Improved Weighted *k*-Nearest Neighbor Based on PSO for Wind Power System State Recognition

Chun-Yao Lee *, Kuan-Yu Huang, Yi-Xing Shen and Yao-Chen Lee

Department of Electrical Engineering, Chung Yuan Christian University, No. 200, Zhongbei Road,
Zhongli District, Taoyuan City 320, Taiwan; g10578025@cycu.edu.tw (K.-Y.H.); g9778003@cycu.edu.tw (Y.-X.S.);
g9878036@cycu.edu.tw (Y.-C.L.)
* Correspondence: cyl@cycu.edu.tw; Tel.: +886-3-2654827

Abstract: In this paper, we propose using particle swarm optimization (PSO) which can improve weighted *k*-nearest neighbors (PWKNN) to diagnose the failure of a wind power system. PWKNN adjusts weight to correctly reflect the importance of features and uses the distance judgment strategy to figure out the identical probability of multi-label classification. The PSO optimizes the weight and parameter *k* of PWKNN. This testing is based on four classified conditions of the 300 W wind generator which include healthy, loss of lubrication in the gearbox, angular misaligned rotor, and bearing fault. Current signals are used to measure the conditions. This testing tends to establish a feature database that makes up or trains classifiers through feature extraction. Not lowering the classification accuracy, the correlation coefficient of feature selection is applied to eliminate irrelevant features and to diminish the runtime of classifiers. A comparison with other traditional classifiers, i.e., backpropagation neural network (BPNN), *k*-nearest neighbor (*k*-NN), and radial basis function network (RBFN) shows that PWKNN has a higher classification accuracy. The feature selection can diminish the average features from 16 to 2.8 and can reduce the runtime by 61%. This testing can classify these four conditions accurately without being affected by noise and it can reach an accuracy of 83% in the condition of signal-to-noise ratio (SNR) is 20dB. The results show that the PWKNN approach is capable of diagnosing the failure of a wind power system.

Keywords: particle swarm optimization (PSO); *k*-nearest neighbors (*k*-NN); classification; feature selection

---

1. Introduction

With the emergence of green energy, wind power plays a principle role in energy. However, wind power is a form of intermittent energy without a stable output. In addition, the malfunction of wind generators is a great concern. Downtime leads to heavy cost losses in peak-wind seasons [1–3]. Furthermore, a large number of wind generators in wind farms increases the probability of failure. Without sufficient personnel, an automatic inspection of the wind generator is indispensable [4]. A central system is employed to monitor the operation of wind generators, which can effectively reduce personnel cost. According to [5,6], the most common fault condition was bearing damage, accounting for 45%; followed by stator failure, accounting for 35%; and finally, rotor failure, accounting for 10%; and other damages are 10%. When the motor fails, the running cost increases. Therefore, fault identification has always been one of the topics in industrial applications. Generally, the automatic identification system includes the following three parts: the signal analysis, feature extraction, and condition classification. The signal analysis has been carefully researched in the past, in which the feature extraction and condition classification are crucial factors [7,8]. Feature extraction involves extracting specific and representative features, providing few data to classifiers in classification.

However, the features extracted are not always the effective ones. The valid features are likely to become invalid features if the signals include noise. This research emphasizes how to gain effective features and raise the classification accuracy [9]. Artificial neural network (ANN) has been widely used in different fields, such as machine operation control, automatic fault detection, and image recognition [6,10–14]. Since the first neural network was proposed, different types of networks have been developed one after another, such as feedforward neural network, probabilistic neural network (PNN) [10,11], and backpropagation neural network (BPNN) [6,12–14]. The use of different classifiers for various types of issues have different performance, architecture, and classification effects. BPNN is a supervised learning network, and its recognition ability is higher than the unsupervised learning network. Furthermore, classifiers offer a method of automatic classification. A well-organized classifier can provide a higher classification accuracy, which takes less time for computing, increases computing efficiency, and favors a large quantity, real-time system.

There have been considerable studies on the signal analysis of rotating machines, of which the vibration signal analysis was the focus [1,15]. Vibration signals can show the machine operation clearly. While measuring vibration signals, an accelerometer is a compulsory supplement. Additionally, one accelerometer can merely measure a single position's vibration. Recently, more and more researches have used the fault diagnosis of electric currents and voltage [16–18]. Electric currents and voltage can lessen the number and cost of sensors, complexity of measurement system, and failure rate.

