FedBIP

A Federated Learning Based Model for Wind Turbine Blade Icing Prediction

Zhang, Dongtian; Tian, Weiwei; Cheng, Xu; Shi, Fan; Qiu, Hong; Liu, Xiufeng; Chen, Shengyong

Published in:
IEEE Transactions on Instrumentation and Measurement

Link to article, DOI:
10.1109/TIM.2023.3273675

Publication date:
2023

Document Version
Peer reviewed version

FedBIP: A Federated Learning Based Model for Wind Turbine Blade Icing Prediction

Dongtian Zhang, Weiwei Tian, Xu Cheng, Fan Shi, Hong Qiu, Xiufeng Liu, Shengyong Chen

***Abstract*—Prediction of icing on wind turbine blades is crucial, particularly in high-latitude areas where ice accumulation is a frequent occurrence. Traditional centralized data-driven approaches for predicting blade icing have demonstrated promising performance, but they require a large amount of storage and computational resources and may also raise concerns about data privacy. Federated Learning (FL) presents a potential solution to address these issues. These challenges include redundant features in the collected data, a highly imbalanced data distribution between normal and icing samples, and slow model convergence during FL training. To tackle these challenges, we proposed a novel FL model called FedBIP. FedBIP employs a feature selection approach enhanced with human knowledge to select relevant features, a segmentation-based oversampling method to alleviate class imbalance, and a new aggregation method that takes into account data size, timestamps, and offsets of each participating client. In addition, knowledge distillation is employed in the local model training to accelerate model convergence and speed up the overall training process. The results of comprehensive experiments demonstrate that FedBIP outperforms state-of-the-art FL methods, aggregation methods, and feature extractors. Ablation and sensitivity analysis were also conducted to validate the importance of each component and key parameters in FedBIP.**

***Index Terms*—Federated learning, wind turbine, icing detection, class imbalance, model aggregation.**

I. Introduction

WIND energy has the potential to play a significant role in meeting the growing demand for clean energy due to its abundance and lack of cost. In fact, it is the largest contributor to the renewable energy category, alongside hydroelectric power [1]. However, wind farms are often located in high-altitude areas with cold climates, which can lead to ice accumulation on the blades of wind turbines during the winter season. This can result in a loss of power generation and, in some cases, a reduction of up to 30% in electricity production over the course of a year [2]. In addition, ice accumulation on wind turbines poses a safety risk to nearby facilities. As such, predicting and addressing icing on wind turbines has become an important area of research.

Dongtian Zhang, Xu Cheng, Fan Shi, and Shengyong Chen are with the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China, 300386.

Weiwei Tian is with the Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Aalesund, Norway, 6009.

Hong Qiu is with the college of big data and software engineering, Zhejiang Wanli University, Ningbo, China, 315100.

Xiufeng Liu is with the Department of Technology, Management and Economics, Technical University of Denmark, Produktionstorvet, Denmark, 2800.

Dongtian Zhang and Weiwei Tian are equal contributions. Xu Cheng and Fan Shi are the corresponding authors.

There are three main approaches for predicting icing on wind turbine blades: human observation, direct methods, and indirect methods. Human observation relies on the experience of experts and can be subjective. Direct methods involve measuring the attributes of ice, such as ultrasonic damping [3], piezoelectric sensors [4], [5], or temperature changes [6]. However, some of these methods are impractical or difficult to use in a timely manner. Indirect methods include using double anemometers [7], measuring visibility and cloud base height [8], or monitoring engine room mechanical vibration and power curves [9]. However, these methods may be prone to machine errors and may require the use of weather forecasts to be more accurate. In recent years, model-driven and data-driven methods have emerged as alternatives to these traditional approaches. Model-driven methods use mathematical models, such as the one proposed by Pedersen et al. [10], which combines weather forecasts with a conventional CFD model to predict the power loss caused by icing on wind turbines. However, these methods require assumptions about the icing conditions and may require external tools such as wind tunnels for experimentation. In contrast, data-driven methods utilize the data collected from sensors on wind turbines and do not require a deep understanding of the icing process [11].

Traditional centralized data-driven approaches for model training rely on the consolidation of data in a single location, such as a cloud. However, this requires a significant amount of storage and computing resources and may also raise concerns about data privacy [12]. In addition, the network bandwidth between the server and each wind turbine must meet certain requirements. Federated learning (FL), a new distributed learning paradigm, was developed as a solution to these challenges [13]. It allows the training process to be distributed across multiple clients, such as individual wind farms, rather than being centralized on a server. This not only protects the data from being accessed by others, but also reduces the computing resources required compared to centralized methods. In FL, the model weights are updated by synchronizing with the clients, rather than training on the entire dataset. This enables each wind farm to act as a "data island," protecting their data while still contributing to the overall model training.

However, there are several challenges to using federated learning (FL) for predicting whether wind turbine blades are in icing condition:

1) Feature redundancy: Numerous sensors are installed to monitor the status of wind turbines, and some, such as temperature sensors, may be more important than others for detecting blade icing. Training a model on all available features without selecting the most relevant ones can be

computationally burdensome.

2) Class imbalance: Wind turbines typically operate most of the time and only occasionally shut down due to ice accumulation, which leads to an imbalanced dataset with a disproportionate number of normal data samples compared to abnormal ones. If a model is trained on this imbalanced dataset without addressing the imbalance, it may be biased towards the majority class.

3) Model convergence: Although FL distributes the training process across multiple clients, the training process on each client can still be seen as centralized. The convergence speed of the local models on each client may vary, and their participation in the federated training process may be uncertain. This can affect the convergence speed of the aggregated global model and slow down the overall training process.

To tackle the challenges mentioned above, a **fed**erated learning-based model for **b**lade **i**cing **p**rediction (FedBIP) is proposed. Specifically, a feature selection approach enhanced with human knowledge is used to reduce feature redundancy, and a segmentation-based class imbalance method is employed to address the issue of imbalanced data. To improve model convergence, a new model aggregation method is implemented that takes into account the importance of data size, timestamps, and offsets of each participating client. Additionally, a knowledge distillation (KD) mechanism is used to accelerate the training process and improve the performance of the global model.

The contributions of this paper are listed below:

1) A novel FL-based model, FedBIP, is proposed for wind turbine blade icing prediction. FedBIP addresses the challenges of applying FL to blade icing prediction by incorporating the proposed human-knowledge-enhanced feature selection, segmentation-based oversampling, and a new aggregation method. Through the use of these techniques, FedBIP can select informative features, reduce communication rounds, and consider the importance of data size, timestamps, and offsets of each participating client during FL training.

2) Thorough evaluations have been conducted to assess the effectiveness of the FedBIP model using real-world wind turbine data from two different wind farms. The results of the comparison with state-of-the-art FL methods, aggregation methods, class imbalance processing methods, and local feature extractors, clearly demonstrate the superiority of the FedBIP model. Furthermore, the ablation and sensitivity analysis have been conducted to verify the importance of key components and parameters in the FedBIP model. These evaluations provide strong evidence of the effectiveness and practical value of the proposed FedBIP model for wind turbine blade icing prediction.

