embedding dimension. Both $E_{feature}$ and E_{scale} are randomly initialized and updated during the training phase. The adjacency matrix is then calculated on the i -th scale as follows:

$$\begin{aligned} E^i_{spec} &= E_{feature} \odot E^i_{scale} \\ M^i_1 &= \left[\tanh\left(E^i_{\text{spec}}\theta^i\right)\right]^T \\ M^i_2 &= \tanh\left(E^i_{\text{spec}}\varphi^i\right) \\ A^i_f &= ReLU\left(M^i_1 M_2 k - \left(M^i_1 M^i_2\right)^T\right) \\ A^i &= Sparse\left(Softmax\left(A^i_{full}\right)\right) \end{aligned} \tag{3}$$

where E^i_{spec} is the embedding of the i-th scale adjacency matrix, which is then weighted by two trainable parameters

θ^i and φ^i to generate two matrix factorization vectors M_1^i and M_2^i. We then obtain the raw adjacency matrix on the i-th scale A_{full}^i. A_{full}^i is finally modified to be sparse by the function $Sparse(\cdot)$ in order to make the model more robust, minimize computation, and reduce the impact of noise. :

$$Sparse(A_{lr}^k) = \begin{cases} A_{lr}^k, & A_{lr}^i \in Top\left(A_{l*}^i, \tau\right) \\ 0, & A_{lr}^i \notin Top\left(A_{l*}^i, \tau\right) \end{cases} \tag{4}$$

where τ is the threshold of $Top(\cdot)$ function, i.e., the number of neighbors of each node in the graph.

Thus, we can obtain the adjacency matrix on each scale, which indicates the inter-variate relationships of each wavelet scale $A = [A^0, A^1, A^2, ..., A^L]$.

E. Temporal Graph Convolutional Module

The temporal graph convolutional module, which consists of L TCNs and GCNs for each scale, is introduced with the above-mentioned adjacency matrix set $A = [A^0, A^1, A^2, ..., A^L]$ obtained by the scale-specific correlation learning module, and multiscale wavelet input $X_{input} = [X, X^H(1), ..., X^H(L)]$ generated by the MDWD module. Unlike other RNN-like modules, such as LSTM and GRU, which involve a delay between the end of the forward pass of the previous time step and the start of the forward pass of the next time step, the parallel computing capacity of the TCN submodule could improve the model efficiency in both training and testing stages.

In this module, the input data at each time step is first built by a GCN submodule to obtain the input of the TCN submodule at different scales:

$$h_t^i = GCN_{\text{in}}^i\left(x_t^i, A^i\right) + GCN_{\text{out}}^i\left(x_t^i, \left(A^i\right)^T\right) \tag{5}$$

where h_t^i is the input of the i-th scale TCN at time step t, x_t^i is the wavelet input of the i-th scale at time step t, $GCN(\cdot)_*$ denotes the GCN with the following graph convolution operation:

$$x *_G \theta = \sigma\left(\theta\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}\right)x\right) \tag{6}$$

where G denotes the graph represented by the weighted adjacency matrix A, x is the representation of the nodes that contain the N signals of the specific wavelet scale. σ is the activation function, θ is a trainable parameter matrix, $\widetilde{A} = I_n + A$ is the adjacency matrix with self-connection, and $\widetilde{D}$ is the diagonal degree matrix of $\widetilde{A}$.

We introduce the TCN submodule consisting of l dilated causal convolution layers, and the dilated causal convolution is calculated as follows:

$$x^{(i)} \star_t F^{(i)} = \sum_{c=1}^{C}\sum_{k=1}^{K} F^{(i)}[k,c] \odot x_{t-d\times(k-1)}^{(i)}[c] \tag{7}$$

where $x^{(i)}$ denotes the input of the i-th dilated causal convolution layers, $\star_t$ represents a convolution operation at time step t, $\odot$ is element-wise multiplication, and $F^{(i)}$ is the convolution filter.

Then, the outputs of the GCN submodule at the i-th scale $\left\{h_1^i, h_2^i, ..., h_{\frac{T}{2^i}}^i\right\}$ are fed into the TCN submodule to obtain the scale-specific representations h^i of the i-th wavelet scale:

$$h^i = TCN^i([h_1^i, h_2^i, ..., h_{\frac{T}{2^i}}^i]), \tag{8}$$

where $TCN^i(\cdot)$ is the TCN for the wavelet detail coefficient on the i-th scale, and the size of h^i is set to d_s.

F. Scale-attention Module

The scale-specific features $H = [h^0, h^1, h^2, ..., , h^L] \in \mathbb{R}^{(L+1)\times N\times d_s}$ are processed by the scale attention module to determine the importance of temporal patterns of different wavelet scales and capture cross-scale correlations.

First, an average pooling operation is performed on the scale dimension to aggregate scale-specific information of the N signal variables between different scales, which can reveal the cross-scale correlations for the representations:

$$h_{mean}^j = \frac{\sum_{i=0}^{L} H_j^i}{K} \tag{9}$$

where $h_{mean}^j \in \mathbb{R}^{1\times d_s}$ is the aggregate representation of the signal variable $j = 1, 2, ..., N$, and the aggregate global representation is $h_{mean} \in \mathbb{R}^{N\times d_s}$.

