Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

A Fault Diagnosis Method for Wind Turbine Gearbox under High Noise Based on Adaptive Probability Random Forest

Lipeng Cui[1,2], Yu Yu[3], Mingzhu Tang[3,✉], Zhao Wang[4], Jianyou Ouyang[4]

[1] *School of Automation and Electrical Engineering, Tianjin University of Technology and Education, Tianjin, 300222, China*
[2] *School of Electronic Information and Automation, Tianjin Light Industry Vocational Technical College, Tianjin, 300350, China*
[3] *School of Energy and Power Engineering, Changsha University of Science & Technology, Changsha, Hunan, 410114, China*
[4] *Department of Energy Technology, Changsha Electric Power Technical College, Changsha, Hunan, 410131, China*

ABSTRACT

A fault diagnosis method for wind turbine gearbox based on adaptive probability random forest is proposed to address the issue of noise pollution in SCADA data of wind turbine gearbox. Firstly, SMOTE oversampling is used to balance sample categories, and then CART is trained and classified by constructing multiple balanced subsets. The sample error rate represents the weight of sample ambiguity, and the label uncertainty is determined. Monte Carlo simulation is used to calculate the mean distribution of features, which is fused with each sample instance to obtain the uncertainty of sample features. Utilizing adaptive labels and sample uncertainties as inputs to probabilistic random forest can enhance the ability to manage feature noise and label noise, thereby improving the robustness of fault diagnosis. Conduct an experimental evaluation using the SCADA dataset of wind turbine gearbox. The results show that this model outperforms other methods in terms of false alarm rate, false alarm rate, and F1 rating metrics when dealing with missing values, Gaussian noise, and label noise in the dataset, as compared to other methods. This method is of great significance for improving the accuracy and robustness of wind turbine gearbox fault diagnosis.

Keywords: Wind turbines, Gearbox, Uncertainty, Fault diagnosis, Probabilistic Random Forest

1. Introduction

Wind energy is an infinitely renewable energy source that does not generate greenhouse gas emissions and is beneficial for mitigating global warming and climate change. Using wind energy can improve energy security, reduce dependence on foreign oil and gas, create employment opportunities, and promote economic development [1-2]. A wind turbine is a device that converts wind energy into electrical energy, consisting of two major subsystems: mechanical and electrical components [3].

✉ Corresponding author.
E-mail address: tmz@csust.edu.cn (M. Tang).
Received 25 March 2024; Revised 05 June 2024; Accepted 15 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-408

Mechanical components convert wind power into rotational power and transmit it to generators. In contrast, electrical components convert rotational power into AC or DC power and output it to grid-connected or energy storage systems. The two subsystems are coordinated and regulated through control systems [4]. Among them, the gearbox plays a crucial role in wind turbines. It not only impacts the output power of the wind turbine but also influences its operational efficiency. The primary function of the gearbox is to convert the low-speed rotation of the wind turbine blades into the high-speed rotation needed by the generator, facilitating the conversion of wind energy into electrical energy.

These faults affect wind turbines' performance and lifespan, increase maintenance costs and downtime losses, and even cause safety accidents and environmental pollution. Therefore, diagnosing gearbox faults and implementing timely and effective maintenance measures are crucial for ensuring wind turbines' efficient and stable operation [5-6].

One method to achieve a real-time diagnosis of gearbox faults is to monitor and analyze the SCADA system of wind turbines in real-time. SCADA system is a monitoring and control system that records data from various system sensors and corresponding working condition labels during the operation of wind turbines. By analyzing the characteristic data associated with the labels, wind turbine fault type and operational status can be determined in real time [7]. However, due to factors such as sensor errors, environmental conditions, and communication interference, the data collected by SCADA systems often contains noise or missing values. In addition, manually set labels often exhibit subjectivity and category ambiguity, which means labels may also contain noise. Regarding the noise issue in the application of SCADA systems for fault detection, Zhang [8] designed a robust kernel principal component analysis (SR-RKPCA) model based on subspace reconstruction to address the significant fluctuations in SCADA data. This model can improve the stability of wind turbine fault detection and has strong non-linear feature extraction capabilities. The experimental results demonstrate the robustness and effectiveness of the model. Wang [9] proposed a method based on cascaded SAE abnormal state monitoring and LightGBM abnormal state classification to detect early faults or abnormal states of wind turbine components. This method utilizes the anti-interference characteristics of cascaded SAE to detect early abnormal states of wind turbines. Shi [10] proposed a method that combines a density clustering algorithm and normal power interval estimation to handle abnormal data. This method is combined with the XGBoost algorithm to construct a technique based on multi-feature monitoring parameter information fusion. This approach can provide early warning of the fault status of wind turbines. The literature above presents various anti-interference feature extraction methods for noise issues. However, a lack of targeted optimization design for different types of noise would enable the model to resist noise adaptively.

There is often a class imbalance in the SCADA fault dataset of wind turbines, and an effective and convenient method is oversampling. SMOTE is a synthetic minority class oversampling technique Chawla [11] et al. proposed in 2002. Its main idea is to analyze minority class samples and generate new ones through interpolation based on their k-nearest neighbor samples. However, SMOTE also has some limitations, such as the possibility of generating noisy data or leading to blurred boundaries. Therefore, subsequent research will make some improvements and extensions to SMOTE to enhance its robustness and adaptability, such as SMOTEDNN [12], K-Means-SMOTE-ENN [13], AWSMOTE [14], SP-SMOTE [15], IW-SMOTE [16], etc. However, the noise reduction strategies of these SMOTE variants cannot fundamentally solve the noise problem. Collaborating the noise caused by the data itself and the noise brought about by solving class imbalance problems, indirectly exploring the data distribution, and guiding the model is a new approach.

In addition, the utilization of ensemble algorithms can mitigate the issue of class imbalance to some extent. Random forest is an effective ensemble algorithm that constructs multiple decision trees by randomly sampling training data and features. It combines the results of various base classifiers through voting or averaging [17]. However, the performance of random forests may be affected when

dealing with data containing label noise and feature noise. These noises can result in inaccurate decision tree partitioning, reducing the model's generalization ability and prediction accuracy. Therefore, researchers propose a series of enhanced random forest algorithms to improve their robustness to noisy data. For example, Zhou [18] proposed an improved noise robust random forest (NRRF) model based on the global multi-class noise tolerance loss function, which can optimize the training process of decision trees by utilizing noise information. Agjee [19] proposed a robust classification algorithm for oblique random forest ridge regression (oRF-RR) and applied it to hyperspectral datasets. Experimental results showed that the algorithm is not affected by noisy spectral conditions. Wang [20] proposed a multi-scale adaptive switched random forest (MARF) for the inevitable noise and multi-scale characteristics in laser scanning of leg detectors, which can be applied to human detection and tracking systems. Reis [21] established the Probabilistic Random Forest (PRF) algorithm to address label noise, feature noise, and missing values in data, treating features and labels as probability distribution functions. The above methods explore the improvement of the robustness of random forest algorithms from different perspectives, among which the PRF algorithm achieves collaborative processing of multiple types of noise in data and applies it to multiple fields, such as bioactivity predictions [22], credit card fraud detection [23], etc. However, the label and feature uncertainty calculation still face some challenges. Therefore, to apply the PRF algorithm to fault diagnosis of wind turbines, enhance the robustness of the model, and integrate it with the SMOTE oversampling method. The SMOTE oversampling adaptive probability random forest (SMOTE-APRF) algorithm is proposed to achieve collaborative processing of SCADA data noise and SMOTE-associated noise. The content of this article is arranged as follows:

1) In the second section, the specific details of the SMOTE-APRF algorithm are introduced.