The remainder of the paper is structured as follows. Section II presents an overview of the existing literature on turbine blade icing prediction and federated learning. The proposed methods FL-based model is presented in Section III. In Section IV, experimental results are provided to demonstrate the effectiveness of the proposed model. Finally, section V concludes the paper.

II. Related Work

A. Data-Driven Wind Turbine Blade Icing Prediction

In recent years, researchers have explored various machine learning and deep learning techniques for this task. One common approach is to use supervised learning algorithms to learn a mapping from sensor data to icing conditions. For example, Chen et al. introduced a deep neural network-based model with a bypass component for detecting wind turbine faults [14]. Yuan et al. used a wavelet-enhanced autoencoder model to identify icing conditions [15], while Tian et al. combined wavelet transformation with a multilevel convolutional recurrent model for turbine icing detection [16]. Tong et al. proposed a sample distribution-aware adaptive weighted kernel extreme learning machine algorithm [17], and Cheng et al. developed a temporal attention convolutional model for icing estimation [18]. Wang et al. proposed a wavelet-based multiscale long short-term memory network for wind turbine blade icing detection [19], and Xiao et al. proposed a group method of data handling technique-based selective deep ensemble model [20]. Another direction is to use semi/unsupervised learning algorithms to identify patterns in the sensor data that may indicate the presence of icing. Examples include a statistical model for predicting icing-induced energy losses for wind turbines [21], a clustering-based approach to identify icing patterns in wind turbine blades [11], and a semi-supervised model for blade icing detection [22].

Many existing works have trained models for wind turbine icing prediction by centralizing data from different wind farms, typically in a cloud, and training a model on this data. However, this centralized data-driven approach requires significant storage and computational resources, and wind farm owners may be hesitant to share their data for commercial reasons. In this study, we propose using FL as a distributed training approach to bypass these issues associated with training the model.

B. Federated Learning

FL is a distributed machine learning approach introduced in [13] that allows participants to train a local model using their data, with the global model aggregated on the server without the data being uploaded. FL has been applied to a range of tasks [23]. Cheng et al. were the first to apply FL with a heterogeneous structure between the client and server for detecting wind turbine in icing conditions [12]. They later proposed an improved version that integrates Blockchain with FL for blade icing detection [24], and a version that emphasizes class imbalance learning [25]. Jiang et al. implemented a multi-scale residual attention network enhanced FL framework for wind turbine fault diagnosis [26]. The results of these studies suggest that FL can produce a good performance in these tasks. There are some works of applying FL for fault diagnosis other than wind turbine blade icing detection [27]–[29].

Few existing works have applied FL to wind turbine icing prediction and have used different aggregation strategies, which can impact the effectiveness of the training process.

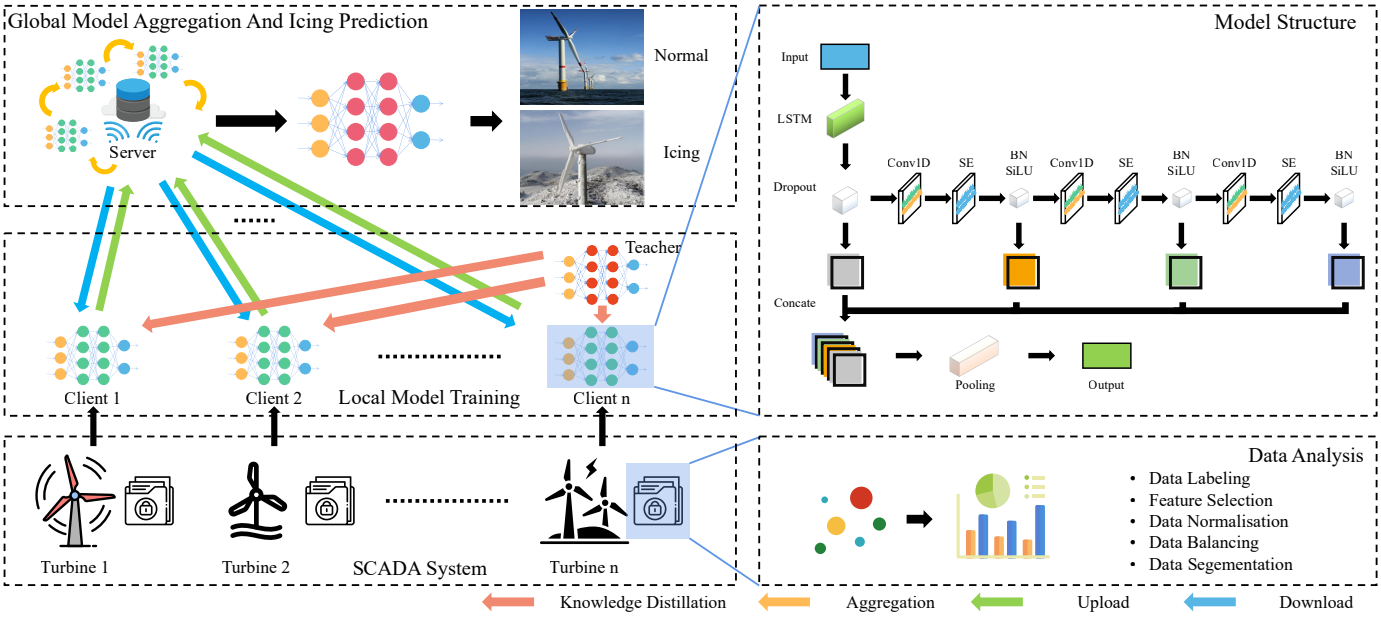

Fig. 1. Overview of the proposed FL-based wind turbine icing prediction. The model consists of three parts: data analysis, local model training, and global model aggregation.

In this paper, we address the issue of inactive and poorly-trained clients with varying amounts of data by considering the importance of data sizes, timestamps, and offsets of each participating client in the model aggregation process.

III. Methods

In this section, we first describe the overview of the whole proposed structure, and then each procedure's detail is separately described.

A. Overview

The schematic representation of the FedBIP framework is illustrated in Figure. 1. The framework is composed of three main components: data analysis, local model training, and global model aggregation.

In the data analysis, raw data acquired from the wind turbines' Supervisory Control and Data Acquisition (SCADA) systems are preprocessed to mitigate potential risks associated with these data. The data analysis mainly includes data labeling, feature selection, normalization, and egementation.

The local model training phase involves each wind farm training a local model using the preprocessed data. A sequential structure, comprising a long short-term memory (LSTM) and a three-layer densely connected convolutional neural network (CNN), is utilized for feature extraction on each client. During each communication round, the local clients train their model with local data and upload their model's parameters to the server.

