JASA ARTICLE

Nonintrusive wind blade fault detection using a deep learning approach by exploring acoustic information[a)]

Hongqing Liu,[1,b)] Wenbin Zhu,[1] Yi Zhou,[1] Liming Shi,[1] and Lu Gan[2]

[1]*School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China*

[2]*College of Engineering, Design and Physical Science, Brunel University, London UB8 3PH, United Kingdom*

ABSTRACT:

Various physical characteristics, including ultrasonic waves, active acoustic emissions, vibrations, and thermal imaging, have been used for blade fault detection. In this work, we propose using the sound produced by spinning wind blades to identify faults. To the best of our knowledge, passive acoustic information has not yet been explored for this task. In particular, we develop three networks targeting different scenarios. The main contributions of this work are threefold. First, when normal and aberrant data are available for supervised learning, an attention-convolutional recurrent neural network is designed to show the feasibility of using passive sound information to conduct fault detection. Second, in the absence of abnormal training data, we build a normal-encoder network to learn the distributions of normal data through semisupervised learning, which avoids the requirement of abnormal training data. Third, when multiple devices are used to collect the data, due to different properties of devices, there is a domain mismatch issue. To overcome this, we create an adversarial domain adaptive network to close the gap between the source and target domains. Acoustic signal datasets of actual wind turbine operations are collected to evaluate our fault detection systems. The findings demonstrate that the proposed systems offer high classification accuracy and indicate the feasibility of passive acoustic signal-based wind turbine blade fault detection with one step close to automatic detection. © *2023 Acoustical Society of America*. https://doi.org/10.1121/10.0016998

(Received 5 July 2022; revised 25 November 2022; accepted 6 January 2023; published online 24 January 2023)

[Editor: Paul J. Gendron] Pages: 538–547

I. INTRODUCTION

Wind energy is one of the fastest-growing renewable energy sources in the world today and has been widely used globally (Zhang *et al.*, 2018). A wind turbine depends on the wind to propel its blades to rotate, and then its rotation is accelerated to promote the generator and facilitate power production. The blade is the key part of a wind power generation system, and its safety and reliability play a crucial role in the operation of wind turbines (Leite *et al.*, 2018). It is well known that wind turbine blades are prone to aging or cracking due to their harsh operational environments. Damaged blades may cause the wind turbine's yaw angle and blade angle to be asymmetrical, which affects the aerodynamic performance and wind energy conversion efficiency of the wind turbine (Chen *et al.*, 2020). The maintenance costs for wind turbine blades can be very high, and fault detection is a challenging task. The operation and maintenance expenses can be decreased, and wind power development can be further advanced if blade problems are identified at an early stage and fixed quickly (Amano, 2017; Liu and Zhang, 2020b). Therefore, reducing the cost of fault detection and improving its efficiency have both economic and research relevance. When a wind turbine blade fails, the rotational speed, power, vibration frequency, and temperature of the generator all change significantly. In view of the changes in these physical characteristics, current nondamaged wind turbine blade fault detection methods mainly include active acoustic emission detection (Chacon *et al.*, 2015; Hongwu *et al.*, 2015; Liu *et al.*, 2021), vibration signal detection (Goyal and Pabla, 2016; Kashfi *et al.*, 2019), and infrared thermal imaging detection (Mori *et al.*, 2007). Tang *et al.* (2016) monitored a 45.7 m long blade and used triangulation to determine the damage location of the blade, verifying that acoustic emission detection technology can provide early warnings of wind turbine blade damage. However, acoustic emission detection technology has poor noise robustness. The detection accuracy will drastically decrease at low signal-to-noise ratios if there are numerous interference sources in the surrounding environments. Liu and Zhang (2020a) used a sparse augmented Lagrange-based algorithm to filter the acoustic emission signal and extract weak defective signals to improve the accuracy of blade fault detection. However, acoustic emission-based methods require a transmitter and receiver, which can increase maintenance costs. Vibration detection technology has high sensitivity and strong practicability in practice. González and Fassois (2016) collected a large amount of vibration data for wind turbine blades in various states. They also proposed a principal component analysis (PCA) statistical method with supervised learning to extract the characteristics of vibration signals. The effectiveness of this

[a)]The preliminary results were presented at the IEEE International Conference on Signal Processing, Communications and Computing, Xi'an, China, 2021.

[b)]Electronic mail: hongqingliu@outlook.com algorithm was demonstrated through intensive experiments. Fitzgerald *et al.* (2010) developed a time-frequency–based damage detection algorithm using blade vibration signals. By tracking the dominant frequency in the model over time, the potential damage is detected. Infrared thermal imaging detection technology is sensitive to defects on the blade surface and can perform long-distance and large-area detection. Hwang *et al.* (2017) developed continuous line laser thermal imaging technology for the damage visualization of wind turbine blades in a revolving state, which realizes full noncontact monitoring of wind turbine blade detection. Thermographic flow visualization is a noninvasive measurement technique used to identify different flow regimes. However, to date, it has not been able to visualize separate flows without explicit additional heating of the measured object. To address this issue, Dollinger *et al.* (2018) introduced a measurement method with improved sensitivity to evaluate the temporal temperature fluctuations of a sequence of thermographic images by standard deviation and analysis of selected Fourier coefficients.

Despite the above efforts, the majority of current technologies for detecting faults in wind turbine blades have limited detection effectiveness, high detection costs, and challenging data-gathering procedures. With the developments of computer hearing technology and artificial intelligence algorithms, the application of intelligent audio is receiving extensive interest, and fault detection methods based on audio signals have been gradually applied in industry (Grollmisch *et al.*, 2019). Although still in its early stages of development, this technology has begun to show promise as audio signals are noncontact and carry rich information. Additionally, the equipment used to capture audio signals is inexpensive, and the acquisition technique is straightforward, which facilitates the data collection process.