Then, several fully connected layers are stacked to compute the attention weights of each scale:

$$\begin{aligned} a_{in} &= ReLU(\boldsymbol{W_{in}} h_{mean} + b_{in}) \\ a_{out} &= Sigmoid(\boldsymbol{W_{out}} a_{in} + b_{out}) \end{aligned} \tag{10}$$

where $a_{out} \in \mathbb{R}^{1\times(L+1)}$ denotes the list of scale-attention weights of scale-specific features $H = [h^0, h^1, h^2, ..., , h^L] \in \mathbb{R}^{(L+1)\times N\times d_s}$. The weights of the fully connected layers are W_{in} and W_{out}, while the biases are b_{in} and b_{out}, respectively. Last, the scale-specific features are weighted by a_{out} to generate the scale-attention representation $h_{att} \in \mathbb{R}^{N\times d_s}$:

$$h_{att} = ReLU(a_{out} \otimes H) \tag{11}$$

G. Class Rebalancing Blade Icing Classifier

The scale-attention representation h_{att} is input to a class rebalancing classifier for the final blade icing detection. This classifier consists of a fully connected network composed of f fully connected layers with the ReLU and Sigmoid activation functions:

$$
\begin{aligned}
o_0 &= ReLU(\boldsymbol{W_0} h_{att} + b_0) \\
o_{i+1} &= ReLU(\boldsymbol{W_i} o_i + b_i) \\
\hat{y} &= Sigmoid(\boldsymbol{W_f} o_f + b_f)
\end{aligned} \tag{12}
$$

where o_0 is the output of the first fully connected layer, o_{i+1} is the output of the i-th fully connected layer, and $\hat{y}$ is the probability of the true label (blade icing) normalized to $(0, 1)$ by $Sigmoid(\cdot)$ function.

In addition, during the operation of wind turbines, wind turbines operate normally most of the time, but with a small period of time when the blades are icing, resulting in a large imbalance between these two labels, normal and icing. If the detection model is unable to manage the data imbalance properly, the model's prediction outputs will be biased. Therefore, it is necessary to alleviate the data imbalance of blade icing detection. Thus, it is necessary to deal with data imbalance.

Data resampling is a data-level approach that includes both undersampling and oversampling [52]. However, undersampling causes information loss in the original datasets, while oversampling can easily lead to model overfitting. Yi et al. proposed the Minority-Clustering Synthetic Minority Oversampling Technique approach (MC-SMOTE) to improve data the oversampling method, which employed minority sample clustering to alleviate the imbalanced label nature of wind turbine blade icing [25], while its performance can be easily affected by data sizes [29]. Recently, some works proposed some class-balanced-loss methods, which do not involve the operations on datasets [53]–[55]. These methods adopt novel losses to improve the generalization of the minority data, and this method can be easily implemented by the loss functions to deep learning models, and do not increase the computational cost. Thus, in this work, we adopt the focal loss [53], which adds a modulating weight to cross-entropy loss to reduce the relative loss for well-classified samples (normal samples) and focus on hard-classified samples (icing samples) to achieve an improvement in class rebalancing.

$$
\mathrm{FL}\left(p_t\right) = -\alpha_t \left(1 - p_t\right)^{\gamma} \log\left(p_t\right) \tag{13}
$$

where p_t is the probability of the true label (blade icing), α_t is the parameter to reconcile the weight ratios of positive and negative samples, γ is the parameter to reduce the loss value for negative samples (normal samples).

IV. Experiments

This section will carry out experiments to evaluate the proposed method. It first presents the experimental settings, followed by data preprocessing, evaluation metrics, implementation details, baseline comparison, ablation study, and sensitivity analysis.

A. Experimental Setups

1) Implementation Settings: The Pytorch deep learning framework (v1.10.2) was used to implement the neural networks of the proposed model. All experiments were carried out on an Ubuntu (v.18.04.5) server equipped with a 2.20GHz Intel Xeon CPU, NVIDIA Tesla T4 GPU, and 16GB of RAM. Adam optimizer was used in model training [56]. We used the same parameters as the corresponding baselines, instead of tuning hyperparameters. The source code is made publicly available on Github[1].

2) Datasets: The experimental data are SCADA data from two wind turbines installed in the Inner Mongolia Autonomous Region, China, which were provided by the Chinese Ministry of Industry and Information Technology (MIIT)[2]. The SCADA data were sampled every 7 seconds, with a total operating time of 696 and 306 hours, respectively. There are more than hundreds of sensors inside wind turbines that provide readings

Algorithm 1 Learning process for MWGCN

Input: $Epoch$, L, X, N, T, Y
Output: Predicted label $\hat{y}$

Initialisation :
Randomly initialise trainable parameters $E_{feature}, E^i_{scale}$
$A = CorrelationLearning(E_{feature}, E^i_{scale})$ /*Eq.(3,4)*/
for $epoch = 1$ to $Epoch$ **do**
  $X = Normalization(X)$ /*Eq.(1)*/
  $X^L(0) = X$
  for $l = 1$ to L **do**
    $X^H(l), X^L(l) = MDWD(X^L(l-1))$ /*Eq.(2)*/
  end for
  $X^H(0) = X$
  $X_{input} = [X^H(0), X^H(1), ..., X^H(L)]$
  for $l = 0$ to L **do**
    for $t = 0$ to $\frac{T}{2^l}$ **do**
      $h^l_t = GCN^l_{in}(x^l_t, A^l) + GCN^l_{out}(x^l_t, (A^l)^T)$
    end for /*Eq.(5,6)*/
    $h^l = TCN^L(h^l_1, h^l_2, ..., h^l_{\frac{T}{2^l}})$ /*Eq.(7,8)*/
  end for
  $H = Concat[h^0, h^1, ..., h^L]$
  $h_{att} = SelfAttention(H)$ /*Eq.(9,10,11)*/
  $\hat{y} = Classifier(h_{att})$ /*Eq.(12)*/
  $Loss = FL(\hat{y}, y)$ /*Eq.(13)*/
  Update the model by descending the gradient of $Loss$.
end for

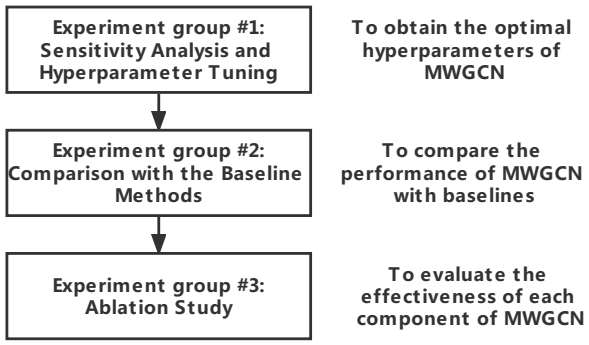


Fig. 2. Experiment groups.

