A Blockchain-Empowered Cluster-based Federated Learning Model for Blade Icing Estimation on IoT-enabled Wind Turbine

Xu Cheng, *Member, IEEE,* Weiwei Tian, *Student Member, IEEE,* Fan Shi, *Member, IEEE,* Meng Zhao, Shengyong Chen, *Senior Member, IEEE* and Hao Wang, *Senior Member, IEEE,*

***Abstract*—Wind energy is a fast-growing renewable energy but faces the blade icing. Data-driven methods provide talented solutions for blade icing detection but a considerable amount of IoT data need to be collected to a central server, which may lead to the leakage of sensitive business data. To address this limitation, this work proposes *BLADE*, a Blockchain-empowered imbalanced federated learning (FL) model for blade icing detection. With the help of Blockchain, the conventional FL is improved without worrying about the failure of the single centralized server and boosts the privacy-preserving. A validation mechanism is introduced into the Blockchain to enhance the defense of poisoning attacks. In addition, a novel imbalanced learning algorithm is integrated into BLADE to solve the class-imbalance problem in the sensor data. The BLADE is evaluated on the 10 wind turbines from two wind farms. The experimental results verify the effectiveness, superiority, and feasibility of the proposed BLADE.**

***Index Terms*—IoT, Blockchain, Federated learning, Blade icing detection, Imbalance learning**

I. Introduction

Wind energy has become a promising and fast-growing renewable energy because of the ample availability and technological maturity of wind turbines [1]. However, the performance degradation caused by blade-icing represents a critical shortcoming to wind turbines, with up to 30% loss in annual power generation in severe cases [2].

Till now, lots of effort has been made to reduce the loss caused by blade icing. The passive method for anti-blade-icing usually uses the special coating [3], but coating alone is insufficient to prevent icing. Active method is thus proposed with the help of some heating equipment [4]. A precondition for passive and active methods is timely and accurate icing detection. Conventionally, icing detection can be conducted according to the physical properties of ice [5] or machine behavior affected by the accretion of ice [6]. Besides, mathematical method is also developed using mathematical or numerical models for blade icing predicting. The above methods can achieve good performance but are limited by the cost or dependent on domain knowledge [7].

In recent years, data-driven methods, especially end-to-end solutions based on deep neural networks, have attracted extensive attention in academia and industry. While training deep learning (DL) models commonly does not rely on prior domain knowledge, it requires a considerable number of data samples. However, a huge amount of data is available thanks to the Internet of Things (IoT) technologies, which continuously stream data from sensors, such as power, temperature, etc.. Combining wind turbines monitoring IoT data of multiple plants for DL model training contributes to performance improvement of the detection model because more spatio-temporal correlation information in time series data can be reflected, thus can encourage more wind farms to actively engage in collaborative detecting services [8].

But there are two key issues for training deep learning models using big data of IoT. Firstly, wind farm owners are reluctant in sharing raw data for privacy and commercial concerns. Some key parameters in monitoring data are valuable for keeping wind turbines operating properly, which is beneficial for wind farms to keep competitive in the market. This may discourage multi-parties from sharing the data. Secondly, conventional machine learning models are generally centralized, so as the data. Collecting multi-parties' data to a central place not only threatens data privacy, but also causes models to become more complex and hard to be trained as datasets generated by machines equipped with IoT technologies grow much larger. So the efficiency of machine learning methods should be further enhanced. Fortunately, federated learning (FL) provides a distributed solution to learn a collaborative machine learning (ML) model without the need of sharing the raw data and only the intermediate gradients of the learned model are uploaded to a central server so as to preserve privacy [9].

Despite these superiorities, employing FL for icing detection still faces some critical challenges: First, the FL is heavily dependent on the robustness of the single centralized server and the single centralized server is assumed to be trustworthy. The failure of the server would cause the entire FL network to collapse. The server is responsible to make decision on the fair participants selection and model aggregation. Nevertheless, a

Xu Cheng, Weiwei Tian, Fan Shi, Meng Zhao, and Shengyong Chen are with Engineering Research Center of Learning-Based Intelligent System (Ministry of Education), the Key Laboratory of Computer Vision and System (Ministry of Education), the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, 300384.

Xu Cheng is also with the Department of Manufacturing and Civil Engineering, Norwegian University of Science and Technology, Gjøvik, 2815, Norway.

Hao Wang is with the Research center for Optical Fibre Sensing of Zhejiang Lab China, and also with the Department of Computer Science, Norwegian University of Science and Technology, Gjøvik, 2815, Norway.

Xu Cheng and Weiwei Tian are equal contribution. Fan Shi and Hao Wang are the corresponding authors. e-mail:shifan@email.tjut.edu.cn; hawa@ntnu.no.

biased server would appear inevitable, thereby hurting the performance of the learned model. Second, the FL is easily influenced by malicious attacks. If malicious clients upload toxic models to the server, the performance of the learned model would decrease. More seriously, the server, which has the access of all clients' information, might be dishonest. In this case, the private information leakage will be more severe. Third, the wind turbine works in the normal status most of the time and blade icing happens in certain periods, such as low temperature. This results in a highly skewed class distribution between icing samples and normal samples. The size of the icing samples is much smaller than the size of normal samples, which poses a great challenge for constructing high-performance detection models. Skewed class distributions can severely affect the performance of classifiers because classifiers tend to be swamped by the majority of classes and ignore the minority.

To address the above limitations of FL, Blockchain which is a secure technology and can tolerate the failure of the single centralized server, is introduced into the FL in this work. To overcome the imbalance of sensor data, a cluster-based imbalance learning method is presented. To summarize, the contributions of this paper are:

1) We propose a **BL**ockchain-empowered imb**A**lanced federate**D** l**E**arning model (BLADE) for wind turbine blade icing detection and overcome the issues raised by centralized server in conventional FL network. A novel cluster-based method for addressing the data imbalance is integrated into the proposed model. Compared with conventional FL, BLADE can enhance the data privacy and model updates within a trust Blockchain network. Moreover, BLADE can be easily applied to the sensor data with imbalanced distribution.
2) The proposed BLADE is evaluated comprehensively on 10 wind turbines from two wind farms. The experimental performance comparisons of global models, imbalance learning capability, defense the poisoning attacks, indicate that BLADE exhibits superiority over others in blade icing detection. In addition, sensitivity and ablation studies show the effectiveness of each component of the proposed model.

The rest of the paper is structured as follows. Section II gives an overview of the literature on wind turbine blade icing detection and federated learning. Section III describes the proposed model BLADE for detecting blade icing of wind turbines. Section IV evaluates the proposed model through comprehensive experiments. Section V concludes the paper and presents future work.

