Cost-Sensitive Extremely Randomized Trees Algorithm for Online Fault Detection of Wind Turbine Generators

Mingzhu Tang[1], Yutao Chen[1], Huawei Wu[2]*, Qi Zhao[1], Wen Long[3], Victor S. Sheng[4]* and Jiabiao Yi[1]

[1]School of Energy and Power Engineering, Changsha University of Science and Technology, Changsha, China, [2]Hubei Key Laboratory of Power System Design and Test for Electrical Vehicle, Hubei University of Arts and Science, Xiangyang, China, [3]Guizhou Key Laboratory of Economics System Simulation, Guizhou University of Finance and Economics, Guiyang, China, [4]Computer Science Department, Texas Tech University, Lubbock, TX, United States

The number of normal samples of wind turbine generators is much larger than the number of fault samples. To solve the problem of imbalanced classification in wind turbine generator fault detection, a cost-sensitive extremely randomized trees (CS-ERT) algorithm is proposed in this paper, in which the cost-sensitive learning method is introduced into an extremely randomized trees (ERT) algorithm. Based on the classification misclassification cost and class distribution, the misclassification cost gain (MCG) is proposed as the score measure of the CS-ERT model growth process to improve the classification accuracy of minority classes. The Hilbert-Schmidt independence criterion lasso (HSICLasso) feature selection method is used to select strongly correlated non-redundant features of doubly-fed wind turbine generators. The effectiveness of the method was verified by experiments on four different failure datasets of wind turbine generators. The experiment results show that average missing detection rate, average misclassification cost and gMean of the improved algorithm better than those of the ERT algorithm. In addition, compared with the CSForest, AdaCost and MetaCost methods, the proposed method has better real-time fault detection performance.

Keywords: fault detection, fault diagnosis, cost-sensitive learning, extremely randomized trees, class imbalance, wind turbine generator

INTRODUCTION

The global capacity of installed wind turbine generators in 2019 reached 60.4 GW, with an annual increment of 19% (Kandukuri et al., 2016). The operation and maintenance costs of wind turbine generators account for approximately 15–30% of their total cost (Artigao et al., 2018). Generator failures account for approximately 4% of total failures, and generator fault identification has attracted considerable attention in recent years (Chen et al., 2016; Quiroz et al., 2018; Lei et al., 2019). Failures in the generator may cause the whole mechanical system to stop functioning, reduce the operation efficiency of the wind turbine and even cause personnel casualties. Wind turbine generators, intermittent operating conditions, and severe weather pose challenges to the safe operation of wind turbines (Judge et al., 2019). Since the generator is the critical component of the wind turbine, wind turbine failure detection can greatly reduce the operation and maintenance costs by reducing unplanned failures (Willis et al., 2018; Yang et al., 2021).

Fault detection methods can be divided into two categories: model-based methods (Cho et al., 2018; Habibi et al., 2019) and data-based methods (Mingzhu et al., 2020; Liming and Bo, 2020; Song et al., 2021). Model-based fault detection methods include a parameter estimation method (Pan et al., 2017), state estimation methods (Shahriari et al., 2020; Ghahremani and Kamwa, 2016), and an equivalent space method (Bakri and Boumhidi, 2018). Bakri et al. proposed a model-based fault detection and isolation technology to solve the early fault detection problem of wind turbines (Bakri and Boumhidi, 2018).These methods can comprehensively examine the essence of dynamic systems for real-time fault detection. However, the structure of wind turbines is complex, with many characteristic parameters, and model-based methods have difficulty obtaining accurate models.

Data-based methods include signal-based methods, statistical analysis-based methods, and machine learning-based methods. Fernandez-Canti et al. proposed a wind turbine fault detection method based on the hybrid Bayesian set membership method (Fernandez-Canti et al., 2015). This method only uses the non-fault behavior model to generate the consistency index and the fault indicator, and detects whether the wind turbine fails by analyzing the noise of the equipment. It is difficult for methods based on statistical analysis to detect the fault of a combination of signal distortion and signal fading. Ibrahim et al. proposed a method based on an effective extended Kalman filter to iteratively estimate a fault signature component (FSC) and track its amplitude to realize fault detection in wind turbine generators (Ibrahim et al., 2018). The state characteristic signal is weak at the initial stage of the fault, which makes it difficult to accurately detect generator faults by the signal-based method.

Machine learning-based methods—for instance, artificial neural networks (ANNs) (Marugan et al., 2018; Hamidreza et al., 2014), support vector machines (Zeng et al., 2019; Li Z. M. et al., 2019), decision trees (Yu et al., 2018), bagging (Breiman, 1996), boosting (Cheki et al., 2016), and random forests (RFs) (Li et al., 2016; Joshuva and Sugumaran, 2017)—are often applied to solve binary classification problems. These methods can effectively predict the operating state of a wind turbine. Chun et al. used RF learning to evaluate the correlation between characteristic variables and target variables and then used a deep neural network (DNN) model to identify wind turbine permanent magnet drop failures. However, DNNs are computationally complex and easily overfit data (Teng et al., 2018). Gao et al. used the integrated extended load mean decomposition multiscale entropy method to extract features and then applied the least square support vector machine (LSSVM) method to perform wind turbine fault detection (Gao et al., 2018). The LSSVM method achieved strong fault detection performance but poor real-time performance when processing big data. Gopinath proposed a method for wind turbine fault detection that combines nuisance attribute projection and the classification and regression tree (CART) algorithm (Gopinath et al., 2016). Disturbance attribute projection was used to extract the frequency domain statistical characteristics of the current signal, and CART was used as a decision model to realize synchronous generator fault detection. Although a decision tree method has various advantages, such as

a simple structure, strong real-time performance, and the ability to handle big data, a single decision tree is impractical. Li et al. adopted the short-term memory network of the residual generator and used an RF to build a detection model (Li M. et al., 2019). This method can effectively detect early faults of wind turbines in harsh environments. The RF model improves the generalization ability *via* integration.

In the actual operation of wind turbines, the number of fault samples is much smaller than the number of normal samples, which is characteristic of typical imbalanced classification problems (Malik and Mishra, 2016; Buda et al., 2018; Longting et al., 2019). Traditional fault detection methods perform poorly when applied to imbalanced data. For class-imbalanced problems, cost-sensitive learning combines misclassification costs and traditional fault detection methods. By introducing different types of cost functions to characterize the importance of a sample, the objective function is transformed from one designed to maximize the classification accuracy into one designed to minimize the misclassification cost. For example, the cost-sensitive decision tree algorithm has been widely used in industrial control processes and detection (Tan, 1993; Lomax and Vadera, 2013; Kim et al., 2018). Because the test cost and misclassification cost of cost-sensitive learning are often similar in scale, Zhang et al. presented a multiscale cost-sensitive decision tree algorithm that combines the misclassification cost and test cost. The approach solves the problem of integrating multiple costs together in cost-sensitive learning (Zhang, 2018). Qi et al. proposed a cost-sensitive decision tree algorithm that incorporates data cleaning algorithms to address poor-quality data, including the high cleaning cost (Qi et al., 2019). However, a single classifier easily leads to overfitting when considering complex industrial problems and the poor model generalization ability.

Ensemble learning combines multiple classifiers to obtain better performance than that achieved by a single classifier. Tree ensemble algorithms can be classified as either boosting or bagging. Masnadi-Shirazi et al. presented a cost-sensitive framework suitable for AdaBoost, RealBoost, and LogitBoost for class-imbalanced problems (Masnadi-Shirazi and Vasconcelos, 2011). Furthermore, Zelenkov et al. proposed a sample-based cost-sensitive adaptive boosting algorithm (Zelenkov, 2019) in which the misclassification cost and sample distributions are combined, and the cost matrix of the sample is corrected based on the training set to improve the overall performance. Because the boosting algorithm uses serial dependence, it is difficult to train data in parallel. The cost-sensitive RF algorithm uses a parallel approach and has strong generalization capabilities (Nami and Shajari, 2018; Siers and Islam, 2015). Siers et al. combined cost-sensitive parameters with an RF model, introduced misclassification costs when building models, and implemented a cost-sensitive forest (CSForest) algorithm based on a decision tree (Siers and Islam, 2015). Lu et al. embedded the cost of misclassification, test cost and rejection cost into a rotating forest algorithm (Lu et al., 2017), which was transformed into a cost-sensitive problem to effectively reduce the classification cost and improve the effectiveness of the algorithm. However, the computational

complexity of the cost-sensitive RF algorithm is high. Geurts et al. proposed an extremely randomized trees (ERT) algorithm based on the RF algorithm (Geurts et al., 2006). By adding random disturbances when nodes are split, the model achieves stronger generalization ability and reduced computational complexity. Moreover, each base classifier uses the complete training dataset for training, which reduces the variance of the ERT algorithm.

Although the ERT algorithm has faster calculation speed and smaller prediction variance (Geurts et al., 2006), the problem of low detection accuracy of failure samples still exists for unbalanced data. For the imbalance problem, many cost-sensitive fault detection methods based on tree ensemble algorithms have been proposed and have made certain achievements in the field of wind turbine generator fault detection. However, these methods make it difficult to meet both high performance and high real-time requirements. Therefore, this paper proposes a wind turbine generator fault detection method based on cost-sensitive extremely randomized trees (CS-ERT). The main contributions of this paper are as follows:

- To solve the class imbalance problem in the actual operation of wind turbine generators, cost sensitive learning was introduced into the ERT algorithm, and the CS-ERT algorithm was proposed to detect the fault of wind turbine generators. The objective function of the algorithm was transformed from minimizing classification error to minimizing misclassification cost. The proposed method was verified by the data of 1.5 MW doubly-fed wind turbine generators.
- The HSICLasso feature selection method was used to remove weak correlation features to address the high feature dimension problem of wind turbine generators. A feature subset composed of strongly correlated non-redundant variables was used to train the fault detection model.