In the global model aggregation phase, the server aggregates the parameters of all participating clients using the proposed aggregation method, which takes into account the importance of data sizes, timestamps, and offsets of each participating client. The aggregated model is then broadcast back to each client for the next communication round. A KD mechanism is employed in the local training process to enhance the performance of local models and accelerate the training process. Once communication rounds have been completed, the final server model is used for icing prediction.

B. Data Analysis

In this work, all of the data are labeled by experienced experts. If there are some uncertain periods in which the experts hard to decide if it is icing or not, these periods are removed.

Considering the large number of features that can be obtained from the SCADA system, we employ a feature selection approach enhanced with human knowledge. With the aim of selecting the same features for different wind turbines, experts first remove most of the irrelevant features, and then the chi-square test is used to calculate and rank the relevance score between the collected features and the labels. As the wind turbines are selected from two different wind farms, the data may vary in different distributions and certain features may be deemed important for one farm but not for the other. Even for turbines on the same farm, relevant features may vary to some extent. In this situation, we try our best to select the common features from the ranking list.

To eliminate the negative impact of individual data samples and reduce the disparity between data values and feature dimensions, the original data undergo a normalization process to transform their numerical range into [0, 1].

C. Segmentation-based Sample Balancing

As mentioned above, wind turbines typically operate for the majority of the time, but may occasionally shut down

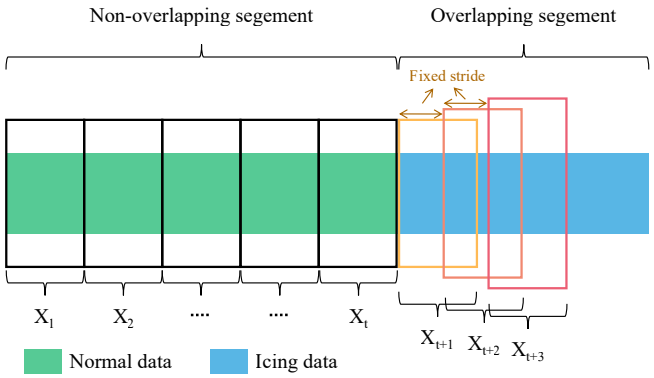

Fig. 2. An illustration of data sample generation for normal samples and icing samples

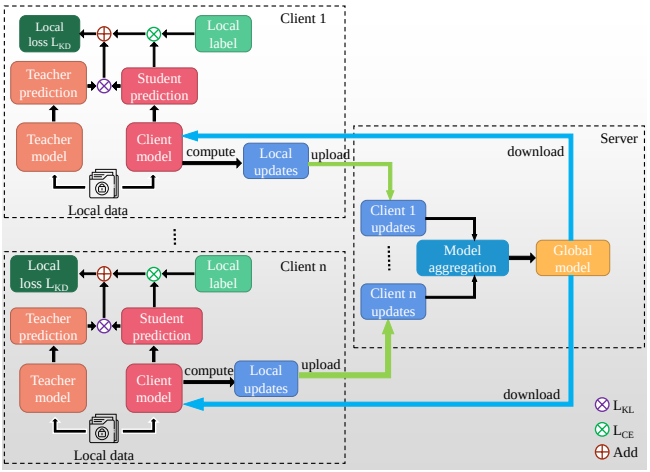

Fig. 3. Procedure of distribute training with knowledge distillation

due to ice accretion or other faults. This results in a large imbalance between normal and abnormal data samples. Imbalanced training data can negatively impact the performance of the trained model, particularly in terms of predicting minority classes [30]. To mitigate the impact of imbalanced data on training processes, we introduce a segmentation-based sample balancing method.

To address the issue of imbalanced data in training, we propose using a segmentation-based sample balancing method instead of traditional oversampling and undersampling techniques. These traditional methods can alter the original timing structure of time-series data, whereas the segmentation-based method preserves the timing structure while still addressing the class imbalance. Specifically, we use a sliding window without overlap for normal samples and a sliding window with overlap for icing samples, as shown in Fig. 2.

During the data balancing process, only the samples used for training undergo re-sampling, while the data used for testing remains unchanged in order to preserve the authenticity of the model testing. After the balancing procedure, we segment the data into fixed-length samples, as the raw data collected from wind turbine SCADA systems contains time series trends over consecutive time periods.

D. Local Model Training

1) Model Structure: The collected data from the SCADA system is time series data essentially, and therefore we propose using a combination of LSTM and CNN to learn both temporal and spatial information. The raw time series data is first input into an LSTM to learn temporal information, and then three sequential CNN layers are used to learn spatial information. In order to account for the harsh and variable working conditions of wind turbines, the CNN layers are equipped with dense connections and Squeeze-and-Excitation (SE) modules [31] to capture multi-scale features and emphasize important features. Additionally, each CNN layer includes a Batch normalization layer and a SiLU layer after the SE module. The SiLU layer, also known as the Swish activation function, is differentiable, which makes it suitable for use in back propagation and then enhances the capability of learning complex features. It is worth noting that the client and server models both share this same structure in our approach.

Assume the output of the LSTM is $X_l \in \mathbb{R}^{C\times L}$, where C represents the channels, and L is the time window size of each sample. For the input X_{in} of each CNN layer, the output of each component of each CNN layer can be formulated as:

$$X_{conv} = Conv1d(X_{in}) \tag{1}$$

$$X_{se} = SE(X_{conv}) \tag{2}$$

$$X_{norm} = Norm1d(X_{se}) \tag{3}$$

$$X_{out} = SiLU(X_{norm}) \tag{4}$$

where X_{conv} denotes the output of the Conv1d layer, X_{se} denotes the output of SE layer, X_{norm} denotes the output of Norm1d layer, and X_{out} denotes the output of SiLU layer. These outputs have the same shape, $\mathbb{R}^{H\times L}$, where H denotes the number of filters in the CNN. The three CNN layers are sequentially connected, and the input of the latter CNN layer is the output of the previous CNN layer and the input of the first CNN layer is the output X_l of the LSTM layer. The outputs of these CNN layers are denoted as X_1, X_2, and X_3. Together with X_l, these four outputs are concatenated by the feature columns.

2) Knowledge Distillation: 'Cumbersome' networks require a significant amount of training time to reach their global or local optima, and also require a large amount of computational resources [32]. The outputs of a model are probabilities indicating the likelihood of an input being classified into each class. However, the highest probability of an input does not necessarily mean that the input belongs to that class, but rather that it resembles that class closely. This is knowledge that can be transferred from a teacher model to a student model through the process of knowledge distillation.

In FL, the model is trained distributively by splitting the centralized server training into collaboration among clients. Although each local client training procedure can be considered as an individual centralized training, the number of samples on each client is typically smaller than if all the samples were centralized on one server. As a result, training from scratch can take a long time to reach the 'global'

optimum. Additionally, model updates can be extremely large if they contain a large number of parameters, and many communication rounds may be needed to generate a high-quality global model [33]. However, using knowledge distillation can accelerate the training process.