2) In the third section, the text introduces the gearbox data of wind turbines and the gearbox fault diagnosis method based on SMOTE-APRF.

3) In the fourth section, performance was tested using manual noise addition and compared with other methods.

4) In the fifth section, summarize the methods proposed in this article.

2. Methods and Materials

2.1. SMOTE Oversampling

SMOTE (Synthetic Minority Oversampling Technique) is an oversampling method used to address class imbalance issues. The class imbalance problem is when the number of minority class samples is much smaller than the number of majority class samples. This imbalance can result in the classifier's insufficient learning ability for minority class samples and easily lead to bias towards the majority class. SMOTE balances the dataset by synthesizing new minority class samples to enhance the classifier's ability to learn from minority class samples.

For a minority class sample X_i, assume that its K nearest neighbors are $X_1, X_2, ..., X_k$. The synthesized new sample is:

$$X_{new} = X_i + rand(0,1) \times (X_k - X_i) \tag{1}$$

where, $rand(0,1)$ is a random number between 0 and 1, used to control the position of the synthesized sample. This formula allows K new samples to be synthesized for each minority class sample to balance the dataset.

2.2. Random Forest

The Random Forest (RF) algorithm is an ensemble learning method that constructs multiple decision trees during training. Decision trees are non-parametric models represented by a tree graph for classification and regression tasks. The construction of a decision tree starts with the entire training

dataset and a root node. The algorithm searches for the "best" partition, which combines a set of features and thresholds to maximize the degree of separation between two categories in a node. Gini impurity is typically used as the criterion for determining the "best" partition, and its calculation formula is:

$$Gini(n) = 1 - \left(P_{n,A}{}^2 + P_{n,B}{}^2 \right) \tag{2}$$

Among them, $P_{n,A}$ and $P_{n,B}$ represents the score of A and B class objects in the group of node n (class probability).

2.3. Probabilistic Random Forest

PRF is an enhanced random forest classification algorithm that leverages the uncertainty information of input data to enhance the model's resistance to noise. The difference between it and traditional random forests lies in the form of input data. The traditional random forest inputs a set of predetermined feature-label pairs:

$$(x_1, y_1), ..., (x_n, y_n) \tag{3}$$

These pairs follow an unknown relationship $h : X \to Y$. Then model this relationship and predict the y value of a given x. And PRF receives a set of quadruplets with uncertainty:

$$(x_1, y_1, \Delta x_1, \Delta y_1), ..., (x_n, y_n, \Delta x_n, \Delta y_n) \tag{4}$$

where Δx and Δy represent the uncertainty of features and labels. PRF will simultaneously consider the accuracy of each feature and label to model the relationship *h*.

The input in RF is a supervised classification task that involves two types of uncertainty. One type is feature uncertainty, i.e., representing the feature values' measurement error. Another type is label uncertainty, which represents the classification confidence of the object type. PRF considers features and labels as probability distributions rather than deterministic values to handle uncertainty in data. Features are transformed into probability density functions (PDFs) with expected values and variances, while labels are converted into probability mass functions (PMFs) with probability allocation. In PRF, feature uncertainty can affect how objects split in tree nodes. In traditional decision trees, each object is divided into left or right branches based on splitting criteria. In the presence of feature uncertainty, splitting is probabilistic. That is to say, at each node, each object may be divided into two branches with a certain probability, as illustrated in Figure 1. The object in the PRF tree can be split with a certain probability across all tree nodes. Therefore, the expected value of the object in node n is used to calculate impurities:

$$P_{n,A} \to \overline{P}_{n,A} = \frac{\sum_{i \in n} \pi_i(n) \cdot P_{i,A}}{\sum_{i \in n} \pi_i(n)} \tag{5}$$

$$P_{n,B} \to \overline{P}_{n,B} = \frac{\sum_{i \in n} \pi_i(n) \cdot P_{i,B}}{\sum_{i \in n} \pi_i(n)} \tag{6}$$

The corrected Gini impurity is converted to:

$$Gini(n) \to \overline{Gini}(n) = 1 - \left(\overline{P}_{n,A}{}^2 + \overline{P}_{n,B}{}^2 \right) \tag{7}$$

Define the cost function as the weighted average of the corrected impurities at two nodes:

$$\overline{Gini}(n,r) \times \frac{\sum_{i \in (n,r)} \pi_i(n,r)}{\sum_{i \in n} \pi_i(n)} + \overline{Gini}(n,l) \times \frac{\sum_{i \in (n,l)} \pi_i(n,l)}{\sum_{i \in n} \pi_i(n)} \tag{8}$$

Uncertainty in labeling can impact the classification results of objects in tree nodes. The randomness caused by PMF differs from the user randomness injected in RF, as it reflects the information content relevant to classification tasks. This label uncertainty can propagate into the splitting criteria, such as Gini impurities, and affect the prediction model during the tree construction. To reduce the complexity of the computational model, PRF introduces a selective splitting scheme. Splitting is only performed by setting a probability threshold p_{th} when the probability of the object being divided into a certain node is greater than this threshold. This approach can reduce the running time. The PRF prediction model is implemented by taking the weighted average of the prediction probabilities of all leaf nodes. When the input uncertainty approaches zero, the prediction of PRF converges to the traditional random forest, and the predictions are obtained through majority voting.