EXTREMELY RANDOMIZED TREES

ERT (Geurts et al., 2006) is an ensemble algorithm with high randomness in which a set of nonpruned decision trees is established via a top-down process. In contrast to the RF algorithm, bagging is not used by the ERT model to train each basic classifier. Each tree of ERT uses the complete training samples for learning to minimize the deviation in the model. In the traditional ensemble method, the best feature and cut-point of a node are obtained by evaluating the Gini coefficient, Shannon entropy of each feature value of each feature, etc. ERT is different.

Given the dataset D (X, Y), the m-dimensional vector f_i represents the feature vector of the sample x_i. In the extreme decision tree splitting process, a value a_c^k is randomly selected from the maximum a_{max}^k to the minimum a_{min}^k for attribute k as the cut-point of this feature. Then, the score measure of feature k is calculated according to **Eq. 1**.

TABLE 1 | Cost matrix of two classification problems.

Predict class	Actual class	
	Normal	**Fault**
Normal	C_{TN}	C_{FN}
Fault	C_{FP}	C_{TP}

Note that C_{FN} is the cost of predicting the fault sample as a normal class, C_{FP} is the cost of predicting the normal sample as a fault class, and C_{TN} and C_{TP} represent the cost of correct detection. The larger the misclassification cost parameters, the more important the classification. For practical wind turbine generators, the economic losses caused by false negatives are far greater than those caused by missing detection. Therefore, the misclassification cost parameter C_{FN} of the fault class is greater than the misclassification cost parameter C_{FP} of the normal class ($C_{FN} > C_{FP}$).

$$\text{Score}_c(k, S) = \frac{2I_c^k(S)}{H_k(S) + H_c(S)} \quad (1)$$

where $I_c^k(S)$ represents the mutual information of the two subsets with respect to the class after node S is split according to attribute k and cut-point a_c^k. $H_k(S)$ represents the split entropy of attribute k. $H_c(S)$ represents the information entropy of node S. Each candidate feature of the node is traversed according to the above method, and the feature and cut-point with the largest score measure $\text{Score}_c(k, S)$ are selected to split the node. Then, the samples with a value of feature k less than the cut-point are placed in the left leaf node, and the remaining samples are placed in the right leaf node. The above steps are repeated recursively until the stop splitting condition is satisfied. The simplicity of the tree growth process makes the space complexity of ERT lower than that of other ensemble methods.

The final result of the ERT algorithm is determined by voting by all base classifiers, as follows.

$$P_{(c|f_i)} = \frac{1}{M} \sum_{t=1}^{M} P_t(c|f_i) \quad (2)$$

$$\hat{c} = \text{argmax}_c P(c|f_i) \quad (3)$$

where M is the total number of trees, f_i is the feature vector of sample x_i, and P_t represents the conditional probability that the sample belongs to class c under the condition of vector f_i. For regression problems, **Eq. 2** defines the classification probability of the sample. For classification problems, the voting method is used to make decisions according to **Eq. 3**. In the fault detection method, **Eq. 3** is used to realize the fault detection of the sample.

COST-SENSITIVE EXTREMELY RANDOMIZED TREES

In this section, the CS-ERT algorithm is proposed, and the computational complexity of the algorithm is analyzed.

The Principle of Cost-Sensitive Extremely Randomized Trees

CS-ERT is a derivative of the ERT algorithm. CS-ERT combines cost-sensitive learning with the ERT algorithm, which solves the problem of low accuracy in the failure samples of traditional ERT
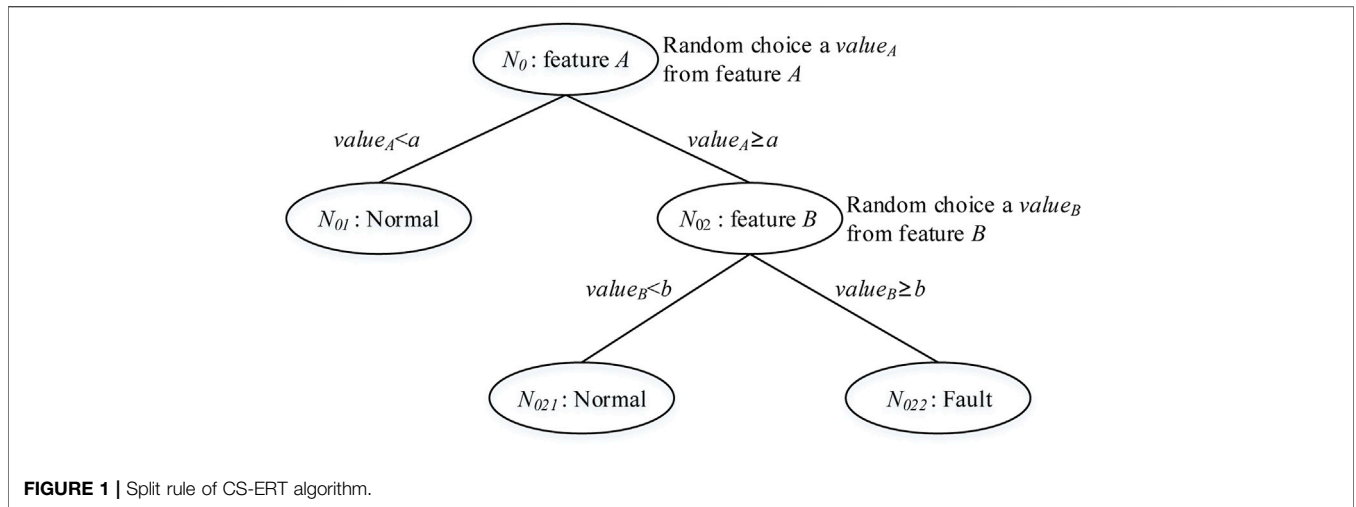
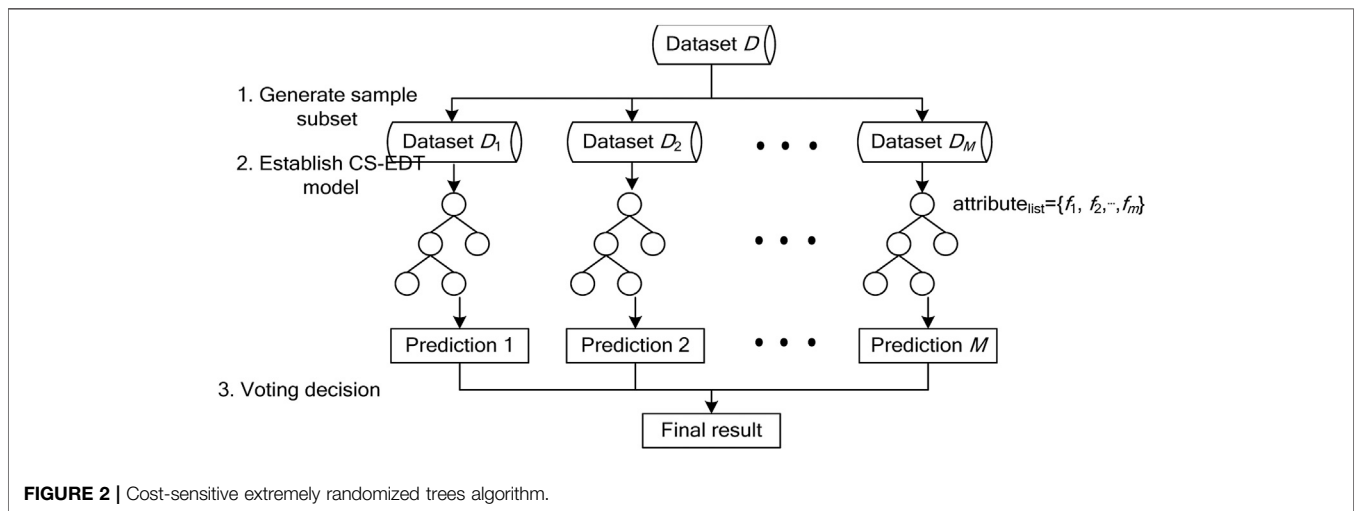**FIGURE 1 |** Split rule of CS-ERT algorithm.

FIGURE 2 | Cost-sensitive extremely randomized trees algorithm.

algorithms in imbalanced data. The cost matrix is introduced to represent the misclassification cost in the fault detection field, as shown in **Table 1**.

The CS-ERT algorithm is composed of multiple cost-sensitive extreme decision trees (CS-EDT). Each CS-EDT model has a chain structure similar to a decision tree, which includes a finite set and edge set that constitute the root node, branch nodes and leaf nodes, as shown in **Figure 1**.

In **Figure 1**, Ni represents the i-th node. If Ni is a branch node, the cut-point is randomly selected for each feature of the node. To solve the problem of category imbalance, this paper proposes the MCG as the score measure of the branch node. The MCG G_k for attribute k is defined as follows:

$$G_k = C(parent\_node) - \frac{N_L}{N_L + N_R} C(left\_child\_node)$$
$$- \frac{N_R}{N_L + N_R} C(right\_child\_node) \qquad (4)$$

where $C(parent\_node)$ represents the misclassification cost of the parent node; $C(left\_child\_node)$ and $C(right\_child\_node)$ are the misclassification costs of the left and right child nodes, respectively; and N_L and N_R represent the numbers of the left and right child nodes, respectively. According to **Eq. 4**, the misclassification cost gain is calculated for each candidate feature. Then, the attribute and random value with the largest MCG is selected as the split feature and cut-point of the branch node.

