Energy 254 (2022) 124441

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

Wind turbine blade icing detection: a federated learning approach

Xu Cheng [a], Fan Shi [b], Yongping Liu [a], Xiufeng Liu [c, *], Lizhen Huang [a, **]

[a] Department of Manufacturing and Civil Engineering, Norwegian University of Science and Technology, Gjøvik, 2815, Norway
[b] School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, 300384, PR China
[c] Department of Technology, Management and Economics, Technical University of Denmark, Produktionstorvet, 2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Article history:
Received 10 December 2021
Received in revised form
8 May 2022
Accepted 31 May 2022
Available online 9 June 2022

Keywords:
Wind turbine
Blade icing detection
Federated learning
Imbalanced learning
Time series classification

ABSTRACT

Wind farms are often located at high latitudes, which entails a high risk of icing for wind turbine blades. Traditional anti-icing methods rely primarily on manual observation, the use of special materials, or external sensors/tools, but these methods are limited by human experience, additional costs, and understanding of the mechanical mechanism. Model-based approaches rely heavily on prior knowledge and are subject to misinterpretation. Data-driven approaches can deliver promising solutions but require large datasets for training, which might face significant challenges with respect to data management, e.g., privacy protection and ownership. To address these issues, this paper proposes a federated learning (FL) based model for blade icing detection. The proposed approach first creates a prototype-based model for each client and then aggregates all client models into a globally weighted model. The clients use a prototype-based modeling method to address the data imbalance problem, while using the FL-based learning method to ensure data security and safety. The proposed model is comprehensively evaluated using data from two wind farms, with 70 wind turbines. The results validate the effectiveness of the proposed prototype-based client model for feature extraction, and the superiority over the five baselines in terms of icing detection accuracy. In addition, the experiment demonstrates the promising result of online blade icing detection, with almost 100% accuracy.

1. Introduction

Wind energy has developed rapidly in recent years as a result of the energy crisis, and more importantly, the inexhaustible nature of wind energy and the mature technology of wind turbines [1]. Wind turbines are generally installed at high altitudes to better exploit the wind source [2]. The performance of wind turbines can be affected by many factors, including the physical design of the blades, the technical parameters of the turbine, the location and climatic factors [3]. Among them, blade icing is a critical factor that limits wind turbine performance, as turbines are usually installed in high-altitude areas to make full use of wind energy. In severe cases, almost 30% of annual power generation is lost due to blade icing [4]. More seriously, blade icing can, to some extent, cause casualties and production losses. Therefore, detection of blade icing is of the utmost importance.

Traditionally, human observation, passive methods and active methods are the main solutions for wind turbine blade icing detection. Human observation is too subjective, and observations are highly dependent on the experience of the observers. Passive methods use special materials such as black paint and coating to prevent icing [5,6]. Although passive methods are inexpensive, coating alone does not produce the best results and is difficult to maintain. The active method proposed in Ref. [4] is an effective anti-icing method, but requires additional power and mechanical replacement, which can damage wind turbines.

To compensate for the above traditional methods, model-based approaches, including mathematical and data-based methods, have received increasing attention in recent years. Mathematical methods make predictions about wind turbine blade icing by developing mathematical or numerical models, for example [7], but their disadvantages are obvious. First, they are often highly dependent on different assumptions, which can lead to a misidentification of the blade icing conditions. Second, external experimental tools, such as a wind tunnel, are required to obtain an accurate mathematical model. Third, domain knowledge is often needed to model the icing process [8]. In recent years, with the wide spread of sensor technologies in wind farms, a large amount

* Corresponding author.
** Corresponding author.
E-mail addresses: xiuli@dtu.dk (X. Liu), lizhen.huang@ntnu.no (L. Huang).

https://doi.org/10.1016/j.energy.2022.124441

of data has been collected, making it possible to construct data-driven methods for blade icing detection. Data-driven models detect the icing state based on the mining of hidden information from history data [9]. The advantage is that they do not rely on prior domain knowledge, but rather on the available data, which is more efficient and cost-effective. They are favored by both academia and industry, and in particular, end-to-end solutions based on deep neural networks to detect blade icing have received significant interest.

Nevertheless, some challenges remain in the development of data-driven models for blade icing detection. First, it often requires large amounts of data for training, especially for deep learning models. Traditional centralized modeling approaches require not only large computing resources, but also large storage capacity, which in many cases is prohibitively expensive. In addition, when data are collected to a central storage, an additional communication cost is required. Second, since wind farms are usually located in different regions, such as mountainous areas and offshore, sending data over the network is vulnerable to attacks [10]. Third, the quality of wind turbine data is characterized by diversity, i.e., heterogeneity. When data are used directly for model training, this may lead to low efficiency in operations, due to a too general model and insufficient performance. Last but not least, there is a high skewed class distribution between icing and non-icing samples, as a wind turbine operates in a normal state most of the time. The icing samples are typically much smaller than normal samples, which presents a great challenge for constructing high-performance detection models. A highly skewed distribution can severely affect classification performance because the resulting classification model tends to favor the majority of classes while ignoring the minority.

To address these challenges, this paper proposes a federated learning (FL) based model for detecting blade icing state of wind turbines, named *IcingFL*. With IcingFL, it becomes possible to learn a common model collaboratively (global model), without the need to collect data to the cloud; instead, the model is trained locally at the site of each wind farm (client model). This approach pushes down computing from the cloud to the edge, which not only reduces the burden, but also enhances the data safety due to no-sharing data. However, the federated approach may not be able to address skewed data problems like the centralized approach in the cloud, which can affect model performance. To address this problem, we introduce a novel latent space to the client-side modeling, and use a weighted learning method to create the global model by aggregating the client models. This method differs not only from traditional approaches to data imbalance problems, i.e., oversampling the minority classes or undersampling majority classes, but also from traditional FL-based methods. Traditional FL methods generally do not consider data imbalance. Instead, it uses only the raw data space or specially developed loss functions to create client models.

In summary, the contributions of this paper are threefold:

- Proposes a federated learning-based model, IcingFL, for wind turbine blade icing detection. IcingFL optimizes the generation of global models, considering the importance of each client model.
- Proposes a novel method to address data imbalance problem at each client. The proposed method uses a specially designed feature extractor to extract features from imbalanced raw data, and then balances the features in a latent space based on the prototype of each class.
- Evaluates the proposed model comprehensively by comparing with other baselines. The results demonstrate the superiority of the proposed model over the others for blade icing detection. Furthermore, ablation studies show the effectiveness of the model network structure and validate the feasibility of online estimation.