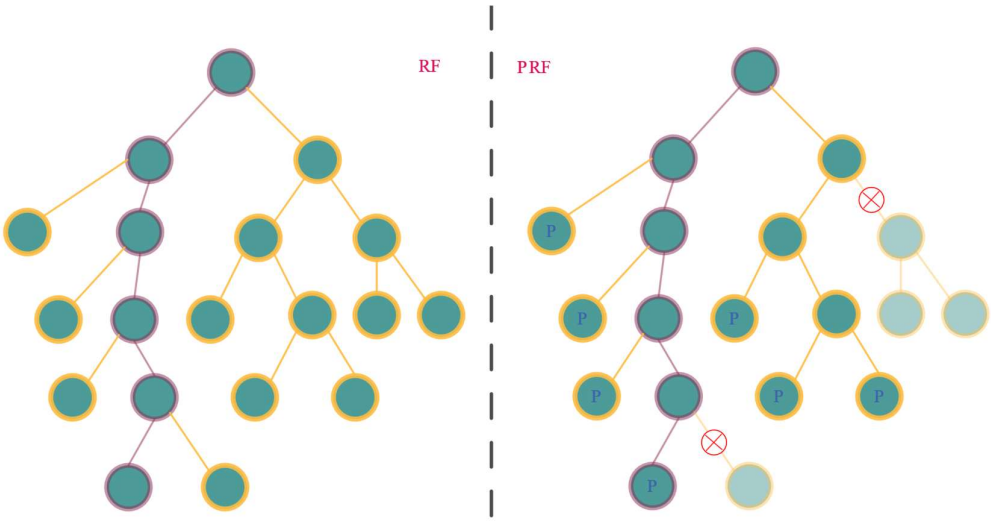


Fig. 1. Explanation of object propagation in RF and PRF trees.

2.4. SMOTE-APRF

Considering the issues of class imbalance and noise in the data, we combined SMOTE oversampling with PRF and introduced adaptive uncertainty calculation to propose the SMOTE-APRF model. The process is illustrated in Figure 2. In the SMOTE-APRF model, SMOTE directly samples the data categories, establishing a foundation for data balance in the following processing steps. The adaptive uncertainty calculation module calculates Δx and Δy on the balanced data, and finally, it uses x, Δx, Δy as inputs to the probabilistic random forest, thereby improving the robustness and applicability of the probabilistic random forest.

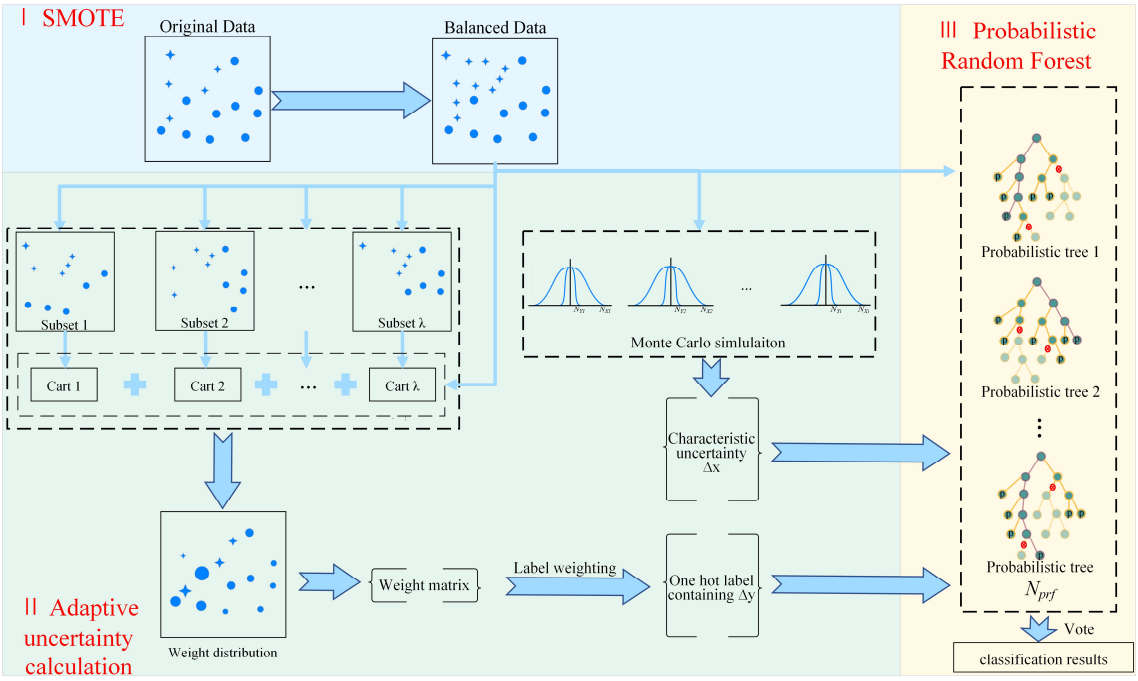


Fig. 2. SMOTE-APRF algorithm flowchart.

2.4.1. Calculation of Label Uncertainty. The uncertainty label of a probabilistic random forest refers to the fact that in the training dataset, the true category of each sample is not completely determined. Instead, it is represented by a probability distribution to indicate its possible values. In medical image classification, the label of each image may be provided by multiple experts, but inconsistencies or errors among experts may exist. Therefore, the label of each image can be represented by a probability vector to indicate its likelihood of belonging to different categories. The uncertainty of labels can be calculated through human professional knowledge and judgment ability. Still, it will require more time and resources, as well as the development of reasonable voting rules and evaluation standards. Therefore, it is necessary to study methods for calculating uncertainty labels based on data-driven integration strategies.

To explore the distribution of data, an ensemble algorithm is used to indirectly calculate the uncertainty labels in PRF. Sample weights are calculated based on the distribution of confusion information output from multiple Classification and Regression Trees (CARTs) trained on balanced subsets. Then, the weights are used to refine the uncertainty labels, resulting in the one-hot form of the uncertainty labels. Specifically, taking binary classification as an example, APRF uses the following steps to generate uncertainty labels, with corresponding pseudocodes shown in Algorithm 1.

Step 1: Perform multiple self-sampling on the original dataset to obtain λ Balanced subsets, with N_{subset} of samples for each balanced subset:

$$N_{subset} = d \times N_{one\text{-}class} \tag{9}$$

Among them, $N_{one\text{-}class}$ is the total number of single class samples; d is the segmentation coefficient.

Step 2: Train a CART model on each balanced subset and use the model to classify all samples in the original dataset to obtain a confusion information distribution, representing the result of each sample being classified into different categories under the model.

Step 3: Add up the output results of all CART models to obtain a total confusion information distribution, with a size of (N, λ). This matrix represents the number of times each sample is classified into different categories under all models.

Step 4: Calculate each sample's error rate based on the confusion information distribution. The error

rate refers to the proportion of times a sample is classified into an incorrect category across all models, divided by the total number of classification attempts, creating a weight matrix. The visualization diagram of weights is shown in Figure 3, where the size of the label on the sample represents its weight information. The larger the size, the higher the error rate and it can be considered a sample with blurred labels. The weight matrix w is represented by a one-dimensional matrix with a size of $(N,1)$.