The MCG is essentially the difference between the misclassification cost of the parent node and the weighted sum of the costs of all child nodes. The misclassification cost of the leaf *node* is defined as follows:

$$C(node) = C_P + C_N \qquad (5)$$

where C_P is the cost of the fault class at *node*, and C_N is the cost of the normal class at *node*, as shown in **Eqs. 6, 7**:

$$C_P = C_{FP} \cdot N_{FP} + C_{TP} \cdot N_{TP} \tag{6}$$

$$C_N = C_{FN} \cdot N_{FN} + C_{TN} \cdot N_{TN} \tag{7}$$

where N_{FP} is the number of false alarm samples, and N_{FN} is the number of missing detection samples. N_{TP} and N_{TN} are the numbers of samples correctly predicted as faults and normal, respectively. As shown in **Table 1**, C_{FN}, C_{FP}, C_{TN} and C_{TP} are the misclassification cost parameters.

The score measure of the branch node is affected by the sample distribution. Thus, to reduce the impact of class imbalance, the class distribution is added to the calculation of the misclassification cost function. In addition, C_{TP} and C_{TN} are usually regarded as zero in industry. The expression of the misclassification cost function is as follows:

$$C_P = p_P \cdot C_{FP} \cdot N_{FP} \tag{8}$$

$$C_N = p_N \cdot C_{FN} \cdot N_{FN} \tag{9}$$

where $p_P = N_P/(N_P + N_N)$ represents the proportion of faulty samples in the node, and $p_N = N_N/(N_P + N_N)$ is the proportion of normal samples in the node. N_P and N_N are the numbers of samples classified as faults and normal, respectively.

If N_i is a leaf node in **Figure 1**, according to Bayes' theorem, the classification with the minimized misclassification cost is selected as the category of the leaf node. The definition is as follows.

$$\hat{c} = \underset{i=0,1}{arg\min} \left\{ \sum_j p\left(c_j|x\right)c_{ij} \right\} \tag{10}$$

where $p\left(c_j|x\right)$ represents the posterior probability that sample x belongs to class c_j, and c_{ij} represents the cost of a sample of class i being classified as belonging to class j.

The CS-ERT model is developed through generating sample subsets, establishing the CS-EDT method, and making decisions. A structure diagram of the CS-ERT method is shown in **Figure 2**.

D(X, Y) is the dataset, where $X = [x_1, x_2, \ldots, x_m]$ is the m-dimensional feature space, and $Y \in [0, 1]$ represents the target variables. First, **Figure 2** shows that one of the differences between ERT and a traditional random forest is that it generates M subsets that are the same as the original dataset D. Then, CS-EDT models $\{h(X, \theta_m), m = 1, \cdots, M\}$ are trained with these subsets, where M represents the number of CS-EDT models. Notably, the candidate features of the root node are all the features of the sample subset in the process of tree growth, and the leaf node is established recursively. Finally, the classification results of multiple CS-EDTs are integrated by means of the CS-ERT method, and the predicted category of the sample is determined according to majority voting, as shown in **Eq 11**:

$$H(x) = \underset{y}{arg\max} \sum_{m=1}^{M} I\left[h(x, \theta_m) = y\right] \tag{11}$$

where $h(x, \theta_m)$ is a CS-EDT model, y is the classification result of the base classifier, and I(●) is an exponential function.

Pseudocode of CS-ERT is presented as follows.

Algorithm 1 CS-ERT approach.

Input: Training dataset D, Number of base classifiers M, Base classifier CS-EDT, Cost matrix $C = [C_{FP}, C_{FN}, C_{TP}, C_{TN}]$, Candidate attribute set attribute $_{list}$ = $\{f_1, \cdots, f_m\}$

---

1   **for** m=1,2,…,M **do**
2   Obtain the same dataset D_m as the train set D
3   Create node N
4   **if** the samples of node N have the same class c, **then**
5   **Return** node N is a leaf node, node N classification is c;
6   **End if**
7   **if** attribute $_{list}$ is empty, **then**
8   Calculate the misclassification cost of node N marked as normal or fault according to (5);
9   **Return** node N is a leaf node, and node N is marked as a class with a low misclassification cost;
10   **End if**
11   Select the attribute A_{best} with the highest MCG in attribute $_{list}$;
12   **for** each attribute A in attribute $_{list}$, **do**
13   Randomly select a value of the attributeA_i as the cut-point a_i^c, and the MCG G_i is calculated according to (4);
14   **Return** Select the attribute A_{best} with the largest G_i;
15   **End for**
16   attribute$_{list}$←attribute$_{list}$ − A_{best}
17   Put the samples with $a_{best} < a_{best}^c$ into the left node N_L, and put the samples with $a_{best} \geq a_{best}^c$ into the right node N_R;
18   Add node CS-ERT(N_L, C = $[C_{FP}, C_{FN}, C_{TP}, C_{TN}]$, attribute$_{list}$ − A_{best}) and CS-ERT(N_R, C = $[C_{FP}, C_{FN}, C_{TP}, C_{TN}]$, attribute$_{list}$ − A_{best});
19   **return** $h(x, \theta_m)$ // Each base classifier is trained with a complete training set
20   **End for**
21   **return** $H(x) = \sum_m^M I[h(x, \theta_m) = y]$

Output: CS-ERT$H(x)$

The Computational Complexity of Cost-Sensitive Extremely Randomized Trees

The computational complexity of the RF algorithm is $O(M(mnlogn))$, where M represents the number of base classifiers, m represents the number of features, and n represents the number of samples. Compared with RFs, the CS-ERT algorithm introduces randomness in the process of tree growth. When a node selects a split feature, a random value for each feature is used as the cut-point for that attribute. Therefore, the computational complexity of CS-EDT is $O(mlogn)$, and the computational complexity of the CS-ERT algorithm is $O(M(mlogn))$, according to (11). The CS-ERT algorithm has better real-time performance.

WIND TURBINE GENERATOR FAULT DETECTION

In wind turbine generator fault detection, there are generally two types of erroneous predictions: 1) missed detection, where a

FIGURE 3 | Flowchart of wind turbine generator fault detection based on CS-ERT.
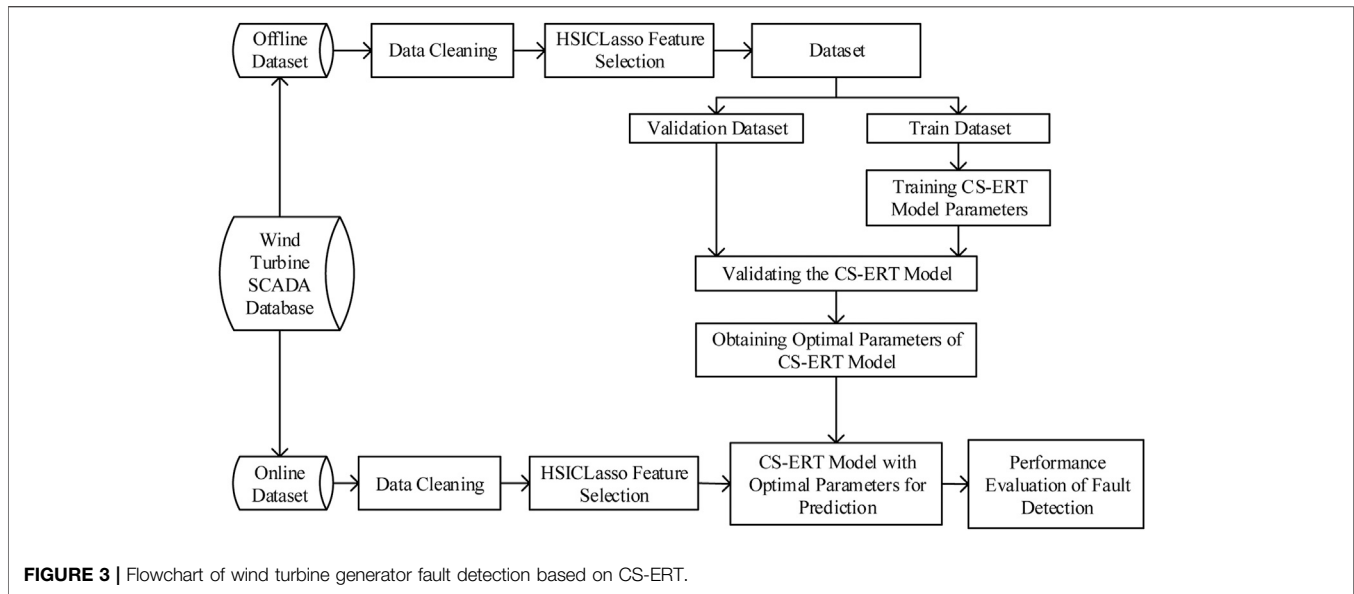
system in the fault state is predicted to be working normally, and 2) false alarm, where a system in the normal working state is predicted to be in a fault state. Clearly, the economic loss caused by missed detection is far greater than the loss caused by false alarms. CS-ERT can be used for fault detection of wind turbine generators to minimize the missing detection rate.