The rest of the paper is structured as follows. Section 2 reviews the literature on wind turbine blade icing detection and federated learning. Section 3 describes the proposed FL-based model in detail. Section 4 evaluates the proposed model through experiments. Section 5 concludes the paper and presents future work.

2. Related work

2.1. Wind turbine blade icing detection

For blade icing detection of wind turbines, passive and active de-icing systems, model-based and data-driven methods have been proposed in the literature. Passive methods use special materials such as liquid-infused surfaces [11] and porous superhydrophobic/ polyvinylidene fluoride coatings [12], while active methods use external sensors or tools, e.g., Refs. [13–15]. Model-based methods create mathematical models based on certain assumptions, which are then verified by experiments, e.g., Ref. [16]. These methods require human knowledge and assumptions for the icing process. In recent years, data-driven methods have received increasing attention, especially with the emergence of deep learning, which has a strong ability to extract features. For example, Jiménez et al. used machine learning algorithms, including decision tree and support vector machines, to implement a classifier that can identify the presence and thickness of blade ice of wind turbines from ultrasonic signals [17]. Liu et al. proposed an ensemble depth-learning model to extract multilevel features from SCADA data [18]. Yuan et al. presented a wavelet-based CNN model [19], and demonstrated that CNN can achieve competitive performance over traditional machine learning methods.

All existing data-driven approaches build models based on the data in a centralized server, which were collected from different wind farms geographically located at different locations. These approaches have a number of drawbacks, including the need for large data storage and computational capacity of the central server, vulnerability to cyberattacks and data leakage, network overhead associated with data transfer, unwillingness of data owners to share their data, etc. In this paper, we address these limitations by proposing the FL framework to train the deep learning model in a distributed manner. To our knowledge, this is the first attempt to build the wind turbine blade icing detection model while addressing data management issues associated with wind farms.

2.2. Federated learning

The proposed IcingFL uses the federated learning framework to address data management issues during model construction. The idea of federated learning was originally proposed in Ref. [20] to solve data communication bottlenecks during distributed model training. In federated learning, data are not collected on a central server. Instead, all participating clients train their partial models locally using their own data, and then the partial models are aggregated to a global model by a central server, thus preserving data privacy. In recent years, due to the emergence of deep learning, the computational resources and data required for model training are considerable. In addition, data protection has received unprecedented attention. For example, the European Union has implemented GDPR for data privacy protection. As a result, researchers have started to explore more effective and efficient learning approaches, among which federal learning has emerged as a promising solution. In the energy section, Wang et al. [21]

proposed a federated learning approach for the identification of household profiles. Lin et al. [22] proposed a novel edge-based federated learning framework for FDI attack detection in power grid state estimation, which shows great potential in real-world applications with unknown system parameters. Su et al. [23] proposed a secure and efficient federated-learning-enabled AIoT scheme for private energy data sharing in smart grids with edge-cloud collaboration. With the advent of Industry 4.0, a large number of devices are interconnected in smart manufacturing, and data security and privacy become an important issue [24]. Federated learning can be a promising solution to this challenge, due to its nature without the need for data sharing [20,25]. Several studies have been found for FL in Industry 4.0. Among others, Yu et al. proposed a novel data-driven cognitive computing platform for Industry 4.0 smart manufacturing by combining federal learning and blockchain technologies [26], and Hao et al. proposed a privacy-enhanced efficient FL model for Industry 4.0 [10].

The construction of icing detection models must take into account the climatic differences of different wind farms. Climatic factors, such as humidity and light distribution, vary from region to region, which can influence model performance. Thus, traditional centralized approaches that build a single model based on aggregated data may not be sufficiently representative of the wind farm in each region and therefore may have low performance. However, if independent models are built entirely from their own data, although most of the problems related to data management can be eliminated, the model may also face the problem of using imbalanced data. Therefore, aggregation of partial models trained by different wind farms should be the optimal solution; thus, federated learning comes into play in this regard.

3. Proposed IcingFL model

In this section, we will first present the FL-based architecture proposed in IcingFL. Then, we will present the data preprocessing, the imbalanced learning method to build the client model, and the weighted aggregation method to create the global model.

3.1. Overview

The proposed IcingFL aims to address the following challenges. The first is how to train the model using imbalanced data. Wind turbines operate in a normal state most of the time, i.e., the blades are free of ice, and only in a few cases the blades are icing. Therefore, the data collected from the SCADA (Supervisory Control and Data Acquisition) system is imbalanced in terms of distribution, and models trained using imbalanced data will result in biased results. The second is how to minimize the impact on the model due to the differences between the clients. As each client uses its own data to train the local model, the resulting blade icing detection model is highly dependent on local climatic factors, such as temperature. FL usually assumes that the used data are independent and identically distributed, whereas in reality this is not the case. The performance of the global model will be compromised if we simply aggregate the client models without considering their differences. The third is how to ensure data safety and security during model training. As FL model training requires frequent communication between the central server and each client, i.e., updating the model parameters, this process is vulnerable to cyberattacks. Attackers can tamper with client models to invalidate the aggregated model or obtain the trained model to recover the original data.

The proposed IcingFL addresses these challenges in two ways, imbalanced learning and federated learning. Fig. 1 shows the architecture of the IcingFL. It consists of three key components: local client model training (client side), gradient data encryption (communication) and global model aggregation (server side). Each client first pre-processes the raw sensor data to reduce the negative impact of noise on its model quality. Then a local prototype-based modeling method is used to address the data imbalance problem in the training (see the upper part of Fig. 1). Instead of using traditional oversampling or undersampling methods, we address this problem by balancing the extracted features in a latent space (described in Section 3.3). The training process begins with initializing the model parameters on the central server, followed by training the local model at each client. During the training, all clients upload their local models to the central server for aggregation, and the server updates all clients with the aggregated global model. This process is an iterative process until the model is convergent or reaches a pre-set number of iterations. To ensure data security, we apply homomorphic encryption for communication, which allows the computation to be directly based on encrypted data, i.e., without the need for decryption [27]. The lower part of Fig. 1 shows the homomorphic encryption method based on linear transformation. The server aggregates all encrypted vectors from clients based on the proposed weighted aggregation method (detailed in Sections 3.4 and 3.5). The server adjusts the weight of each local model based on the training sample size of each client (described in Section 3.4). When the entire training process is over, the final global model will be used for online blade icing detection.