In this paper, the procedure of KD is shown in Fig. 3. By introducing the teacher model into the training process, we first calculate the Kullback-Leibler divergence loss (L_{KL}), then, combined the L_{KL} with the originally used loss function cross-entropy loss (L_{CE}), to get the total loss (L_{KD}) for local model training. The mathematical processing are given below:

$$L_{KL} = KL(log(\frac{p_s}{T})||\frac{p_t}{T}) = log(\frac{p_s}{T}) \cdot \frac{log(\frac{p_s}{T})}{\frac{p_t}{T}} \tag{5}$$

$$L_{CE} = -y \cdot log(p_s) + (1-y) \cdot log(1-p_s) \tag{6}$$

$$L_{KD} = \alpha \cdot T^2 \cdot L_{KL} + (1-\alpha) \cdot L_{CE} \tag{7}$$

where p_s is the value of a student model's prediction and p_t is the value of a teacher model's prediction, and y is the true value. α is hyperparameter and T resembles the temperature that is used to distil the 'soft target'.

In this paper, each client trains its local model using traditional centralized training methods, such as using the collected data and the Adam optimization algorithm to update the model's parameters via gradient descent. For a client with the learning rate of ϵ and local gradient g_i, the local client update, W_c^i is calculated by:

$$W_{c-1}^i - \epsilon \cdot g_i \longrightarrow W_c^i \tag{8}$$

E. Global Model Aggregation

In FL, the server is tasked with aggregating model parameters from multiple clients. This process presents several challenges. Firstly, clients with more data tend to be better trained, and if all clients are treated equally, those with fewer data may negatively impact the training procedure. To address this, it is necessary to assign different weights to each client based on the number of data they contain. Secondly, the clients participating in each communication round are randomly selected from the entire group, which may result in some clients not participating in training for extended periods of time. This can lead to a lack of fitness in these clients and may slow the training process [34]. To address this, the server should also consider the amount of time each participating client has been involved in training. Finally, during each federated communication round, not all updates may be effective, and some models may not perform as well as previously. In this case, if a client's model update is better than the previous one, it is defined as "positive," otherwise it is "negative." When calculating the final global updates by adding the value of each parameter, those with larger absolute values may significantly impact the aggregation process. As such, the server should update in the direction of the majority of participating models' updates.

To address these challenges, we propose a novel aggregation method that takes into account the data size, timestamp, and offset of each participating client. The data size is used to give different weights to the model updates from different clients. The timestamp is used to determine how recently the client has participated in the training process. The offset is used to measure the improvement or decline of the client's model performance compared to the previous communication round. By considering these three factors, the proposed aggregation method can effectively address the challenges mentioned above and improve the performance of the global model.

The procedure of the central server aggregating the client models is as follows. For each communication round c, first, there are K client models randomly selected from all the clients. K is calculated with $K = pr \cdot NC$, where pr is the participation rate and NC is the total number of clients. Then, the local model of each participating client is trained using its own local data that consists of N_i samples where i is the index of the i-th participating client. During the training process, the server together with all the client models is randomly initialized in the beginning.

After each participating client calculated the local gradient and used the before gradient to minus this one, the server then aggregates all the weights of participating client models to generate a new global model W_c. The aggregation processing can be presented mathematically as below:

$$\eta \cdot sign(\sum_{i=i}^{K} \frac{N_i}{N} \cdot (\frac{e}{2})^{-(c-ts_i)} \cdot sign(W_c^i)) \rightarrow W_c \tag{9}$$

where c denotes the current round, whereas ts_i denotes the round of i-th client's previous participation. The logarithm base e is used to represent the time effect. And η is the hyperparameter. The data size weights of participating clients are calculated using $\frac{N_i}{N}$, which represents the ratio of the sample size of a particular client, denoted as N_i, to the total sample size N. Clients with larger sample sizes are assigned higher weights. $(\frac{e}{2})^{-(c-ts_i)}$ will determine the time weights, the clients that are newer participated in the federated training will be set with larger weights. Finally, $sign(W_c^i)$ will help the global model update towards the positive or negative base on the update direction of most of the participating clients. The $sign$ function's definition is as follows:

$$sign(x) = \begin{cases} 1, x > 0 \\ 0, x = 0 \\ -1, x < 0 \end{cases} \tag{10}$$

IV. Experiments

A. Dataset

The study employs a wind turbine dataset from 20 turbines across 2 different farms. One is in Yanling, Henan province and the other is in Hongshimao, Shaanxi province around 600 kilometres of distance. The dataset contains raw data gathered with a 30-second granularity from two different wind farms for around 15 days from 11st, Feb 2019 to 26th, Feb 2019. In Yanling, there have been two days of icing status and in Hongshimao, there is only one day of icing status, which makes the dataset imbalanced and the imbalance ratio

TABLE I
SPECIFICATION OF SELECTED PARAMETERS

No.	Parameter	No.	Parameter
1	Generator speed	18	Torque of generator
2	Speed of pithc1	19	Temperature of shaft cabinet1
3	Speed of pithc2	20	Temperature of shaft cabinet2
4	Angle of pitch1	21	Temperature of shaft cabinet3
5	Angle of pitch2	22	Temperature of stator of generator U1
6	Angle of pitch3	23	Temperature of stator of generator U2
7	Yawing angle	24	Temperature of stator of generator V1
8	Temperature of hub1	25	Temperature of stator of generator V2
9	Temperature of hub2	26	Temperature of stator of generator W1
10	Temperature of hub3	27	Temperature of stator of generator W2
11	Power performance1	28	60s mean of wind speed
12	Power performance2	29	Current1
13	Power performance3	30	Current1
14	Temperature of pitch motor1	31	Current1
15	Temperature of pitch motor2	32	Temperature of cooling pad1
16	Temperature of pitch motor3	33	Temperature of cooling pad2
17	Active power	34	Temperature of cooling pad3

of data collected from the wind farm in Yanling is around 7:1 and around 15:1 in Hongshimao. Before labeling, all raw data underwent consultation with experienced engineers. The sensor data columns consist of approximately 300 variables, falling into three categories: string, floating-point, and boolean. Table I shows the final selected columns. The training set and test set were partitioned in a ratio of 7:3. The 70% of data in each client is utilized for the local model training. The left 30% of data is combined into a big testing dataset for the performance evaluation of the learned global model.

B. Experiment Settings

In the data analysis process, the size of the time window TW is 128 and the stride for over-sampling the minority class is 70. During segmentation, the time window slides three times to over-sample the data of the minor class. 20 clients are involved with an initial participation rate of 0.5, implying that for each communication round the server randomly selects 10 clients to participate in the federated training. Each client loads the data of a single turbine. The optimization algorithm used is Adam with a learning rate of 0.01. During each experiment, a server went through 100 communication iterations to generate an optimal global model. The learning rate ϵ in the local client update is set to 0.01. The hyperparameter η in model aggregation is set to 0.04. As for KD, the hyperparameter α is set to 0.9 and the temperature T is set to 20. The details of the experimental settings can be found in TABLE II. All methods are repeated five times with different random seeds.