II. Related work

A. Blade icing detection

Blade icing detection of wind turbines can be mainly divided into direct and indirect methods. The direct approaches are usually conventional methods, which monitor the ice accumulation through additional devices. Shajiee et al. developed an optical sensing method for direct icing detection on wind turbine blades, combining with distributed resistive heating for ice mitigation [10]. Wang et al. leveraged ultrasonic guided waves for ice monitoring as the propagation features of the elastic waves can be altered by ice acceleration [5]. By contrast, indirect methods detect icing events through operational data of wind turbines instead of employing extra devices, which are more cost-effective and easily be maintained [11]. Indirect approaches include model- and data-driven methods. Model-driven methods establish mathematical or numerical models based on certain assumptions. In [12], a model-based method was proposed for blade icing diagnosis by detecting the change of the rotor angular speed. However, these methods are dependent heavily on human or domain knowledge. Instead, data-driven approaches, especially deep learning models, can bypass this problem and map the correlations between operational signals to obtain competitive performances of detecting icing events. Tian et al. integrated the discrete wavelet decomposition into a special-designed multilevel convolutional recurrent neural network for blade icing detection [13]. Cheng et al. combined CNN with a temporal attention module to automatically determine the important sensors and extract discriminative information from Supervisory Control and Data Acquisition (SCADA) data [14]. They also proposed an enhanced version for blade icing detection by using the learning strategy of semi-supervised learning [7]. In recent years, extensive data of multiple wind farms are available due to the rapid development of IoT. However, these centralized models are insufficient to take the advantage of these big data due to commercial and privacy reasons, as well as the limitation of the models themselves. In this paper, we propose a decentralized privacy-preserving model by employing the FL framework which can be trained at different physical sites to fully utilize the data of multiple plants.

B. Federated learning

Privacy issue of data used for analysis is increasingly concerned with the development of IoT [15]. More efficient and intelligent methods need to be investigated for data privacy-protecting. In recent studies, FL has been proved to be an effective method [16]–[18]. Saputra et al. proposed a federated energy demand predicting method for electric vehicle networks, in which heavy data privacy issues and communication overhead can be mitigated [19]. Zhang et al. introduced a probabilistic prediction method for solar irradiation, leveraging FL for data privacy and security issues [20]. These studies show the importance and validity of FL for data privacy protection. However, an equally important issue, the security of models in the distributed framework, is not considered. Some studies leveraged Blockchain technology to protect models from attacks. Lu et al. presented a federated privacy-preserved data sharing scheme in industrial IoT that integrates blockchain technology to achieve model security and reliability [21]. In addition, a hybrid blockchain architecture was developed to enhance the security of the federated model for data sharing on the Internet of Vehicles [22]. Zhang et al. presented a Blockchain-driven federated learning model for the failure detection based on IoT data [23]. Chen et

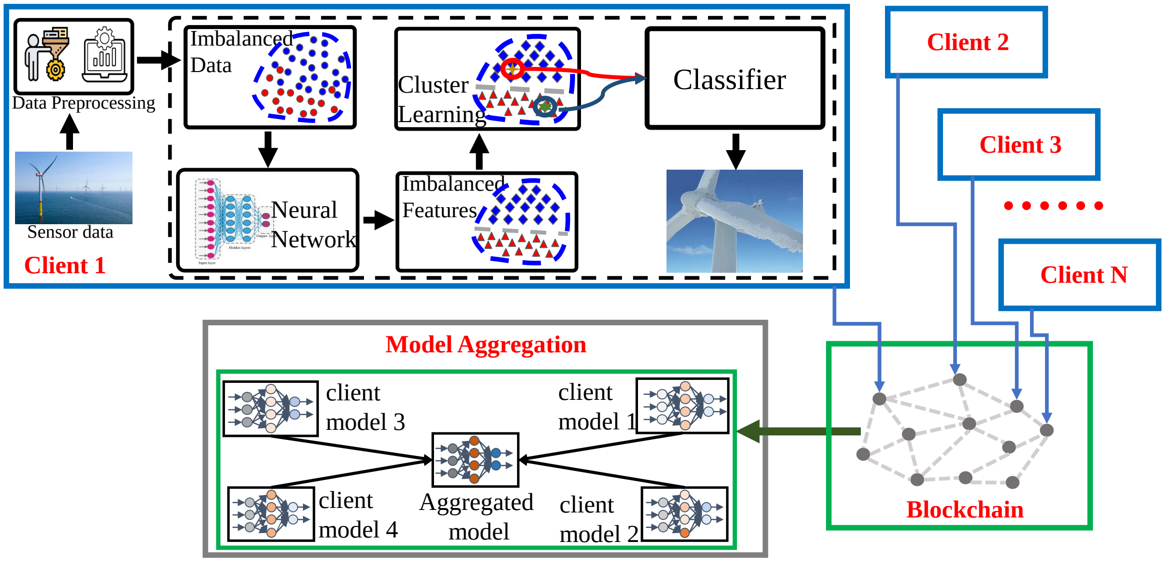


Fig. 1: Architecture of blockchain-based federated learning systems for blade icing detection of wind turbine using IoT.

al. introduced a Blockchain-based federated learning method by integrating a novel validation mechanism and proof-of-stake inspired consensus [24]. However, to the author's best knowledge, there are few works by applying federated learning to blade icing detection. Furthermore, there is still room for the improvement of the security issues in applying FL to blade icing detection.

III. Blockchain-empowered Cluster-based Federated Learning

A. System overview

Blockchain can be considered as a growing list of blocks, and different blocks are linked using cryptographic techniques [25], [26]. Blockchain can enhance the FL from the following technical aspects: 1) Blockchain offers a completely decentralized place for round delineation, client selection and model aggregation. By using the distributed consensus mechanism, the Blockchain can enhance the trust of FL. 2) Blockchain provides a peer-to-peer network which can improve the fault tolerance of the single server in FL. And this could enhance the computing flexibility and integrity of FL. 3) Blockchain can incentivize the enthusiasm of participants by providing rewards to share the model with customized smart contracts.

Although there are lots of advantages of applying Blockchain to FL, there are still some challenges for using Blockchain-based FL to blade icing detection: First, how to overcome the sensor data imbalance. It is because wind turbines operate in a normal state most of the time, i.e., the blades are ice-free, and only a few times the blades are iced. Therefore, the data collected from the SCADA system itself are imbalanced in distribution, and models trained using imbalanced data will result in biased detection. Second, wind turbine is a very important asset both for wind farms and society, especially in the future low-carbon society. Therefore, how to further improve the safety and reliability of Blockchain-based FL for blade icing detection is significant. It is necessary to promote FL within only legitimate participants while excluding malicious participants [24]. Moreover, it is also necessary to provide a flexible strategy of consensus mechanisms so that users can choose according to their needs. Third, the distribution and the size of training data might be different, which results in heterogeneity between different clients. For example, the wind turbine works under different weather conditions and operate by different users. This will pose a great challenge to aggregating the obtained local models to obtain a global model for the blade icing detection.