3.2. Data preprocessing

The data used in this paper are collected by the SCADA system and have some quality issues, including anomalies, redundant or missing values, etc. To minimize the impact of data quality issues, we perform the following preprocessing, including labeling, visual analysis, segmentation, and normalization.

- **Data labeling.** All raw data are labeled as normal or icing by experienced turbine engineers. All uncertain intervals that are difficult to label were removed.
- **Visual analysis.** We ensure the data quality using the visual analysis approach. First, we simply eliminate redundant data. This is because redundant information in the data collected by the sensors cannot contribute to the detection of blade icing. In contrast, they can weaken the feature representation capability. Second, we identify outliers by visual analysis and fix them. Third, we also investigate the correlation between different signals through visual analysis, such as power output and wind speed. Through correlation analysis, we can have a deep understanding of the features and the output.
- **Segmentation and normalization.** The data from SCADA system are time series collected by the censors at regular intervals. The data fed into the training or testing of the model must satisfy a certain length. In addition, the raw data represent the signal with a certain unit and magnitude. To reduce the impact of these factors on the model, we first perform a mix-max normalization of the data and then segment the time series according to a fixed-length time window.

3.3. Imbalanced learning-based client model

Class imbalance in the training data can deteriorate the performance of the model, and thus the classification ability, especially for the identification of minority classes. The imbalance problem can generally be addressed at two levels: at the data level and at the algorithm level. Data-level approaches, as the name implies, require some processing of raw data, such as resampling and data

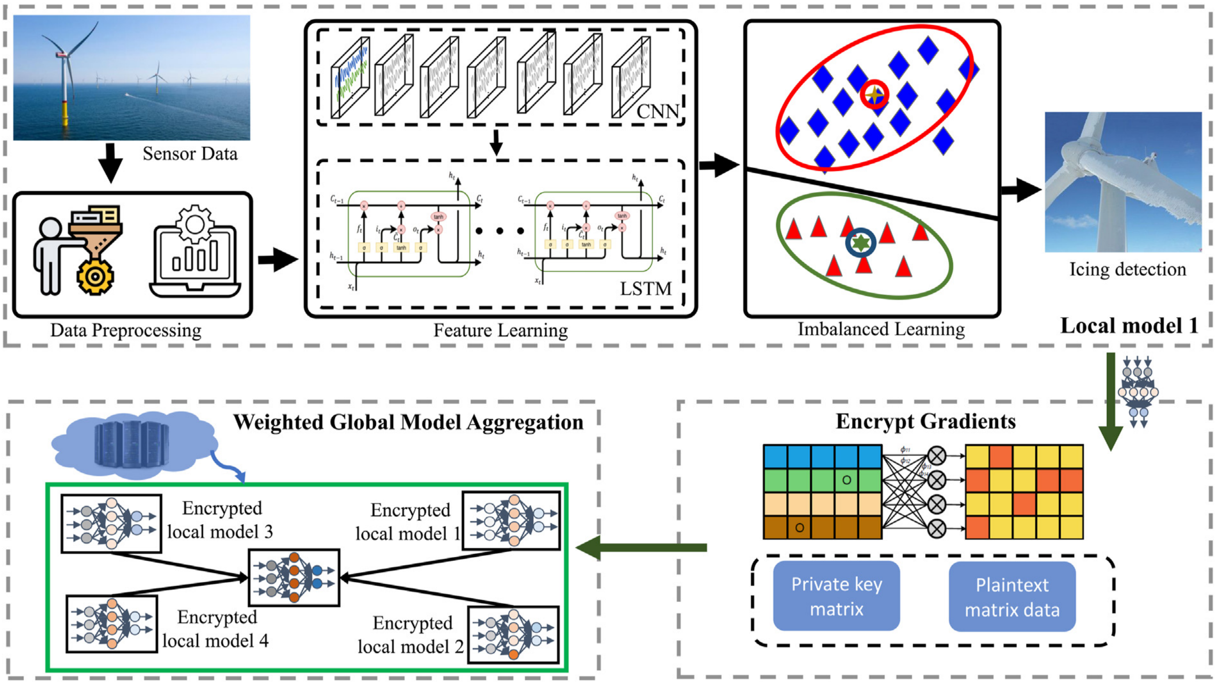

Fig. 1. Overview of IcingFL for blade icing detection of wind turbines.

enrichment, which involves increasing the number of minority classes or reducing the number of majority classes [28,29]. Data-level approaches also require additional information in modeling, such as the distribution of the training data. Since this may violate the privacy of the data [30] and does not conform to the idea of LF, they are not used in this study. On the contrary, algorithm-level approaches focus on algorithm innovation for training, e.g., designing new neural network structures [31,32], or tuning model training parameters. Algorithm-level approaches, therefore, require no or much less processing on the data, which is favored in this study. Fig. 2 illustrates the imbalanced learning process at a client where a deep learning model is trained. In this approach, the features are first extracted from the imbalanced data in a latent space. Then the features/classes are balanced by learning the prototype of each class, and finally, the classifier is built based on the resulting prototypes. Here, an attention-based method is used to improve the prototype learning ability [33]. The following subsections describe the whole learning process in more detail:

3.3.1. Feature extraction

In this work, we implement the feature extractor f based on the Convolutional Neural Network (CNN) and the Long Short-Term Memory (LSTM) as shown in Fig. 2. CNN consists of a convolution layer (Conv1D), an attention layer (SE) [34], a normalized layer (Norm1D), and an activation layer (ReLU). Assuming that the input is $X_{raw} \in \Re^{d \times T}$, where d and T are the input dimension and the window size of the samples, respectively, the output of CNN can be represented by:

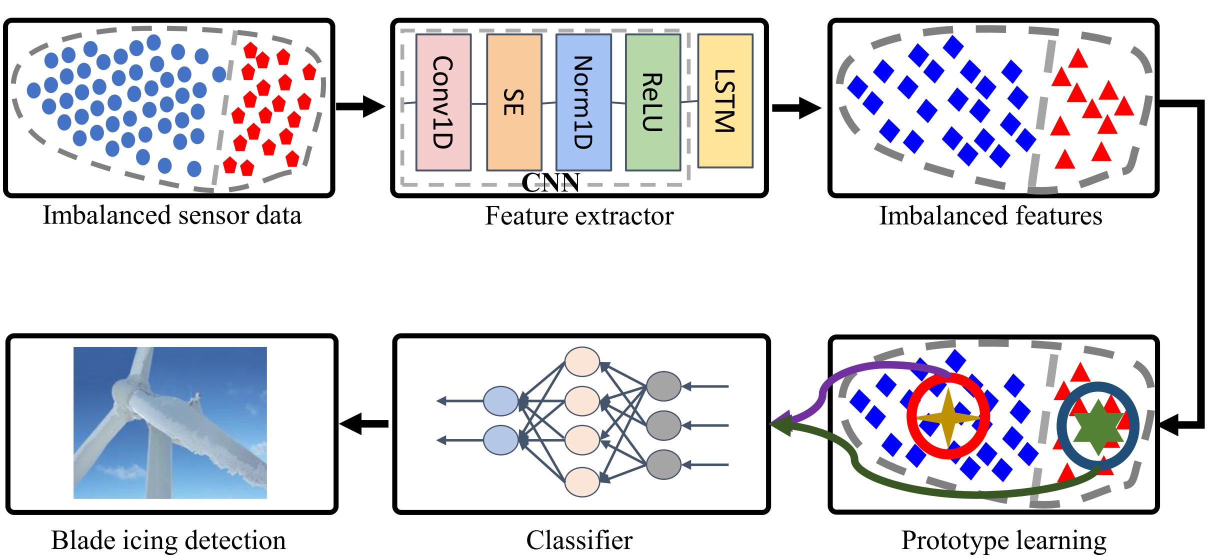

Fig. 2. Imbalanced learning-based client model.

$$\begin{aligned} X_c &= Conv1D(X_{raw}) \\ X_{SE} &= SE(X_c) \\ X_{Norm} &= Norm1D(X_{SE}) \\ X_{CNN} &= ReLU(X_{Norm}) \end{aligned} \tag{1}$$

where X_c, X_{SE}, X_{Norm}, and X_{CNN} are the output of the Conv1D layer, SE layer, Norm1D layer, and ReLU layer, respectively. The values of three variables have the same shape, $\Re^{F\times T}$, where F is the number of filters in CNN. The transposition of X_{CNN} is then fed into the LSTM to capture temporal features. The calculation of the LSTM can be represented as follows [35]:

$$\begin{aligned} g^i &= \sigma\left(W^i h^{t-1} + I^i x_l^t\right) \\ g^f &= \sigma\left(W^f h^{t-1} + I^f x_l^t\right) \\ g^o &= \sigma\left(W^o h^{t-1} + I^o x_l^t\right) \\ g^c &= \tanh\left(W^c h^{t-1} + I^c x_l^t\right) \\ m^t &= g^f \otimes m^{t-1} + g^i \otimes g^c \\ h^l &= \tanh(g^o \otimes m^t) \end{aligned} \tag{2}$$

where g^i, g^f, g^o, g^c are the output of the input, forget, output, and cell state after the activation layer (sigmoid function σ), respectively; W^i, W^f, W^o, W^c represents the recurrent weight matrices; and I^i, I^f, I^o, I^c are projection matrices. The hidden state of the LSTM is h^t and $\otimes$ is the element-wise multiplication. The output of the LSTM is $\Re^{T\times dl}$, where dl is the feature dimension in the latent feature space.

3.3.2. Feature balancing through attentional prototype learning

Prototype-based method can be used to address the data imbalance problem [36]. Let $X_k = [x_1, \ldots, x_m] \in \Re^{m\times dl}$ be a matrix of time series embeddings for class k, where m is the total number of data samples with a class label, k. In this paper, we use the pooling to transfer the latent feature to embeddings. Then, the attentional prototype of the class, C_k, can be presented by a weighted sum of individual sample embeddings [33]:

$$C_k = \sum_i A_{k,i} \cdot X_{k,i} \tag{3}$$

where $A_{k,i}$ is the weight of the i-th data sample in class k; and $X_{k,i}$ denotes the embeddings of the data sample.

The attention weights A_k for the k-th class can be computed using the following equation:

$$A_k = softmax(\omega_k^T sigmoid(V_k X_k^T)) \tag{4}$$

where $\omega_k \in \Re^{u\times 1}$ and $V_k \in \Re^{u\times dl}$ are trainable parameters for the attention model and u is the hidden dimension size for both trainable parameters. In this work, we use separate parameters ω_k and V_k for each class because different classes may pay distinct attention to their feature spaces.

By Equation (3), the imbalanced features are compensated for due to the classifier constructed with the same number of prototypes for each class. The resulting prototypes are used to compute the squared Euclidean distance between a class prototype and a time series, i.e., $dist$. The distance is used to construct the classifier represented by the following equation:

$$\mathbb{P}(y=k|x) = \frac{exp(-dist(f(X), C_k))}{\sum_i exp(-dist(f(X), C_i))} \tag{5}$$

3.4. Global model generation

With the FL framework, each client updates the central server with its trained local model, instead of training data. The central server generates a global model by aggregating local models from the clients, which can perform well with respect to the data points available at different clients. The strategy of how to generate a global model is vital for federated learning. In this work, there might be two strategies of obtaining the global model: 1) averaging the model parameters of each participant and 2) weighted averaging the model parameters of each participant.

We assume that there are N clients in our IcingFL, and there is a random selection of K client models to generate the global model, where $K = rN$, r is the participation rate for each client model. For a selected client, i, the local model is trained based on S_i samples from its own dataset. The i-th client model calculates the gradient g_i with the model W_c^i using gradient descent techniques, where c is the c-th communication round. It should be noted that all participating clients start with the same global model that was randomly initialized in the first training round. For a client learning rate of ξ, the local client update, W_{c+1}^i, is given by:

$$W_c^i - \xi g_i \rightarrow W_{c+1}^i \tag{6}$$

The central server aggregates all client models to create a new global model, W_{c+1}. As mentioned above, there are different methods for generating the global model, which are described as follows in detail:

1) **Arithmetic mean**

This method performs a simple average of the received client models to generate the global model [20], which is defined as follows:

$$\sum_{i=1}^{K} \frac{1}{K} W_{c+1}^i \rightarrow W_{c+1} \tag{7}$$

where W_{c+1}^i is the i-th client model in the $c+1$ communication round, K is the number of models involved in the learning process and W_{c+1} is the global model in the same round. This method is effective when the differences between the client models are minor.

2) **Weighted average**

The weighted average method takes into account the number of training samples of each client and gives a higher weight to the client with the more training samples, which is defined as follows:

$$\sum_{i=1}^{K} \frac{S_i}{K\sum_{i=1}^{K} S_i} W_{c+1}^i \rightarrow W_{c+1} \tag{8}$$

where W_{c+1}^i is the i-th client model in the $c+1$ communication round, W_{c+1} is the global model in the same round, S_i is the sample size of the i-th client and K is the number of models involved in the learning process. This method accounts for the difference in the training sample size of each client model. Compared to the simple averaging method that treats all client models the same, the weighted averaging method can take into account the influence of training sample sizes. This method is favored in this study.