C. Evaluation Metrics

The proposed methods are evaluated using two metrics: F_β Score and Balanced Accuracy (BA). The choice of these metrics is motivated by the imbalanced nature of the testing dataset used in the research, which has an imbalance ratio of 8:1, and the focus on the sensitivity of icing prediction. The accuracy metric alone may not be effective in evaluating the performance of the proposed model, as predicting all test samples as normal can still result in high accuracy due to the class imbalance. Therefore, Balanced Accuracy (BA) is used as the evaluation metric. The equations are listed below:

TABLE II
EXPERIMENTAL SETTINGS

Description	Setting or value
Time window size	128
Stride for over-sampling the minority class	70
Number of LSTM layer's hidden nodes	34
p in Dropout layer	0.5
Convolutional layers' filter sizes	128,256,128
Convolutional layers' kernel sizes	9,5,3
Number of communication rounds	100
Number of clients	20
Participation rate	0.5
Learning rate of optimizer Adam	0.01
Learning rate in local client update	0.01
η in model aggregation	0.04
α in KD	0.9
Temperature T in KD	20

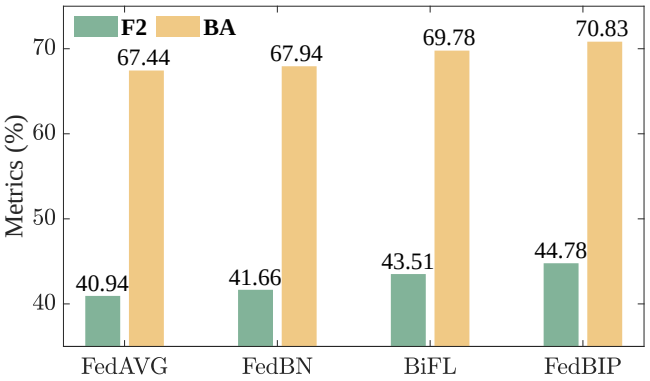

Fig. 4. Comparison of state-of-the-art FL methods on over-sampled dataset

$$F_\beta = (1+\beta^2)\frac{Precision \cdot Recall}{\beta^2 \times Precision + Recall} \tag{11}$$

$$Precision = \frac{TP}{TP+FP} \quad Recall = \frac{TP}{TP+FN} \tag{12}$$

where TP, FP, TN and FN represent True Positive, False Positive, True Negative and False Negative. β here is set to 2 and then the F_β is called as F_2.

D. Comparison with state-of-the-art FL methods

The following state-of-the-art FL methods are utilized for the evaluation of our proposed FedBIP. The details of these models are as follows:

- **Federated averaging (FedAVG)**: FedAVG aggregates the parameters of all the participating models through averaging based on the loaded sample size of each client [13].
- **Blade Icing Federated Learning (BiFL)**: BiFL proposed a heterogeneous FL model for blade icing detection. In this model, the exchange between clients and server is the encoded feature map rather than gradient [12].
- **Federated Learning Batch Normalization (FedBN)**: FedBN is proposed to overcome the feature shift between clients. In this work, batch normalization is integrated into FL to alleviate the feature shift [35].

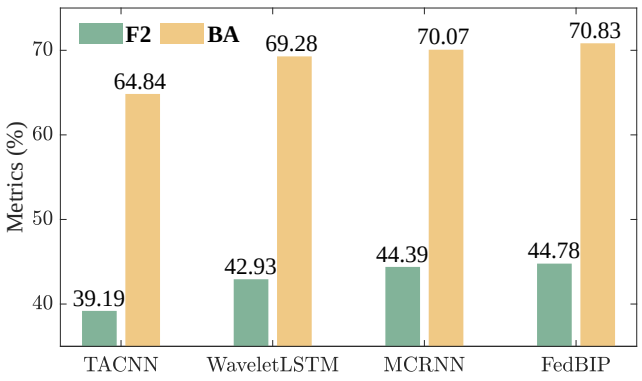

Fig. 5. Performance comparison with centralized learning methods

All methods are performed on the over-sampled dataset processed by our processed segmentation-based sample balancing method. All methods were tested using the default settings, with the exception of the use of our proposed local model in FedBIP.

Our proposed method, FedBIP, outperforms other methods in terms of both F_2 and BA measures, as shown in Fig. 4. When compared to BiFL, which ranks second in performance, FedBIP shows a 2.92% improvement in F_2 and a 1.50% improvement in BA. In comparison to FedBN, FedBIP demonstrates a 7.49% improvement in F_2 and a 4.25% improvement in BA. Additionally, FedBIP outperforms FedAVG, which showed the worst performance, with a 9.38% improvement in F_2 and a 5.03% improvement in BA.

BiFL addresses class imbalance through prototype learning, which is heavily dependent on the local model's ability to learn. Additionally, the global model learned through BiFL is trained on the learned feature map, which means that the imbalance ratio may vary among clients and may not be well-handled by BiFL. FedBN was designed to address feature shift rather than class imbalance, and FedAVG simply aggregates model parameters based on sample size, which also lacks the ability to effectively address class imbalance. This may be due to the fact that the data remains class imbalanced even after balancing efforts. In contrast, our model performs better because it incorporates a class balancing method, knowledge distillation, and a new aggregation method.

E. Comparison with state-of-the-art centralized method

The following centralized methods are included for evaluating our proposed FedBIP. The details of these models are as follows:

- **MCRNN**: This method utilized discrete wavelet decomposition to extract multilevel features from both the time and frequency domains. It also proposed a parallel structure that combines an LSTM branch and a CNN branch for feature extraction [16].
- **TACNN**: This model integrated a convolutional neural network with a temporal attention module, enabling the identification of discriminative features from raw sensor data [18].
- **WaveletLSTM**: This method integrated wavelet-based multiscale learning into the conventional LSTM structure, enabling simultaneous learning of global and local temporal features from multivariate SCADA signals [19].

Since these models are trained in a centralized way, we combine all the data on the local client to get one training dataset for these methods, and the results are shown in Fig. 5.

According to the results, FedBIP has shown significant improvements compared to other models. When compared to MCRNN, which ranked second in performance, FedBIP demonstrated an improvement of 0.88% in F_2 and 1.08% in BA. Compared to WaveletLSTM, FedBIP displayed a 4.31% improvement in F_2 and a 2.24% improvement in BA. In comparison with TACNN, which demonstrated the worst performance, FedBIP showed an impressive 14.26% improvement in F_2 and a 9.24% improvement in BA.

Although MCRNN and WaveletLSTM employed wavelet decomposition for feature extraction from both the time and frequency domains, they do not tackle the issue of imbalance while training the model. TACNN incorporates a temporal module to learn temporal information from data, but its performance may not be satisfactory when the data is imbalanced, and the imbalance ratio varies for different wind turbines.