The model *BLADE* is proposed with the aim of addressing the above challenges, as shown in Fig. 1. Different from other Blockchain systems, the Blockchain used in this work proposes the concept of a validation mechanism to enhance the security. There are three kinds of roles: worker $\mathcal{W}$, validator $\mathcal{V}$, and miner $\mathcal{M}$ rather than only two kinds of roles: worker $\mathcal{W}$ and miner $\mathcal{M}$ in conventional Blockchain [24]. In BLADE, we assume that there are $\mathcal{N}$ wind turbines with equal computing capability. The details of BLADE can be summarized as follows: **1) Role assignment**: Each wind turbine may have three different roles: worker $w_i \in \mathcal{W}$, validator $v_i \in \mathcal{V}$, and miner $m_i \in \mathcal{M}$, where $\mathcal{W} + \mathcal{V} + \mathcal{M} = \mathcal{N}$. The role assignment in BLADE could be random or by users. **2) Local imbalanced learning**: each worker w_i performs the imbalanced learning from the local dataset by using the proposed cluster-based method, as illustrated in the upper left of Fig. 1. **3) Validation**: Once the local learning is implemented, each worker w_i broadcasts the local model to the associated validator v_i. Validators will share their obtained model with other validators and examine the legitimacy of local model updates. **4) Mining**: The verified local model is then sent to miners, where the legitimacy block would be appended to its own Blockchain. And each miner requests its associated worker and validator to download the legitimacy block to their own Blockchain. Once the newly generated block is verified, the verified model in the block is immutable. Without any central server intervention, each client performs global aggregation to update its local model by using all shared models in the verification block.

B. Cluster-based federated learning

1) Neural network: Extracting informative features from the sensor data is a key task to the construction of blade icing detection model. Designing the model should not only consider the model size because there is a need to exchange the model gradient and larger size occupying more bandwidth, but also the learning capability of relevant features. In this work, the model of the feature extractor is implemented through the convolutional neural network (CNN). The structure of the local model is shown in Fig. 2, and there are three stacked CNN blocks. Each CNN block consists of a convolution layer (Conv1D), an attention layer (Attention), a batch normalized layer (BN), and the activation layer (RELU).

For the input of $X_{raw} \in \mathbb{R}^{m\times T}$, m and T are the input dimension and the window size of the samples, respectively, the output of each CNN block component can be represented by:

$$\begin{aligned} X_c &= Conv1D(X_{raw}) \\ X_{attn} &= Attention(X_c) \\ X_{out} &= RELU(BN(X_{attn})) \end{aligned} \tag{1}$$

where X_c, X_{attn} and X_{out} are the output of Conv1D layer, Attention layer, BN layer, and RELU layer, respectively. The values of these variables have the shape of $\mathbb{R}^{F\times T}$, where F is the number of filters in Conv1D layer.

The attention layer used in this paper is depicted in Fig. 3. As illustrated in Eq. (1), the input of the attention layer is the output of Conv1D layer, which has the shape of $\mathbb{R}^{F\times T}$, F stands for the number of filters, also called channels. In Conv1D layer, all channels have the same importance, and this leads to the loss of important information [27], [28]. Therefore, the attention mechanism is designed for the selection of important channels.

Let the output of Conv1D layer be $X_c = [x_1, x_2, ..., x_F]$, where $x_i \in \mathbb{R}^{T\times 1}$ denotes the i-th channel. For the identification of informative channels, we apply global average and max pooling to obtain the distinctive features, X_{ap} and X_{mp}, respectively (see Fig. 3). Both features after average and max pooling are then forwarded to a Conv1D layer (not shown in Fig. 3) to obtain the channel attention. Finally, the channel attention can be utilized for the calibration of channel features. The whole process for the attention layer can be illustrated mathematically as follows:

$$\begin{aligned} X_{mp} &= Conv1D(MaxPool(X_{raw})) \\ X_{ap} &= Conv1D(AvgPool(X_{raw})) \\ \alpha &= \sigma(X_{mp} + X_{ap}) \\ \widetilde{X} &= X_{raw} + \alpha \otimes X_{raw} \end{aligned} \tag{2}$$

where X_{raw} is the input of the attention layer, X_{mp} and X_{ap} are the re-weighted features of the output for average and max pooling layer, α is the calculated attention weights, σ is the sigmoid activation function, $\otimes$ is the element-wise multiple, $\widetilde{X}$ is the weighted features.

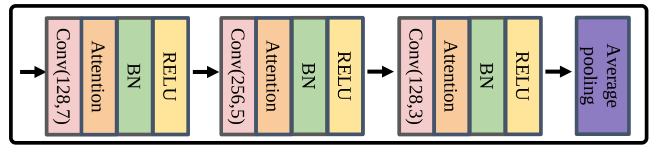


Fig. 2: Details of neural network in deep learning model.

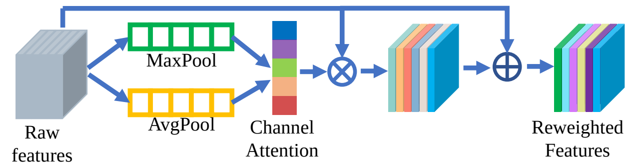


Fig. 3: Illustration of channel calibration attention module.

2) Cluster-based imbalanced learning: The imbalance in training data can worsen the performance of the trained models and especially affect the classification capability for minority classes. The common approaches address the imbalance problem at two levels: data level and algorithm level [16]. The data-level approaches generally need to process the raw data through the methods such as data resampling and data enhancement. These approaches can help to obtain balanced data by increasing or decreasing the samples of a certain class [29]. However, employing data-level approaches are inappropriate for the FL framework as extra information in the training phase of the model is needed, such as data distribution, which may lead to data leakage and violate data privacy [16]. In contrast, the algorithm level methods alleviate the data imbalance by modifying the algorithm, e.g. such as the neural network structure [30], [31], or adjusting the model parameters in the training stage. As there is no or little need to process the raw data, the algorithm-level methods are more suitable for the FL framework to protect data privacy.

In BLADE, we propose the cluster-based approach for imbalanced learning which follows the concept of algorithm-level methods, as shown in the training processing of Client 1 in Fig. 1. The imbalanced raw data leads to imbalanced features extracted by the neural network, so we first obtain a cluster for each class, which is then employed for the establishment of the classifier.