Considering these facts, we propose a wind turbine fault detection algorithm based on sound event detection. Specifically, we first investigate the detection of wind turbine blade faults using supervised learning with an attention-convolutional recurrent neural network (attention-CRNN) algorithm. The attention mechanism combined with the CRNN effectively improves fault detection accuracy. To address the scarcity problem of faulty samples during training, we then applied semisupervised learning through the normal-encoder to detect wind turbine blade flaws. By comparing the original spectrogram and the reconstructed spectrogram, the normal-encoder method offers an effective fault detection method in a higher-level abstract space. Furthermore, note that there are offsets between the audio data recorded by various devices due to different system response functions. However, deep learning techniques lack the "transfer learning" capacity to use the same model on other datasets, as they are often data-dependent (Wang and Deng, 2018). When the training and evaluation datasets come from different devices, the detection accuracy may drop significantly (Kośmider, 2021). To overcome this problem, we employ an adversarial domain adaptive network, where the network learns both domain-invariant and class-discriminative features, thereby eliminating offsets between data domains.

The rest of the paper is organized as follows. In Sec. II, we first introduce the supervised learning algorithm attention-CRNN and the semisupervised learning algorithm using the normal-encoder. In Sec. III, we study the fault detection task of the power component under equipment mismatch conditions. Next, we introduce our core dataset collection process in Sec. IV. Simulation results of extensive experiments are presented in Sec. V, followed by conclusions in Sec. VI.

II. WIND TURBINE BLADE FAULT DETECTION

The general detection system is shown in Fig. 1. Here, the extracted audio signals' features are used as the neural network model's input. The defect detection result is obtained after postprocessing the network's output. We create the attention-CRNN based on supervised learning and the normal-encoder method based on semisupervised learning to detect blade faults and test the viability of wind turbine blade fault identification based on sound event detection.

A. Feature extraction

Deep neural network has a strong ability to extract features from the original waveform. The end-to-end processing method also avoids the manual extraction of audio features. However, one-dimensional audio signals are less robust than manually extracted two-dimensional feature noise. Moreover, the end-to-end processing method requires a large amount of audio data to support network training, so the end-to-end fault detection method directly using the original waveform is the least effective. The spectrogram is also a commonly used audio signal feature, but it has a large amount of redundant information, and the linear distribution of the spectrogram will not be useful enough for feature extraction, resulting in excessive model computation and low accuracy. The process of calculating logarithmic Mel spectra

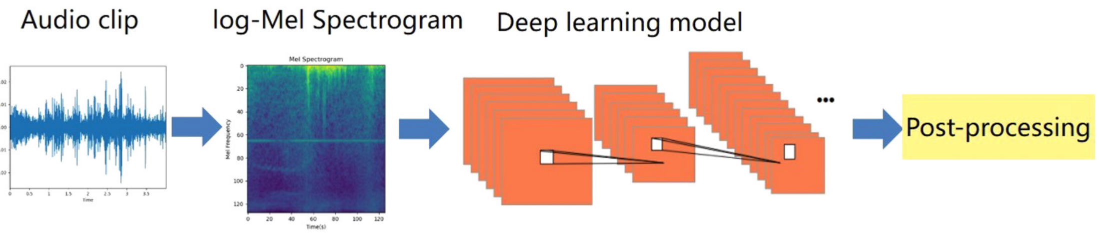

FIG. 1. (Color online) Block diagram of the wind turbine blade fault detection system.

and mel-scale frequency cepstral coefficients (MFCCs) is similar, and MFCCs are based on the logarithmic Mel spectrum to perform discrete cosine transformations to decorrelate the filter bank coefficients, so as to obtain a compressed representation of the filter bank. Many machine learning algorithms are based on the discorrelation between dataset samples, and MFCCs are very suitable for such machine learning algorithms. Yet, many deep learning algorithms are less sensitive to highly correlated inputs, and discrete cosine transform (DCT) is a linear transformation that loses a lot of useful information in the audio signal during calculation. Thus, log-Mel spectrogram is more suitable for deep neural networks.

In this work, we extract the log-Mel spectrogram of the audio signal and apply it as the input to the neural network. The log-Mel spectrogram is widely used in environmental sound recognition and the differences between normal and abnormal sounds are more distinct in the Mel domain than in the time domain. The typical frame size in speech processing ranges from 20 to 40 ms, with 50% overlap between consecutive frames (Mohamed, 2014). Here, overlap means two neighboring observation time blocks intersect each other to ensure smooth reconstruction. In this work, the size of each frame is set to 20 ms with a 10 ms stride (10 ms overlap) to produce audio signal information. After framing the input signal, a Hamming window is used to prevent spectrum leakage. We use a 512-point short-time Fourier transform (STFT) to calculate the power spectrum. After passing through the filter banks, the log-Mel spectrogram is obtained by logarithmic operation of the Mel spectrum. In our work, the 4 s long audio signal is converted into (40 128) two-dimensional features, where 128 and 40 represent the dimension of the Mel frequency and the time frame length, respectively.

B. Attention-CRNN

We first utilize an attention-CRNN (Shen *et al.*, 2018) for blade fault detection. As shown in Fig. 2, the network structure mainly includes two parts: the frequency domain attention mechanism model and the convolutional neural network-gated recurrent unit (CNN-GRU) network model.

The attention mechanism (Vaswani *et al.*, 2017) effectively improves the efficiency in image recognition, target detection, speech recognition, and speech detection. Here, the attention module was designed to ignore input spectrum frames that have minimal bearing on the detection results in favor of the more significant ones, which are multiplied with larger weights before being fed into the neural network. The frequency domain attention model consists of a fully connected layer with N hidden units and a sigmoid activation function. The input feature passes through a fully connected layer with 64 hidden units, followed by a sigmoid operation. We then normalize the weights obtained along with the frequency axis function. Finally, the frequency attention weights and the input feature are multiplied elementwise. The weighted feature $\bar{\mathbf{X}}$ is calculated by

$$\hat{\mathbf{W}}_{n,t} = \sigma(\mathbf{V_n}\mathbf{X} + b_n), \tag{1}$$

$$\mathbf{W}_{n,t} = N_f \frac{\hat{\mathbf{W}}_{n,t}}{\sum_n} \hat{\mathbf{W}}_{n,t}, \tag{2}$$

and

$$\bar{\mathbf{X}} = \mathbf{W}_{n,t} \otimes \mathbf{X}, \tag{3}$$

where $\sigma(x) = 1/(1+e^{-x})$ represents the sigmoid activation function that introduces non-linearity to increase the modeling capacity, $\mathbf{X}$ is the log-Mel spectrogram, $\mathbf{V_n}$ and b_n represent the weights and bias for the n-th hidden unit, respectively, $\hat{\mathbf{W}}_{n,t}$ is the frequency attention weight without normalization, $\mathbf{W}_{n,t}$ is the normalized result, N_f is the number of frequency points in the Mel domain, and $\otimes$ represents elementwise multiplication.