F. Comparison of class imbalance processing method

To demonstrate the effectiveness of our proposed method for addressing the class imbalance, we compare it with two commonly used imbalance learning methods. The descriptions are as follows:

- **Imbalance**: In this study, the model was evaluated on raw, class-imbalanced data without utilizing any methods specifically designed to address the class-imbalance issue.
- **Focal (focal loss)**: Focal loss is a well-known method for addressing the class imbalance in machine learning. In our study, we replaced the cross entropy loss function in each client with focal loss to evaluate its effectiveness.
- **SMOTE**: SMOTE is an over-sample method that is based on each minority class sample and creates synthetic samples along the line segments joining any/all of the k nearest neighbors from the minority class [36]. In this scenario, k is set to 5.

Model performance comparisons are shown in Fig. 6. The results show that all three methods (SMOTE, Focal loss, and FedBIP) improve upon the model trained on original, imbalanced data. SMOTE generates synthetic samples to balance the number of samples in each class, which may not accurately represent the real icing condition and could affect the authenticity of the experiment. Focal loss demonstrates some effectiveness for addressing class imbalance, but it is heavily dependent on the selection of hyperparameters. Additionally, there is no guarantee that every client will be able to effectively synchronize to address the imbalance using Focal loss. In contrast, FedBIP utilizes the existing icing data and achieves higher performance without requiring oversampling of the minority class. Oversampling the minority class too much may lead to overfitting to the icing data, which can have negative effects on training.

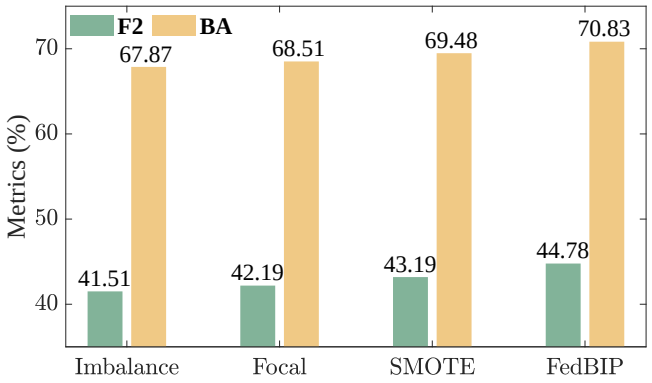

Fig. 6. Performance comparison of class imbalance processing methods

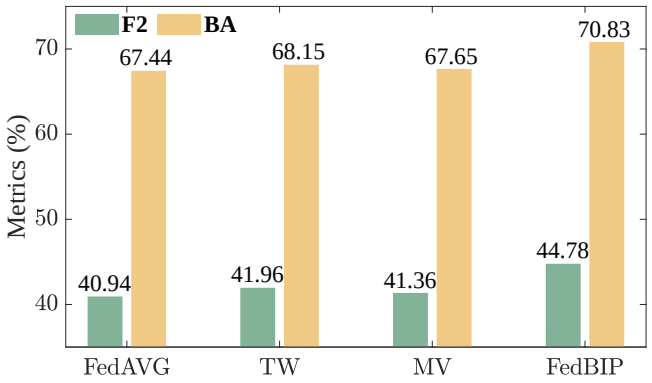

Fig. 7. Comparison of model aggregation method

G. Comparison of model aggregation methods

For the evaluation of the proposed model aggregation mentioned above, we compare it with other aggregation methods, including:

- **Federated averaging (FedAVG)**: FedAVG aggregates the parameters of all the participating models through averaging based on the loaded sample size of each client [13].
- **Temporally weighted aggregation asynchronous (ASTW)**: ASTW is modified based on FedAVG and takes timestamp into consideration. Since our built model does not have that many layers, so during the comparison experiment, we only take temporally weighted (TW) module into account. The code is programmed base on the equations given in the paper [34].
- **Majority vote (MV)**: MV aggregates the model parameters base on the positive or negative values of most of the participating clients [37].

The results are shown in Fig. 7. According to the results, our aggregation method gets the best performance compared with the other aggregation algorithms. Compared with MV, which ranks second place, our method gains an improvement of 1.88% on F_2 and 1.09% on BA. Compared with TW, our method gains an improvement of 3.27% on F_2 and 2.4% on BA. As for the FedAVG, which gets the worst performance, our model gains an improvement of 4.28% on F_2 and 2.68% on BA.

Each of these aggregation methods has its own strengths, but they all only consider a single aspect of the problem. FedAVG focuses on sample size, ASTW only considers the temporal order of model aggregation, and MV is only concerned with reducing the amount of data transmitted during aggregation. In contrast, our aggregation method takes all of these factors into account.

H. Comparison of the local feature extractor

To evaluate the designed local feature extractor's performance, we select seven baseline methods to conduct the comparison experiment, including 1) **BiLSTM**: a bidirectional LSTM implemented by adding a backward LSTM. The hidden nodes is set to 128; 2) **CNN_LSTM**: a CNN layer followed by one LSTM layer. The filter size of both CNN and LSTM is set to 128; 3) **SSENET**: a strong baseline for classifying multivariate time series data. The model is built with dense connections. During the comparison experiment, the attention mechanism is removed since the comparison is to testify model structure's performance [38]; 4) **FCN**: a fully convergent network in time series classification [39]. The filter sizes of each layer are set to $\{128, 256, 128\}$; 5) **Gated Recurrent Unit (GRU)**: a light-weight variant of LSTM, but its performance is not inferior to LSTM. Hidden nodes are set to 128; 6) **LSTM**: commonly used networks in time series data classification, hidden nodes are set to 128; 7) **MultiLayer Perceptron (MLP)**: a three-layer model with a dropout layer added between every two consecutive layers. Each MLP layer has 128 hidden nodes. During this comparison, seven different models were tested as a replacement for the local model that we designed. The result is shown in TABLE III.

The results show that FedBIP outperform other models on the oversampled dataset, but is slightly outperformed by FCN on the original, imbalanced dataset. On the oversampled dataset, FedBIP shows a 1.91% improvement in F_2 and a 0.74% improvement in BA compared to the best baseline, CNN_LSTM. On the imbalanced dataset, FCN performs similarly to FedBIP in terms of F_2, but shows a slight absolute improvement of 0.04% in BA. Among the models tested, SSENET, which has a complex model structure and requires a longer training time, performs the worst on the original imbalanced dataset and not so well on the over-sampled dataset. This suggests that when solving real-world problems, it is important to choose a model structure that is appropriate for the task, rather than simply opting for a more complex model. Complex models may not necessarily lead to better performance and can require more computing resources and longer training times to reach optimal performance.

I. Ablation and sensitivity analysis

To illustrate the importance and study the key parameters in the FedBIP, the ablation and sensitivity analysis are performed in this section.