3.5. Privacy analysis

With the FL framework, privacy can be preserved as follows. First, each client uses its own data X_{raw} to train the local model of the deep neural network $\Omega_i(\forall i)$, and the data are encoded. Therefore, it is not possible to control the operations of a wind turbine from the data (note that each wind turbine allows users to physically control its operations). Second, each wind turbine calculates its own gradient descent updates $g_i(\forall i)$ locally and uploads only the model parameters $W_i(\forall i)$ to the central server. Thus, no raw data (with possibly sensitive information) are sent over the network. Each client must also send the number of training samples N_i to the central server for the weight calculation. Homomorphic encryption is used to secure communication between the central server and each client, so that the model parameters and the training sample size will not be tampered with. Last, the central server will never store the local model $W_i(\forall i)$ and N_i of each client. They are immediately discarded when the global model has been generated. When the model training is completed, the global model W will be available for online blade icing detection.

4. Evaluation

4.1. Experimental settings and data

We simulate FL with one aggregation server and 70 clients. All experiments were performed on a server equipped with Tesla T4, 16 GB. PyTorch was used as the deep learning framework for model implementation. The following hyperparameters were set for model learning: the Adam optimization learning rate was set to 0.01, and the server and client epochs were set to 200 and 100, respectively.

The experimental data are from two wind farms with 60 and 10 turbines, respectively. The two wind farms are located in northern China, in Shanxi and Henan provinces, about 700 km apart. Wind farm SCADA systems are equipped with hundreds of sensors that record turbine status at a frequency of 5 s. Wind turbine experts have identified 16 variables related to blade icing, as shown in Table 1.

The data from the two wind farms have a length of 360 and 384 h, respectively. In this experiment, we will evaluate the performance of the global model. 90% of the data are used for training, and the remaining 10% for testing. To evaluate the robustness of IcingFL, we randomly set the imbalance ratio of each client's training samples with the highest ratio of 20:1 and the lowest ratio of 4:1. The imbalance ratio of the test data is 12:1. All experiments are repeated five times, and the average performance is reported.

Table 1
Specification of the SCADA data.

No.	Variable name	Description
1	wind_speed	Wind speed
2	wind_direction	Wind direction
3	generator_speed	Generator speed
4	power	Active power
5	yaw_position	Yaw position
6	pitch1_angle	Angle of pitch 1
7	pitch2_angle	Angle of pitch 2
8	pitch3_angle	Angle of pitch 3
9	pitch1_speed	Speed of pitch 1
10	pitch2_speed	Speed of pitch 2
11	pitch3_speed	Speed of pitch 3
12	environment_temp	Environment temperature
13	internal_temp	Internal temperature of nacelle
14	pitch1_moto_tmp	Temperature of pitch motor 1
15	pitch2_moto_tmp	Temperature of pitch motor 2
16	pitch3_moto_tmp	Temperature of pitch motor 3

4.2. Evaluation metrics

The following metrics are used for the evaluation, including **AUC**, **F1**, and **Matthews correlation coefficient (MCC)**. The definitions of **F1** and **MCC** are as follows:

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \end{aligned} \tag{9}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{10}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{11}$$

where TP, FP, FN, and TN represent true positive, false positive, false negative, and true negative, respectively.

AUC is another robust indicator to assess the performance of imbalanced learning. The AUC value ranges from 0.5 to 1, where 0.5 is a random estimate and 1 is an excellent classifier. The task in this paper is a binary classification problem and the AUC can be calculated using the following equation.

$$\mathtt{AUC} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \tag{12}$$

4.3. Evaluation of model aggregation

To evaluate the proposed model aggregation of IcingFL, we compare it with FedAvg [20], which aggregates the parameters of all clients by averaging. Both methods employ the same feature extractor to make the comparison fair. We vary the participation rate (PR), and measure the metrics including F1, AUC and MCC. The window size of the sensor data is 128 (almost 11 min). In IcingFL, the number of filters in the CNN part and the number of hidden nodes in the LSTM part are set to 128, and the participation rate of each client is set to 50%. The results are shown in Table 2.

Based on the results, we can observe that our model outperforms FedAvg for all metrics by varying the RP value, with the only exception of AUC, which FedAvg is better for $PR = 0.7$. The results also validate the effectiveness of the proposed weighted approach for global model generation by weighting the sample size of each client. Specifically, there is an absolute improvement of 0.44%, 1.23%, and 2.11% for F1, AUC, and MCC, respectively. When the PR is 0.3. When PR is 0.5, the absolute improvements of our method for the three metrics are 0.19%, 0.65%, and 2.96%, respectively. The AUC of FedAvg is slightly higher than ours when the PR is 0.7. From the results, we can also observe that the performance of

Table 2
Performance of model aggregation, % (Numbers in bold indicate the better performance).

PR	F1		AUC		MCC	
	FedAvg	IcingFL	FedAvg	IcingFL	FedAvg	IcingFL
0.3	93.12	**93.56**	78.23	**79.46**	46.21	**48.32**
0.5	93.68	**93.87**	79.38	**80.03**	47.15	**50.11**
0.7	93.26	**93.70**	**81.19**	79.96	48.76	**49.88**

IcingFL increases with PR from 0.3 to 0.5. This is because the more clients participate in the training, the more information they provide, and this information helps to improve performance. However, we observe a slight drop in performance from 0.5 to 0.7, which can be explained by the fact that it becomes difficult to further improve the performance when more clients join. Therefore, there is a pivot to the increasing value of PR.

4.4. Evaluation of feature extraction

To evaluate the performance of the proposed feature extractor, we select five commonly used neural networks as baselines for comparison, including 1) **MLP** (MultiLayer Perceptron), a three-layer MLP with 128 hidden nodes in each layer and with a dropout layer between two layers of the MLP [37]; 2) **LSTM**, which is one of the most popular neural networks for time series modeling. The performance with the number of hidden nodes of {8, 16, 32, 64, 128} will be evaluated, and the best will be selected for comparison; 3) **CNN**, which will be evaluated with the filter number in {16, 32, 64, 128}, and the best one will be selected; 4) **FCN**, which is a fully convergent network for time series classification [37]. The model with the same structure and hyperparameters as in Ref. [37] will be evaluated; and 5) **DenseNet**, which has the state-of-the-art performance according to Ref. [9]. The model with the same network structure as in Ref. [9] will be used, but all the attention layers will be removed.