To provide a clearer structure, this section introduces three evaluation indicators for fault detection in advance. The missing detection rate, average misclassification cost, and gMean are abbreviated as MDR, AMC, and gMean, respectively. The evaluation index calculation equation is as follows.

$$MDR = FN/(TP + FN) \tag{12}$$

$$AMC = \frac{FN \cdot C_{FN} + FP \cdot C_{FP} + TP \cdot C_{TP} + TN \cdot C_{TN}}{FN + FP + TP + TN} \tag{13}$$

$$gMean = \sqrt{Recall^*Specificity} \tag{14}$$

Referring to **Eqs 12**, **13** TP represents true positives, FN represents false negatives, FP represents false positives, and TN represents true negatives. C_{FN}, C_{FP}, C_{TP}, and C_{TN} are the cost matrices. In **Eq 14**, $Recall = TP/(TP + FN)$ represents the probability of correct detection of fault samples, and $Specificity = TN/(TN + FP)$ represents the probability of correct detection of normal samples. The MDR refers to the ratio of the number of missed detection samples to the total number of samples when the wind turbine generator fails. The AMC considers not only the failure recognition rate but also the case where the misclassification cost is unequal. The gMean refers to the square root of the product of the failure detection rate and the normal detection rate, which is typically used as an evaluation of performance for class-imbalanced problems. The running time is closely related to the computational complexity of the algorithm. In this experiment, the running time is the mean value of the model's 10-fold cross-validation.

Figure 3 is a flowchart of a fault detection method based on the CS-ERT model. Offline wind turbine generator data are first

collected from the SCADA database, and data cleaning is performed. Data cleaning includes normalization and removal of missing and null values. Expert experience and the HSICLasso method are used to select features and generate feature subsets to avoid the impact of weakly correlated features and redundant features on the fault detection performance. In addition, the offline data are divided into a train dataset and a validation dataset. The train set is used to train the CS-ERT model. The validation dataset is used to adjust the hyperparameters of the model and initially evaluate the performance of the model. The optimal hyperparameters of the CS-ERT model are obtained through offline data, and the CS-ERT model with optimal parameters is established according to the optimal hyperparameters. In the last step, wind turbine data is collected online, and data preprocessing is performed. The processed online data are then used as the input for the optimal CS-ERT model, which is used to predict the real-time working status of wind turbine generators. If a fault is predicted, an alarm is triggered. Finally, the performance of the fault detection model on online data is analyzed according to **Eqs 12–14**.

The pseudocode of the large-scale wind turbine generator fault detection method based on CS-ERT is described as follows. **Algorithm 2** represents the process of obtaining the optimal CS-ERT model on the offline dataset. **Algorithm 3** realizes online fault detection of wind turbine generators.

Algorithm 2 Offline implementation of the CS-ERT fault detection method.

Input: Wind turbine SCADA dataset D_{off};

1 Perform data cleaning on dataset D_{off}, and normalize it using **(15)**
2 Use HSICLasso method for feature selection, divide D_{off} into the train dataset D_{train} and the validation dataset D_{vali}
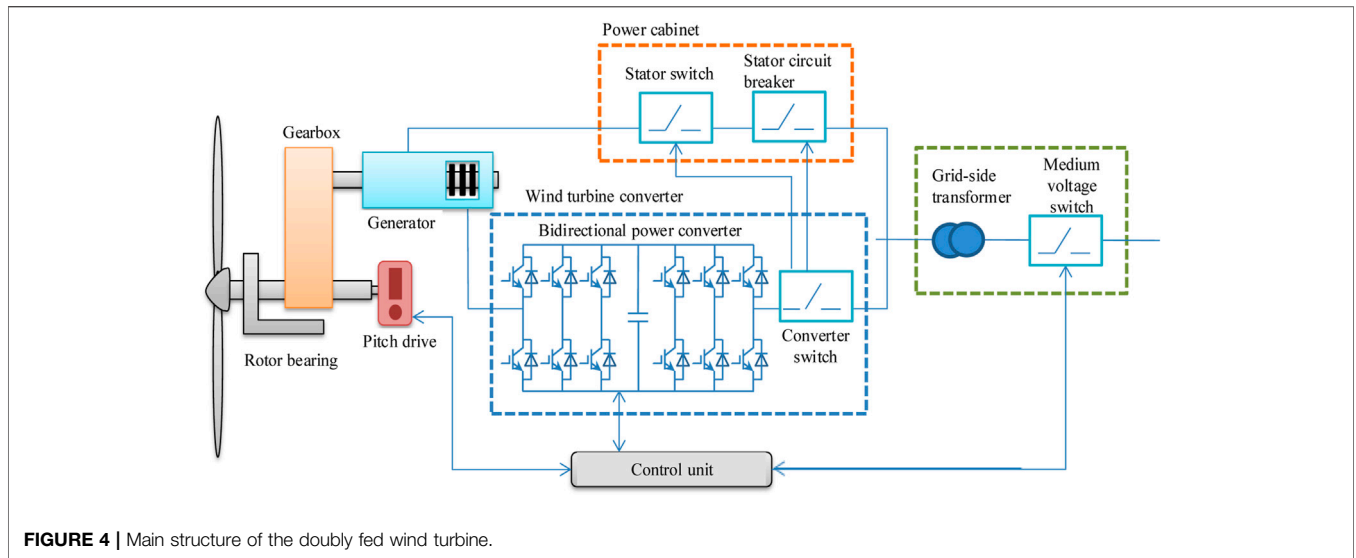3 The CS-ERT model M was established by train dataset D_{train}

FIGURE 4 | Main structure of the doubly fed wind turbine.

TABLE 2 | Wind turbine generator fault type and sensitive parameters.

Dataset	Fault type	Failure mechanism	Sensitive parameters
Data 1	Generator winding temperature error	The winding temperature exceeds 165°C for 5 s	Winding temperature, bearing temperature, gearbox bearing temperature, cooling air temperature
Data 2	Generator bearing temperature error	The temperature of the front axle D-END of bearing a or the rear axle N-END of bearing b is greater than 95°C for 5 s	Bearing temperature, winding temperature, cooling air temperature
Data 3	Generator fan pump heater protection error	Protection switch of generator fan or heater trips	Cold wind temperature, wind speed, bearing temperature
Data 4	Generator brush error	The generator carbon brush is badly worn and lasts 5 s	Bearing temperature, engine room temperature, winding temperature

4   Taking AMC as the evaluation index, the hyperparameters of the model are adjusted by the validation dataset D_{vali} to obtain the optimal hyperparameters θ

5   the CS-ERT model M⋆ with optimal parameters is established according to the optimal hyperparameters θ

Output: model M⋆

Algorithm 3 Online implementation of CS-ERT fault detection method.

Input: CS-ERT model M*, Online data D_{on};

1   Perform data cleaning and feature selection on the online data D_{on} and normalize it using **(15)** to obtain D'_{on}

2   Begin timing

3   Obtain model M* from **Algorithm 1**, and use M* and D'_{on} to predict the operating state of the wind turbine

4   **If** online data D_{on} is predicted to be a failure, **then**

5   Trigger alarms

6   **End if**

7   End timing

8   running time = Ending time - Beginning time

9   According to **(12)**–**(14)**, analyze the performance of the model M⋆ on the online data D_{on}

Output: Trigger alarms, missing detection rate, gMean, AMC and running time

EXPERIMENTAL ANALYSIS

In this section, data preprocessing is first performed on the data in the SCADA database. Then, the HSICLasso feature selection method extracts the main features and verifies the effectiveness of the method. Finally, the operating data of a 1.5 MW wind turbine in a wind farm in Shandong is used as experimental data, the effectiveness of the proposed method in the wind turbine generator fault detection problem is verified, and its superiority is emphasized by comparison.

Data Description and Data Cleaning

A generator fault detection experiment was conducted on a 1.5 MW doubly fed wind turbine in a wind farm in Shandong, China, which proved the effectiveness of the method. The main structure diagram of the doubly fed wind turbine is shown in **Figure 4**. Wind turbines are mainly composed of generators, gearboxes, pitch systems, etc. Fan blades convert wind energy into

mechanical energy, and generators convert mechanical energy into electrical energy. The electrical energy generated by the generator is integrated into the power grid through components such as converters, power cabinets and transformers.

The research object of this paper is a doubly-fed wind turbine generator. The doubly-fed wind turbine generator is mainly composed of a generator and a cooling system. The generator is composed of a stator, a rotor, a bearing, etc. The stator winding of the generators is directly connected to the power grid, and the rotor winding is connected to the power grid through a frequency converter. The equipment realizes variable-speed and constant-frequency power generation, which meets the requirements of the grid connection. Due to the AC excitation characteristics, the doubly-fed wind turbine can accurately adjust the output voltage of the generator by adjusting the excitation current. However, the power factor of doubly-fed wind turbines is low and requires additional power compensation. Therefore, in order to ensure the normal operation of the wind turbine, it is very important to perform fault detection on the generator.

Four kinds of defects (i.e., generator winding temperature error (F1), generator bearing temperature error (F2), generator fan pump heater protection error (F3) and generator brush error (F4) are generated in the actual operation of the generator. **Table 2** shows the fault mechanism and sensitive parameters of the four types of faults of the generator. The failure mechanism indicates the cause of the failure. Sensitive parameters are features that have a greater impact on faults through manual analysis. Wind turbine generator data are obtained from the SCADA database. Each sample has 213 features. The starting sampling point is half an hour before the start of a fault. The ending sampling point is half an hour after the end of a fault, and the data sampling interval is 2 s.