1) Influence of KD: To evaluate the effectiveness of the KD module, we compare the models' results with/without KD. First, a model that gets an overall optimum performance is trained on the imbalanced and over-sampled dataset from

TABLE III
PERFORMANCE COMPARISON OF LOCAL MODEL

	Imbalanced dataset		Over-sampled dataset	
	F_2(%)	BA(%)	F_2(%)	BA(%)
BILSTM	43.27±0.25	69.55±0.25	43.34±0.41	69.57±0.37
CNN_LSTM	43.08±0.22	69.41±0.22	43.94±0.55	70.31±0.45
SSENET	42.21±0.48	68.54±0.52	43.48±1.21	69.79±1.19
FCN	**44.00±0.48**	**70.32±0.47**	43.13±0.37	69.44±0.29
GRU	42.36±0.63	68.64±0.67	40.08±0.46	66.37±0.48
LSTM	42.87±0.67	69.15±0.62	43.32±0.22	69.62±0.20
MLP	43.59±0.16	69.86±0.15	42.94±0.34	69.26±0.34
FedBIP	43.99±0.22	70.28±0.15	**44.78±0.58**	**70.83±0.44**

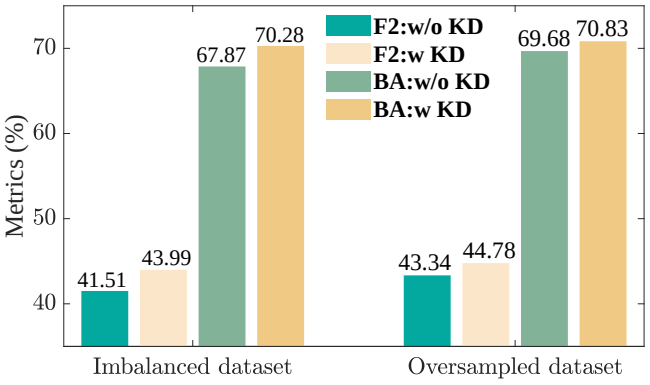

Fig. 8. Ablation study of KD module

scratch. Then, this model is used as the teacher model to help the other models' training. The results are shown in Fig. 8.

According to Fig. 8, it can be seen that the introduction of knowledge distillation (KD) during training consistently improves model performance on both imbalanced and over-sampled datasets. Specifically, on the imbalanced dataset, there are improvements of 5.97% in F_2 and 3.55% in BA, while on the oversampled dataset, there are improvements of 3.32% in F_2 and 1.65% in BA. These results demonstrate the effectiveness of KD in improving model performance for both F_2 and BA measures on both types of datasets.

In addition to evaluating model performance, we also compare the number of communication rounds required to train the global model until it reaches a target balanced accuracy (BA) of 63%. This allows us to assess the impact of the KD procedure on convergence speed. To eliminate the randomness introduced by the Adam optimizer, we run the experiment multiple times and calculate the average number of communication rounds for each model with and without the KD module. TABLE IV lists the results.

Looking over the table, all of the models require fewer communication rounds to reach the target BA when KD is used. Specifically, LSTM shows the greatest improvement, requiring 71.21% fewer communication rounds with KD activated. SSENET also shows a significant reduction in communication rounds, with a 42.27% decrease. While FedBIP only shows a 5.16% reduction in communication rounds, KD still helps to speed up the training process. Overall, the use of KD leads to a significant decrease in the number of communication rounds required to reach the target BA for all models.

TABLE IV
THE AVERAGE NUMBER OF COMMUNICATION ROUNDS TO REACH THE TARGET BA OF 10 TIMES WITH OR WITHOUT KD ON THE OVER-SAMPLED DATASET (THE FEWER COMMUNICATION ROUNDS, THE FEWER COMMUNICATION COSTS)

	w/o KD	w/ KD
BILSTM	48.2±28.7	32.0±12.6
CNN_LSTM	41.8±24.0	26.6±12.9
SSENET	63.4±20.6	36.6±21.5
FCN	50.6±22.0	47.5±23.1
GRU	54.0±33.5	48.5±28.0
LSTM	64.6±17.6	18.6±9.98
MLP	51.1±13.0	45.3±25.3
FedBIP	**31.0±12.4**	**29.4±22.9**

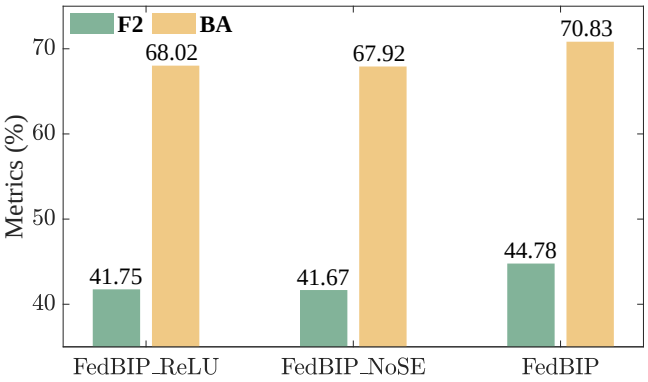

Fig. 9. Ablation analysis of local model on over-sampled dataset

2) Influence of the modules in local model: We undergo the ablation experiments to verify the effectiveness of the proposed model by comparing it with the following architectures: 1) FedBIP_ReLU: where ReLU is used as the activation function; 2) FedBIP_noSE: where SE blocks are removed from the model; The results are shown in Fig. 9.

According to the figure, when SE blocks are equipped , the model performance increased by 7.46% of F_2 and 4.28% of BA. And when changing the activation function from ReLU into SiLU, the performance rose around 7.26% of F_2 and 4.13% of BA.

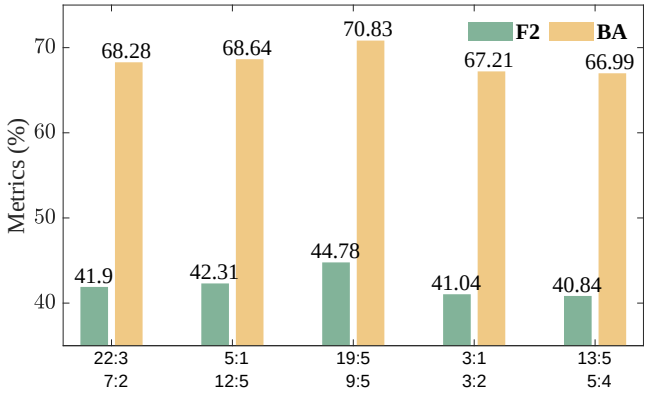

Fig. 10. Comparison on over-sampled dataset with different imbalance ratios (The first row is the imbalance ratio of data of Hongshimao, and the second row is the imbalance ratio of data of Yanling)

3) Impact of data balancing with different imbalance ratios: To investigate the effect of the over-sampling method sliding window, we conduct experiments on the data with different imbalance ratios by generating a different number of samples with the sliding window. There is the same number of icing samples as the number of icing samples of the imbalanced dataset generated each time. And the results are shown in Fig. 10.