Given the learned features $\mathcal{H} = \{X^i\}_{i=1}^{C}$ where C is the number of classes, $X^i \in \mathbb{R}^{N_i\times F\times L}$ is a sample collection of class i, N_i is the sample number of class i. The class centroid $\mathbf{c}_i$ is simply calculated as:

$$\mathbf{c}_i = mean(H^i) = \frac{1}{N_i}\sum^{N_i} H^i, \quad \mathbf{c}_i \in \mathbb{R}^{F\times L} \tag{3}$$

Once the calculation of the centroids for each class is implemented, the probability of a given time series sample to one class can be computed. Suppose the learned feature of a time series is represented as $H_s \in \mathbb{R}^{F\times L}$, the probability belongs to class i is calculated according to the Euclidean distance:

$$p(y=i|H_s) = \frac{exp(-Eucli\_dist(H_s, \mathbf{c}_i)))}{\sum_k exp(-Eucli\_dist(H_s, \mathbf{c}_i)))} \tag{4}$$

where $Eucli\_dist$ is the distance function for H_s and $\mathbf{c}_i$.

And then each client model is trained with the guide by the categorical cross-entropy function.

$$\mathcal{L} = -\sum_{(x_k,y_k)} log(p(y=y_k|x_k,\mathbf{c})) \tag{5}$$

where (x_k, y_k) stands for one sample and $\mathbf{c}$ represents the centroid for this sample. In this work, the local model parameters can be derived by minimizing the average loss using the gradient descent fashion based on the back-propagation algorithm. The Adam algorithm [32] is utilized to optimize the loss function, and the learning rate and weight_decay in the local training are fixed to 1e-4, respectively.

3) Federated aggregation: Within the FL framework, each client trains model locally using its dataset without the need of sending the local dataset to a central server. The k-th local model gradient in the c-th communication round can be calculated as follows:

$$\nabla\boldsymbol{\varphi_{k,c}} = \frac{\partial\mathcal{L}_k(\boldsymbol{\varphi_{k,c}})}{\partial\boldsymbol{\varphi_{k,c}}} \tag{6}$$

where $\boldsymbol{\varphi_{k,c}}$ is the gradient of the k-th local model in the c-th communication round. L_k is the categorical cross-entropy function used to guide the local model training, which is defined as shown in Eq.(5).

All clients send their model gradients to the central server for generating the global model.

$$\nabla\boldsymbol{\varphi_c} = \sum_{i=1}^{N} \alpha_k \nabla\boldsymbol{\varphi_{k,c}} \tag{7}$$

where N is the number of client models, α_k denotes the weight of the k-th local gradient.

The weight in this work is proportional to the data size in each client $\alpha_k = S_k / \sum_i^N S_i$, by considering that the data size might be different for different clients in this work.

C. Security-enhanced Blockchain

In BLADE, we focus on the collaborative learning of blade icing detection, in which $\mathcal{K}$ wind farms located in different locations work together to achieve the task of blade icing detection. All participants (wind farms) are considered dishonest, so we believe that there are multiple threats. The first is the quality of the model updates provided. Dishonest participants may provide biased or inaccurate model updates to other participants, thereby reducing the usability of aggregated models. The second is data privacy. All participants may try to infer others' private data from model updates, which may lead to potential leakage of sensitive data. A worse situation might be a group of participants trying to extrapolate data from other participants. The third is data authority management. Once the local model is shared, the owner will lose its control, and the model may be shared by dishonest participants to other unauthorized entities. All attacks mentioned above almost are from the local side. This motivates us to employ the Blockchain to avoid the attacks of malicious clients.

1) Validation mechanism: The conventional Blockchain is equipped with weak capability for the identification of malicious nodes. Therefore, a validation mechanism proposed by Chen et al. is utilized in this work [24]. The validation mechanism in the Blockchain is designed to employ the majority vote for excluding those distorted models which may be sent from malicious or compromised workers in the process of global model construction. The idea behind the validation mechanism is very simple: the validator can receive local updates from its associated workers, if the local updates drop very severely in the current communication round than last round, it means that this local update is contaminated, otherwise, the new local updates should be better than the one in last round or the decline is small. In this work, the threshold T for determining the accuracy-drop measurement is a hyperparameter that is set subjectively.

2) Consensus mechanism: The consensus mechanism is dedicated to protecting local model updates for legal learning and ensuring that these updates are recorded on the Blockchain and used to update the global model. For this purpose, it is very necessary to avoid the malicious participants becoming a miner since the miner is responsible for aggregating the voting results and recording them in a block. There are two consensus mechanisms utilized in BLADE, i.e., proof-of-work (PoW) and proof-of-stake (PoS).

Pow is a classical consensus mechanism in the Blockchain. Its core idea is that the members (miners) in Blockchain compete for the hash operation (usually SHA-256) using their computing resources. The winner who first finds that the calculated hash value is lower than the published target has the right to append a new block to the Blockchain and obtain a certain number of rewards. An important concept in PoW is "difficulty", which is a measure used to determine the difficulty of mining new blocks. The more difficult the network is, the more computing power the miners need on average to find the next block hash. PoS was proposed to overcome the limitation of PoW. PoS uses the concept of coinage, that is, unused assets are multiplied by the duration from the last winning time to the current time, so as to avoid high resource consumption in the competition process [33].

The reward mechanisms for worker, validator, and miner are as follows [24]. 1) For each worker, the reward is proportional to the number of training samples and the number of local training epochs. Nevertheless, if the local model update does not pass the validation mechanism, there is no reward for the worker. 2) For each validator, the reward is related to the verification of the signatures of the received local model update from the worker and the generation of signature-verified update. 3) For each miner, the reward is calculated by verifying the received signatures from the validators. The reward mechanisms are applying for both consensus algorithms, i.e., PoW and PoS.

D. Privacy analysis

With the BLADE framework, privacy can be protected as follows. First, each participant connects each other through multi-party data retrieval. The Permission Blockchain is used to replace the trusted server, and thus, the high-risk data leakage in the centralized server can be avoided. Second, BLADE protects against invalid data/models provided by dishonest participants. The validation mechanism used can validate the learned data-driven model and keep only qualified models. Third, only learned models are uploaded and transferred via a Permission Blockchain, with local data stored locally. Data owners can control permissions on their own data. In addition, the use of cryptographic algorithms can also enhance the security of data. Fourth, the imbalanced learning module can also protect against data leakage because it uses cluster centers to build classifiers instead of features. Therefore, the ability to defend against attacks is improved.

IV. EXPERIMENTS

All models in this work are implemented by the Pytorch (V1.7.1, cuda 11.0). All experiments are conducted on a server which is equipped with Tesla V100 with 32 GB video memory.

A. Experimental settings and evaluation metrics

1) Data: The data used for this paper are from two wind farms which are located in Shanxi and Henan provinces of China. Both wind farms face the blade icing of wind turbine in winter, and there are approximately 700 km apart for these two wind farms. We choose five wind turbines from each wind farm for the evaluation of the proposed model. There are hundreds of sensors installed in the wind turbine and the data are sampled with a frequency of five seconds. The experienced expert helped us to identify 16 parameters highly associated with blade icing, which is illustrated in TABLE I. The icing detection in this work is considered as a binary classification problem and all of the sensor data are labeled by the expert.