Data cleaning methods include missing value processing outlier value processing, and commonly used methods such as the deletion method and data repair method. To solve this problem, this paper adopts the deletion method to clean the data. This experiment was conducted on the Python 3.6 platform. The multi-duplicated samples and the samples with missing and null values were removed from the dataset. This method can not only reduce the influence of noise on the model performance, but also reduce the data diversity. Furthermore, features that have all 0 values were removed to reduce the dimensionality of the feature space and the model. To ensure the comparability of each feature, z-score normalization was used to eliminate the dimensionality of each feature. The value of each feature was transformed into a dimensionless value in the interval [0, 1].

$$x_i' = \frac{x_i - \mu}{\sigma} \qquad (15)$$

where x_i represents an attribute variable, μ is the mean of attribute x_i, and σ is the variance of attribute x_i. Each dataset of Data 1–Data 4 contains only normal samples and designated failure samples. Each dataset is normalized using the z-score method.

Experimental Results and Analysis

In accordance with the procedure of **Figure 3**, the HSICLasso method is used to select the features of the wind turbine generators

dataset. The HSICLasso feature selection method (Yamada et al., 2014) is a derived algorithm of the least absolute shrinkage and selection operator (lasso) (Tibshirani, 1996). We use non-negative constraints on α to improve the algorithm's ability to select effective features. In addition, the Gaussian kernel function and the triangular kernel function are used on the input vector and output vector of HSICLasso, respectively. We can incorporate structured outputs via kernels. Ren et al. (2020) proved that HSICLasso can effectively analyze the nonlinear relationship between multivariate time series. The F-norm replaces the L2-norm. The HSICLasso algorithm is defined as follows.

$$\min_{\alpha} \frac{1}{2} \left\| \overline{L} - \sum_{k=1}^{m} \alpha_k \overline{K}_{Frob}^{(k)} \right\|^2 + \lambda \|\alpha\|_1, \qquad (16)$$

$$\text{s.t.} \quad \alpha_1, \cdots, \alpha_k \geq 0,$$

where $\overline{L} = \Gamma L \Gamma$ and $\overline{K}^{(k)} = \Gamma K^{(k)} \Gamma$ are centered Gram matrices, and L and $K^{(k)}$ are both Gram matrices. $\Gamma = I_n - \frac{1}{n} 1_n 1_n^T$ is the centering matrix. I_n represents the n-dimensional identity matrix. 1_n represents an n-dimensional matrix with all elements of 1. The first term in the above expression represents the linear set of the input kernel matrix K and the fitting output kernel matrix L, and the last part represents the regular term. The above formula is further expressed as:

$$\min_{\alpha} \left(\frac{1}{2} \text{HSIC}(y, y) - \sum_{k=1}^{d} \alpha_k \text{HSIC}(u_k, y) \right.$$
$$\left. + \frac{1}{2} \sum_{k,l=1}^{d} \alpha_k \alpha_l \text{HSIC}(u_k, u_l) \right) + \lambda \|\alpha\|_1, \qquad (17)$$

where $\text{HSIC}(\cdot)$ is the Hilbert-Schmidt independence criterion (HSIC). $\text{HSIC}(u_k, y)$ represents a measure of independence based on the core. The higher the correlation between u_k and y is, the larger the value of $\text{HSIC}(u_k, y)$ and the smaller the result of **Eq 16**. The strong correlation between the feature and the output vector is ensured. The lower the correlation between u_k and u_l is, the smaller the value of $\text{HSIC}(u_k, u_l)$ and the smaller the result of **Eq 16**. Non-redundancy between features is guaranteed. In this way, the HSICLasso feature selection method is similar to a minimum redundancy maximum relevancy algorithm. The global optimal solution is effectively obtained by **Eq 17**. The method is extended to the high-dimensional feature selection problem. For massive high-dimensional data, the Gaussian kernel in HSIC Lasso is computationally expensive. Yamada et al. (2014) proposed a table lookup approach to reduce the computation time and memory size, reducing the computational complexity from $O(dn^2)$ to $O(dn + B)$, where d is the feature dimension, n is the number of samples, and B is the hyperparameter (we use $B = 20$ in our implementation).

The wind turbine generator dataset contains a large number of nonlinear and nonfunctional relationships. The high-dimensional feature space entails a large amount of calculation and low real-time performance for fault detection. The non-redundant features that have a strong correlation with the output vector are extracted based on expert experience and HSICLasso feature selection. Yamada et al. (2018) used the HSICLasso feature selection

TABLE 3 | HSICLasso feature selection results.

Feature	Tag	HSICLasso feature selection score			
		Data 1	Data 2	Data 3	Data 4
Generator winding temperature W	AW	0.009782277	**0.017599536**	**0.0615362**	0.025937422
Generator winding temperature V	AU	**0.66801703**	0.00001453	**0.10023695**	**0.010824516**
Generator bearing temperature B	BA	**0.033899892**	**0.2168639**	0.022490485	**0.36818618**
Generator cooling air temperature	BB	0.024351638	0.075802557	0.000024650	0.166879070
Gearbox shaft 1 temperature	AL	**0.063766375**	**0.069782838**	**0.037703969**	**0.052285645**
Gearbox shaft 2 temperature	AM	**0.017679539**	0.062894136	0.029099241	**0.14716054**
Gearbox inlet oil temperature	AN	**0.017236305**	**0.018463224**	**0.13771559**	**0.061054502**
30 s average wind speed	V	0.0052472115	0.00007895	**0.35092705**	0.0000784
Wind direction	H	**0.011375366**	0.00005421	**0.076073088**	0.0004825
Yaw deviation	AJ	**0.047915235**	**0.11275150**	0.0000251	0.0000854
Gearbox oil temperature	AO	0.0065203002	**0.004965703**	0.027400982	0.0001254
Nacelle temperature	BE	0.000003215	0.000695982	0.0027198601	**0.10510261**
Nacelle outdoor temperature	BD	0.022269854	0.00000234	0.0000154	**0.026983585**

The features corresponding to the bold values are selected for model construction.

TABLE 4 | Dataset description.

Dataset	Original feature number	Feature number after feature selection	Number of samples	Sample imbalance
Data 1	213	8	5,354	18.83
Data 2	213	8	3,554	24.48
Data 3	213	8	3,568	12.31
Data 4	213	8	3,585	8.98

TABLE 5 | Optimization of the cost parameters of CS-ERT for 4 datasets.

Dataset	C_{FP}	C_{FN}	C_{TP}	C_{TN}
Data 1	1	50	0	0
Data 2	1	74	0	0
Data 3	1	43	0	0
Data 4	1	40	0	0

method for ultrahigh-dimensional big data nonlinear feature selection and achieved good results. The features with the top 8 are selected as inputs for the wind turbine generator fault detection model. The feature selection results are as follows.

According to **Table 3**, the winding temperature, bearing temperature, and cooling air temperature are strongly correlated in the four fault datasets, consistent with the failure mechanism and sensitive parameters in **Table 2**. Therefore, the HSICLasso feature selection method can accurately extract attribute subsets from wind turbine generator data. The feature dimensions, fault types, and sample imbalance of the dataset after applying the HSICLasso feature selection method are shown in **Table 4**.

TheCS-ERT model has 4 hyperparameters: the number of decision trees M, the minimum number of leaf nodes n_{node}, and two misclassification cost parameters C_{FP} and C_{FN}. Because the model has many hyperparameters, the optimal hyperparameters are difficult to determine. Hyperparameter optimization methods include the gray wolf optimizer method (Long et al., 2018), butterfly optimization algorithm (Long et al., 2021; Long et al., 2021), and grid search method. We input the obtained low-dimensional feature set into the cost-sensitive extreme random forest classifier optimized by

the grid optimization method to realize automatic fault identification of wind turbines. Four key parameters (n_{node}, M, C_{FN} and C_{FP}) of the CS-ERT classifier are selected through a grid search method using 10-fold cross-validation. To simplify the experimental process, C_{FN} is regarded as 1. The variation range of the parameter C_{FP} is [0, 200]. As shown in **Table 5**, the results of the cost parameters of the CS-ERT model are optimized for each dataset.

Comparison Among Different Methods

In this subsection, comparative studies among different methods are performed to verify the efficacy and superiority of the proposed algorithm. According to the procedure mentioned in *Experimental Results and Analysis* , different features are extracted to form four feature sets of four faults, and then these feature sets are input to the model to identify wind turbine generator faults. To evaluate the effectiveness of the CS-ERT fault detection method, three points should be emphasized. First, nonredundant features with strong correlation are selected via the HSICLasso method to reduce the feature dimensionality. Then, the parameters of different classifiers are selected based on grid optimization for each dataset. Finally, the experiment compares RF (Hsu et al., 2020; Jia et al., 2018) with XGBoost (Zhang et al., 2018), ERT (Janssens et al., 2016), CS-EDT (base classifier for CS-ERT), MetaCost (Kim et al., 2012), AdaCost (Yin et al., 2013), CSForest (Siers and Islam, 2015), and CS-ERT. To eliminate the contingency of the experiment, all methods use the 10-fold cross-validation method. During performance analysis, MDR, gMean, AMD and Time are used to evaluate the performance of the model.
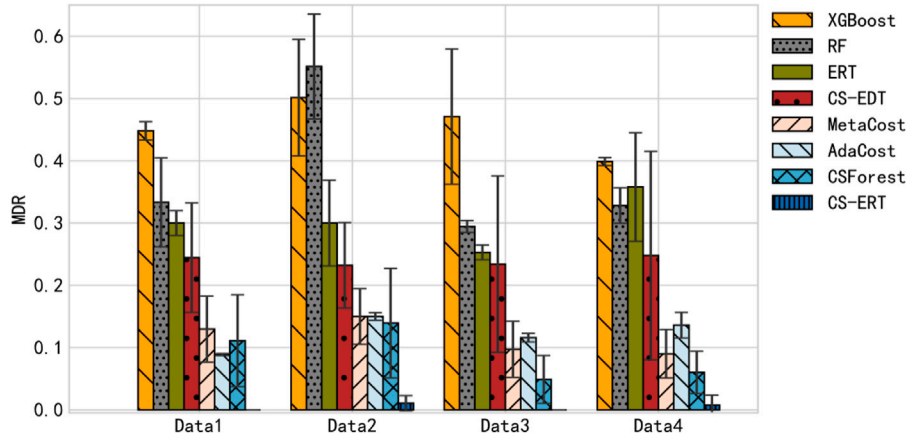
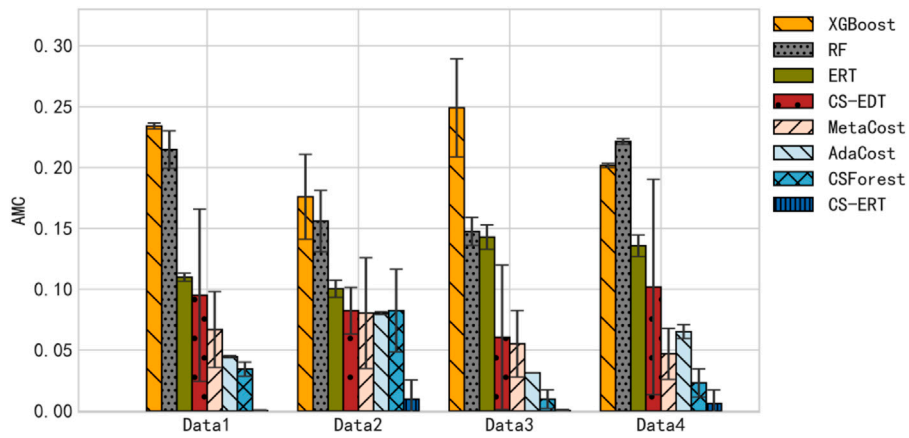**FIGURE 5 |** Comparison of MDR for the eight algorithms on the four datasets.