The CNN-GRU network is a two-stage model that is widely applied in sound event detection or classification tasks, and it utilizes the information of input features in time series and spatial locations. The two-dimensional convolutional neural network effectively extracts the features of the input spectral map in the spatial position, whereas the GRU gate structure enhances the generalization ability of the model, and it also works well for learning information in time series. The fully connected layer is used to classify feature information and output the final detection result.

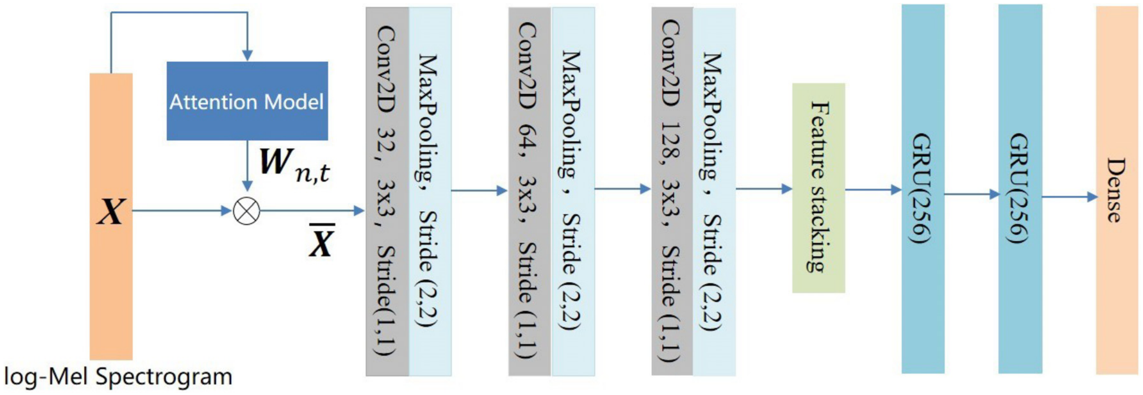
X
Attention Model
$W_{n,t}$
$\bar{X}$
log-Mel Spectrogram
Conv2D 32, 3x3, Stride(1,1)
MaxPooling, Stride (2,2)
Conv2D 64, 3x3, Stride (1,1)
MaxPooling , Stride (2,2)
Conv2D 128, 3x3, Stride (1,1)
MaxPooling , Stride (2,2)
Feature stacking
GRU(256)
GRU(256)
Dense

FIG. 2. (Color online) Attention-CRNN structure framework.

In our work, the model contains three convolutional layers with 3×3 filters and 1×1 strides. Each convolutional layer is followed by a max-pooling layer with 2×2 filters. After convolution and max-pooling, there are two forward-only GRU layers with 256 units, recurrent in the time dimension. The batch normalization layer (Ioffe and Szegedy, 2015) is added between all convolutional layers and activation functions to normalize the input of the activation function. We use the rectified linear unit (ReLU) function (Glorot *et al.*, 2011) to improve the non-linearity for all of the network layers, and ℓ_2 regularization and dropout are utilized to prevent overfitting. Here, dropout means temporarily dropping neural network units out of the network with a certain probability during training. By doing so, it forces one neural unit to work with other neural units picked at random to achieve good results. As it is a binary classification task, the model uses cross-entropy, given below as the loss function:

$$Loss = \frac{1}{N}\sum -[y_i \log(p_i) + (1 - y_i)log(1 - p_i)], \quad (4)$$

where y_i represents the true label of sample i, with 1 and 0 corresponding to abnormal and normal conditions, respectively, and p_i represents the probability of the failure of the i-th sample.

C. Normal-encoder

While anomaly detection tasks often only have access to a small amount of aberrant data, supervised learning frequently needs a substantial amount of data for model training, both in normal and abnormal conditions. In the training phase, if there are not enough training samples, the classifier has insufficient ability to describe the sparse class samples (Koizumi *et al.*, 2019). Hence, it is difficult to classify them effectively, leading to decision bias. There will be numerous unknown faults on the blades as a result of the harsh operating environments of wind turbines, and there is a lack of training data for these faults. Therefore, in practical applications, the classification algorithm may miss these unknown defects in time.

To solve the dataset imbalance problem in anomaly detection tasks, we utilize a normal-encoder method based on semisupervised learning. Deep neural networks learning is a representation learning, and its process is to learn data features through spatial transformation. When we fit a given data distribution under the framework of a probability plot, we need to use an unknown variable (latent variable) to fit the function. Latent space can transform more complex forms of raw data into a simpler data representation, which is more beneficial for data processing. The network learns the characteristics of the audio signal by compressing and reconstructing the log-Mel spectrum. In the process of reconstructing compressed data, the model must learn to store all relevant information and ignore noise, so that the network can eliminate irrelevant information and focus only on the most important features.

In the training phase, only normal audio data are used for training so that the neural network can effectively learn normal high-dimensional features and latent space features of audio. In the testing phase, the test samples without labels are input into the network, and after neural network coding and reconstruction, it is judged by calculating whether the weighted sum of the reconstruction error and coding error is greater than the predefined threshold to determine whether the sample is faulty.

As shown in Fig. 3, the normal-encoder network structure is divided into two subnetworks. The first part is regarded as a conventional autoencoder network that consists of an encoder G_E and a decoder G_D. The function of the encoder is to map the input log-Mel spectrogram $\mathbf{M}$ into a high-dimensional feature vector $\mathbf{z}$, and the decoder reconstructs $\mathbf{z}$ into the original log-Mel spectrogram $\hat{\mathbf{M}}$. The encoder network consists of five CNN layers with convolution kernel sizes (16, 32, 64, 128, and 128). The activation function is used after each layer of the CNN. The activation effect of the Swish function (Nader and Azar, 2020) is better than that of the ReLU function in the deep model. Therefore, the network uses Swish as the activation function, given by

$$f(x) = x \cdot \sigma(\beta \cdot x), \quad (5)$$

where β represents a constant or trainable parameter. When $\beta = 0$, the Swish function is a linear activation function, and when $\beta \to \infty$, Swish becomes a ReLU function. After convolution, the network uses global average pooling (GAP) to pool the features. GAP can map the category information to