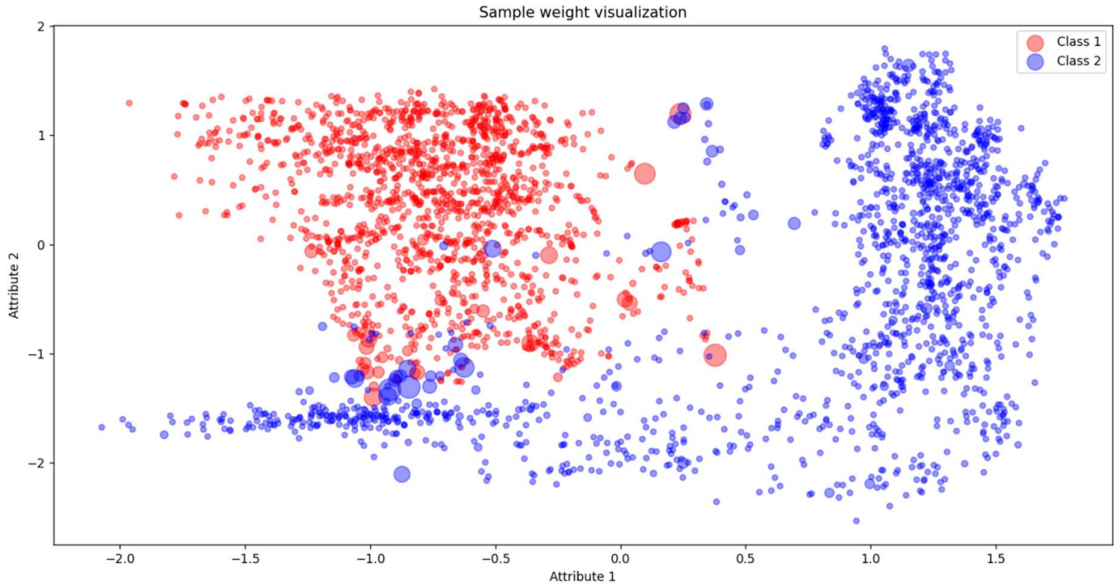


Fig. 3. Sample Weight Distribution Map Derived from Confused Information.

Step 5: After obtaining the weight matrix, it needs to be normalized before being used to weight deterministic labels. The purpose of weighting is to adjust the values in the deterministic label vector of each sample based on its error rate in different categories, transforming it into an uncertainty label. The mathematical expression used to represent the calculation process can be written as:

$$\Delta y = f_w\left(y_{one-hot}, w\right) \tag{10}$$

Among them, $y_{one-hot}$ is the One Hot encoding of the deterministic label y; $f_w(y,w)$ is a weighted function, specifically:

$$f_w\left(y,w\right)_{ij} = \begin{cases} 1-0.5w_i \ \ if \ \ y_{ij}=1 \\ 0.5w_i \ \ \ if \ \ y_{ij}=0 \\ y_i \ \ \ \ \ if \ \ w_i=0 \end{cases} \tag{11}$$

Algorithm1: Calculation of label uncertainty integrated Cart algorithm.	
Input: an balanced data set $\Phi = \{(x_1, y_1),(x_2, y_2),...,(x_N, y_N)\}$ and its one-hot label $y_{one\text{-}hot}$, where N denotes the number of instances, and $y_i \in \{+,-\}$; the parameter d is the division factor; the parameter λ which indicates the number of training subsets; the single cart model c; The w matrix is the label uncertain weight.	
Output: Δy; which is the One-Hot code for the uncertainty label.	
1:	Divide Φ into two sets, $\Phi+$ only contains the minority instances, while $\Phi-$ contains all majority instances;
2:	For i=1: λ:
3:	randomly extract $d*N/2$ instances from $\Phi+$ and $\Phi-$, respectively;
4:	train the model c on the extracted samples and classify all the data Φ;
5:	End

6:	Integrate the classification results of all λ classifiers, calculate the error rate of each sample, and obtain the weight matrix;
7:	For *j*=1: *N*:
8:	calculate Δy using Equation (11)
9:	End
10:	Return the modified labels Δy.

2.4.2. Calculation of Feature Uncertainty. The input of a probabilistic random forest includes not only the numerical values of the data points but also the uncertainty Δx of the data points. Δx is a measure that represents the difference between data points and feature means, reflecting the distribution of data points in the feature space. The larger Δx, the greater the difference between the data point and the feature mean, and the higher the feature uncertainty. The smaller the uncertainty, the smaller the difference between the data points and the feature mean, and the lower the feature uncertainty. Feature uncertainty is an important type of information that can help probabilistic random forest models better handle data changes and noise and improve the model's classification ability.

Δx can be analyzed through manual observation or instrument accuracy. Different data sources and generation processes may have varying error models and distributions. However, this method is challenging to quantify as it considers the correlation or dependency between data, leading to incomplete or inconsistent estimation results. Statistical methods leverage the information within the data itself, independent of external assumptions or knowledge. They can accommodate various data types and distributions without being constrained by specific error models or assumptions, rendering them more flexible and robust. Therefore, the following introduces the Monte Carlo simulation method to calculate uncertainty.

Monte Carlo simulation is a numerical calculation method based on random numbers. It estimates the outcomes of complex probability distributions and functional relationships. Calculating uncertainty in Monte Carlo simulation involves utilizing the simulation outcomes to assess the reliability and precision of statistical measures. Assuming that the input variable X follows a known or unknown probability distribution $P(X)$, the output variable Y is a function of the input variable X, $Y = f(X)$. Assuming you want to estimate the mean and variance of the output variable Y:

$$\mu_Y = E(Y) \tag{12}$$

$$\sigma_Y^2 = Var(Y) \tag{13}$$

Define the number of times N_s for Monte Carlo simulation, which determines the number of random samples X_N and output samples Y_N to be generated, as well as the accuracy and stability of the simulation results. Generally speaking, the larger the number of simulations, the closer the simulation results are to the true values, but it also increases the calculation time and space.

For each simulation process, loop through each feature. Assuming the number of features is M, for each feature, randomly select a sample value from the data to form a random sample $X_M^y = \left\{X_1^y, X_2^y, ..., X_M^y\right\}$. The bootstrap method is utilized here, involving placing the sample values back into the dataset after each extraction. This ensures that each extraction is independent and identically distributed. Then, calculate the mean and variance of the random samples and obtain the statistics for each feature as follows:

$$\Theta_M = \left\{\hat{\mu}_Y, \hat{\delta}_Y^2\right\} \tag{14}$$

Calculate the mean and standard deviation of multiple simulation results for each feature and use them to estimate the standard error and confidence interval of the mean for each feature:

$$\overline{SE}(\hat{\mu}_Y) = \frac{\hat{\delta}_Y}{\sqrt{M}} \tag{15}$$

$$\overline{CI}(\hat{\mu}_Y) = \hat{\mu}_Y \pm E \tag{16}$$

Among them, E is the half-width of the feature mean within the confidence interval, and the calculation formula is as follows:

$$E = Z_{\frac{\alpha}{2}} \overline{SE}(\hat{\mu}_Y) \tag{17}$$

Among them, $Z_{\frac{\alpha}{2}}$ is the critical value of the standard normal distribution.