The time length of collected data for these two wind farms is 360 and 384 hours, respectively. To better illustrate the model performance, we just use 60% of the data for training and the remaining 40% for testing. It is worth noting that the performance is evaluated on all testing data from each wind turbine.

2) Metrics: In order to quantify the model performance, we use the following metrics: F_β, **Precision**, **Recall**, and **Matthews correlation coefficient (MCC)**.

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F_\beta = \frac{(1 + \beta^2) \times Precision \times Recall}{\beta^2 Precision + Recall} \tag{9}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{10}$$

where TP, FP, FN, and TN represent true positive, false positive, false negative and true negative, respectively. The value of β in F_β is set to 2. The reason for selecting these four metrics is that they are widely used for performance comparison of blade icing detection algorithms in the literature [14]. More importantly, the F_β and MCC are good at evaluating the performance of imbalance classification. To validate the robustness of the proposed model, the average values of the four metrics for all participants in all communication rounds are used for performance comparison.

TABLE I: Specification of the used input parameters

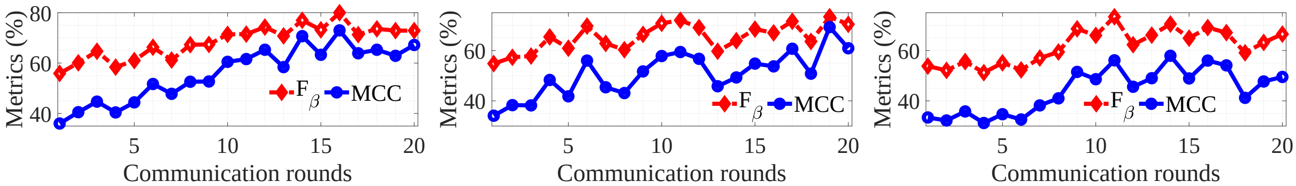
No.	Variable name	Description
1	wind_speed	Wind speed
2	wind_direction	Wind direction
3	generator_speed	Generator speed
4	power	Active power
5	yaw_position	Yaw position
6	pitch1_angle	Angle of pitch 1
7	pitch2_angle	Angle of pitch 2
8	pitch3_angle	Angle of pitch 3
9	pitch1_speed	Speed of pitch 1
10	pitch2_speed	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

3) Settings: The learning rate for the local model training is set to 1e-4. Both the communication rounds and the training epochs for the local client are set to 20. The used Blockchain is adopted from the work [24]. To evaluate the model performance in different imbalance ratios, three imbalance ratios are used in this work: 20, 50, and 100. The imbalance ratio is defined as: $\rho = \frac{S_n}{S_i}$, where the S_n and S_i are the number of samples in normal and icing status, respectively. It is worth noting that all test sets are with the imbalance ratio of 10:1. All the experiments are repeated three times.

B. Convergence analysis

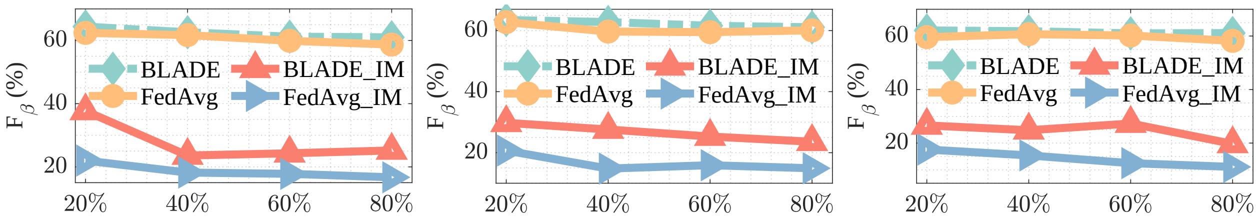
This section focuses on the convergence analysis of the learned model in the three imbalance ratios, which is depicted in Fig. 4. In this section, there is no malicious client and the number of workers, validators, miners are selected randomly in each global communication round. The validation threshold is set to 1. From Fig. 4, we can see that the proposed model can achieve acceptable results with the increase of communication rounds, even there are more fluctuations for $\rho = 50:1$ and $\rho = 100:1$ during the training. We can also see that when the communication round reaches a certain number ($\geq$ 15), the performance of the model does not improve significantly. This is why we set the communication round to 20. In reality, since wind farms are located in remote regions, communication costs are still a factor that needs to be considered.

From Fig. 4, we can also know that there is a strong relationship between the accuracy and imbalance ratios. With the increase of imbalance ratio, the accuracy of the global model obtained decreases. Obviously, the greater of imbalance ratio, the lesser icing information can be used for the learning by the model, which makes the model performance is limited.

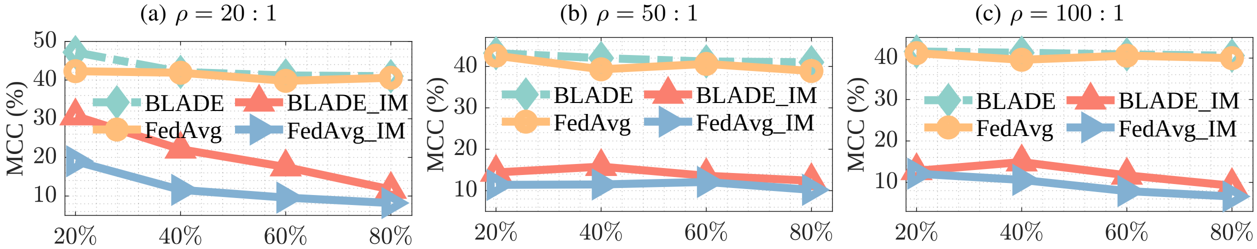


(a) $\rho = 20:1$ (b) $\rho = 50:1$ (c) $\rho = 100:1$

Fig. 4: Convergence analysis in different imbalance ratios



(a) $\rho = 20:1$ (b) $\rho = 50:1$ (c) $\rho = 100:1$



(d) $\rho = 20:1$ (e) $\rho = 50:1$ (f) $\rho = 100:1$

Fig. 5: Performance comparison of different malicious devices in the three imbalance ratios

Another interesting thing we can find is that the performance of the global model is close when the imbalance ratio is 50 and 100.

C. *Effectiveness of Blockchain for federated learning*

To illustrate the effectiveness of Blockchain for federated learning. This section focuses on the model performance under attacks, the performance comparison for the two strategies: PoW and PoS, and the effectiveness of the validation mechanism used in this work.