The signal analysis can detect the unusual signals; the traditional signal analysis is Fourier transforms (FT) [19] which cannot obtain transient messages. However, rotating machines often have intermittent messages. A multi-resolution analysis (MRA) can be a good method of analyzing unusual signals [20]. Features extracted from signals in automatic classification can reduce the input number of classifiers and computing time. Feature extraction probably includes invalid features and its number affects computing. In addition, invalid features lower the classification accuracy of classifiers [21–23].

Although *k*-nearest neighbor (*k*-NN) is a simple and fast classifier, its procedure and structure need some improvement. The greedy method, used in [24], is to adjust *k*-NN with weight features. In [25], the *k* parameters should be adjusted according to different categories of training sets. Additionally, the setting of parameter k and identical probability of classification remain to be solved.

In this paper, we propose one approach to recognize the operational conditions of power generators. The approach is based on structure and consists of the following two parts: First, the feature extraction technique is used, where the wavelet transform method extracts the features from the 300 W power system. Second, the correlation coefficients are used to optimize the feature set. Finally, the proposed method is used to recognize the operation of the wind power system. The proposed method combines *k*-NN and PSO which can improve the accuracy, namely PWKNN. Experiments show that the performance of the PWKNN model is better than the other classification methods, and the average classification accuracy of PWKNN is higher than that of other classifiers.

This paper is organized as follows: In Section 2, we introduce the signal analysis and feature extraction approach; in Section 3, we present the procedure and structure of improved weight *k*-NN based on particle swarm optimization (PWKNN); in Section 4, we discuss the types of malfunctions of a wind power system, the feature selection; in Section 5, we present the simulation results; and in Section 6, we provide the conclusions.

2. Signal Processing and Feature Extraction

2.1. Wavelet Transform

The French geophysicist Jean Morlet invented the wavelet transform for analyzing local properties of seismic waves [26]. He found that the fast Fourier transform (FFT) could not meet the requirements, therefore, he introduced a wavelet into the signal analysis and decomposed signals. The wavelet transform of signals can be observed at different resolutions. Therefore, local properties can be examined effectively. It is a signal transform analysis between time and frequency domains.

Continuous Wavelet Transform

The scale parameter and translation parameter can adjust the mother wavelet $\psi(t)$ shown in Equation (1), where a is the scale parameter and b is the translation parameter.

$$\psi(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t - b}{a}\right) \tag{1}$$

The function $f(t)$ of continuous wavelet transform is defined in Equation (2), where the asterisk represents the operation of complex conjugate.

$$CWT(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi^*\left(\frac{t - b}{a}\right) dt \tag{2}$$

2.2. Multi-Resolution Analysis

The function $f(t)$ is decomposed by the scaling function and the mother wavelet function through the scaling change and wavelet transform. The scaling function is considered to be a low-pass filter and the mother wavelet function is considered to be a high-pass filter. The structure shown in Figure 1 decomposes the signal $f(t)$ into a high- and low-frequency signal. The low-frequency signal is decomposed for the second time and the decomposed signals, d_1, ..., d_n and a_n can be yielded as follows:

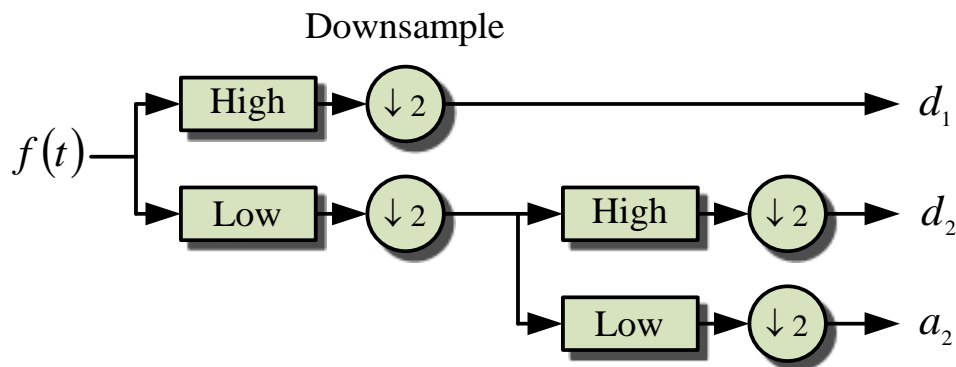$$f(t) = d_1 + d_2 + \cdots + d_n + a_n \tag{3}$$

Figure 1. The structure of the multi-resolution analysis (MRA).