The comparison will use the same experimental parameters as in Section 4.3. Table 3 shows the results, where the best value is shown in bold. As can be seen, IcingFL has the best performance, except for AUC, which ranks second, a little lower than CNN. For MCC, the performance of our model shows a 9.91% improvement over the second ranked method, DenseNet, and 51.30% over the last ranked method, MLP. As for F1, the results of all methods are close. LSTM has the worst AUC value, but its F1 is second. This means that although LSTM can identify whether the blade is icing or not, it is not competitive in terms of values of all metrics. Another interesting result is that the performance does not scale with the complexity of the model structure. For example, DenseNet has the most complex network, but its performance is not the best. This suggests that we should choose the appropriate model, instead of increasing the model complexity, i.e. the number of layers of a neural network. The results have validated the effectiveness of our model structure that combines CNN and LSTM to extract spatial and temporal information.

4.5. Evaluation of imbalance learning

We evaluate the proposed imbalanced learning method by comparing it with two other famous baselines: 1) **WeightedCE (weighted cross entropy)**, in which different weights are assigned to classes. With cross-entropy, the ability to learn features can be greatly improved from imbalanced data; 2) **Focalloss**, which has a popular loss function for imbalanced learning. It compensates for imbalanced features using a specific-designed loss function [31]. To make a fair comparison, in our method we replace the prototype component with the two loss functions and set the imbalance ratio to \{6, 8, 10\}. The experiments use the following two widely used metrics for imbalance learning evaluation, AUC and MCC. Table 4 shows the results.

From the results, we can see that the proposed imbalanced learning method is more effective than the others. In terms of AUC, our method achieves an improvement of 3.83%, 2.99%, and 5.23% over the WeightedCE method for the three imbalance ratios, and an improvement of 8.41%, 10.80%, and 20.74% over the Focalloss method for the three ratios. In terms of MCC, our method shows an improvement of 26.77%, 32.64%, and 41.22% over WeightedCE for the three ratios and a slight improvement over Focalloss. From the results, we also find that the performance of WeightedCE and Focalloss decreases with an increasing imbalance ratio. In contrast, our method is more stable. Therefore, we can safely conclude that the proposed method is effective in addressing the imbalance problem.

4.6. Ablation study

To assess the effectiveness of the key components of our model, we perform ablation and sensitivity analysis in this section. We compare the following two variants of IcingFL: 1) **IcingFL_No_SE**, where the SE module was removed; 2) **IcingFL_No_Imbalance**, where the feature compensation module was removed, but a fully connected layer was added. We use the same settings as in Section 4.3 for the experiments.

Fig. 3 shows the results of the ablation study. We can observe that when no attention layer is used, the performance of F1, AUC, and MCC decreases by 1.35%, 2.58%, and 2.85% (absolute value) compared to IcingFL (see IcingFL_No_SE line). When no feature balancing module is used, the performance of F1, AUC and MCC decrease by 0.16%, 8.6% and 10.35%, respectively (see the IcingFL_No_Imbalance line).

4.7. Sensitivity study

We perform a sensitivity analysis to assess the impact of hyperparameters, including the window size for training data segmentation and the participation rate of clients. Sensitivity analysis is evaluated based on three metrics, including F1, AUC, and MCC.

Fig. 4 shows that performance varies with the window size used, but in general, the accuracy for all three metrics increases steadily with increasing window size. It reaches its maximum when the window size is 128, and then slowly decreases.

Fig. 5 shows the performance with the participation rate increased from 0.1 to 1 with a step size of 0.1. The results show that the performance of the three metrics increases steadily with increasing participation rate before 0.5, but the performance tends to be stable after 0.5. However, when the participation rate is low, i.e., below 0.2, the performance is not as good as when it is high. This may be because what IcingFL has learned is not sufficient to support the construction of a good model. Another possible explanation is that the knowledge learned by participating clients is not transferred well to nonparticipating clients. Since the test set contains data from 70 clients and not all of them can be learned by the model. Therefore, the accuracy of the final test results will remain low if the knowledge has not been transferred well.

4.8. Online blade icing detection

We train the model offline and then use it for online detection in

Table 3
Performance of feature extractor, % (Numbers in bold indicate the best performance).

Method	F1	AUC	MCC
MLP	92.86	75.84	33.12
LSTM	93.84	74.37	43.51
CNN	92.34	**80.57**	44.61
FCN	93.24	77.08	42.56
DenseNet	93.71	78.14	45.59
IcingFL	**93.87**	80.03	**50.11**

Table 4
Performance of model aggregation, % (Numbers in bold indicate the best performance).

Imbalance Ratio	AUC			MCC		
	WeightedCE	Focalloss	IcingFL	WeightedCE	Focalloss	IcingFL
6	78.10	74.80	**81.09**	40.42	49.39	**51.24**
8	77.80	72.32	**80.13**	37.74	47.34	**50.06**
10	76.91	67.22	**81.16**	35.35	43.02	**49.98**

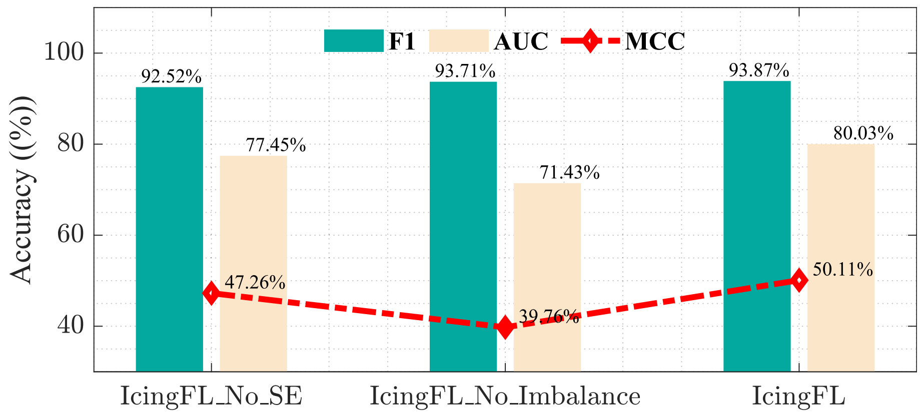

Fig. 3. Ablation study.