1) Performance comparison of defense against poisoning attacks: Three models are utilized for the performance comparison with the proposed model BLADE. 1) **FedAvg**: FedAvg is the most classical federated learning model, which uses the gradients of all clients to generate a global model. In this comparison, the client model is the same with the proposed BLADE [9]. 2) **BLADE_IM**: we remove the proposed imbalance module from the BLADE model. 3) **FedAvg_IM**: we remove the imbalance module from the FedAvg model. The number of malicious devices is set to $\{20\%, 40\%, 60\%, 80\%\}$ and the other settings are the same with the previous section. The attack is set to add Gaussian noise with a variance of 2 to the client model and the performance of the final aggregated model is selected for comparison.

The results are depicted in Fig. 5. Surprisingly, our proposed BLADE only slightly outperforms FedAvg rather than the expected significant improvement. Comparing the two models, BLADE_IM and FedAvg_IM, after removing the imbalanced learning module, the BLADE_IM is also just slightly better than FedAvg_IM in the three cases. We believe this slightly better performance is due to the Blockchain. On the contrary, comparing BLADE and BLADE_IM, FedAvg and FedAvg_IM, we find that the performance decreases significantly, and we can conclude that our proposed imbalanced learning module has the ability to deal with attacks as well as imbalance data. Our cluster-based imbalanced learning approach essentially utilizes the cluster centers of the features learned for classification. The simulated attack aims to affect the client model gradient, but does not impact much in the cluster centers used for constructing the classifier. While the attack affects the gradient of the model, the final response does not change much in the cluster centers used for classification. From Fig. 5, we can know that the model performance decreases with the number increase of malicious clients. It is worth noting that there are only 10 clients used in this work. Therefore, the influence of malicious clients needs to be further investigated.

2) Performance comparison of PoW and PoS: As mentioned above, there are two mechanisms used in BLADE: PoW and PoS. This section, therefore, compares their performance. The difficulty of PoW is set to 2 and the number of malicious clients is set to 50%, following the settings of [24]. The number of workers, validators, and miners is determined randomly in each communication round. The validation threshold is set to 1. The results are shown as TABLE II. From the results, we can easily know that the PoS has better performance in the

TABLE II: Performance comparison of PoW and PoS

Metrics	20:1		50:1		100:1	
	PoW	PoS	PoW	PoS	PoW	PoS
F_β	**66.8**	64.5	60.6	**62.4**	59.4	**60.6**
Precision	**34.4**	29.2	26.4	**27.5**	24.7	**26.1**
Recall	87.4	**92.4**	90.1	**92.3**	**92.2**	90.7
MCC	**48.3**	44.5	40.3	**42.2**	38.7	**40.2**

TABLE III: Performance comparison in different validation threshold ('T' means validation threshold)

Metrics	20:1		50:1		100:1	
	T=1	T=10	T=1	T=10	T=1	T=10
F_β	**62.8**	60.6	**61.4**	59.9	**60.9**	59.3
Precision	**27.5**	26.5	**27.3**	25.6	**26.0**	24.9
Recall	88.1	**90.0**	**90.8**	90.2	**92.1**	91.2
MCC	**42.9**	40.3	**41.4**	39.5	**40.5**	38.7

case of 50:1 and 100:1.

3) Effectiveness of the validation mechanism: As mentioned above, there is a novel validation mechanism used in the Blockchain. To illustrate its effectiveness, we compare the performance in two different validation thresholds, i.e., T=1 and T=10. We set the number of workers, validators, and miners to $\{6, 2, 2\}$, respectively. The number of malicious devices is set to 50%. The other settings are the same with the previous sections. It is worth noting that the smaller T can result in better performance of identifying the malicious devices. As shown in TABLE. III, the higher performance happens when the T is smaller, as expected, in the three cases.

D. Comparison with state-of-the-art class-imbalance methods

One of the concerns in this work is the imbalance of learning due to the data nature of the blade icing. Therefore, it is very necessary to compare our proposed BLADE with the state-of-the-art class-imbalance methods. These methods are illustrated as follows:

1) **Focalloss**: is a famous algorithm-level loss function for class-imbalance learning [30]. The hyper-parameters for the focal loss are set: the α is 1, and γ is 2. 2) **Class-balance**: is an improved version of focal loss by introducing the concept of effective number of samples. In this work, the data overlap is taken into account, and according to the calculated effective number of samples in each class, the final balanced loss of each class can be obtained [31]. 3) **LDAM** (label-distribution-aware margin): is a recently proposed class-balance algorithm for deep learning models by minimizing a margin-based generalization bound. In this method, the CE loss can be enhanced with the prior strategies, and a simple but effective training schedule is applied [34]. 4) **WeightedCE**: is an easily improved variant of CE loss by considering the number of samples in each class is different. According to the different number of samples, different weights are assigned to standard CE loss for the imbalance learning.

All the above methods are used for the model in which the proposed module of imbalance learning is removed while the other components are remained. The details of the training settings are the same with the previous sections, such as the same learning rate, training epochs, etc. All clients are assumed to be honest and the validation threshold is set to 1. The communication rounds and the local epochs are set to 20. The results are shown in TABLE IV. It is easy to see that our proposed BLADE achieves the highest accuracy in terms of MCC, except in the case of 100:1 where our model is slightly lower than WeightedCE. Regarding F_β, BLADE has the best performance in the three cases. WeightedCE ranks second, which means that even the simple method can achieve good performance. Another reason that the WeightedCE can have a better performance than others is that all of the clients have the same imbalance ratio, it is convenient for WeightedCE to assign the weights for each class. If the imbalance ratio is different for different clients, the result might change. These designed algorithm-level imbalance learning methods (i.e., Focalloss, Class-balance, and LDAM) do not have the expected results. The explanation might be that they are designed for central learning not for distributed learning. In the new scenario, performance degradation exists.

E. Impacts of different settings in the neural network

1) Impacts of the structure for the neural network: The learning capability is also an important factor for the proposed BLADE. Therefore, the following five widely used baseline models are utilized for the illustration of the impacts of the local model. 1) **MLP (multilayer perceptron)**: there are three fully connected (FC) layers and one dropout layer inserted between FC layers. The hidden number for each FC is set to 500. 2) **LSTM** (Long Short-Term Memory): a simple one-layer LSTM model with the hidden number 8 is used. 3) **CNN**: one-layer CNN with filter size 128 is employed. 4) **DenseNet**: is another state-of-the-art model for time series classification problem [27]. We just use the model structure with the attention modules excluded. 5) **MLSTM-FCN**: we adopt the same settings with the original paper [35]. These models are used to replace our proposed neural network and the other settings are the same with Section IV-B.