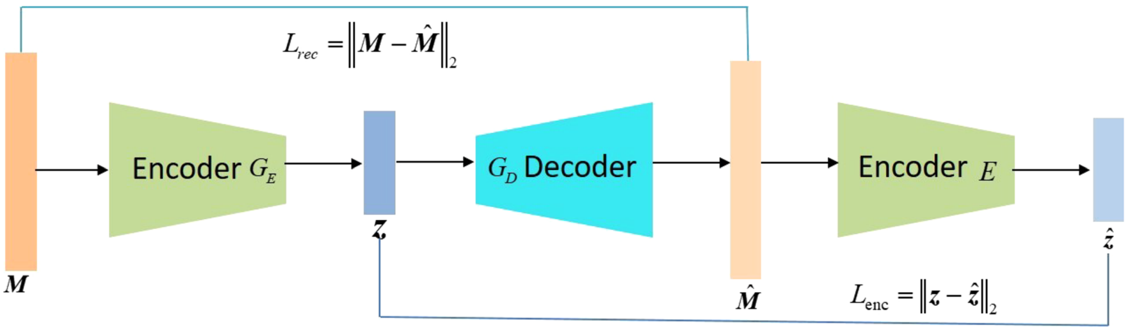

FIG. 3. (Color online) Normal-encoder structure framework.

the feature map of the convolution layer and integrate the global spatial information. The network structure of the decoder includes one fully connected layer and five deconvolution layers. The sizes of the convolution kernel are 128, 64, 32, 16, and 1.

The second part consists of the encoder E, which maps the reconstructed spectrogram $\hat{\mathbf{M}}$ into a high-dimensional feature vector $\hat{\mathbf{z}}$, which is the core part of the entire network. It is different from the traditional autoencoder-based anomaly detection method, and we only compare the differences between the original input spectrogram and the reconstructed spectrogram. The network additionally adds a way to infer anomalies by comparing the differences between the original spectrum and the reconstructed spectrum in a higher-level abstract space, and this additional abstraction layer improves the noise immunity of the network and learns a robust anomaly detection model. The encoder E and the encoder G_E are structurally identical, but the parameters they learn are completely different.

To make the network fully learn the features of normal audio data, two loss functions are used to optimize each subnetwork. The loss function of the first part of the subnetwork calculates the reconstruction loss of the autoencoder. The distance between the original spectrum $\mathbf{M}$ and the reconstructed spectrum $\hat{\mathbf{M}}$ is measured by the ℓ_2 loss. Through such constraints, the reconstructed spectrum can be closer to the real spectrum, given by

$$L_{rec} = \mathbb{E}_{\mathbb{M}\sim\mathbb{N}}|\mathbf{M} - \hat{\mathbf{M}}|_2, \tag{6}$$

where N represents the normal audio dataset.

The loss function of the second molecular network calculates the coding error of the original spectrum and the reconstructed spectrum in abstract space, and the ℓ_2 loss is also used to measure the distance between the high-dimensional features of the original spectrum and those of the reconstructed spectrum, given by

$$L_{enc} = \mathbb{E}_{\mathbb{M}\sim\mathbb{N}}|\mathbf{z} - \hat{\mathbf{z}}|_2. \tag{7}$$

By weighting the loss function of the two parts, the total loss function of the model is

$$L = \alpha L_{rec} + (1-\alpha)L_{enc}, \tag{8}$$

where α is the weighting parameter that controls the contribution of L_{rec} and L_{enc}. Through training, the network model learns how to encode and reconstruct normal samples. For abnormal data samples, the distance between the input spectrogram and the reconstructed spectrogram cannot be minimized, and the difference between them is high.

D. Postprocessing

To detect whether the input audio contains fault information, an anomaly score $\mathbb{S}(x_t, \theta)$ is calculated using an already trained model, where $x_t \in R$ is an input vector calculated from the observed sound indexed on $t \in [1,2, \ldots,\mathrm{T}]$ for time and θ is the set of the parameters of the trained model. In this study, x_t is composed of handcrafted acoustic features, such as log-Mel spectrograms. The anomaly score is calculated by the trained model. The input of x_t is determined to be anomalous when the anomaly score exceeds a predefined threshold value λ, and the binary decision $\mathbb{D}$ is given by

$$\mathbb{D}(x_t, \theta, \lambda) = \begin{cases} 0\,(\text{Normal}), & \mathbb{S}(x_t, \theta) \le \lambda, \\ 1\,(\text{Abnormal}), & \mathbb{S}(x_t, \theta) > \lambda. \end{cases} \tag{9}$$

III. DOMAIN MISMATCH

Various recording devices have different system frequency responses during training. The original audio signal will be corrupted to different degrees when using different recording equipment, which will result in the data domain shift. The term "data domain offset" refers to the fact that when a model is trained using data from the source domain, its performance on those data is typically strong, but its performance on the target domain is often inferior. The model developed on the source domain can also be applied to the target domain if the feature distributions of the two domains can be aligned. The feed-forward network can be fitted to the target domain without being affected by the offset between the two domains when the neural network can learn similar features between two data distributions. To solve this issue, we utilize an adversarial domain adaptive network (ADAN) (Ganin and Lempitsky, 2015), and its structure is depicted in Fig. 4. It is mainly divided into a pretraining stage and an adversarial domain adaptation stage and its detail is provided below.

A. ADAN

The purpose of pretraining is to allow the feature extractor F and the category classifier C to correctly determine whether the input samples have faults. For each input feature matrix $\mathbf{Q}$ with class label $y = 0, 1$, the feature extractor F maps $\mathbf{Q}$ into an N-dimensional feature vector $\mathbf{f}$, and θ_f

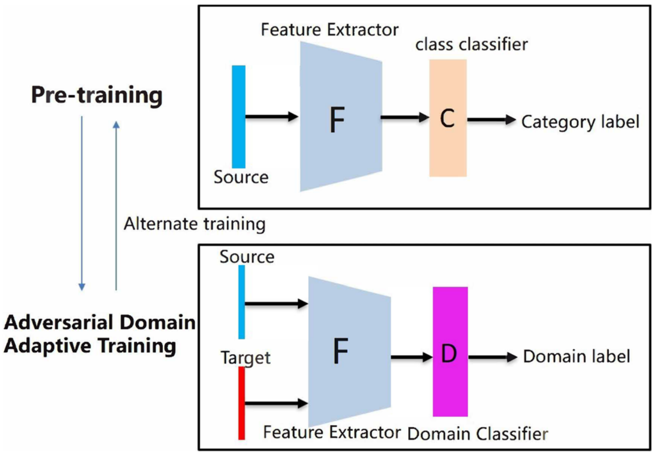

FIG. 4. (Color online) ADAN structure framework.