[1]http://github.com/ryanlaics/MWGCN
[2]http://106.38.3.217/pages/datadown/dataset/dataSet06.jsp

to the SCADA system. Wind turbine engineers have identified 26 signals associated with blade icing conditions based on their industrial experience and knowledge. Table I shows the details of the datasets. In the following, we refer to the two datasets as Turbine-1 and Turbine-2, respectively.

TABLE I
DESCRIPTION OF THE DATASETS.

No.	Parameter	No.	Parameter
1	wind speed	14	temperature of pitch motor 1
2	generator speed	15	temperature of pitch motor 2
3	active power	16	temperature of pitch motor 3
4	live wind direction	17	switching temperature of pitch 1
5	nacelle temperature	18	switching temperature of pitch 2
6	yaw position	19	switching temperature of pitch 3
7	yaw speed	20	charger's DC current of pitch 1
8	angle of pitch 1	21	charger's DC current of pitch 2
9	angle of pitch 2	22	charger's DC current of pitch 3
10	angle of pitch 3	23	horizontal acceleration
11	speed of pitch 1	24	vertical acceleration
12	speed of pitch 2	25	environment temperature
13	speed of pitch 3	26	average wind direction with 25s

The data processing approach used in this study is based on the relative literature [8]. As shown in Table II, wind turbines operate normally most of the time, but with a small period of time when blades are icing, resulting in a large imbalance between these two labels, normal and icing. In addition, due to some unidentified errors, there are some data without labels. Therefore, we first clean up these data.

TABLE II
LABEL DISTRIBUTION OF THE DATASETS.

Dataset	Total	Normal	Icing	Invalid
Turbine-1	393,886	350,255	23,846	19,785
Turbine-2	190,494	168,930	10,638	10,926

Also, we segment the SCADA signals by fixed time steps, which will be used as input to the model. The fixed-length time step is referred to as the window size in the following. Each segmented sample is paired with a binary label that indicates whether the blades are iced or not during window size.

3) Metrics: A turbine operates normally most of the time, with only a small percentage of the time the blades are icing, resulting in significantly more negative samples (normal states) than positive samples (icing states). As a result, the positive and negative samples in the datasets collected by the SCADA system are imbalanced (with a ratio of approximately 1:15). In this study, precision is defined as the percentage of correctly detected blade icing samples in all detected positive samples (icing states), while recall (i.e., True Positive Rate, TPR) is defined as the percentage of correctly detected blade icing samples in all positive samples. Since too many false positive samples can lead to unnecessary manual maintenance and checks performed on the turbine, we adopt fall-out (i.e., False Positive Rate, FPR). Beside, we refer to the literature [8], [33], [57], for such imbalanced datasets, F1 score, which is obtained by calculating the harmonic mean of precision and recall, can increase the balance of precision and recall where two metrics are equally important, and provide a more comprehensive model evaluation. As a result, we calculate precision, recall, fall-out, and F1 score to evaluate the model performance, with the F1 score serving as the primary assessment metric in the experiments.

The results of blade icing detection can be described by the confusion matrix, which is a 2×2 matrix consisting of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Note that the samples labeled with the icing condition are designated as positive samples, while the normal samples are designated as negative samples. Then, the evaluation metrics of blade icing detection models are calculated based on TP, TN, FP and FN, including precision, recall (TPR), fall-out (FPR) and F1 score, which are described in the following:

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ Recall = TPR &= \frac{TP}{TP+FN} \\ Fall\text{-}out = FPR &= \frac{FP}{FP+TN} \\ F1score &= \frac{2 \times Precision \times Recall}{Precision + Recall} \end{aligned} \tag{14}$$

4) Model Settings: The window size is set to 32 for data preprocessing. The input multivariate time series segments are decomposed at 3 scales using the Haar wavelet transform for the multiscale wavelet decomposition module. The embedding size d_e for the scale-specific correlation learning module is set to 4, and the maximum number of neighbors for each node τ is set to 10. The number of graph convolution layers is set to 2 for the GCN submodule, and the kernel sizes of the TCN convolution layers at different scales are set to 32, 16, 8, and 4, respectively. The class rebalancing blade icing classifier has 3 fully connected layers with 26, 13 and one node(s), and the focal loss function's α_t and γ are set to 0.25 and 3, respectively. Sensitivity analysis of important hyperparameters is presented in Section IV-C.

5) Computational Complexity: During the test phase, the computationally intensive components of the MWGCN are the MDWD module, the GCN submodule, the TCN submodule, the scale-attention module, and the class rebalanced classifier, and their computational complexity can be defined as $O(N \cdot T \cdot L)$, $O(N^2 \cdot T \cdot L)$, $O(T^2 \cdot N \cdot d_s \cdot L)$, $O(d_s^2)$ and $O(d_s^2)$, where N is the number of signals, T is the length of the window size, L is the number of MDWD levels, d_s is the embedding size of h^i and h_{att}, respectively.