The results are illustrated in TABLE V and indicate that our proposed BLADE performs the best in terms of F_β except $\rho = 100 : 1$, where the score is second only to Densenet. Regarding MCC, our proposed BLADE also achieves competitive performance except $\rho = 100 : 1$, where our method is in the third place, ranking behind the DenseNet and LSTM. It is interesting that the MLP and LSTM are almost identical for the three cases. More importantly, we find that the performance has not improved significantly with the increased complexity of the model structure. For example, the DenseNet and MLSTM_FCN have the most complicated model structure in this comparison, but their performance is not always the best. These results indicate that designing a suitable model is even more significant than using a complex model.

2) Comparison with other attention mechanisms: To demonstrate the effectiveness of the proposed attention module, we compare it with the other four attention modules, as shown in TABLE VI. In this comparison, we just replace our proposed attention module with these attention modules. The other settings are the same with Section IV-B. The details of these four attention modules are as follows: **SE**: is the first and

TABLE IV: Performance comparison of imbalance learning algorithms in different number of imbalance ratios

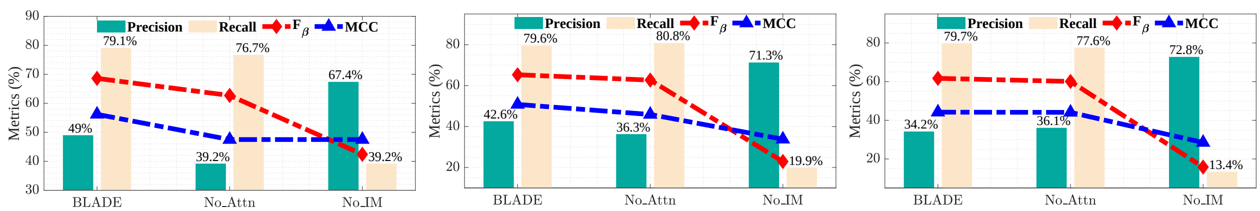
Method	$\rho = 20:1$				$\rho = 50:1$				$\rho = 100:1$			
	F_β	Precision	Recall	MCC	F_β	Precision	Recall	MCC	F_β	Precision	Recall	MCC
Focalloss	37.8	66.6	34.4	43.9	27.1	**75.3**	23.6	38.6	17.0	**70.0**	14.6	27.7
Class-balance	42.5	13.4	**99.6**	16.7	48.6	16.4	**99.1**	26.3	44.7	14.3	**99.3**	21.2
LDAM	48.3	**67.2**	45.6	51.4	28.8	62.9	26.0	35.8	21.1	64.4	18.6	29.6
WeightedCE	53.0	54.3	53.2	48.8	51.9	58.0	51.3	49.7	47.4	56.8	46.0	**46.4**
Ours	**68.6**	49.0	79.1	**56.2**	**65.3**	42.6	79.6	**50.8**	**61.7**	34.2	79.7	44.2

TABLE V: Performance comparison of local model

Method	$\rho = 20:1$				$\rho = 50:1$				$\rho = 100:1$			
	F_β	Precision	Recall	MCC	F_β	Precision	Recall	MCC	F_β	Precision	Recall	MCC
MLP	61.2	42.1	70.9	48	58.8	39.2	68.9	45.0	57.1	37.6	68.1	42.9
LSTM	61.5	48.8	68.0	51.5	58.5	**45.4**	66.1	47.8	59.2	**40.4**	71.0	45.9
CNN	43.9	17.1	72.5	21.9	45.8	17.6	76.9	23.8	44.1	16.4	76.7	21.9
DenseNet	66.1	34.7	**88.0**	47.9	64.2	32.6	**88.6**	45.7	**66.9**	35.0	**90.0**	**48.8**
MLSTM_FCN	64.1	45.0	76.1	51.1	63.3	34.6	83.7	45.6	59.2	34.8	78.0	42.9
Ours	**68.6**	**49.0**	79.1	**56.2**	**65.3**	42.6	79.6	**50.8**	61.7	34.2	79.7	44.2

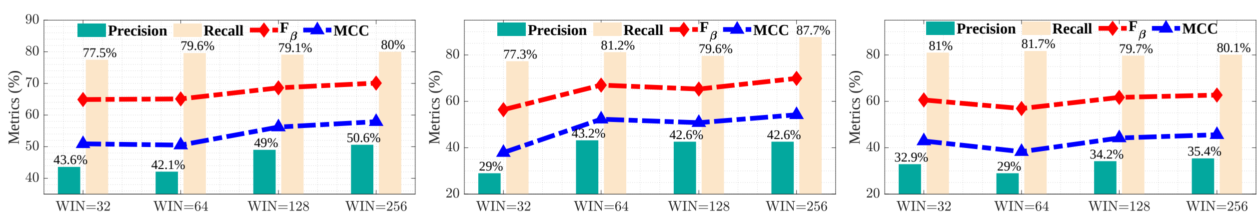
TABLE VI: Performance comparison of attention modules

Method	$\rho = 20:1$				$\rho = 50:1$				$\rho = 100:1$			
	F_β	Precision	Recall	MCC	F_β	Precision	Recall	MCC	F_β	Precision	Recall	MCC
SE	67.1	50.5	76.5	55.8	**65.7**	38.5	**83.2**	49.1	60	31.9	80.9	42.1
GC	64.7	42.5	78.3	50.2	62.8	38	79.3	46.8	**64.1**	**38.7**	**81.3**	**48.3**
TR	53.5	**51.0**	55.8	47.4	57.5	36.2	70.8	42.5	49.3	38.4	58.6	38.5
CBAM	57.1	36.0	75.5	41.5	55	33.4	75.4	38.4	55.4	30.8	77.2	37.8
Ours	**68.6**	49.0	**79.1**	**56.2**	65.3	**42.6**	79.6	**50.8**	61.7	34.2	79.7	44.2



(a) $\rho = 20:1$ (b) $\rho = 50:1$ (c) $\rho = 100:1$

Fig. 6: Ablation analysis



(a) $\rho = 20:1$ (b) $\rho = 50:1$ (c) $\rho = 100:1$

Fig. 7: Sensitivity analysis

famous attention module [36]. **GC**: calibrates the raw learned features by CNN using the concept of global context [37]. **TR**: is a recently proposed attention module for time series classification problem [38]. **CBAM**: is a step-wise attention module which uses the channel and spatial attention module sequentially [39].

To implement the comparison, we simply replace the used attention module by these four attention modules, and the other settings are the same with Section IV-B. TABLE VI clearly shows that the proposed BLADE performs the best in the case of $\rho = 20:1$. In the cases of $\rho = 50:1$ and $\rho = 100:1$, our proposed BLADE is not the best except for the MCC in the case of $\rho = 50:1$. In the case of $\rho = 100:1$, GC shows the best performance, and our proposed BLADE ranks in the second place.