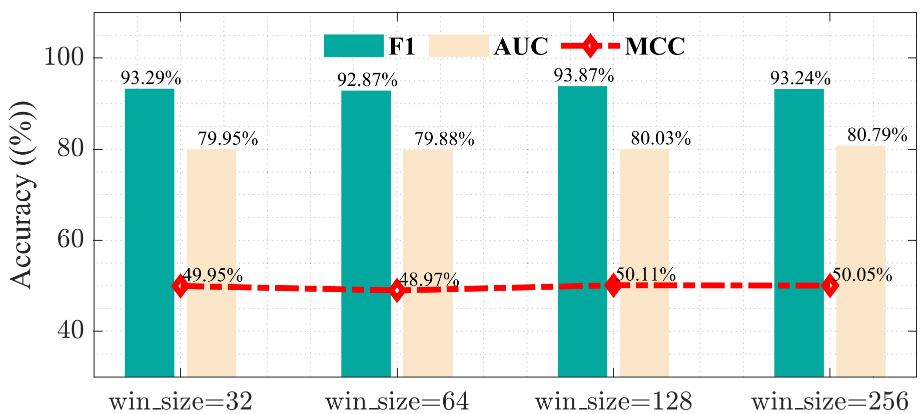

Fig. 4. Sensitivity analysis of varying window size.

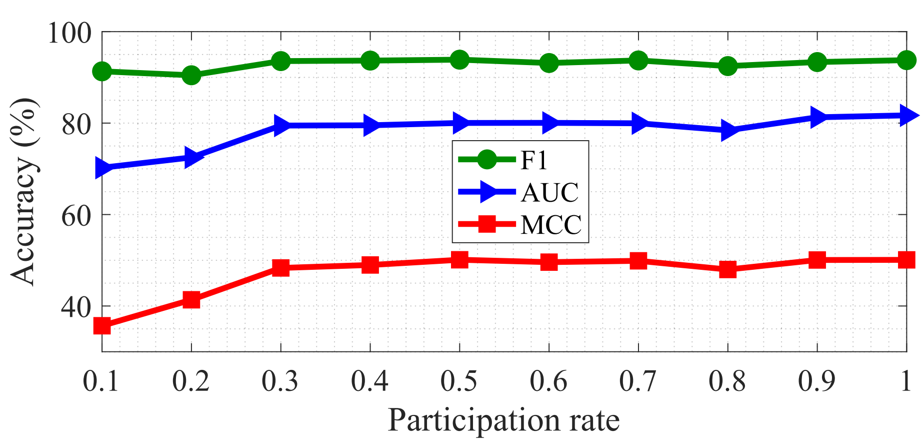

Fig. 5. Sensitivity analysis of varying participation rate.

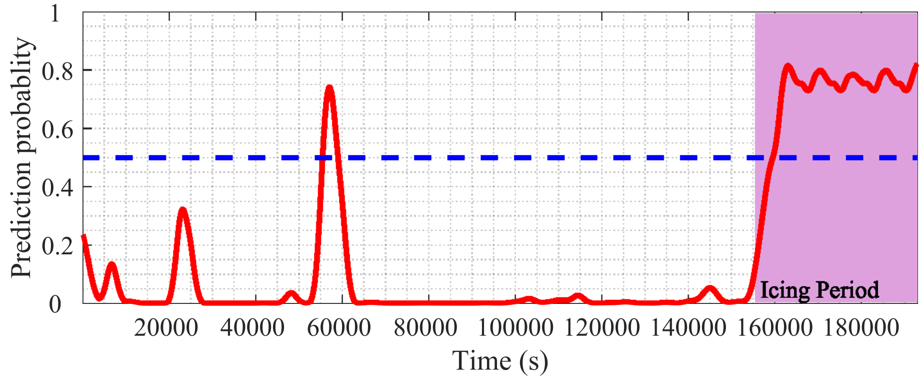

Fig. 6. Online blade icing detection of wind turbines.

wind farms. Online detection uses the sliding-window approach, which divides the time series into equally sized windows as the training data segmentation. The model calculates the probability of blade icing for the input data. If the probability value is greater than a given threshold value, it means that the blade is icing, otherwise it is not icing. To ensure the robustness of the online detection, we use the majority voting algorithm [19] to further improve the accuracy of the detection.

Fig. 6 shows the results of the online detection. The magenta-colored area represents the icing period identified by a wind farm engineer, and the blue dotted line represents the given threshold to identify blade icing. From the results, the proposed model can estimate the blade icing condition with almost 100% accuracy using SCADA data.

5. Conclusion and future work

Data management related issues are becoming critical for wind farms due to privacy, ownership, competition and technical barriers. This paper proposes an imbalanced federated learning model for wind turbine blade icing detection, IcingFL. This model allows different clients to distributively train local models without sharing their data, then the trained local models are aggregated to obtain a global model by a central server. In this paper, we addressed the data imbalance problem by proposing a prototype learning method, which can balance the learned prototypes in a latent space and generate the classification model for detecting blade icing state. We also proposed averaging and weighted averaging methods to generate the global model. The weighted averaging method is more effective as it takes into account the sample size of each client. We comprehensively evaluated the model by comparing it with five state-of-the-art models. The results showed that the proposed IcingFL model outperforms the others remarkably. We compared the proposed imbalance learning method with two baselines and the results indicated the superiority of our model in terms of accuracy. Finally, we performed ablation and sensitivity studies, as well as online experiments, which validate the effectiveness of model design, parameter setting, and its capability for real-world blade icing detection.

There are several directions for future work. First, we will study server-side imbalance learning in FL, while in this paper, we focus more on client-side learning. Second, we will develop a method capable of identifying the severity of wind turbine blade icing. Finally, we also plan to explore model update policies for online detection.

Credits author statement

Conceptualization, ALL; methodology, X. C.; investigation and resources, X. C., L.H., X. L.; writing—original draft preparation, X. C.; writing—review and editing, ALL; project administration L. H; funding acquisition, L. H. and X. L. All authors have read and agreed to submit the manuscript.