The procedure mentioned above is an MRA [26]. Through the viewpoint of spectral spectrum, the MRA can process the detailed analysis of low-frequency signals shown in Figure 2. Therefore, more decomposition implies higher resolution.
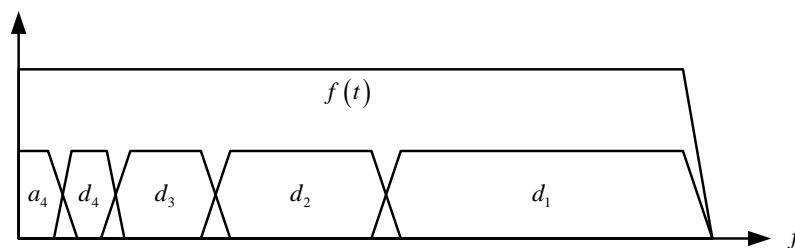

Figure 2. The spectral distribution of wavelet coefficient.

2.3. Feature Extraction

Scales of wavelet coefficients can show the signals in Equation (3). By using Parseval's theorem, Equation (3) can be reconstructed as follows:

$$\int_{n=-\infty}^{\infty} |f(t)|^2 dt = \sum_{l=-\infty}^{\infty} |a_j(l)|^2 + \sum_{j=0}^{\infty} \sum_{z=-\infty}^{\infty} |d_j(z)|^2 \qquad (4)$$

The first component refers to the average power of a signal. The latter component is the sum of the average power of coefficients after wavelet transformation. The simplified formula is shown as follows:

$$E_f = E_{a_j} + \sum_{j=0}^{j-1} E_{d_j} \qquad (5)$$

The energy spectrum indicates the energy for each frequency band of signal. The different distributions of energy spectrum can easily show operational conditions of wind generator. Hence, the energy spectrums are considered to be the features in this paper.

3. PWKNN and Feature Selection

3.1. PWKNN

The *k*-NN is a supervised learning method and also among the simplest of machine learning algorithms [27]. The method classifies an unknown sample class according to the nearest *k* samples of training datasets. The nearest *k* samples are called *k*-nearest neighbors which can be obtained through the distance computation. The class having the maximum number is considered to be the unknown category. The classification decision is as follows:

$$D(x_i) = argmax_m \sum_{t_j \in kNN} y(t_j, c_m) \qquad (6)$$

The equation means that the predictor will be the class that has the largest number of members in the *k*-nearest neighbors. For example, the left part of Figure 3 indicates three classes of samples (square, circle, and triangle) and an unknown sample (rhombus). The nearest four samples are selected. The numeric means the order of the distance 1 is the nearest and 4 is the farthest. The figure shows that Class A is a strong probability. Therefore, the unknown sample is classified to Class A.
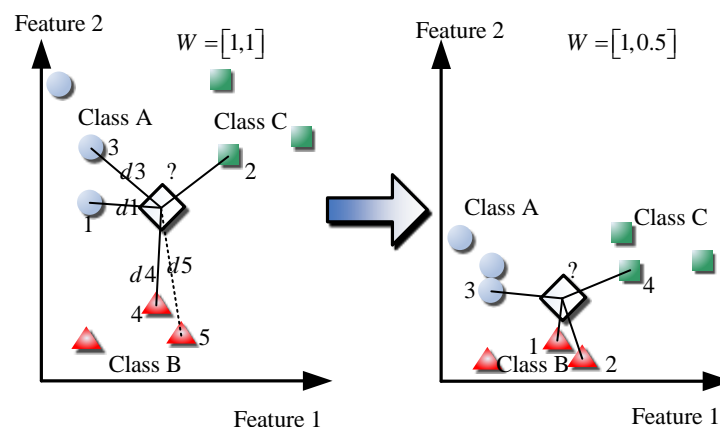

Figure 3. The diagram of classification by weighted *k*-NN (WKNN).