B. Experimental Results and Discussion

1) Baseline Models: We will use the following eight state-of-the-art models as a baseline to evaluate our model.

We will use the following eight state-of-the-art models as a baseline to evaluate our model. **MSPCA** is a moving window generalized likelihood ratio test method based on multiscale principal component analysis [58]. **RF-XGBoost** is a wind turbine fault detection approach that combines random forest and XGBoost [59]. **LSTM** is an LSTM-based fault detection network model. **MLSTM-FCN** is an efficient

multivariate time series classification model [60]. **FCNN** is a fully convolutional neural network [61]. **WaveletFCNN** is a wavelet based fully convolutional neural network [33]. **SSENET** is built on the basis of stacked convolutional neural network (CNN) blocks with dense connections, channel and feature attention modules [29]. **MRCNN** is a wavelet based multilevel convolutional recurrent neural network [8].

2) Comparison with the Baseline Methods: Table III presents the blade icing detection performance in terms of precision (P), recall (R), fall-out (F) and F1 score (F1), of the proposed MWGCN and the baselines on the Turbine-1 dataset. The results show that the proposed MWGCN has the highest performance in terms of precision, fall-out and the main metric F1 score among all models. Compared to the best baseline, MRCNN, MWGCN improves the F1 score, precision and fall-out by 17.2%, 43.4% and 66.7%, respectively. FCNN obtains the highest recalls, however, the its low precision and highest fall-out indicate that this model tends to output positive labels, which leads to excessive false alarms. Similarly, although FCNN, MRCNN and MLSTM-FCN outperform MWGCN in terms of recall, they also have poor performance in precision and fall-out, resulting in the misclassification of many wind turbine's regular operation periods as blade icing states, resulting in significant additional maintenance costs. Table III shows the blade icing detection accuracy of the proposed MWGCN and the baselines on the Turbine-2 dataset. The proposed MWGCN also outperforms the best baseline MRCNN in terms of precision, fall-out and the main metric F1 score, with improvements of 27.5%, 50.0% and 11.3%, respectively. The high fall-outs of other baseline models indicate that these models do not adequately address the balance between precision and recall. In contrast, the proposed MWGCN has balanced performance and consequently achieves the best F1 score in both datasets.

TABLE III
BASELINE COMPARISON ON THE DATASETS

Dataset	Turbine-1				Turbine-2			
	P	R	F	F1	P	R	F	F1
MSPCA	0.33	0.30	0.05	0.31	0.30	0.66	0.11	0.41
RF-XGBoost	0.33	0.79	0.13	0.47	0.31	0.71	0.12	0.43
LSTM	0.27	0.57	0.12	0.37	0.32	0.30	0.05	0.31
MLSTM-FCN	0.37	0.78	0.11	0.50	0.37	**0.95**	0.12	0.53
FCNN	0.26	**0.82**	0.18	0.39	0.38	0.73	0.09	0.50
WaveletFCNN	0.21	0.60	0.18	0.31	0.19	0.84	0.25	0.31
SSENET	0.53	0.59	0.04	0.56	0.35	0.90	0.12	0.50
MRCNN	0.53	0.81	0.06	0.64	0.69	0.72	0.02	0.71
MWGCN	**0.76**	0.74	**0.02**	**0.75**	**0.88**	0.72	**0.01**	**0.79**

C. Sensitivity Analysis and Hyperparameter Tuning

This section performs a sensitivity analysis to examine the impact of key hyperparameters on the MWGCN, which is to find the optimal hyperparameters and optimize its structure. We evaluated the following three key hyperparameters using the Turbine-1 and Turbine-2 datasets: 1) the MDWD scale; 2) the maximum neighbor number of each node in the scale-specific correlation learning module; and 3) the number of graph convolution layers in the GCN submodule. We define the range of possible values for all hyperparameters based on the literature [8] and practical experience.

To study the impact of the MDWD scale, we implemented four variants of MWGCN with different MDWD scales (1, 2, 3, and 4). To explore the influence of the maximum number of neighbors in the scale-specific correlation learning module, we choose five different parameters that range from 5 to 26. Finally, to evaluate the importance of the number of graph convolution layers in the GCN submodule, we set the number of GCN layers to 1, 2, 3, 4, and 5. Therefore, there are a total of 14 groups of sensitivity experiments for the Turbine-1 and Turbine-2 datasets.

The results of the sensitivity analysis on two datasets are shown in Figures 3, respectively. On the Turbine-1 dataset, the highest F1 score is obtained when the MDWD scale is 2 and 3. If some blade-icing states were not detected, this could lead to economic loss. Therefore, we adopt the MDWD scale of 3 because its recall is higher than the scale of 2's. In addition, the F1 score is the best when the number of neighbors is set to 5. Moreover, with two GCN layers, the proposed MWGCN achieves the best performance on the Turbine-1 dataset. For the Turbine-2 dataset, we can observe similar results for the MDWD scale and the number of GCN layers, while the F1 score reaches its highest value when the number of neighbors is set to 10. Based on the above results, we were able to find the optimal hyperparameter settings of MWGCN for each dataset.

D. Ablation Study

The ablation study is to see if some of the model's components are effective and indispensable. For example, if a new component is introduced to the model, we should compare the results obtained by the model with that component removed to the results obtained by the model with that component added to evaluate whether that component is effective.

Specifically, we implement several variants of the MWGCN by removing some components, and then observe their impact on the performance of the MWGCN. The variants of MWGCN are constructed as follows:**MWGCN-1**: MWGCN without the multiscale wavelet decomposition module; **MWGCN-2**: MWGCN without the scale-specific correlation learning module; **MWGCN-3**: MWGCN without the scale-attention module; **MWGCN-4**: MWGCN without the class rebalancing loss function.