Acknowledgements

The research was supported by the National Natural Science Foundation of China (62020106004) and by the Flexible Energy Denmark project (8090-00069B) funded by Innovationsfonden.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Ren Z, Jiang Z, Skjetne R, Gao Z. Development and application of a simulator for offshore wind turbine blades installation. Ocean Eng 2018;166:380–95.
[2] Ali QS, Kim M-H. Design and performance analysis of an airborne wind turbine for high-altitude energy harvesting. Energy 2021;230:120829.
[3] Fei Z, Tengyuan W, Xiaoxia G, Haiying S, Hongxing Y, Zhonghe H, Yu W, Xiaoxun Z. Experimental study on wake interactions and performance of the turbines with different rotor-diameters in adjacent area of large-scale wind farm. Energy 2020;199:117416.
[4] Wei K, Yang Y, Zuo H, Zhong D. A review on ice detection technology and ice elimination technology for wind turbine. Wind Energy 2020;23(3):433–57.
[5] Laakso T, Holttinen H, Ronsten G, Tallhaug L, Horbaty R, Baring-Gould I, et al. State-of-the-art of wind energy in cold climates. IEA annex XIX 2003;24:53.
[6] Dalili N, Edrisy A, Carriveau R. A review of surface engineering issues critical to wind turbine performance. Renew Sustain Energy Rev 2009;13(2):428–38.
[7] Yin C, Zhang Z, Wang Z, Guo H. Numerical simulation and experimental validation of ultrasonic de-icing system for wind turbine blade. Appl Acoust 2016;114:19–26.
[8] Shu L, Qiu G, Hu Q, Jiang X, McClure G, Yang H. Numerical and field experimental investigation of wind turbine dynamic de-icing process. J Wind Eng Ind Aerod 2018;175:90–9.
[9] Cheng X, Li G, Ellefsen AL, Chen S, Hildre HP, Zhang H. A novel densely connected convolutional neural network for sea state estimation using ship motion data. IEEE Trans Instrum Meas 2020;69(9):5984–93.
[10] Hao M, Li H, Luo X, Xu G, Yang H, Liu S. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. IEEE Trans Ind Inf 2019;16(10):6532–42.
[11] Zhang M, Yu J, Chen R, Liu Q, Liu J, Song D, Liu P, Gao L, Wang J. Highly transparent and robust slippery lubricant-infused porous surfaces with anti-icing and anti-fouling performances. J Alloys Compd 2019;803:51–60.
[12] Peng C, Xing S, Yuan Z, Xiao J, Wang C, Zeng J. Preparation and anti-icing of superhydrophobic pvdf coating on a wind turbine blade. Appl Surf Sci 2012;259:764–8.

[13] Muñoz CQG, Márquez FPG, Tomás JMS. Ice detection using thermal infrared radiometry on wind turbine blades. Measurement 2016;93:157–63.
[14] Shajiee S, Pao LY, McLeod RR. Optimizing the layout of heaters for distributed active de-icing of wind turbine blades. Wind Eng 2014;38(6):587–600.
[15] Zeng J, Song B. Research on experiment and numerical simulation of ultrasonic de-icing for wind turbine blades. Renew Energy 2017;113:706–12.
[16] Corradini ML, Cristofaro A, Pettinari S. A model-based robust icing detection and estimation scheme for wind turbines. In: 2016 European control conference (ECC). IEEE; 2016. p. 1451–6.
[17] Jiménez AA, Márquez FPG, Moraleda VB, Muñoz CQG. Linear and nonlinear features and machine learning for wind turbine blade ice detection and diagnosis. Renew Energy 2019;132:1034–48.
[18] Liu Y, Cheng H, Kong X, Wang Q, Cui H. Intelligent wind turbine blade icing detection using supervisory control and data acquisition data and ensemble deep learning. Energy Sci Eng 2019;7(6):2633–45.
[19] Yuan B, Wang C, Jiang F, Long M, Yu PS, Liu Y. Waveletfcnn: a deep time series classification model for wind turbine blade icing detection; 2019. arXiv; 2019. arXiv:1902.05625.
[20] McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. PMLR; 2017. p. 1273–82.
[21] Wang Y, Bennani IL, Liu X, Sun M, Zhou Y. Electricity consumer characteristics identification: a federated learning approach. IEEE Trans Smart Grid 2021;12(4):3637–47.
[22] Lin W-T, Chen G, Huang Y. Incentive edge-based federated learning for false data injection attack detection on power grid state estimation: a novel mechanism design approach. Appl Energy 2022;314:118828.
[23] Su Z, Wang Y, Luan TH, Zhang N, Li F, Chen T, Cao H. Secure and efficient federated learning for smart grid with edge-cloud collaboration. IEEE Trans Ind Inf 2021;18(2):1333–44.
[24] Diez-Olivan A, Del Ser J, Galar D, Sierra B. Data fusion and machine learning for industrial prognosis: trends and perspectives towards industry 4.0. Inf Fusion 2019;50:92–111.
[25] Sattler F, Wiedemann S, Müller K-R, Samek W. Robust and communication-efficient federated learning from non-iid data. IEEE Transact Neural Networks Learn Syst 2019;31(9):3400–13.
[26] Qu Y, Pokhrel SR, Garg S, Gao L, Xiang Y. A blockchained federated learning framework for cognitive computing in industry 4.0 networks. IEEE Trans Ind Inf 2020:1.
[27] Kumar R, Wang W, Yuan C, Kumar J, Qing H, Yang T, et al. Blockchain based privacy-preserved federated learning for medical images: a case study of covid-19 ct scans. Med Image Anal 2021. arXiv:2104.10903.
[28] Xie Y, Qiu M, Zhang H, Peng L, Chen Z. Gaussian distribution based oversampling for imbalanced data classification. IEEE Trans Knowl Data Eng 2022;34(2):667–79.
[29] Liu C-L, Hsieh P-Y. Model-based synthetic sampling for imbalanced data. IEEE Trans Knowl Data Eng 2019;32(8):1543–56.
[30] Wang L, Xu S, Wang X, Zhu Q. Towards class imbalance in federated learning; 2020. arXiv; 2020. arXiv:2008.06217.
[31] Lin T-Y, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 2980–8.
[32] Cui Y, Jia M, Lin T-Y, Song Y, Belongie S. Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2019. p. 9268–77.
[33] Zhang X, Gao Y, Lin J, Lu C-T. In: Tapnet: multivariate time series classification with attentional prototypical network. AAAI; 2020. p. 6845–52.
[34] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 7132–41.
[35] Tian W, Cheng X, Li G, Shi F, Chen S, Zhang H. A multilevel convolutional recurrent neural network for blade icing detection of wind turbine. IEEE Sensor J 2021;21(18):20311–23.
[36] Cheng X, Shi F, Liu X, Zhao M, Chen S. A novel deep class-imbalanced semi-supervised model for wind turbine blade icing detection. IEEE Transact Neural Networks Learn Syst 2021;33(6):2558–70.
[37] Wang Z, Yan W, Oates T. Time series classification from scratch with deep neural networks: a strong baseline. In: 2017 international joint conference on neural networks (IJCNN). IEEE; 2017. p. 1578–85.