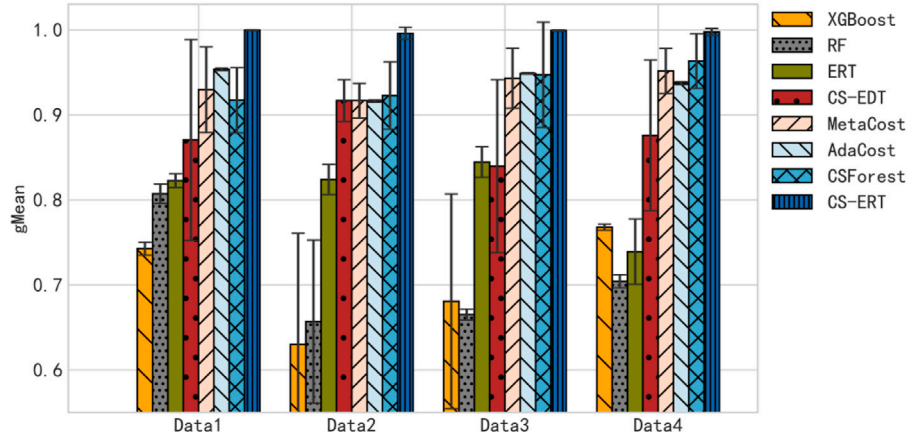**FIGURE 6 |** Comparison of AMC for the eight algorithms on the four datasets.

FIGURE 7 | Comparison of gMean for the eight algorithms on the four datasets.
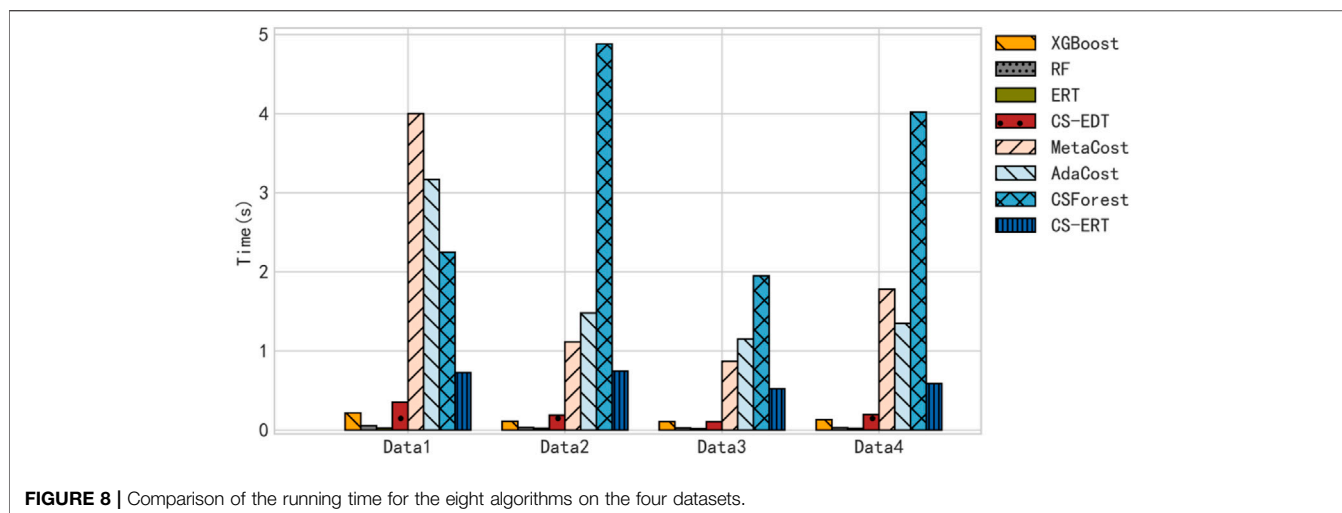
FIGURE 8 | Comparison of the running time for the eight algorithms on the four datasets.

A higher gMean and lower MDR, AMC, and Time indicate better performance of the fault detection method.

Figures 5, **6** represent the diagnosis results of different fault detection methods for the four faults of the wind turbine generator. As observed in **Figures 5**, **6**, the MDR (average MDR is 0.45%) and AMC (average AMC is 0.41%) of the proposed method are much lower than those of other fault detection methods in the four fault types. We can also see that the missing detection rate and average misclassification cost of traditional fault detection methods are higher than those based on cost-sensitive fault detection methods. The MDR and AMC of ERT, RF and XGBoost methods are all greater than 20 and 10%. Moreover, missing detection rate and AMCs below 20 and 10%, respectively, are attained by the cost-sensitive methods. This means that cost-sensitive fault detection methods give a higher misclassification cost to minority classes when dealing with imbalanced data than traditional methods (ERT, RF, XGBoost). Furthermore, it helps reduce the false negative rate and average misclassification cost of fault detection methods. Namely, the superiority of the cost-sensitive method is confirmed through experimental analysis.

The average MDR and average AMC of CS-EDT are 23.54 and 8.07%, respectively. The performance of CS-ERT is obviously better than that of CS-EDT, which proves the necessity and advantage of adopting the ensemble algorithm. In addition, **Figures 5**. **6** show that the average MDR and average AMC of the CS-ERT method are 0.45 and 0.41%, respectively, on the four types of faults of wind turbine generators. The average MDRs of other cost-sensitive methods—namely, MetaCost, AdaCost and CSForest—are 11.67, 15.18, and 9.14%, respectively, and the average AMCs are 6.24, 6.37, and 3.97%, respectively. The results demonstrate the efficacy and benefits of the CS-ERT classifier. The HSICLasso feature extraction method is proved to effectively reduce the impact of weakly correlated features and redundant features on model performance. This proves the superiority of the proposed method for fault detection on wind turbine generators.

To further analyze the effectiveness of the proposed method, gMean is used as an indicator to evaluate the performance of the above fault detection method. The experimental results are shown in **Figure 7**. The gMean value is composed of the missing detection rate and the false alarm rate. It is mostly used for model performance evaluation when addressing imbalanced data and can effectively evaluate the performance of the model. The experimental results show that the average gMean of the proposed method is 99.68%, which is higher than the gMean value of the other 7 methods (70.48, 70.83, 73.15, 83.36, 93.53, 92.06, and 93.92%). This shows that while the method improves the failure detection rate, it also maintains a high false alarm rate. There are several reasons that could explain this: First, compared with the standard ERT algorithm, CS-ERT considers the cost of misclassification to improve the detection accuracy of fault classes. Then, compared with CSForest, the proposed method uses complete features for training and can make more reliable decisions. In addition, it reduces the interference of weakly correlated features and improves model performance. Therefore, we can conclude that the proposed method achieves the best classification performance in this experiment.

In wind turbine generator fault detection, the running time of the model is also an important index. How to meet both high fault detection performance and short running time has always been a research hotspot (Barrios Aguilar et al., 2020; Falehi, 2020).The objective function of CS-ERT only focuses on fault detection performance compared to the multiple objective optimization approach. The advantage of the running time is reflected in its unique structure. The above methods are used to process the generator fault dataset and record its running time. The result is shown in **Figure 8**. Each method sets hyperparameters with the goal of optimal performance. The average calculation time of the CS-ERT method is 0.646 s, which is shorter than the calculation times of MetaCost, AdaCost and CSForest (1.941, 1.787, and 3.425 s, respectively). The running time on the 4 datasets is better than those of these three algorithms. The reason for this result is that CS-ERT randomly selects a value for each feature, reducing one level of looping in the model. The average computation time of CS-EDT is 0.21 s, which is lower than that of the CS-ERT algorithm,

which verifies that the ensemble algorithm increases the computational complexity while improving the model performance. The average calculation times of the XGBoost, RF and ERT methods are 0.141, 0.036, and 0.021 s, respectively. Although the calculation speed of traditional algorithms is faster, they do not consider the cost of misclassification. This leads to a low failure detection rate, which seriously affects the economic benefits of the wind turbine.