Table IV shows the results of the ablation analysis on the two datasets. We can observe that the complete MWGCN outperforms all other variants in terms of F1 score. This result demonstrates the effectiveness of all components in the proposed MWGCN. As illustrated in Table IV, it indicates that the performance of all four MWGCN variants is worse than that of the complete MWGCN, implying that all the primary components of the proposed MWGCN have been shown to improve the performance of the model, and the performance of MWGCN will degrade as one of its components is removed. Furthermore, there are some variances in impacts in different datasets. MWGCN-3 has the worst performance on the Turbine-1 dataset, suggesting that the scale-attention module, which has been removed from MWGCN-3, may have a significant impact on the results. However, in Turbine-2,

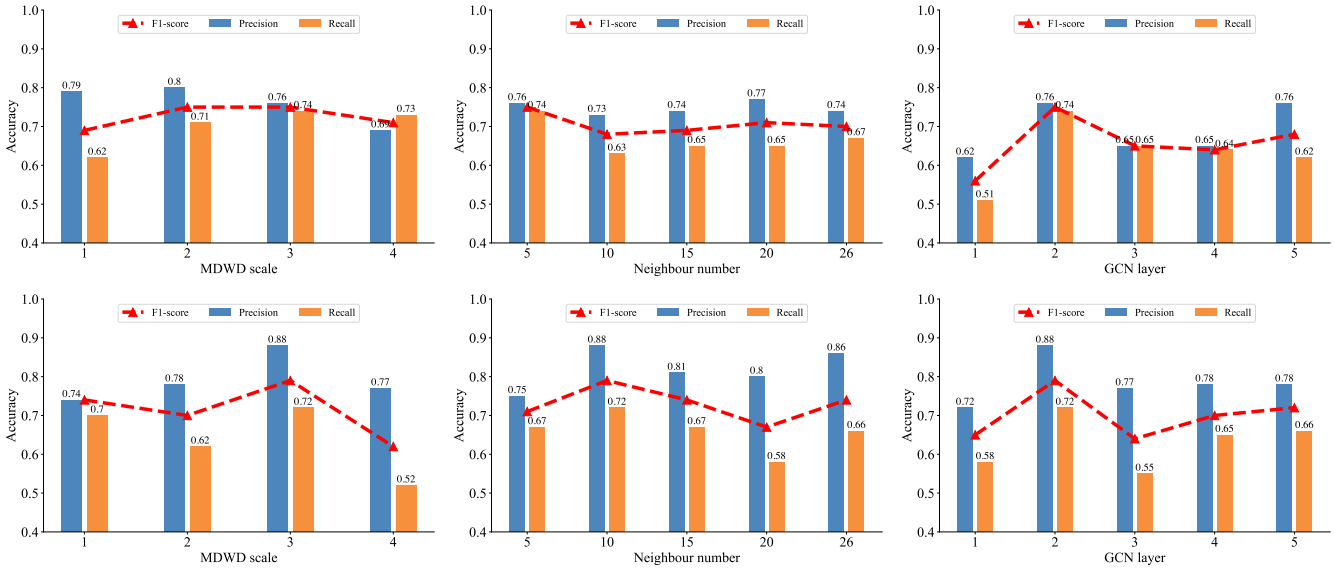


Fig. 3. Results of the sensitivity analysis on the Turbine-1 dataset(upper line) and the Turbine-2 dataset(lower line).

this module has the second most important impact of all components. The explanation for this could be that Turbine-1's decomposed input from multiple scales has more variation than Turbine-2, which requires the scale-attention module to determine the weights of each scale to improve the classifier's performance. Furthermore, MWGCN-2 performs the worst on the Turbine-2 dataset, suggesting that inter-variate correlations at different scales might be quite different, implying that the scale-specific correlation learning module may critical in this turbine dataset.

TABLE IV
ABLATION ANALYSIS OF THE PROPOSED MWGCN

Dataset	Turbine-1				Turbine-2			
	P	R	F	F1	P	R	F	F1
MWGCN-1	0.80	0.65	0.01	0.71	0.77	0.70	0.02	0.74
MWGCN-2	0.80	0.67	0.01	0.73	0.67	0.55	0.02	0.60
MWGCN-3	0.76	0.60	0.02	0.67	0.69	0.75	0.03	0.72
MWGCN-4	0.80	0.62	0.01	0.70	0.78	0.59	0.01	0.67
MWGCN	0.76	0.74	0.02	**0.75**	0.88	0.72	0.01	**0.79**

E. Discussion

1) What are the main differences between the proposed MWGCN and the others?: In this paper, we have developed a blade icing detection model based on the deep learning method that takes into account rich and implicit information on different scales of the time and frequency domains. Compared to MSPCA, RF-XGBoost, LSTM, MLSTM-FCN, FCN and SSENET, which directly model the temporal pattern of the original signals, the proposed MWGCN uses the discrete wavelet decomposition module to be able to capture rich and implicit information in the time and frequency domains. More specifically, traditional convolutional neural networks can learn information only within a fixed receptive field constrained by the filter size, while the multisacle wavelet module provides much more helpful information among multiple time and frequency scales for the convolutional neural networks.

WaveletFCNN and MRCNN both use a similar discrete wavelet decomposition module, but they were not designed to exploit the intervariable correlations of multiscale SCADA data. In contrast, the proposed MWGCN uses the graph convolution module and the scale attention module to model scale-specific correlations. These modules are incorporated with the structural information and the information of each scale. Therefore, the proposed MWGCN has achieved a greater ability to capture the signal information from multivariate time series in both the time and frequency domains. These are the main reasons for the effectiveness and superiority of the proposed MWGCN over other methods.