represents the parameter vector of the feature extractor, given by

$$\mathbf{f} = F(\mathbf{Q}, \theta_f). \tag{10}$$

The network structure of the feature extractor F consists of four CNN layers, and each CNN layer is followed by max-pooling. The category classifier C maps the feature vector $\mathbf{f}$ to obtain the category prediction probability p_y and uses θ_c to represent the parameter vector of C, and the relationship is

$$p_y = C(F(\mathbf{Q}, \theta_f); \theta_c). \tag{11}$$

The class classifier C consists of two fully connected layers and a sigmoid activation function.

In the pretraining stage, we use source domain data with class labels for training. Since the problem is a binary classification problem, to optimize the network parameters, the binary cross entropy (BCE) function below is used as the loss function

$$Loss = \frac{1}{N}\sum -[y_i \log(p_i) + (1 - y_i)log(1 - p_i)], \tag{12}$$

where y_i represents the true label of the ith sample and p_i represents the class prediction probability of the ith sample.

In the process of adversarial domain adaptive optimization, it is necessary to make the feature distribution learned by the feature extractor F from the target domain and the feature distribution learned from the source domain as similar as possible. The domain classifier D maps the feature vector $\mathbf{f}$ to obtain the predicted probability p_d of the domain label, given by

$$p_d = D(F(\mathbf{Q}, \theta_f); \theta_d). \tag{13}$$

The domain classifier D also consists of two fully connected layers and a sigmoid activation function.

During the domain adaptation training phase, domain classifier D is an already trained discriminator, and D can correctly determine whether the input samples are from the source domain or the target domain. If the input samples are from the source domain, the value of p_d is close to 1, and if the input samples are from the target domain, the value of p_d is close to 0. The goal of training is to prevent D from determining whether the input samples are from the source domain or the target domain. When the predicted probability is close to 0.5 (similar to random classification), the features learned by the feature extractor from the two domains are very similar. The loss function for domain adversarial adaptive training is

$$L_d = \frac{1}{N}\sum_{i=1}^{N} \log(p_{s_i}) + \frac{1}{N}\sum_{j=1}^{N} \log(1 - p_{t_j}), \tag{14}$$

where p_{s_i} represents the predicted probability of the source domain samples and p_{t_j} represents the predicted probability of the target domain samples, and N is the number of samples. In the adaptive training process of the cutting domain, only a small amount of source domain data and target domain data with domain labels is used.

Through such a training method, the feature extractor can simultaneously learn features with class-discriminative properties and domain-invariance properties. In the testing phase, we use the pretrained model as the final system model.

IV. DATASET COLLECTION

To verify the feasibility of the proposed networks, we recorded the audio signals generated by the wind turbines in real time, and these datasets are currently mainly used by developing the algorithms. The recording equipment and the production of the dataset are described in detail below.

A. Recording equipment overview

In this study, three recording devices are used to collect the data and each device is described below.

Device A: The SS-ALIM6S2M1–002 voice algorithm motherboard is a six-microphone array motherboard product developed by SureSence Technology. The product is externally connected to AC108, and the output format is a 6-channel pulse code modulation (PCM) signal with a sampling rate of 16 kHz and a depth of 16 bits. The equipment is shown in Fig. 5.

Device B: The Zoom H1n recording device uses a stereo X/Y microphone configuration and supports 24 bit/96 kHz audio signals. The device features a low-cut filter that reduces pops, wind noise, and other unwanted low-frequency noise to improve recording quality, and a compressor can be used to obtain maximum sound pressure, up to 120 dB SPL, to ensure that the recording audio is undistorted.

Device C: Sogo AI recorder E1 has a total of eight microphones, six of which are omnidirectional microphones, and the remaining two are Harman 10 mm pointing microphones. In addition, the device uses the clairVoice 8 microphone array algorithm and pureVoiceAI noise reduction algorithm for noise reduction in improving the recording quality.

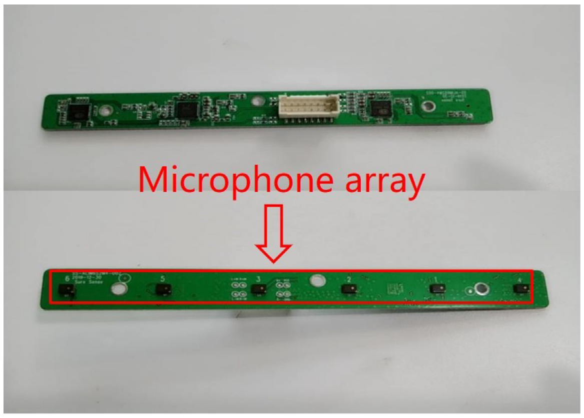

FIG. 5. (Color online) SS-ALIM6S2M1-002 voice algorithm motherboard.

FIG. 6. (Color online) Wind turbines.

B. Dataset overview

With a wind speed of 3–6 m/s and an air pressure of 976 hPa, we recorded the audio signals generated by the real-time operation of the wind generator at the wind farm. The wind turbine is shown in Fig. 6, the height of each wind turbine is 140 m, and the diameter length of the wind turbine blade is 56 m. The dataset recording scene is shown in Fig. 7. The recording equipment is installed in front of the tower at a height of approximately 2 m above the ground. We recorded of 60 h of audio data, including 30 h of normal audio data and 30 h of faulty data. The recorded raw data were cut into 4 s segments, followed by data cleaning to remove invalid data, and finally, corresponding labels were manually added to the audio data of each segment.