In summary, the CS-ERT-based wind turbine generator fault detection method has the performance of low MDR, low AMC and high gMean in four kinds of generator faults. Compared with MetaCost, AdaCost and CSForest, the proposed method has a faster calculation advantage.

CONCLUSION

A generator is one of the energy conversion components of a doubly fed wind turbine. The long time operation results in the generator fault data are far less than the normal data. To deal with this problem, we proposed a novel method (CS-ERT) for wind turbine generator fault detection with imbalanced data in this paper. First, the HSICLasso feature selection method is used to select strongly correlated non-redundant features to form feature subsets to reduce the dimension of the dataset. Then, the fault detection model of doubly-fed wind turbine generators based on CS-ERT is established. Finally, the feature subset is used as the input of the model, and the working state of the generator is taken as the output of the model to detect the actual working condition of the generator. A practical application of a wind farm in Shandong, China, verified the effectiveness of CS-ERT. The results showed that the CS-ERT method outperformed other fault detection methods (XGBoost, RF, ERT, CS-EDT, MetaCost, AdaCost and CSForest) in MDR, AMC and gMean. The MDR of the proposed method is over 30% higher than that of ERT. The gMean of CS-ERT is more than 15% higher than that of CS-EDT, proving the advantages of the ensemble algorithm. Compared with MetaCost, AdaCost and CSForest, the proposed method has better computational speed and fault detection performance. The proposed method has good fault detection performance for wind turbine generators. We believe that CS-ERT is applicable not only to wind turbine generator fault detection but also to other large-scale industrial fault detection applications. However, the proposed method has some constraints in the detection of hybrid faults and the optimization of hyperparameters, and is sensitive to the SCADA data quality. In future work, we can further study the following:

- There are many hyperparameters in CS-ERT. It is difficult to obtain a global optimal solution by tuning these hyperparameters. The optimization algorithm is combined with the CS-ERT algorithm to achieve the optimal parameters of the adaptive search model.
- For multiple fault problems, we can extend the CS-ERT algorithm from binary classification to multi-classification in the future.
- A data-driven approach applies to low noise data. Poor quality data in SCADA systems will inevitably affect the performance of the model. In the future, we need to further consider the cleaning method for poor quality data and the impact of noise on the model.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding authors.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

FUNDING

REFERENCES

Artigao, E., Martín-Martínez, S., Honrubia-Escribano, A., and Gómez-Lázaro, E. (2018). Wind Turbine Reliability: A Comprehensive Review towards Effective Condition Monitoring Development. *Appl. Energ.* 228, 1569–1583. doi:10.1016/j.apenergy.2018.07.037

Bakri, A. E., and Boumhidi, I. (2018). Fuzzy Model-Based Faults Diagnosis of the Wind Turbine Benchmark. *Proced. Comput. Sci.* 127, 464–470. doi:10.1016/j.procs.2018.01.144

Barrios Aguilar, M. E., Coury, D. V., Reginatto, R., and Monaro, R. M. (2020). Multi-objective PSO Applied to PI Control of DFIG Wind Turbine under Electrical Fault Conditions. *Electric Power Syst. Res.* 180, 106081. doi:10.1016/j.epsr.2019.106081

Breiman, L. (1996). Bagging Predictors. *Mach Learn.* 24 (2), 123–140. doi:10.1007/bf00058655

Buda, M., Maki, A., Mazurowski, M. A., and Mazurowski (2018). A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks. *Neural Networks* 106, 249–259. doi:10.1016/j.neunet.2018.07.011

Cheki, M., Jazayeri-Rad, H., and Karimi, P. (2016). Enhancing the Noise Tolerance of Fault Diagnosis System Using the Modified Adaptive Boosting Algorithm. *J. Nat. Gas Sci. Eng.* 29, 303–310. doi:10.1016/j.jngse.2015.12.029

Chen, J., Pan, J., Li, Z., Zi, Y., and Chen, X. (2016). Generator Bearing Fault Diagnosis for Wind Turbine via Empirical Wavelet Transform Using Measured Vibration Signals. *Renew. Energ.* 89, 80–92. doi:10.1016/j.renene.2015.12.010

Cho, S., Gao, Z., and Moan, T. (2018). Model-based Fault Detection, Fault Isolation and Fault-Tolerant Control of a Blade Pitch System in Floating Wind Turbines. *Renew. Energ.* 120, 306–321. doi:10.1016/j.renene.2017.12.102

Falehi, A. D. (2020). An Innovative Optimal RPO-FOSMC Based on Multi-Objective Grasshopper Optimization Algorithm for DFIG-Based Wind Turbine to Augment MPPT and FRT Capabilities. *Chaos Solitons Fractals* 130, 109407. doi:10.1016/j.chaos.2019.109407

Fernandez-Canti, R. M., Blesa, J., Tornil-Sin, S., and Puig, V. (2015). Fault Detection and Isolation for a Wind Turbine Benchmark Using a Mixed Bayesian/Set-Membership Approach. *Annu. Rev. Control.* 40, 59–69. doi:10.1016/j.arcontrol.2015.08.002

Gao, Q. W., Liu, W. Y., Tang, B. P., and Li, G. J. (2018). A Novel Wind Turbine Fault Diagnosis Method Based on Intergral Extension Load Mean Decomposition Multiscale Entropy and Least Squares Support Vector Machine. *Renew. Energ.* 116, 169–175. doi:10.1016/j.renene.2017.09.061

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely Randomized Trees. *Mach Learn.* 63 (1), 3–42. doi:10.1007/s10994-006-6226-1

Ghahremani, E., and Kamwa, I. (2016). Local and Wide-Area PMU-Based Decentralized Dynamic State Estimation in Multi-Machine Power Systems. *IEEE Trans. Power Syst.* 31 (1), 547–562. doi:10.1109/tpwrs.2015.2400633

Gopinath, R, Santhosh Kumar, C, Ramachandran, K. I., Upendranath, V., and Sai Kiran, P. V. R. (2016). Intelligent Fault Diagnosis of Synchronous Generators. *Expert Syst. Appl.* 45 (1), 142–149. doi:10.1016/j.eswa.2015.09.043

Habibi, H., Howard, I., and Simani, S. (2019). Reliability Improvement of Wind Turbine Power Generation Using Model-Based Fault Detection and Fault Tolerant Control: A Review. *Renew. Energ.* 135, 877–896. doi:10.1016/j.renene.2018.12.066

Hamidreza, M., Mehdi, S., Hooshang, J., and Aliakbar, N. (2014). Reconstruction Based Approach to Sensor Fault Diagnosis Using Auto-Associative Neural Networks. *J. Cent. South Univ.* 21 (06), 161–169., no. v.

Hsu, J.-Y., Wang, Y.-F., Lin, K.-C., Chen, M.-Y., and Hsu, J. H.-Y. (2020). Wind Turbine Fault Diagnosis and Predictive Maintenance through Statistical Process Control and Machine Learning. *Ieee Access* 8, 23427–23439. doi:10.1109/access.2020.2968615

Ibrahim, R. K., Watson, S. J., Djurovic, S., and Crabtree, C. J. (2018). An Effective Approach for Rotor Electrical Asymmetry Detection in Wind Turbine DFIGs. *IEEE Trans. Ind. Electron.* 65 (11), 8872–8881. doi:10.1109/tie.2018.2811373

Janssens, O., Noppe, N., Devriendt, C., Walle, R. V. d., and Hoecke, S. V. (2016). Data-driven Multivariate Power Curve Modeling of Offshore Wind Turbines. *Eng. Appl. Artif. Intelligence* 55, 331–338. doi:10.1016/j.engappai.2016.08.003

Jia, R., Ma, F., Dang, J., Liu, G., and Zhang, H. (2018). Research on Multidomain Fault Diagnosis of Large Wind Turbines under Complex Environment. *Complexity* 2018, 1–13. doi:10.1155/2018/2896850

Joshuva, A., and Sugumaran, V. (2017). Fault Diagnosis for Wind Turbine Blade through Vibration Signals Using Statistical Features and Random forest Algorithm. *Int. J. Pharm. Technol.* 9 (1), 28684–28696.