E can represent the uncertainty of each feature, but it must be combined with each sample instance to derive the final Δx. Δx is used to represent the uncertainty of each data point on a specific feature, indicating the distance from the overall mean. The greater the distance, the greater the uncertainty; the smaller the distance, the lower the uncertainty. If the data point falls within the confidence interval, then its distance from the population mean is small, and the uncertainty is low. If the data point falls outside the confidence interval, then its distance from the population mean is large, and the uncertainty is high.

So, using the following formula to correct *E*, the Δx_{ij} of the i-th feature and j-th sample is:

$$\Delta x_{ij} = \frac{\max\left(\left|x_{ij} - \overline{CI}(\hat{\mu}_Y)_{iL}\right|, \left|x_{ij} - \overline{CI}(\hat{\mu}_Y)_{iU}\right|\right)}{\max\left(\left|\overline{CI}(\hat{\mu}_X)_{iU} - \overline{CI}(\hat{\mu}_Y)_{iL}\right|, \left|\overline{CI}(\hat{\mu}_X)_{iU} - \overline{CI}(\hat{\mu}_Y)_{iU}\right|\right)} E_i \tag{18}$$

where x_{ij} is the value of the point; $\overline{CI}(\hat{\mu}_Y)_{iL}$ is the lower limit of the confidence interval for the i-th feature; $\overline{CI}(\hat{\mu}_X)_{iU}$ is the upper limit of the confidence interval for the i-th feature; $\overline{CI}(\hat{\mu}_X)_{iU}$ is the upper limit of the confidence interval for the distribution of the i-th feature's raw data sample; E_i is the half width of the feature mean of the i-th feature within the confidence interval.

Finally, utilizing Δx as an input for the adaptive probability random forest can enhance the accuracy and robustness of classification.

2.5. Fault diagnosis of wind turbine gearbox based on SMOTE-APRF

2.5.1. SMOTE-APRF. The main components of wind turbines include wind turbines, gearbox, generators, frequency converters, contactors, circuit breakers, grid-side transformers, etc. The main structure is shown in Figure 4. Wind turbines achieve optimized operation and safety protection through integrated control systems, which include wind speed and direction detection, pitch control, generator control, grid connection control, etc. Among them, the gearbox is an important component of variable speed transmission in wind turbines, primarily consisting of gears, bearings, and transmission shafts. The gearbox can transform the low-speed rotation of the wind turbine into the high-speed rotation required by the generator, thereby enhancing power generation efficiency. Gearbox are prone to damage, with faults frequently occurring in gears and bearings. Therefore, it is necessary to measure the temperature, vibration, oil and other parameters of the gearbox through sensors and transmit them to the integrated control system. The integrated control system will analyze the parameters and determine the type of fault in the gearbox.

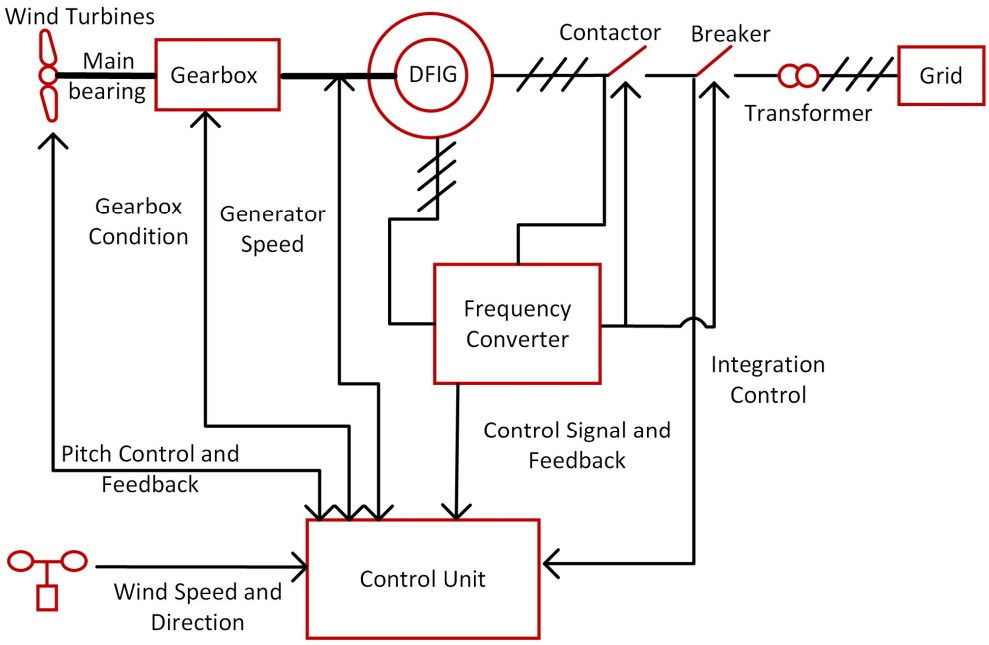


Fig. 4. The main structure of a typical wind turbine.

To verify the effectiveness of the proposed gearbox fault diagnosis model, a case study was conducted using the SCADA dataset of a 1.5MW wind turbine in a Chinese wind farm. Two datasets from different wind turbines were used to avoid randomness in the experiment. Each type of fault data was combined with normal data to form two binary fault datasets. Dataset1's fault type is "Gearbox input shaft 1 temperature fault", with 4585 normal samples and 2603 fault samples. Dataset2's fault type is "Gearbox oil pool heater malfunction", having 2803 normal samples and 2251 fault samples. Each dataset contains several characteristic variables, such as wind speed, gearbox inlet oil temperature, and gearbox input shaft temperature, along with a target variable indicating whether the system is normal or faulty. The ratio of training to testing sets in the dataset is 7:3.