K-NN employs a distance measure, usually Euclidean distance, to show the similarity. If $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ are n-dimensions vectors, the following shows how Euclidean distance computes:

$$dist(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \tag{7}$$

However, k-NN has some weakness as follows: (1) All feature weights are equal. Not all features to the classified results have the positive correlation. The invalid features lead to the error classified results. (2) Identical classified probabilities probably occur. As shown on the left side of the Figure 3, if a nearest neighbor is added to the classification decisions, the probabilities of Class A, Class B, and Class C are 40%, 40%, and 20%, respectively. Class A and Class B have the same probability.

Refers to the k-NN weakness (1), the classification result of k-NN is sensitive to features. Invalid features reduce the classification accuracy. The weighted k-NN (WKNN) is implemented to modify this drawback. Different weights are used according to different features. The weight, $W = [w_1, w_2, \ldots, w_n]$, is taken into account in (x) to compute the weighted distance as follows:

$$dist(X, Y) = \sqrt{w_1(x_1 - y_1)^2 + w_2(x_2 - y_2)^2 + \cdots + w_n(x_n - y_n)^2} \tag{8}$$

From Figure 3, the weight is $W = [1, 1]$ in the left diagram and the weight is $W = [1, 0.5]$ in the right diagram. The equation mentioned above shows that the weight change can influence the classification results. Because, initially, Class A is a high probability, the probability of Class B will be stronger than that of Class A after the changed weight. Feature 2 has less effect on the classification results. The weight adjustment can alter the effect of different features on classification results and eliminate certain invalid features to raise the classification accuracy.

Concerning the classified result of k-NN producing identical probability, this research suggests the distance decision to determine the classification. In the left diagram of Figure 3, Class A and Class B have the identical probability. The method computes the distance from Class A and Class B to each two test points of their k-nearest neighbors. The distances from Class A to its test points of k-nearest neighbors are $d1$ and $d3$. The distances from Class B to its test points of k-nearest neighbors are $d4$ and $d5$. If D is the decision result, the following shows that the sum of shorter distances to the class is considered as the predictor in classification:

$$\begin{cases} D = \text{Class A if } d1 + d3 < d4 + d5 \\ D = \text{Class B if } d1 + d3 > d4 + d5 \end{cases} \tag{9}$$

This method can overcome the identical probability and enhance the correctness of classification.

PWKNN is a classifier of upgrading weighted k-nearest neighbors based on particle swarm optimization (PSO). The PSO optimizes weights and k value and estimates the predictive classification accuracy of PWKNN with leave one out cross-validation (LOOCV). In the LOOCV, each sample data is a class. One class is a testing sample in each computation, others are training samples. All classes are measured, and the sum of classified accuracy are computed as predictive classification accuracy pca_{CV}. If the number of the correct classification is $N_{Correctly}$ and the number of total training samples is N_{Total}, pca_{CV} is shown as follows:

$$pca_{CV} = \frac{N_{Correctly}}{N_{Total}} \times 100\% \tag{10}$$

3.2. Particle Swarm Optimization (PSO)

PSO is a computational method on optimizing a problem presented by R. Eberhart and J. Kennedy [28]. PSO has become a popular optimization algorithm and widely used in practical problem solving by researchers [29–31]. PSO generates particles randomly from the search space.

The movement of a particle is influenced by its best position and better positions founded by other particles, which optimizes a solution in the space. The optimal result is that all particles are found in the same position of the search space. This position is the best solution.

Unlike the genetic algorithm (GA), the procedure of PSO works according to a few simple formula and conceptions [28,32]. The object function or fitness function $Fit(\bullet)$ must be maximized. To avoid the solution converging toward the local best, the suitable number of particles is set, and the position is initialized randomly. Then, the velocity of particle is determined by:

$$v_i(t+1) = w(t) \cdot v_i(t) + \phi_1 \cdot r_1 \cdot (pbest_i - x_i) + \phi_2 \cdot r_2 \cdot (gbest - x_i) \tag{11}$$

In addition, the right choice of inertial weight provides a balance between global exploration and local exploitation. In general, the inertial weight, w, can be set to change depending on the current number of iterations, t, as follows:

$$w = w_{max} - \frac{w_{max} - w_{min}}{T} \times t \tag{12}$$

The equation is influenced by the following three factors: the previous particle's velocity $v_i(t)$, self-best position of particle $pbest_i$ and the all particle's best position $gbest_i$. Next, update the particle's position, shown as follows:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{13}$$