Judge, F., McAuliffe, F. D., Sperstad, I. B., Chester, R., Flannery, B., Lynch, K., et al. (2019). A Lifecycle Financial Analysis Model for Offshore Wind Farms. *Renew. Sustain. Energ. Rev.* 103, 370–383. doi:10.1016/j.rser.2018.12.045

Kandukuri, S. T., Klausen, A., Karimi, H. R., and Robbersmyr, K. G. (2016). A Review of Diagnostics and Prognostics of Low-Speed Machinery towards Wind Turbine Farm-Level Health Management. *Renew. Sustain. Energ. Rev.* 53, 697–708. doi:10.1016/j.rser.2015.08.061

Kim, A., Oh, K., Jung, J.-Y., and Kim, B. (2018). Imbalanced Classification of Manufacturing Quality Conditions Using Cost-Sensitive Decision Tree Ensembles. *Int. J. Comput. Integrated Manufacturing* 31 (8), 701–717. doi:10.1080/0951192X.2017.1407447

Kim, J., Choi, K., Kim, G., and Suh, Y. (2012). Classification Cost: An Empirical Comparison Among Traditional Classifier, Cost-Sensitive Classifier, and MetaCost. *Expert Syst. Appl.* 39 (4), 4013–4019. doi:10.1016/j.eswa.2011.09.071

Lei, J., Liu, C., and Jiang, D. (2019). Fault Diagnosis of Wind Turbine Based on Long Short-Term Memory Networks. *Renew. Energ.* 133, 422–432. doi:10.1016/j.renene.2018.10.031

Li, C., Sanchez, R.-V., Zurita, G., Cerrada, M., Cabrera, D., and Vásquez, R. E. (2016). Gearbox Fault Diagnosis Based on Deep Random forest Fusion of Acoustic and Vibratory Signals. *Mech. Syst. Signal Process.* 76-77, 283–293. doi:10.1016/j.ymssp.2016.02.007

Li, M., Yu, D., Chen, Z., Xiahou, K., Ji, T., and Wu, Q. H. (2019). A Data-Driven Residual-Based Method for Fault Diagnosis and Isolation in Wind Turbines. *IEEE Trans. Sustain. Energ.* 10 (2), 895–904. doi:10.1109/tste.2018.2853990

Li, Z.-M., Gui, W.-H., and Zhu, J.-Y. (2019). Fault Detection in Flotation Processes Based on Deep Learning and Support Vector Machine. *J. Cent. South. Univ.* 26 (9), 2504–2515. doi:10.1007/s11771-019-4190-8

Liming, S., and Bo, Y. (2020). Nonlinear Robust Fractional-Order Control of Battery/SMES Hybrid Energy Storage Systems. *Power Syst. Prot. Control.* 48 (22), 76–83. doi:10.1016/j.energy.2019.116510

Lomax, S., and Vadera, S. (2013). A Survey of Cost-Sensitive Decision Tree Induction Algorithms. *ACM Comput. Surv.* 45 (2), 1–35. doi:10.1145/2431211.2431215

Long, W., Jiao, J., Liang, X., and Tang, M. (2018). An Exploration-Enhanced Grey Wolf Optimizer to Solve High-Dimensional Numerical Optimization. *Eng. Appl. Artif. Intelligence* 68, 63–80. doi:10.1016/j.engappai.2017.10.024

Long, W., Jiao, J., Liang, X., Wu, T., Xu, M., and Cai, S. (2021). Pinhole-imaging-based Learning Butterfly Optimization Algorithm for Global Optimization and Feature Selection. *Appl. Soft Comput.* 103, 107146. doi:10.1016/j.asoc.2021.107146

Long, W., Wu, T., Xu, M., Tang, M., and Cai, S. (2021). Parameters Identification of Photovoltaic Models by Using an Enhanced Adaptive Butterfly Optimization Algorithm. *Energy* 229, 120750. doi:10.1016/j.energy.2021.120750

Longting, C., Guanghua, X., Qing, H., and Xun, Z. (2019). Learning Deep Representation of Imbalanced SCADA Data for Fault Detection of Wind Turbines. *Measurement* 139, 370–379. doi:10.1016/j.measurement.2019.03.029

Lu, H., Yang, L., Yan, K., Xue, Y., and Gao, Z. (2017). A Cost-Sensitive Rotation forest Algorithm for Gene Expression Data Classification. *Neurocomputing* 228, 270–276. doi:10.1016/j.neucom.2016.09.077

Malik, H., and Mishra, S. (2016). Proximal Support Vector Machine (PSVM) Based Imbalance Fault Diagnosis of Wind Turbine Using Generator Current Signals. *Energ. Proced.* 90, 593–603. doi:10.1016/j.egypro.2016.11.228

Marugan, A. P., Marquez, F. P. G., Perez, J. M. P., and Ruiz-Hernandez, D. (2018). A Survey of Artificial Neural Network in Wind Energy Systems. *Appl. Energ.* 228, 1822–1836. doi:10.1016/j.apenergy.2018.07.084

Masnadi-Shirazi, H., and Vasconcelos, N. (2011). Cost-sensitive Boosting. *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (2), 294–309. doi:10.1109/tpami.2010.71

Mingzhu, T., et al. (2020). An Improved LightGBM Algorithm for Online Fault Detection of Wind Turbine Gearboxes. *Energies* 13 (4), 807. doi:10.3390/en13040807

Nami, S., and Shajari, M. (2018). Cost-sensitive Payment Card Fraud Detection Based on Dynamic Random forest and K -nearest Neighbors. *Expert Syst. Appl.* 110, 381–392. doi:10.1016/j.eswa.2018.06.011

Pan, X., Ju, P., Wu, F., and Jin, Y. (2017). Hierarchical Parameter Estimation of DFIG and Drive Train System in a Wind Turbine Generator. *Front. Mech. Eng.* 12 (no. 3), 367–376. doi:10.1007/s11465-017-0429-y

Quiroz, J. C., Mariun, N., Mehrjou, M. R., Izadi, M., Misron, N., and Mohd Radzi, M. A. (2018). Fault Detection of Broken Rotor Bar in LS-PMSM Using Random Forests. *Measurement* 116, 273–280. doi:10.1016/j.measurement.2017.11.004

Ren, W., Li, B., and Han, M. (2020). A Novel Granger Causality Method Based on HSIC-Lasso for Revealing Nonlinear Relationship between Multivariate Time

Series. *Physica A: Stat. Mech. its Appl.* 541, 123245. doi:10.1016/j.physa.2019.123245

Shahriari, S. A. A., Mohammadi, M., and Raoofat, M. (2020). Enhancement of Low-Voltage Ride-Through Capability of Permanent Magnet Synchronous Generator Wind Turbine by Applying State-Estimation Technique. *Compel-the Int. J. Comput. Math. Electr. Electron. Eng.* 39 (2), 363–377. doi:10.1108/compel-11-2018-0478

Siers, M. J., and Islam, M. Z. (2015). Software Defect Prediction Using a Cost Sensitive Decision forest and Voting, and a Potential Solution to the Class Imbalance Problem. *Inf. Syst.* 51, 62–71. doi:10.1016/j.is.2015.02.006

Song, D., Chang, Q., Zheng, S., Yang, S., Yang, J., and Hoon Joo, Y. (2021). Adaptive Model Predictive Control for Yaw System of Variable-Speed Wind Turbines. *J. Mod. Power Syst. Clean Energ.* 9 (1), 219–224. doi:10.35833/mpce.2019.000467

Tan, M. (1993). Cost-Sensitive Learning of Classification Knowledge and its Applications in Robotics. *Mach Learn.* 13 (1), 7–33. doi:10.1007/bf00993101

Teng, W., Cheng, H., Ding, X., Liu, Y., Ma, Z., and Mu, H. (2018). DNN-based Approach for Fault Detection in a Direct Drive Wind Turbine. *Iet Renew. Power Generation* 12 (10), 1164–1171. doi:10.1049/iet-rpg.2017.0867

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *J. R. Stat. Soc. Ser. B (Methodological)* 58 (1), 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x

Qi, Z. X., Wang, H., Zhou, X., Li, J. Z., and Gao, H. (2019). Cost-sensitive Decision Tree Induction on Dirty Data. *J. Softw.* 30 (no. 3), 604–619. doi:10.13328/j.cnki.jos.005691

Willis, D. J., Niezrecki, C., Kuchma, D., Hines, E., Arwade, S. R., Barthelmie, R. J., et al. (2018). Wind Energy Research: State-Of-The-Art and Future Research Directions. *Renew. Energ.* 125, 133–154. doi:10.1016/j.renene.2018.02.049

Yamada, M., Jitkrittum, W., Sigal, L., Xing, E. P., and Sugiyama, M. (2014). High-Dimensional Feature Selection by Feature-Wise Kernelized Lasso. *Neural Comput.* 26 (1), 185–207. doi:10.1162/NECO_a_00537

Yamada, M., Tang, J., Lugo-Martinez, J., Hodzic, E., Shrestha, R., Saha, A., et al. (2018). Ultra High-Dimensional Nonlinear Feature Selection for Big Biological Data. *IEEE Trans. Knowl. Data Eng.* 30 (7), 1352–1365. doi:10.1109/tkde.2018.2789451

Yang, J., et al. (2021). Review of Control Strategy of Large Horizontal-axis Wind Turbines Yaw System. *Wind Energy* 24 (2), 97–115. doi:10.1002/we.2564

Yin, Q.-Y., Zhang, J.-S., Zhang, C.-X., and Liu, S.-C. (2013). An Empirical Study on the Performance of Cost-Sensitive Boosting Algorithms with Different Levels of Class Imbalance. *Math. Probl. Eng.* 2013, 1–12. doi:10.1155/2013/761814

Yu, F., Li, G., Chen, H., Guo, Y., Yuan, Y., and Coulton, B. (2018). A VRF Charge Fault Diagnosis Method Based on Expert Modification C5.0 Decision Tree. *Int. J. Refrigeration* 92, 106–112. doi:10.1016/j.ijrefrig.2018.05.034

Zelenkov, Y. (2019). Example-dependent Cost-Sensitive Adaptive Boosting. *Expert Syst. Appl.* 135, 71–82. doi:10.1016/j.eswa.2019.06.009

Zeng, B., Guo, J., Zhu, W., Xiao, Z., Yuan, F., and Huang, S. (2019). A Transformer Fault Diagnosis Model Based on Hybrid Grey Wolf Optimizer and LS-SVM. *Energies* 12 (21), 4170–4218. doi:10.3390/en12214170

Zhang, D., Qian, L., Mao, B., Huang, C., Huang, B., and Si, Y. (2018). A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost. *Ieee Access* 6, 21020–21031. doi:10.1109/access.2018.2818678

Zhang, S. (2018). Multiple-scale Cost Sensitive Decision Tree Learning. *World Wide Web* 21 (6), 1787–1800. doi:10.1007/s11280-018-0619-5