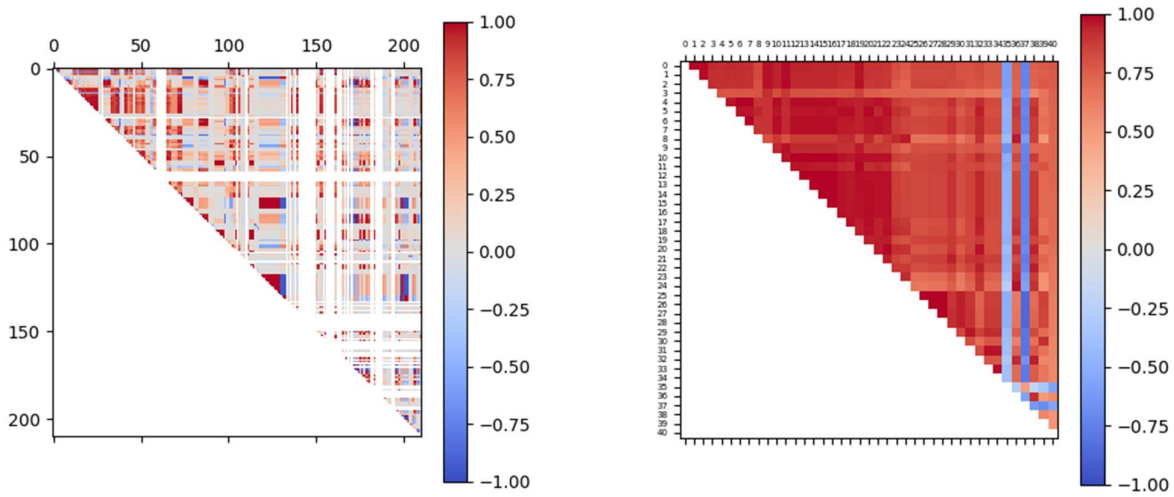
2.5.2. Data preprocessing. Data preprocessing is a crucial step in machine learning that can enhance the quality and availability of data, thus improving classification performance. Data preprocessing mainly includes data cleaning and data standardization. Data cleaning involves removing or correcting incorrect or invalid information in the original data, such as outliers, duplicate values, and missing values. These pieces of information can affect the distribution and statistical characteristics of the data, potentially resulting in inaccurate or unstable classification outcomes. Data standardization refers to transforming data from different ranges or units into a unified form. This process can eliminate dimensional and scale differences in data and enable effective comparison and calculation between different features. Convert wind turbine fault data into a dataset with a mean of 0 and a variance of 1, specifically:

$$x' = \frac{x - \mu}{\delta} \tag{19}$$

Among them, x is the original feature value, μ is the feature mean, and δ is the characteristic standard deviation.

2.5.3. Feature selection. Feature selection is a method of reducing data dimensionality, which enhances the performance and efficiency of a model by filtering out the most relevant features related to the target variable. To evaluate the correlation between different features and fault features in the dataset, the Pearson correlation coefficient is used as a measurement indicator. The Pearson

correlation coefficient is a statistical measure that reflects the degree of linear correlation between two variables. It ranges from -1 to 1, where -1 represents a complete negative correlation, 1 represents a complete positive correlation, and 0 represents no correlation. Based on expert experience, the top 40 most relevant features were selected as the experimental dataset to simplify the complexity of model calculations. Figure 5 shows the Pearson correlation coefficient matrix between different features in the original and feature selection datasets. The graph shows that some features exhibit a strong positive or negative correlation with fault features, while others are nearly unrelated to fault features.



(a) (b)

Fig. 5. (a) Correlation matrix of raw data; (b) Correlation matrix selected selection.

2.5.4. Hyperparameter settings. In the proposed SMOTE-APRF model, there are three types of hyperparameter settings: (i) the hyperparameters required to calculate Δx, including the number of Monte Carlo simulations N_s, (ii) the hyperparameters required to calculate Δy, including the number of balanced subsets λ, segmentation coefficient d, and (iii) the hyperparameters of the probability random forest, including the number of probability trees N_{prf}, probability threshold p_{th}, and other parameters. In this experiment, artificial experience and grid search methods were used. The hyperparameter settings are as follows: N_s is 5000, λ is 400, d is 0.4, N_{prf} is 50, p_{th} is 0.05.

3. Result discussions

A series of experiments were conducted to verify the diagnostic performance of the SMOTE-APRF fault diagnosis model for wind turbine SCADA data and the effectiveness of this method in different noise environments. The experiment used a PC configured with an Intel i5 2.9GHz CPU, 16GB RAM, GTX 1650 GPU, and a 64-bit Windows 11 operating system, using Python as the programming language.

3.1. Evaluating indicators

In order to evaluate the classification accuracy of the model, this article uses the false alarm rate (FAR), missing alarm rate (MAR), and F1-score (F1) as evaluation indicators. The definitions of FAR and MAR are as follows:

$$FAR = \frac{FP}{FP + TN} \tag{20}$$

$$MAR = \frac{FN}{FN + TP} \tag{21}$$

Among them, FP is false positive, indicating the number of samples with detected abnormalities but no actual abnormalities; TN is true negative, indicating the number of samples with no detected abnormalities and no actual abnormalities; FN is false negative, indicating the number of samples with no detected abnormalities but actual abnormalities; TP is true positive, indicating the number of samples that have been detected as abnormal and are actually abnormal.

The definitions of accuracy and recall are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

The definition of F_1 is as follows:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{24}$$

3.2. Comparative experiment

In the SCADA system of wind turbines, there is data presence, label noise, and feature noise. Therefore, whether the classification model has good noise resistance will significantly impact the performance and results of the entire monitoring system. Therefore, comparative experiments were conducted with other algorithms to verify further the proposed model's noise resistance in class-imbalanced noisy datasets. In the comparison algorithm, SMOTE-PRF, SMOTE-RF, SMOTE-LightGBM, kmeans-SMOTE-APRF, kmeans-SMOTE-PRF, and APRF fault diagnosis models are selected for performance comparison in detection. Each experiment is repeated 10 times to avoid randomness, and the average value is taken as the experimental result. Artificial noise is tested in the following three ways.

3.2.1. Randomly empty. Due to the fact that the SCADA data of wind turbines often have values with certain characteristics that are empty, the test data was randomly left blank to test the performance of various models under null noise. When the vacancy rate is 80%, the test results are shown in Table 1. Taking Dataset1 as an example, the F_1 performance of the model varies with the vacancy rate, as shown in Figure 6.

Table 1. Performance Comparison of Various Models with a Vacancy Rate of 80%.

Dataset	Model	F1-Score	FAR	MAR
Dataset1	OURS	**0.9910±0.0022**	**0.0043±0.0007**	0.0095±0.0044
	SMOTE-PRF	0.9885±0.0021	0.0107±0.0022	0.0017±0.0009
	SMOTE-RF	0.4464±0.0492	0.2616±0.0109	**0.0000±0.0000**
	SMOTE-LightGBM	0.5016±0.0000	0.2490±0.0000	0.0064±0.0000
	Kmeans-SMOTE-APRF	0.9906±0.0016	0.0088±0.0012	0.0013±0.0011
	Kmeans-SMOTE-PRF	0.9885±0.0012	0.0105±0.0023	0.0022±0.0033
	APRF	0.9803±0.0077	0.0075±0.0021	0.0241±0.0180
Dataset2	OURS	**0.9991±0.0008**	**0.0011±0.0009**	0.0005±0.0009
	SMOTE-PRF	0.9989±0.0000	0.0018±0.0000	**0.0000±0.0000**
	SMOTE-RF	0.6179±0.0344	0.3015±0.0141	0.0009±0.0018
	SMOTE-LightGBM	0.3340±0.0000	0.3850±0.0000	**0.0000±0.0000**
	Kmeans-SMOTE-APRF	0.9987±0.0004	0.0021±0.0007	**0.0000±0.0000**