Additionally, the testing judges whether or not the output of fitness function is optimal to replace the particle self-best position. If the output is greater than the previous one, the new position replaces the particle's best position. If not, the best position of particle remains in the same position. The equation is shown as follows:

$$pbest_i(t+1) = \begin{cases} pbest(t) \text{ if } Fit(x_i(t+1)) \leq Fit(pbest_i(t)) \\ x_i(t+1) \text{ if } Fit(x_i(t+1)) > Fit(pbest_i(t)) \end{cases} \tag{14}$$

Moreover, the best position of all particles can be found to select the greatest output of fitness function. Finally, the position is considered to be the best solution, and shown in the following equation:

$$gbest(t+1) = \begin{cases} pbest(t+1) \text{ if } \max(Fit(pbest(t+1))) \leq Fit(gbest(t)) \\ gbest(t) \text{ if } \max(Fit(pbest(t+1))) > Fit(gbest(t)) \end{cases} \tag{15}$$

The equation of particle movement is to decide the ability of search solution and convergence. The formula mentioned above is commonly used after improvement.

To ensure exploration ability at the start of the algorithm, and then to ensure exploitation ability in the search area. Therefore, the inertia weight factor in the studies is adjusted to linearly decrease. The values w_{max} and w_{min} are the upper and lower limits of the inertial weight and are set at 0.9 and 0.4 in (12), respectively. The maximum number of iterations, T, is set in 1000. Additionally, by adjusting the learning coefficients φ_1 and φ_2, are set as 2.0, the particle self-best position and all particle's best positions can influence particle movement. The random number is to increase the interference and avoid the solution getting stuck into local best in the optimization process [33].

In this paper, weight $W = [w_1, w_2, \ldots, w_n]$ and k value are taken for particle position $x_i = [w_1, w_2, \ldots, w_n, k]$. The predictive accuracy of PWKNN P_{CV} is regarded as the fitness function $Fit(x_i) = P_{CV}$. The optimization process is shown in Figure 4. A set of optimum parameters $[w_1, w_2, \ldots, w_n, k]_{best}$ can generate after PSO optimization, which PWKNN can use to classify categories to gain the highest classified correction accuracy.
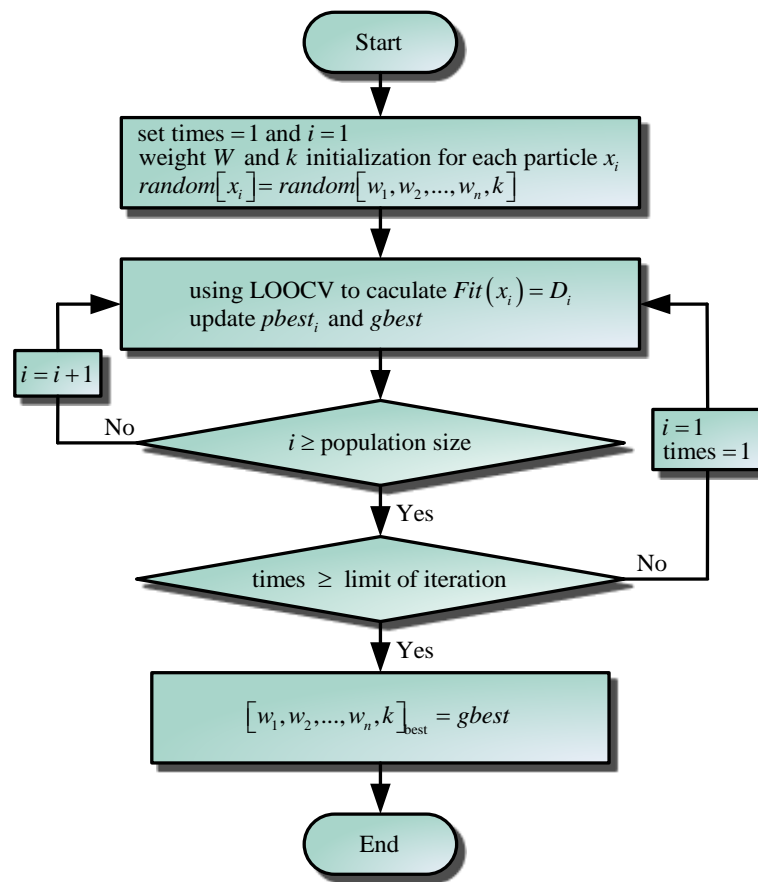
Figure 4. Flowchart of particle swarm with WKNN (PWKNN).