F. Ablation and sensitivity analysis

To further illustrate the importance of each component, the ablation analysis is conducted. In this work, there are two variants are created. 1) **No_IM**: the imbalance learning module is removed in this variant. 2) **No_Attn**: the proposed channel attention module is not used. In this section, the experiment setting is the same with Section IV-B. The results are illustrated in Fig. 6. From the results, we can know that there is the highest performance decrease when the imbalance learning module is removed in the three cases. When the attention module is not used, there is just a slight performance decrease in the three cases. Compared with **No_IM** and **No_Attn**, we can conclude that our proposed imbalance learning module has better performance than the attention module.

The sensitivity analysis is performed to identify the influence of window size on detection accuracy. The results are shown in Fig. 7. From the results, we can easily know that the highest performance happens when the window size is 256 in the three cases. The reason behind might be that the larger window contains more information and results in higher accuracy.

V. Conclusion

Wind energy offers a promising solution for human's future sustainable development. Blade icing is the main factor that limits the performance of wind turbines, especially in winter. This work investigates the Blockchained imbalanced federated learning model BLADE for blade icing detection of wind turbine. BLADE is proposed to address the limitation of centralized servers in conventional federated learning. This paper also introduces a novel cluster-based imbalance learning module by considering that there is a heavy data imbalance in the collected data. To enhance the capability of defending the poisoning attacks, a validation mechanism is introduced in Blockchain. This paper comprehensively evaluated the performance of BLADE by comparing the performance of the aggregated model and four state-of-the-art class-imbalanced learning methods. The performance comparison between different neural networks and attention mechanisms are explored. The effectiveness of Blockchain and each component in BLADE is also investigated.

However, the heterogeneity of each participant is not considered, and the existed heterogeneity might reduce the performance of the model proposed in this paper. A heterogeneous Blockchain-based federated learning model will be designed. In the future, we would like to identify the severity of icing on wind turbine blades. Potential solutions might include interpreting the output probability or labeling the icing severity, based on domain knowledge.

Acknowledgment

The research is partly supported by a grant from the International Cooperation Project of Guangdong Province (Project nr: 2021A0505030017) and partly supported by the National Natural Science Foundation of China (62020106004).

References

[1] Z. Ren, A. S. Verma, Y. Li, J. J. Teuwen, and Z. Jiang, "Offshore wind turbine operations and maintenance: A state-of-the-art review," *Renewable and Sustainable Energy Reviews*, vol. 144, p. 110886, 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, 2020.

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

[4] L. Battisti, P. Baggio, and R. Fedrizzi, "Warm-air intermittent de-icing system for wind turbines," *Wind Engineering*, vol. 30, no. 5, pp. 361–374, 2006.

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

[6] P. Roberge, J. Lemay, J. Ruel, and A. Bégin-Drolet, "A new atmospheric icing detector based on thermally heated cylindrical probes for wind turbine applications," *Cold Regions Science and Technology*, vol. 148, pp. 131–141, 2018.

[7] 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.

[8] C. Goncalves, R. J. Bessa, and P. Pinson, "Privacy-preserving distributed learning for renewable energy forecasting," *IEEE Transactions on Sustainable Energy*, 2021.

[9] 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.

[10] S. Shajiee, L. Y. Pao, and R. R. McLeod, "Monitoring ice accumulation and active de-icing control of wind turbine blades," in *Wind Turbine Control and Monitoring*. Springer, 2014, pp. 193–230.

[11] X. Dong, D. Gao, J. Li, Z. Jincao, and K. Zheng, "Blades icing identification model of wind turbines based on scada data," *Renewable Energy*, vol. 162, pp. 575–586, 2020.

[12] M. L. Corradini, A. Cristofaro, and S. Pettinari, "A model-based robust icing detection and estimation scheme for wind turbines," in *2016 European Control Conference (ECC)*. IEEE, 2016, pp. 1451–1456.

[13] 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.

[14] 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.

[15] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, 2019.

[16] L. Wang, S. Xu, X. Wang, and Q. Zhu, "Towards class imbalance in federated learning," *arXiv preprint arXiv:2008.06217*, 2020.

[17] L. Cui, Y. Qu, G. Xie, D. Zeng, R. Li, S. Shen, and S. Yu, "Security and privacy-enhanced federated learning for anomaly detection in iot infrastructures," *IEEE Transactions on Industrial Informatics*, 2021.

[18] X. Xu, H. Peng, M. Z. A. Bhuiyan, Z. Hao, L. Liu, L. Sun, and L. He, "Privacy-preserving federated depression detection from multi-source mobile health data," *IEEE Transactions on Industrial Informatics*, 2021.

[19] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, "Energy demand prediction with federated learning for electric vehicle networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[20] X. Zhang, F. Fang, and J. Wang, "Probabilistic solar irradiation forecasting based on variational bayesian inference with secure federated learning," *IEEE Transactions on Industrial Informatics*, 2020.

[21] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2019.

[22] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4298–4311, 2020.

[23] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, S. Chen, X. Xu, and L. Zhu, "Blockchain-based federated learning for device failure detection in industrial iot," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5926–5937, 2020.

[24] H. Chen, S. A. Asif, J. Park, C.-C. Shen, and M. Bennis, "Robust blockchained federated learning with model validation and proof-of-stake inspired consensus," *arXiv preprint arXiv:2101.03300*, 2021.

[25] D. V. Medhane, A. K. Sangaiah, M. S. Hossain, G. Muhammad, and J. Wang, "Blockchain-enabled distributed security framework for next-generation iot: an edge cloud and software-defined network-integrated

approach," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6143–6149, 2020.

[26] Y. Wu, H.-N. Dai, and H. Wang, "Convergence of blockchain and edge computing for secure and scalable iiot critical infrastructures in industry 4.0," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2300–2317, 2020.

[27] 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.

[28] J. Zhang, J. Chen, J. Cao, R. Liu, L. Bian, and S. Chen, "Dual attention granularity network for vehicle re-identification," *Neural Computing and Applications*, pp. 1–12, 2021.

[29] Y. Xie, M. Qiu, H. Zhang, L. Peng, and Z. Chen, "Gaussian distribution based oversampling for imbalanced data classification," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

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

[31] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9268–9277.

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

[33] B. Cao, Z. Zhang, D. Feng, S. Zhang, L. Zhang, M. Peng, and Y. Li, "Performance analysis and comparison of pow, pos and dag based blockchains," *Digital Communications and Networks*, vol. 6, no. 4, pp. 480–485, 2020.

[34] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *Advances in Neural Information Processing Systems*, 2019, pp. 1567–1578.

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

[36] 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.

[37] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 1971–1980.

[38] Y. Hao and H. Cao, "A new attention mechanism to classify multivariate time series." in *IJCAI*, 2020, pp. 1999–2005.

[39] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.