	Kmeans-SMOTE-PRF	0.9989±0.0000	0.0018±0.0000	**0.0000±0.0000**
	APRF	0.9984±0.0011	0.0014±0.0007	0.0014±0.0018

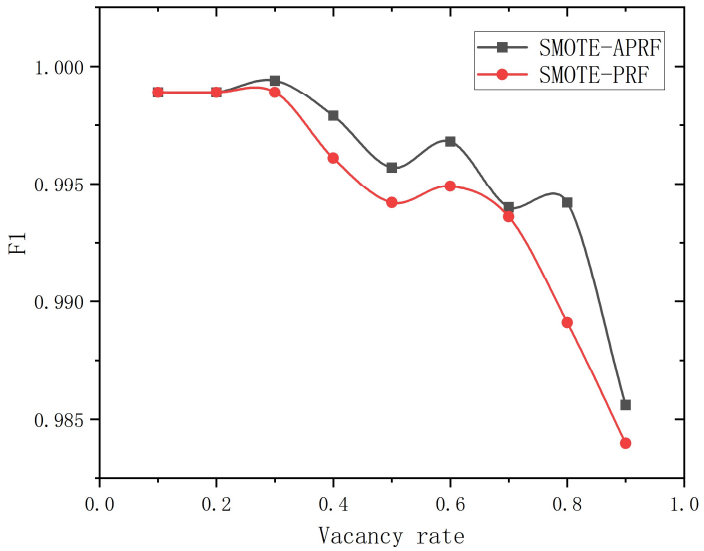


Fig. 6. F1 Performance and Vacancy Relationship Curve.

3.2.2. Gaussian noise. Gaussian noise is a type of random noise that follows a normal distribution and can be used to simulate uncertainty or interference in real data. The proportion of Gaussian noise addition can be expressed as signal-to-noise ratio (SNR), which is the ratio of signal power to noise power, usually expressed in decibels (dB). After adding Gaussian noise with SNR=15 to the test data, the test results are shown in Table 2.

Table 2. Performance Comparison of Various Models at SNR=15.

Dataset	Model	F1-Score	FAR	MAR
Dataset1	OURS	**0.9462±0.0039**	**0.0120±0.0035**	0.0804±0.0101
	SMOTE-PRF	0.9379±0.0027	0.0223±0.0016	0.0774±0.0065
	SMOTE-RF	0.9319±0.0054	0.0399±0.0040	**0.0544±0.0068**
	SMOTE-LightGBM	0.8811±0.0000	0.0543±0.0000	0.1300±0.0000
	Kmeans-SMOTE-APRF	0.9436±0.0047	0.0156±0.0046	0.0787±0.0096
	Kmeans-SMOTE-PRF	0.9404±0.0035	0.0234±0.0059	0.0710±0.0047
	APRF	0.9414±0.0045	0.0333±0.0092	0.0751±0.0078
Dataset2	OURS	**0.9829±0.0013**	0.0123±0.0027	0.0185±0.0017
	SMOTE-PRF	0.9823±0.0019	0.0162±0.0042	0.0149±0.0056
	SMOTE-RF	0.9819±0.0016	**0.0089±0.0025**	0.0248±0.0051
	SMOTE-LightGBM	0.9706±0.0000	0.0263±0.0000	0.0250±0.0000
	Kmeans-SMOTE-APRF	0.9825±0.0019	0.0131±0.0018	0.0144±0.0018
	Kmeans-SMOTE-PRF	0.9827±0.0023	0.0169±0.0024	**0.0131±0.0022**
	APRF	0.9827±0.0035	0.0166±0.0045	0.0135±0.0059

3.2.3. Label noise. To test the model's performance in the presence of label noise, random label permutation is performed on the training data through the permutation rate parameter. When the replacement rate is 30%, the test results are shown in Table 3. Taking Dataset1 as an example, the F1 performance of the model varies with the displacement rate, as shown in Figure 7.

Table 3. Performance Comparison of Various Models at a Replacement Rate of 30%.

Dataset	Model	F1-Score	FAR	MAR
Dataset1	OURS	**0.7806±0.0082**	0.2109±0.0137	0.0890±0.0078
	SMOTE-PRF	0.7332±0.0062	0.2238±0.0078	0.1613±0.0056
	SMOTE-RF	0.7324±0.0126	0.0931±0.0078	0.3513±0.0128
	SMOTE-LightGBM	0.7805±0.0000	**0.0615±0.0000**	0.3078±0.0000
	Kmeans-SMOTE-APRF	0.7711±0.0066	0.2071±0.0105	0.1118±0.0065
	Kmeans-SMOTE-PRF	0.7286±0.0054	0.2167±0.0063	0.1776±0.0113
	APRF	0.7624±0.0106	0.2439±0.0129	**0.0826±0.0096**
Dataset2	OURS	**0.7840±0.0110**	0.2247±0.0120	0.1703±0.0108
	SMOTE-PRF	0.7341±0.0034	0.2095±0.0054	0.2649±0.0053
	SMOTE-RF	0.7271±0.0075	0.2144±0.0061	0.2704±0.0114
	SMOTE-LightGBM	0.7785 ±0.0119	**0.1562 ±0.0128**	0.2551±0.0113
	Kmeans-SMOTE-APRF	0.7766±0.0049	0.2455±0.0138	**0.1662±0.0127**
	Kmeans-SMOTE-PRF	0.7231±0.0117	0.2208±0.0114	0.2739±0.0165
	APRF	0.7741±0.0138	0.2487±0.0123	0.1680±0.0152

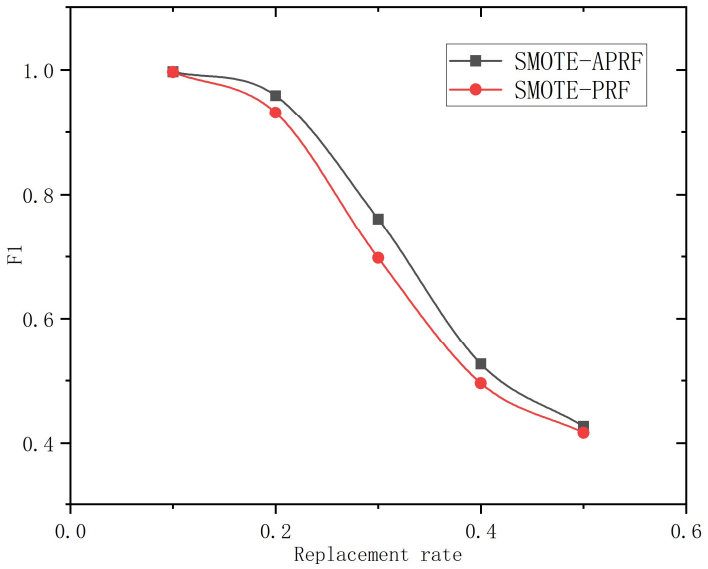


Fig. 7. F1 Performance and Vacancy Relationship Curve.