2) How does the multiscale wavelet transformation work?: In this work, we adopt the scale-specific correlation learning module to capture scale-specific intervariable correlations of input data. Thus, we could collect the adjacency matrice of the scale-specific intervariable correlations on the datasets, and further analyze the difference of the adjacency matrice at different scales. We take the wind speed for example, to analyze how the scale-specific correlation learning module works at different scales. At scale 0, the five most important correlations are the temperatures of pitch motor 1,2 and 3, the yaw position, and the switching temperature of pitch 3; at scale 1, are the angles of pitch 1, 2, and 3 and the temperatures of pitch motor 1 and 2; at scale 2 and 3, are the angles of pitch 1, 2 and 3, the environment temperature, and the nacelle temperature.

We can see that, at scale 0, the wind speed is more related with the temperature of pitch motors, while the correlations between the wind speed and the angles of pitches become more important at higher scales. The reason may lie that the information at scale 0 presents the prominent temporal trend variations of the multivariate time series, while the information at higher scales is the detail coefficients that may capture the transient localized dynamic patterns. In our work, the scale-specific correlation learning module automatically learns the inter-variate relationships of the input data at multiple wavelet

scales, thus improving the classification performance.

3) How does the hyperparameters influence the performance?: From Figure. 3, we can simply analyze how these hyperparameters influence performance.

When it comes to the MDWD scale, the proposed MWGCN performs best when the MDWD scale is set to 3, and it degrades when the MDWD scale is increased. The reasons for this could be that the model may be unable to process and use the information of higher-level detail coefficients, which becomes extra noise, thus degrading the deep learning model's performance.

As for the maximum number of neighbors in the scale-specific correlation learning module, we can see that the sparsification strategy can be effective, with less computation cost of the graph convolution, reduce the impact of noise, and make the model more robust. It shows that the information aggregation of only a few nodes may present even better performance than all nodes, indicating that a lot of noise may be involved in the aggregated information of a node from redundant nodes, and the sparsification strategy can alleviate this kind of noise and make the model robust.

In terms of the number of GCN layers, the experimental results show that the model with deeper GCN layers does not perform better. [62], [63] investigated the graph representation learning and discovered that GCN models cannot be stacked as deeply as CNN models. The reasons could be that the GCN's information aggregation mechanism allows it to be considered as a low-pass filter [64], and as the number of the GCN layers increases, the signals of each node become smoother, which is an inherent advantage of the GCN. However, performing this signal smoothing operation too many times may cause the signals to converge and become indistinguishable, resulting in loss of node characteristics diversity. As a result, the number of GCN layers should be carefully chosen.

V. Conclusion and Future Work

Blade icing detection is crucial to the efficiency of wind turbine power generation. In this paper, we presented a wavelet-driven multiscale graph convolutional network, MWGCN, to detect the icing condition of wind turbines. To improve detection performance, we introduced a discrete wavelet decomposition into the proposed MWGCN, which is capable of capturing multiscale features of SCADA data in both the time and frequency domains. In addition, we implemented a scale-specific correlation learning module that can infer scale-specific intervariable relationships of SCADA data at multiple time and frequency scales. In addition, we used a temporal convolution sub-module to improve the efficiency of parallel processing without sacrificing accuracy. A scaling attention module was also added to capture inter-scale correlations. The proposed model incorporates a class rebalancing classifier to address the class imbalance problem of the input data. Finally, we conducted comprehensive experiments to evaluate the proposed model using real-world datasets, and the results verify its effectiveness and superiority over others. There are three interesting directions for future work. First, we will continue to improve the model, for example, by investigating how to effectively exploit cross-scale signal correlations, as this can be useful for multivariate feature extraction. Second, hyperparameter tuning is crucial to create efficient models for different types of wind turbines and to adapt to different environmental conditions. This can be done automatically and adaptively, for example, by reinforcement learning to learn the hyperparameter settings through the interaction between the environment and the agent. This will be the other direction of future research. Third, because the proposed MWGCN is a supervised batch approach that requires the whole labeled dataset to be a priori, we will improve and modify our model in unsupervised and semi-supervised settings. We leave these directions for future work seeking to improve the model.