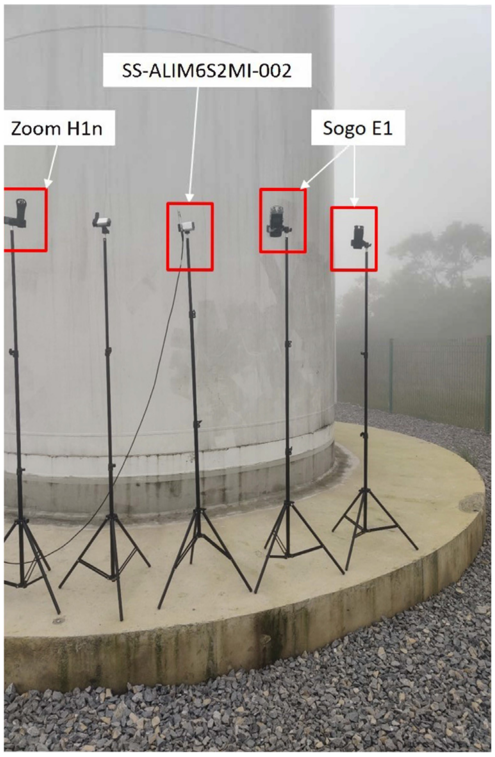

FIG. 7. (Color online) Recording scene.

The waveforms and spectrograms of normal audio and abnormal audio are shown in Fig. 8. Due to the surface faulty of a wind turbine blade, when the wind turbine is running, the sound of the faulty blade cutting down the air sounds is sharper. As can also be seen from the spectrogram, the abnormal audio has more high-frequency components than normal audio data. This is the important feature that distinguishes normal audio from abnormal audio.

In the fault detection task under supervised learning conditions, a total of 40 000 labeled data samples are used, half of which are normal audio data and half of which are

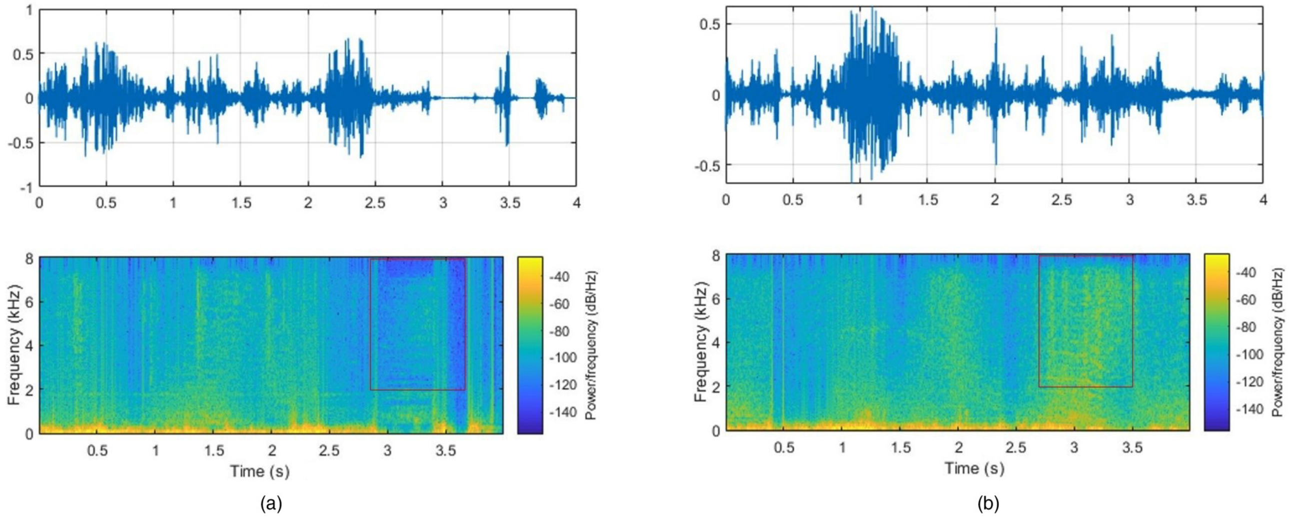

FIG. 8. (Color online) Waveform and spectrogram comparison of audio data: (a) normal audio data, (b) abnormal audio data. The red rectangular highlights the difference between the normal and abnormal sounds in Mel domain.

https://doi.org/10.1121/10.0016998

abnormal. The length of each sample is 4 s, and the total duration is approximately 40 h. Among them, 70% of the data are used for network model training, 20% of the data are used for the validation set, and 10% of the data are used for the final result test.

In the fault detection task under semisupervised learning conditions, the dataset size of the training stage is 20 000 labeled normal data samples, and the total duration is approximately 20 h. In the test phase, 5000 normal data samples and 5000 abnormal data samples were used to test the model.

In the fault detection task under device mismatch conditions, the datasets recorded by three recording devices, SS-ALIM6S2M1-002, Zoom H1n, and Sogo E1, are used to verify the feasibility and effectiveness of the proposed algorithm. The datasets of these three devices are used as the source domain data to train the model, and the datasets of the other two devices are used as the target domain data. In the training phase, a large amount of source domain data and a small amount of target domain data are used for training, and then, the trained model is tested on the target domain data. The training dataset consists of 18 000 source domain audio files (20 h in duration) and 1800 target domain audio files (2 h duration), and the size of the testing dataset is 3600 target domain audio files (4 h in duration).

TABLE I. Performance comparison results of different neural network algorithms.

Network models	Features	Accuracy (%)	Model size (Mb)
Attention-CRNN	Original waveform	76.8	13.4
	Spectrogram	83.2	
	MFCCs	92.1	
	log-Mel spectrogram	**99.6**	
CRNN	Original waveform	77.9	13.1
	Spectrogram	79.2	
	MFCCs	91.7	
	log-Mel spectrogram	95.1	
Resnet-50	Original waveform	76.9	46.2
	Spectrogram	85.2	
	MFCCs	91.6	
	log-Mel spectrogram	94.3	
Mobilenet	Original waveform	80.9	**8.9**
	Spectrogram	80.2	
	MFCCs	88.6	
	log-Mel spectrogram	92.7	
CNN	Original waveform	78.9	14.8
	Spectrogram	73.2	
	MFCCs	90.6	
	log-Mel spectrogram	91.8	
RNN	Original waveform	84.9	19.9
	Spectrogram	72.2	
	MFCCs	83.6	
	log-Mel spectrogram	89	

V. EVALUATIONS

A. Supervised learning

We compared the performances of the attention-CRNN, CRNN, ResNet-50, MobileNet, CNN, and recurrent neural network (RNN) methods. The influence of different audio characteristics on the detection results are also compared. The convergence speed of the model training phase, the average test set accuracy rate, and the size of the model are mainly used as the main evaluation indicators. As shown in Fig. 9, in the training phase, attention-CRNN reached convergence at approximately the 10th epoch, the accuracy rate reached 99%, and the convergence speed was significantly better than that of the other algorithms. As shown in Table I, the test set accuracy of attention-CRNN reached 99.6%, which is higher than that of other models. Compared with the CRNN algorithm without the attention mechanism, the frequency domain attention model effectively improves the convergence speed during training and the accuracy of fault detection, which shows the benefits of adding the attention mechanism. As can be seen from Table I, the performance using log-Mel spectrum is superior, which is consistent with our early analysis.