3.3. Result analysis

From the above experiments, the following conclusions can be drawn:
1) The wind turbine gearbox fault diagnosis model based on SMOTE-APRF has the optimal F1-score, and the FAR and MAR indicators are also better than most models when dealing with missing values, Gaussian noise, and label noise in SCADA data.
2) Compared with SMOTE-APRF, the kmeans-SMOTE-APRF model can synergistically process the noise of the data and the noise introduced by SMOTE, indirectly improving the diagnostic ability under imbalanced data categories.
3) The SMOTE-APRF model demonstrates superior robustness against three different types of noise compared to other models, like LightGBM. While LightGBM performs better in the presence of label noise, it shows poorer performance in the face of other types of noise.

4. Conclusions

The fault diagnosis of wind turbine gearboxes is crucial to ensuring the safe operation of wind turbines. This article proposes a combination of SMOTE oversampling and adaptive probability random forest to enhance the robustness of gearbox fault diagnosis under varying noise conditions. The approach introduces ensemble learning and Monte Carlo simulation to calculate uncertainty. This article's main contributions are threefold.

1) Adopting a data-driven approach to adaptively calculate the uncertainty of wind turbine SCADA data, combined with probabilistic random forest, improves the performance of gearbox fault diagnosis under high noise conditions.

2) Introducing adaptive probability random forest as a classifier indirectly addresses the issues of noise pollution and boundary ambiguity associated with the SMOTE method. Compared with SMOTE variants, using SMOTE can reduce model complexity.

3) The proposed SMOTE-APRF wind turbine gearbox fault diagnosis model exhibits optimal robustness against three types of noise pollution, enhancing the operational and maintenance standards of wind turbines.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62173050); the Natural Science Foundation of Hunan Province, China (Grant No. 2023JJ30051); the Major Scientific and Technological Innovation Platform Project of Hunan Province (2024JC1003); the Energy Conservation and Emission Reduction Hunan University Student Innovation and Entrepreneurship Education Center (Grant No. 2020-19); and the Postgraduate Scientific Research Innovation Project of Hunan Province (Grant No. QL20230214).

References

[1] Dorrell, J. and K. Lee. (2020). The Cost of Wind: Negative Economic Effects of Global Wind Energy Development. Energies, 13(14), 3667.

[2] Nwaigwe, K.N. (2021). Assessment of wind energy technology adoption, application and utilization: a critical review. International Journal of Environmental Science and Technology, 19(5), 4525-4536.

[3] Freire, N.M.A. and A.J.M. Cardoso. (2021). Fault Detection and Condition Monitoring of PMSGs in Offshore Wind Turbines. Machines, 9(11), 260.

[4] Gao, Z. and X. Liu. (2021). An Overview on Fault Diagnosis, Prognosis and Resilient Control for Wind Turbine Systems. Processes, 9(2), 300.

[5] Salameh, J.P., S. Cauet, E. Etien, A. Sakout and L. Rambault. (2018). Gearbox condition monitoring in wind turbines: A review. Mechanical Systems and Signal Processing, 111, 251-264.

[6] Nie, Y., F. Li, L. Wang, J. Li, M. Sun, M. Wang and J. Li. (2022). A mathematical model of vibration signal for multistage wind turbine gearboxes with transmission path effect analysis. Mechanism and Machine Theory, 167, 104428.

[7] Dao, P.B. (2022). Condition monitoring and fault diagnosis of wind turbines based on structural break detection in SCADA data. Renewable Energy, 185, 641-654.

[8] Zhang, K., B. Tang, L. Deng and X. Yu. (2021). Fault Detection of Wind Turbines by Subspace Reconstruction-Based Robust Kernel Principal Component Analysis. IEEE Transactions on Instrumentation and Measurement, 70, 1-11.

[9] Wang, L., S. Jia, X. Yan, L. Ma and J. Fang. (2022). A SCADA-Data-Driven Condition Monitoring Method of Wind Turbine Generators. IEEE Access, 10, 67532-67540.

[10] Shi, Y., Y. Liu and X. Gao. (2021). Study of Wind Turbine Fault Diagnosis and Early Warning Based on SCADA Data. IEEE Access, 9, 124600-124615.

[11] Chawla, N.V., K.W. Bowyer, L.O. Hall and W.P. Kegelmeyer. (2002). SMOTE: synthetic minority over-sampling technique. AI Access Foundation, 16(1), 321-357.

[12] Haq, M.A. (2021). SMOTEDNN: A Novel Model for Air Pollution Forecasting and AQI Classification.

[13] Arjun, P. and K.G. Manoj. (2021). Improved Hybrid Bag-Boost Ensemble With K-Means-SMOTE–ENN Technique for Handling Noisy Class Imbalanced Data. The Computer Journal(1), 1.

[14] Wang, J.B., C.A. Zou and G.H. Fu. (2021). AWSMOTE: An SVM-Based Adaptive Weighted SMOTE for Class-Imbalance Learning. Scientific Programming.

[15] Li, Y., Y. Wang, T. Li, B. Li and X. Lan. (2021). SP-SMOTE: A novel space partitioning based synthetic minority oversampling technique. Knowledge-Based Systems, 228, 107269-.

[16] Zhang, A., H. Yu, S. Zhou, Z. Huan and X. Yang. (2022). Instance weighted SMOTE by indirectly exploring the data distribution. Knowledge-Based Systems, 249.

[17] Breiman. (2001). Random forests. MACH LEARN, 45(1), 5-32.

[18] Zhou, X., P.L.K. Ding and B. Li, Improving Robustness of Random Forest Under Label Noise, in 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). 2019. p. 950-958.

[19] Agjee, N.e.H., O. Mutanga, K. Peerbhay and R. Ismail. (2018). The Impact of Simulated Spectral Noise on Random Forest and Oblique Random Forest Classification Performance. Journal of Spectroscopy, 2018, 1-8.

[20] Wang, T., F. Xue, Y. Zhou and A. Ming. (2022). MARF: Multiscale Adaptive-Switch Random Forest for Leg Detection With 2-D Laser Scanners. IEEE Trans Cybern, 2168-2267.

[21] Reis, I., D. Baron and S. Shahaf. (2018). Probabilistic Random Forest: A machine learning algorithm for noisy datasets. Astronomical Journal, 157(1).

[22] Mervin, L.H., M.A. Trapotsi, A.M. Afzal, I.P. Barrett, A. Bender and O. Engkvist. (2021). Probabilistic Random Forest improves bioactivity predictions close to the classification threshold by taking into account experimental uncertainty. J Cheminform, 13(1), 62.

[23] Lin, T.-H. and J.-R. Jiang. (2021). Credit Card Fraud Detection with Autoencoder and Probabilistic Random Forest. Mathematics, 9(21), 2683.