According to the results, the performance gradually rises when the data tend to be more balanced. When the imbalance ratios are 19:5 on the Hongshimao dataset and 9:5 on the Yanling dataset, FedBIP reaches the highest performance. However, with more icing data generated, the performance starts to fall. This is mainly because the over-sampling technique mostly depends on the minority data. Moreover, when additional icing samples are generated based on the original icing samples, the model may exhibit overfitting towards these icing data, which can negatively impact the model's performance in predicting icing conditions.

V. Conclusion

This paper presents an FL structure FedBIP for predicting wind turbine blade icing. It comprises a feature selection approach enhanced with human knowledge, a segmentation-based class imbalance method, and a new aggregation method. And it allows individual clients to train their local models using their own data, which are then uploaded to the server without sharing any data. The server aggregates all the participating clients' models with weighted formulations to generate the final g lobal m odel. W hat's m ore, k nowledge distillation helps improve our proposed model's performance. To assess the effectiveness of our methods, we conduct our experiments through turbine data from 20 turbines on 2 different farms and compared our proposed model with others. The results demonstrate that our proposed model achieves better performance.

For our future work, there are several directions. First, we will try to address the imbalance problem on the algorithm level, while in this paper we focused on the data level. Furthermore, the feature selection method employed in this study prioritizes the correlation of individual columns to the class. However, hidden correlations might exist within these data columns. Thus, another of our future works is to develop an algorithm for finding c onnections a mong d ata columns.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (62020106004, 62272342, 92048301).

References

[1] P. Sadorsky, "Wind energy for sustainable development: Driving factors and future outlook," *Journal of Cleaner Production*, vol. 289, p. 125779, 2021.

[2] K. Wei, Y. Yang, H. Zuo, and D. Zhong, "A review on ice detection technology and ice elimination technology for wind turbine," *Wind Energy*, vol. 23, no. 3, pp. 433–457, 2020.

[3] P. Wang, W. Zhou, Y. Bao, and H. Li, "Ice monitoring of a full-scale wind turbine blade using ultrasonic guided waves under varying temperature conditions," *Structural control and health monitoring*, vol. 25, no. 4, p. e2138, 2018.

[4] B. Xu, F. Lu, S. J. Dyke, and X. Guo, "Icing monitoring for a wind turbine model blade with active pzt technology," in *Earth and Space 2014*, 2014, pp. 703–710.

[5] X. Bin, G. Xueyi, and C. Hongbing, "Active icing monitoring for wind turbine blade models with pzt technology," *Piezoelectrics & Acoustooptics*, vol. 39, p. 02, 2017.

[6] X. Zhao and J. L. Rose, "Ultrasonic guided wave tomography for ice detection," *Ultrasonics*, vol. 67, pp. 212–219, 2016.

[7] S. V. Venna, Y.-J. Lin, and G. Botura, "Piezoelectric transducer actuated leading edge de-icing with simultaneous shear and impulse forces," *Journal of Aircraft*, vol. 44, no. 2, pp. 509–515, 2007.

[8] R.-Z. Szasz, M. Ronnfors, and J. Revstedt, "Influence of ice accretion on the noise generated by an airfoil section," *International Journal of Heat and Fluid Flow*, vol. 62, pp. 83–92, 2016.

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

[10] M. C. Pedersen and C. Yin, "Preliminary modelling study of ice accretion on wind turbines," *Energy Procedia*, vol. 61, pp. 258–261, 2014.

[11] L. Shu, G. Qiu, Q. Hu, X. Jiang, G. McClure, and H. Yang, "Numerical and field experimental investigation of wind turbine dynamic de-icing process," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 175, pp. 90–99, 2018.

[12] X. Cheng, F. Shi, Y. Liu, J. Zhou, X. Liu, and L. Huang, "A class-imbalanced heterogeneous federated learning model for detecting icing on wind turbine blades," *IEEE Transactions on Industrial Informatics*, 2022.

[13] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

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

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

[16] W. Tian, X. Cheng, G. Li, F. Shi, S. Chen, and H. Zhang, "A multilevel convolutional recurrent neural network for blade icing detection of wind turbine," *IEEE Sensors Journal*, vol. 21, no. 18, pp. 20 311–20 323, 2021.

[17] R. Tong, P. Li, X. Lang, J. Liang, and M. Cao, "A novel adaptive weighted kernel extreme learning machine algorithm and its application in wind turbine blade icing fault detection," *Measurement*, vol. 185, p. 110009, 2021.

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

[19] X. Wang, Z. Zheng, G. Jiang, Q. He, and P. Xie, "Detecting wind turbine blade icing with a multiscale long short-term memory network," *Energies*, vol. 15, no. 8, p. 2864, 2022.

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

[21] L. Swenson, L. Gao, J. Hong, and L. Shen, "An efficacious model for predicting icing-induced energy loss for wind turbines," *Applied Energy*, vol. 305, p. 117809, 2022.

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

[23] X. Cheng, C. Li, and X. Liu, "A review of federated learning in energy systems," *2022 IEEE/IAS Industrial and Commercial Power System Asia (I&CPS Asia)*, pp. 2089–2095, 2022.

[24] X. Cheng, W. Tian, F. Shi, M. Zhao, S. Chen, and H. Wang, "A blockchain-empowered cluster-based federated learning model for blade icing estimation on iot-enabled wind turbine," *IEEE Transactions on Industrial Informatics*, 2022.

[25] X. Cheng, F. Shi, Y. Liu, X. Liu, and L. Huang, "Imbalanced federated learning model for blade icing detection of wind turbines," *Energy*, p. 124441, 2022.

[26] G. Jiang, W. Fan, W. Li, L. Wang, Q. He, P. Xie, and X. Li, "Deepfedwt: A federated deep learning framework for fault detection of wind turbines," *Measurement*, vol. 199, p. 111529, 2022.

[27] X. Ma, C. Wen, and T. Wen, "An asynchronous and real-time update paradigm of federated learning for fault diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8531–8540, 2021.

[28] Z. Zhang, C. Guan, H. Chen, X. Yang, W. Gong, and A. Yang, "Adaptive privacy-preserving federated learning for fault diagnosis in internet of ships," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 6844–6854, 2021.

[29] S. Lu, Z. Gao, Q. Xu, C. Jiang, A. Zhang, and X. Wang, "Class-imbalance privacy-preserving federated learning for decentralized fault diagnosis with biometric authentication," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 9101–9111, 2022.

[30] L. Wang, S. Xu, X. Wang, and Q. Zhu, "Towards class imbalance in federated learning," *arXiv e-prints*, pp. arXiv–2008, 2020.

[31] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[32] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4133–4141.

[33] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature communications*, vol. 13, no. 1, pp. 1–8, 2022.

[34] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, pp. 4229–4238, 2020.

[35] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," *arXiv preprint arXiv:2102.07623*, 2021.

[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[37] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signsgd: Compressed optimisation for non-convex problems," in *International Conference on Machine Learning*. PMLR, 2018, pp. 560–569.

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

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