References

[1] L. Gao, T. Tao, Y. Liu, and H. Hu, "A field study of ice accretion and its effects on the power production of utility-scale wind turbines," *Renewable Energy*, vol. 167, pp. 917–928, 2021.
[2] E. W. E. Association *et al.*, *EU Energy policy to 2050*. EWEA, 2011.
[3] L. Battisti, *Wind turbines in cold climates: Icing impacts and mitigation systems*. Springer, 2015.
[4] R. W. Gent, N. P. Dart, and J. T. Cansdale, "Aircraft icing," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 358, no. 1776, pp. 2873–2911, 2000.
[5] P. Wang, W. Zhou, Y. Bao, and H. Li, "Ice monitoring of a full-scale wind turbine blade using ultrasonic guided waves under varying temperature conditions," *Structural control and health monitoring*, vol. 25, no. 4, p. e2138, 2018.
[6] C. Q. G. Muñoz, F. P. G. Márquez, and J. M. S. Tomás, "Ice detection using thermal infrared radiometry on wind turbine blades," *Measurement*, vol. 93, pp. 157–163, 2016.
[7] C. Q. G. Muñoz, A. A. Jiménez, and F. P. G. Márquez, "Wavelet transforms and pattern recognition on ultrasonic guides waves for frozen surface state diagnosis," *Renewable Energy*, vol. 116, pp. 42–54, 2018.
[8] W. Tian, X. Cheng, G. Li, F. Shi, S. Chen, and H. Zhang, "A multilevel convolutional recurrent neural network for blade icing detection of wind turbine," *IEEE Sensors Journal*, vol. 21, no. 18, pp. 20 311–20 323, 2021.
[9] K. Wei, Y. Yang, H. Zuo, and D. Zhong, "A review on ice detection technology and ice elimination technology for wind turbine," *Wind Energy*, vol. 23, no. 3, pp. 433–457, 2020.
[10] P. Roberge, J. Lemay, J. Ruel, and A. Bégin-Drolet, "A new atmospheric icing detector based on thermally heated cylindrical probes for wind turbine applications," *Cold Regions Science and Technology*, vol. 148, pp. 131–141, 2018.
[11] G. d. N. P. Leite, G. T. M. da Cunha, J. G. dos Santos Junior, A. M. Araújo, P. A. C. Rosas, T. Stosic, B. Stosic, and O. A. Rosso, "Alternative fault detection and diagnostic using information theory quantifiers based on vibration time-waveforms from condition monitoring systems: Application to operational wind turbines," *Renewable Energy*, vol. 164, pp. 1183–1194, 2021.
[12] J. M. P. Pérez, F. P. G. Márquez, and D. R. Hernández, "Economic viability analysis for icing blades detection in wind turbines," *Journal of Cleaner Production*, vol. 135, pp. 1150–1160, 2016.
[13] T. Laakso, H. Holttinen, G. Ronsten, L. Tallhaug, R. Horbaty, I. Baring-Gould, A. Lacroix, E. Peltola, and B. Tammelin, "State-of-the-art of wind energy in cold climates," *IEA annex XIX*, vol. 24, p. 53, 2003.
[14] O. Parent and A. Ilinca, "Anti-icing and de-icing techniques for wind turbines: Critical review," *Cold regions science and technology*, vol. 65, no. 1, pp. 88–96, 2011.
[15] M. L. Corradini, A. Cristofaro, and S. Pettinari, "A model-based robust icing detection and estimation scheme for wind turbines," in *2016 European Control Conference (ECC)*. IEEE, 2016, pp. 1451–1456.
[16] L. Hu, X. Zhu, J. Chen, X. Shen, and Z. Du, "Numerical simulation of rime ice on nrel phase vi blade," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 178, pp. 57–68, 2018.
[17] N. N. Davis, Ø. Byrkjedal, A. N. Hahmann, N.-E. Clausen, and M. Žagar, "Ice detection on wind turbines using the observed power curve," *Wind Energy*, vol. 19, no. 6, pp. 999–1010, 2016.
[18] Y. Qiu, Y. Feng, J. Sun, W. Zhang, and D. Infield, "Applying thermophysics for wind turbine drivetrain fault diagnosis using scada data," *IET Renewable Power Generation*, vol. 10, no. 5, pp. 661–668, 2016.

[19] J. Xu, W. Tan, and T. Li, "Predicting fan blade icing by using particle swarm optimization and support vector machine algorithm," *Computers & Electrical Engineering*, vol. 87, p. 106751, 2020.

[20] J. Tautz-Weinert and S. J. Watson, "Using scada data for wind turbine condition monitoring–a review," *IET Renewable Power Generation*, vol. 11, no. 4, pp. 382–394, 2017.

[21] J. Dai, W. Yang, J. Cao, D. Liu, and X. Long, "Ageing assessment of a wind turbine over time by interpreting wind farm scada data," *Renewable Energy*, vol. 116, pp. 199–208, 2018.

[22] P. B. Dao, W. J. Staszewski, T. Barszcz, and T. Uhl, "Condition monitoring and fault detection in wind turbines based on cointegration analysis of scada data," *Renewable Energy*, vol. 116, pp. 107–122, 2018.

[23] P. Song, Z. Yao, and C. Zhao, "Section division and multi-model method for early detection of icing on wind turbine blades," in *2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE, 2019, pp. 749–754.

[24] G. A. Skrimpas, K. Kleani, N. Mijatovic, C. W. Sweeney, B. B. Jensen, and J. Holboell, "Detection of icing on wind turbine blades by means of vibration and power curve analysis," *Wind Energy*, vol. 19, no. 10, pp. 1819–1832, 2016.

[25] H. Yi, Q. Jiang, X. Yan, and B. Wang, "Imbalanced classification based on minority clustering smote with wind turbine fault detection application," *IEEE Transactions on Industrial Informatics*, 2020.

[26] T. Tao, Y. Liu, Y. Qiao, L. Gao, J. Lu, C. Zhang, and Y. Wang, "Wind turbine blade icing diagnosis using hybrid features and stacked-xgboost algorithm," *Renewable Energy*, vol. 180, pp. 1004–1013, 2021.

[27] J. Xiao, C. Li, B. Liu, J. Huang, and L. Xie, "Prediction of wind turbine blade icing fault based on selective deep ensemble model," *Knowledge-Based Systems*, vol. 242, p. 108290, 2022.

[28] X. Liu, Z. Lai, X. Wang, L. Huang, and P. S. Nielsen, "A contextual anomaly detection framework for energy smart meter data stream," in *International Conference on Neural Information Processing*. Springer, 2020, pp. 733–742.

[29] X. Cheng, G. Li, A. L. Ellefsen, S. Chen, H. P. Hildre, and H. Zhang, "A novel densely connected convolutional neural network for sea-state estimation using ship motion data," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 5984–5993, 2020.

[30] G. Jiang, H. He, J. Yan, and P. Xie, "Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3196–3207, 2018.

[31] L. Chen, G. Xu, Q. Zhang, and X. Zhang, "Learning deep representation of imbalanced scada data for fault detection of wind turbines," *Measurement*, vol. 139, pp. 370–379, 2019.