3.3. Feature Selection

Wavelet coefficients of different features can be taken for the features of generator operation and the input of classifier computing. In addition, correlation coefficients are used to select features to reach optimization and reduce computation. The formula of correlation coefficient is shown as follows:

$$R = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{16}$$

This paper determines the feature selection according to correlation coefficients. The predictive classification accuracy of PWKNN is considered to be the stop criterion when all features are selected. Features with minor correlation coefficients are removed first until none is left out. The flowchart of feature selection is shown in Figure 5. The following is the procedure of the feature selection:

Step (1)   Suppose the number of all features is *N*, select all features and use the cross one out cross-validation (LOOCV) to compute the classification accuracy $pca_{CV}^{\max,N}$ of PWKNN, which is the criterion of raising the classification accuracy. W_N is the weight after the PWKNN optimization.

Step (2)   After removing the lowest correlation coefficient, the feature number is $N-1$. Use W_N to recompute the classification accuracy pca_{CV}.

Step (3)   If $pca_{CV} \geq pca_{CV}^{\max,N}$, go back to Step 2, otherwise, features after removal can gain the optimum weight W_{N-1} and classification accuracy $pca_{CV}^{\max,N-1}$ through weight optimization.

Step (4)　　　　If $pca_{CV}^{\max,N-1} \geq pca_{CV}^{\max,N}$, go back to Step 2. Record W_{N-1} and this set of features is optimum; otherwise, come to an end of this procedure.
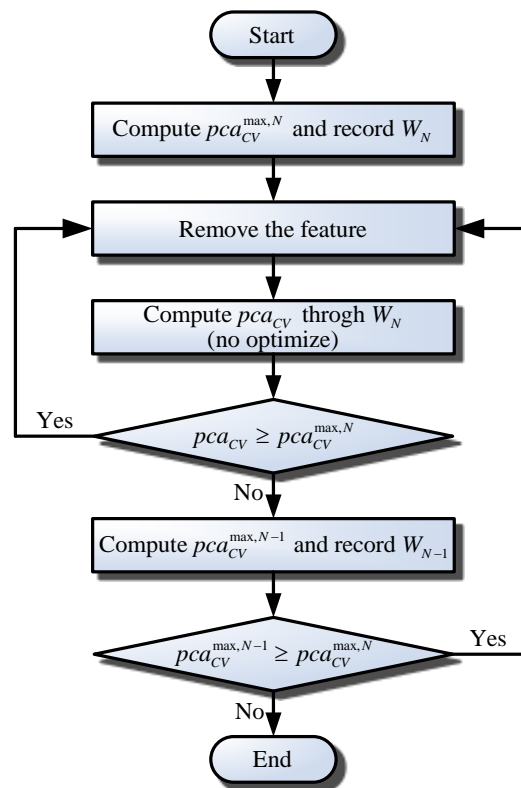
Figure 5. Flowchart of feature selection.