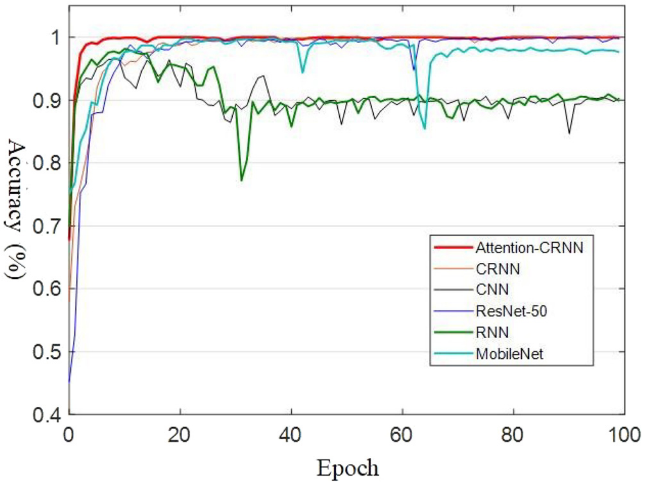

FIG. 9. (Color online) Accuracy versus epoch for different algorithms.

To compare the anti-noise performances of the algorithms, the original signals were added with different intensities of Gaussian white noise to obtain the wind turbine blade audio datasets under different signal-to-noise ratio (SNR) conditions. In this experiment, SNRs of 10, 5, 0, and –5 dB are considered. The experimental results are shown in

TABLE II. Average detection accuracy (%) of each neural network at different SNRs.

Method	SNR = 10 dB	SNR = 5 dB	SNR = 0 dB	SNR = –5 dB
Attention-CRNN	**97.2**	**92.3**	**90.1**	**84.2**
CRNN	94.3	89.6	85.3	79.6
ResNet-50	93.7	90.9	87.5	80.3
MobileNet	90.1	88.7	82.1	75.0
CNN	89.8	87.1	80.9	72.4
RNN	87.9	82.1	75.3	69.1

TABLE III. Performance comparison results of different fault detection algorithms.

Method	AUC	$F1$-Score	Recall	Precision
RBM	0.594	0.502	0.688	0.395
Autoencoder	0.651	0.592	0.795	0.471
VAE	0.687	0.598	0.855	0.460
ANOGAN	0.712	0.645	0.788	0.546
EGBAD	0.755	0.674	0.809	0.578
Normal-encoder	**0.832**	**0.776**	**0.925**	**0.670**

Table II. Under different SNR conditions, the accuracy of attention-CRNN is always higher than that of the other algorithms, which demonstrates the robustness of the proposed network.

B. Semisupervised learning

The neural network algorithm of semisupervised learning aims to solve the data imbalance problem. In this experiment, we compared the performances of the normal-encoder, efficient generative adversarial networks (GAN)-based anomaly detection (EGBAD), anomaly detection with generative adversarial networks (ANOGAN), variational auto-encoder (VAE), autoencoder, and restricted Boltzmann machine (RBM) methods. The area under curve (AUC) and $F1$-score are used as the main evaluation indices of different models. The experimental results are shown in Table III. The AUC and $F1$-score of the normal-encoder are higher than those of the other models. The results show the feasibility and effectiveness of the normal-encoder algorithm for fault detection in the absence of fault samples.

Similarly, to further verify the anti-noise performances of different models, Gaussian white noise with different intensities was added to the original signal. As shown in Fig. 10, the AUC of the normal-encoder remains the highest under different SNR conditions. When the SNR is –5 dB, the AUC values of the other algorithms are all close to 0.5, and the AUC value of the normal-encoder is still 0.637.

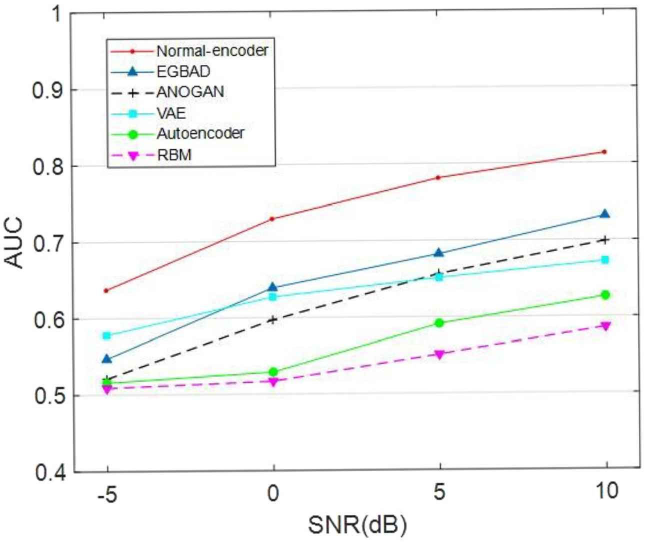

FIG. 10. (Color online) Effect of noise on the AUCs of different algorithms.

C. Domain mismatch

In this task, the data recorded by three recording devices, SS-ALIM6S2M1-002 (device A), Zoom H1n (device B), and Sogo E1 (device C), were used to verify the feasibility and effectiveness of the proposed algorithm. The datasets of these three devices were used as the source domain data to train the model, and the datasets of the other two devices were used as the target domain data.

The experimental results are shown in Table IV, where source only means that no method is used, and the target domain data are directly tested on the model trained in the source domain. ADAN has the highest detection accuracy in the equipment matching fault detection task. The ADAN algorithm enables the model to learn features with domain invariance and category discrimination through joint training, which improves the performance degradation caused by different equipment.