[32] Y. Liu, H. Cheng, X. Kong, Q. Wang, and H. Cui, "Intelligent wind turbine blade icing detection using supervisory control and data acquisition data and ensemble deep learning," *Energy Science & Engineering*, vol. 7, no. 6, pp. 2633–2645, 2019.

[33] B. Yuan, C. Wang, F. Jiang, M. Long, S. Y. Philip, and Y. Liu, "Waveletfcnn: A deep time series classification model for wind turbine blade icing detection," *arXiv preprint arXiv:1902.05625v1*, 2019.

[34] X. Cheng, F. Shi, M. Zhao, G. Li, H. Zhang, and S. Chen, "Temporal attention convolutional neural network for estimation of icing probability on wind turbine blades," *IEEE Transactions on Industrial Electronics*, 2021.

[35] X. Cheng, F. Shi, X. Liu, M. Zhao, and S. Chen, "A novel deep class-imbalanced semisupervised model for wind turbine blade icing detection," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[36] J.-S. Chou, C.-K. Chiu, I.-K. Huang, and K.-N. Chi, "Failure analysis of wind turbine blade under critical wind loads," *Engineering Failure Analysis*, vol. 27, pp. 99–118, 2013.

[37] N. Dalili, A. Edrisy, and R. Carriveau, "A review of surface engineering issues critical to wind turbine performance," *Renewable and Sustainable energy reviews*, vol. 13, no. 2, pp. 428–438, 2009.

[38] A. S. Y. Alsabagh, W. Tiu, Y. Xu, and M. S. Virk, "A review of the effects of ice accretion on the structural behavior of wind turbines," *Wind Engineering*, vol. 37, no. 1, pp. 59–70, 2013.

[39] W. J. Jasinski, S. C. Noe, M. S. Selig, and M. B. Bragg, "Wind turbine performance under icing conditions," 1998.

[40] P. Antikainen and S. Peuranen, "Ice loads, case study," *The Proceedings of BOREAS V, Levi, Finland*, vol. 11, 2000.

[41] B. Tammelin and H. Seifert, "Wind energy production in cold climate," 2000.

[42] M. J. Chambers, "The estimation of continuous time models with mixed frequency data," *Journal of Econometrics*, vol. 193, no. 2, pp. 390–404, 2016.

[43] W. Xu, W. Liu, J. Bian, J. Yin, and T.-Y. Liu, "Instance-wise graph-based framework for multivariate time series forecasting," *arXiv preprint arXiv:2109.06489*, 2021.

[44] J. Wang, Z. Wang, J. Li, and J. Wu, "Multilevel wavelet decomposition network for interpretable time series analysis," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2437–2446.

[45] D. Nalley, J. Adamowski, and B. Khalil, "Using discrete wavelet transforms to analyze trends in streamflow and precipitation in quebec and ontario (1954–2008)," *Journal of hydrology*, vol. 475, pp. 204–228, 2012.

[46] J. Nobre and R. F. Neves, "Combining principal component analysis, discrete wavelet transform and xgboost to trade in the financial markets," *Expert Systems with Applications*, vol. 125, pp. 181–194, 2019.

[47] D. B. Percival, "Analysis of geophysical time series using discrete wavelet transforms: An overview," *Nonlinear time series analysis in the geosciences*, pp. 61–79, 2008.

[48] P. Chaovalit, A. Gangopadhyay, G. Karabatis, and Z. Chen, "Discrete wavelet transform-based time series analysis and mining," *ACM Computing Surveys (CSUR)*, vol. 43, no. 2, pp. 1–37, 2011.

[49] B. Yuan, C. Wang, C. Luo, F. Jiang, M. Long, P. S. Yu, and Y. Liu, "Waveletae: A wavelet-enhanced autoencoder for wind turbine blade icing detection," *arXiv preprint arXiv:1902.05625*, 2019.

[50] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," in *Fundamental Papers in Wavelet Theory*. Princeton University Press, 2009, pp. 494–513.

[51] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 753–763.

[52] F. Last, G. Douzas, and F. Bacao, "Oversampling for imbalanced learning based on k-means and smote," *arXiv preprint arXiv:1711.00837*, 2017.

[53] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[54] M. Hayat, S. Khan, S. W. Zamir, J. Shen, and L. Shao, "Gaussian affinity for max-margin class imbalanced learning," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6469–6479.

[55] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," *Advances in neural information processing systems*, vol. 33, pp. 19 290–19 301, 2020.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[57] K. Leahy, R. L. Hu, I. C. Konstantakopoulos, C. J. Spanos, A. M. Agogino, and D. T. O'Sullivan, "Diagnosing and predictingwind turbine faults from scada data using support vector machines," *International Journal of Prognostics and Health Management*, vol. 9, no. 1, 2018.

[58] M. Z. Sheriff, M. Mansouri, M. N. Karim, H. Nounou, and M. Nounou, "Fault detection using multiscale pca-based moving window glrt," *Journal of Process Control*, vol. 54, pp. 47–64, 2017.

[59] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si, "A data-driven design for fault detection of wind turbines using random forests and xgboost," *Ieee Access*, vol. 6, pp. 21 020–21 031, 2018.

[60] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate lstm-fcns for time series classification," *Neural Networks*, vol. 116, pp. 237–245, 2019.

[61] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1578–1585.

[62] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Thirty-Second AAAI conference on artificial intelligence*, 2018.

[63] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5453–5462.

[64] H. Nt and T. Maehara, "Revisiting graph neural networks: All we have is low-pass filters," *arXiv preprint arXiv:1905.09550*, 2019.