VI. CONCLUSION

To detect wind turbine blade faults, we developed three networks, attention-CRNN, normal-encoder, and ADAN, for supervised, semisupervised, and domain adaptations, respectively. In terms of supervised learning, the proposed method reaches 99.6% accuracy. When background noise is present, the proposed method also outperforms others, demonstrating its anti-noise ability. In the case of semisupervised learning, the proposed method attains 67% accuracy, 10% higher than the second best. When domain mismatch happens, we obtain over 28% accuracy increase than the source only case. From the above results, the proposed networks outperform others, which shows their effectiveness. More importantly, sound information can be used for wind blade fault detection. In the future, we will continue to gather data and create a reliable classification system for different types of faults. The complexity of the algorithms must be further reduced to expand their practical use in industrial applications in future research.

TABLE IV. Fault detection accuracy under device mismatch conditions (%).

Method	Source	A	A	B	B	C	C
	Target	B	C	A	C	A	B
CMSoder		59.83	60.14	54.17	59.14	60.59	63.27
Spectrum Correction		69.13	66.78	70.17	68.72	65.34	70.11
DDC		72.58	74.79	73.64	70.71	75.38	72.79
ADAN		**77.81**	**80.19**	**76.45**	**78.89**	**75.81**	**75.81**
Source only		49.61	50.18	47.26	51.02	47.26	51.02

Amano, R. S. (**2017**). "Review of wind turbine research in 21st century," J. Energy Resour. Technol. **139**(5), 050801–050808.

Chacon, J. L. F., Kappatos, V., Balachandran, W., and Gan, T.-H. (**2015**). "A novel approach for incipient defect detection in rolling bearings using acoustic emission technique," Appl. Acoust. **89**, 88–100.

Chen, J., Song, Y., Peng, Y., Nielsen, S. R., and Zhang, Z. (**2020**). "An efficient rotational sampling method of wind fields for wind turbine blade fatigue analysis," Renew. Energy **146**, 2170–2187.

Dollinger, C., Balaresque, N., Sorg, M., and Fischer, A. (**2018**). "IR thermographic visualization of flow separation in applications with low thermal contrast," Infrared Phys. Technol. **88**, 254–264.

Fitzgerald, B., Arrigan, J., and Basu, B. (**2010**). "Damage detection in wind turbine blades using time-frequency analysis of vibration signals," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–5.

Ganin, Y., and Lempitsky, V. (**2015**). "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning, PMLR*, pp. 1180–1189.

Glorot, X., Bordes, A., and Bengio, Y. (**2011**). "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings*, pp. 315–323.

González, A. G., and Fassois, S. (**2016**). "A supervised vibration-based statistical methodology for damage detection under varying environmental conditions & its laboratory assessment with a scale wind turbine blade," J. Sound Vib. **366**, 484–500.

Goyal, D., and Pabla, B. (**2016**). "The vibration monitoring methods and signal processing techniques for structural health monitoring: A review," Arch. Computat. Methods Eng. **23**(4), 585–594.

Grollmisch, S., Abeßer, J., Liebetrau, J., and Lukashevich, H. (**2019**). "Sounding industry: Challenges and datasets for industrial sound analysis," in *2019 27th European Signal Processing Conference (EUSIPCO)*, pp. 1–5.

Hongwu, Q., Xiaoxue, X., Jincheng, P., Tongli, J., and Weifeng, G. (**2015**). "Using AE testing method for condition monitoring in wind turbine shaft," in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, pp. 173–176.

Hwang, S., An, Y.-K., and Sohn, H. (**2017**). "Continuous line laser thermography for damage imaging of rotating wind turbine blades," Procedia Eng. **188**, 225–232.

Ioffe, S., and Szegedy, C. (**2015**). "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, pp. 448–456.

Kashfi, M., Fakhri, P., Amini, B., and Yavari, N. (**2019**). "Vibration analysis of a wind turbine blade integrated by a piezoelectric layer," in *2019 International Power System Conference (PSC)*, pp. 650–654.

Koizumi, Y., Saito, S., Uematsu, H., Kawachi, Y., and Harada, N. (**2019**). "Unsupervised detection of anomalous sound based on deep learning and the neyman–pearson lemma," IEEE/ACM Trans. Audio. Speech. Lang. Process. **27**(1), 212–224.

Kośmider, M. (**2021**). "Spectrum correction: Acoustic scene classification with mismatched recording devices," arXiv:2105.11856.

Leite, G. d N. P., Araújo, A. M., and Rosas, P. A. C. (**2018**). "Prognostic techniques applied to maintenance of wind turbines: A concise and specific review," Renew. Sust. Energ. Rev. **81**, 1917–1925.

Liu, Z., Yang, B., Wang, X., and Zhang, L. (**2021**). "Acoustic emission analysis for wind turbine blade bearing fault detection under time-varying low-speed and heavy blade load conditions," IEEE Trans. Ind. Applicat. **57**(3), 2791–2800.

Liu, Z., and Zhang, L. (**2020a**). "Acoustic emission analysis for wind turbine blade bearing fault detection using sparse augmented Lagrangian algorithm," in *2020 IEEE Applied Power Electronics Conference and Exposition (APEC)*, pp. 145–151.

Liu, Z., and Zhang, L. (**2020b**). "A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings," Measurement **149**, 107002.

Mohamed, A-r. (**2014**). "Deep neural network acoustic models for Asr," Ph.D. thesis, University of Toronto, Toronto, Canada.

Mori, M., Novak, L., and Sekavčnik, M. (**2007**). "Measurements on rotating blades using IR thermography," Exp. Therm. Fluid Sci. **32**(2), 387–396.

Nader, A., and Azar, D. (**2020**). "Searching for activation functions using a self-adaptive evolutionary algorithm," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pp. 145–146.

Shen, Y.-H., He, K.-X., and Zhang, W.-Q. (**2018**). "Learning how to listen: A temporal-frequential attention model for sound event detection," arXiv:1810.11939.

Tang, J., Soua, S., Mares, C., and Gan, T.-H. (**2016**). "An experimental study of acoustic emission methodology for in service condition monitoring of wind turbine blades," Renew. Energy **99**, 170–179.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (**2017**). "Attention is all you need," arXiv:1706.03762.

Wang, M., and Deng, W. (**2018**). "Deep visual domain adaptation: A survey," Neurocomputing **312**, 135–153.

Zhang, L., Liu, K., Wang, Y., and Omariba, Z. B. (**2018**). "Ice detection model of wind turbine blades based on random forest classifier," Energies **11**(10), 2548.