4. Types of Malfunctions of a Wind Power System

This testing employs a wind power simulation platform, load, data acquisition (DAQ), and personal computer, as shown in Figure 6, as well as a 300 W rated three-phase generator and gearbox with ratio of 1 to 10 driven by a drive motor to simulate wind power. The output of wind power generators links to the load. The load of wye connected single phase is 20 Ω. One phase of current signals can be derived from the NI-PXI 1033 chassis and NI-PXI 4071 digital multimeter (manufactured by National Instruments, Austin, Texas, USA) to the personal computer. The derived signal is segmented with proper length. This research is based on the time of 0.5 s as the length of segment, shown in Figure 7. Finally, we use the computer to analyze and automatically detect the signals.
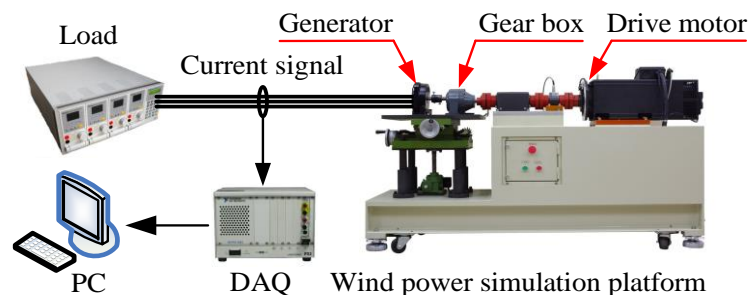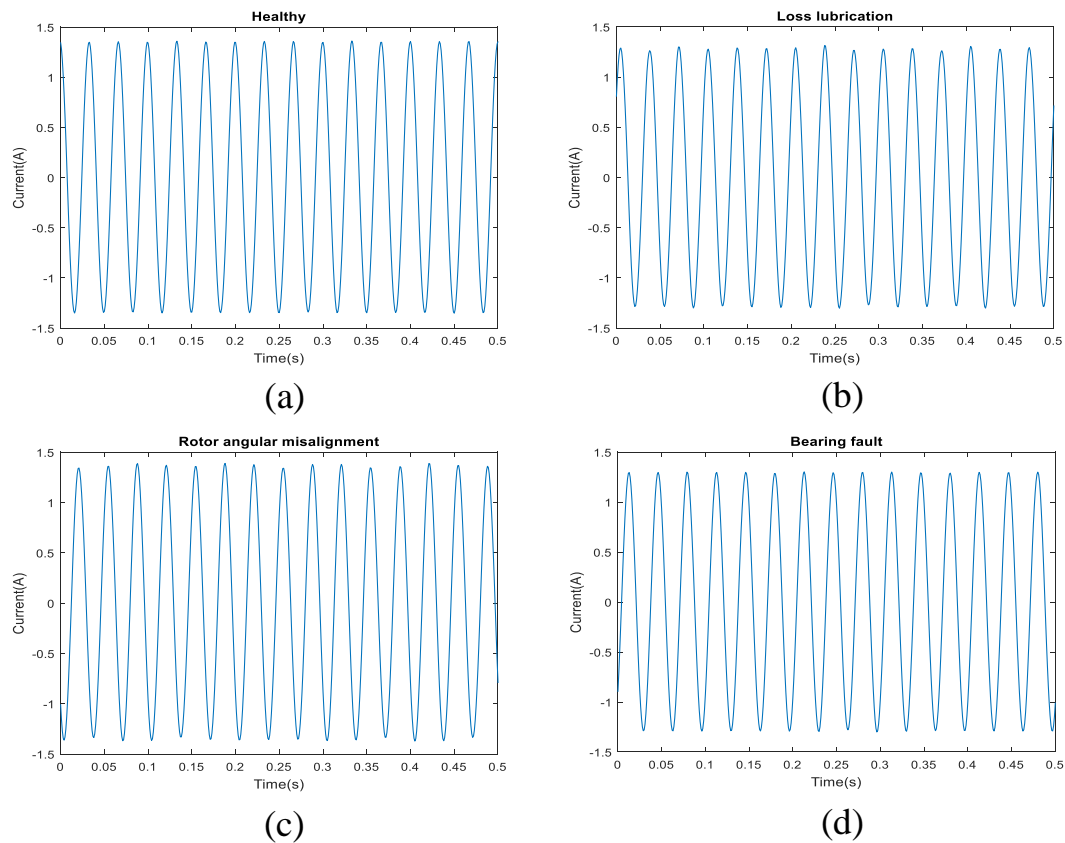
Figure 6. The block diagram of wind power simulation platform.

Figure 7. Generator output current: (**a**) Healthy; (**b**) Loss of lubrication; (**c**) Rotor angular misalignment; (**d**) Bearing fault.

4.1. Loss of Lubrication in the Gearbox

Lubrication leakage leading to malfunction occurs when a gearbox leaks. Usually, the gearbox is filled with eight-tenths lubrication. However, the vibration of power generators is more serious in real scenarios. Vibration is likely to cause the leak valve to loosen and leakage of lubrication. A gearbox is damaged by deficient lubrication for long periods. This research is to examine the leakage of an empty gearbox causing malfunction. The output current of the power generator is shown in Figure 7b.

4.2. Rotor Angular Misalignment

Rotor angular misalignment occurs when the rotor angles of the power generator and gearbox misalign. Normally, two rotor angles are at the same level, as shown in Figure 8. Nevertheless, a large wind turbine is at high altitudes and easily influenced by an external force. The base probably misaligns due to lengthy external vibration, resulting indirectly in rotor angular misalignment. In this research, rotor angular misalignment occurs when the height of the gearbox is lowered, shown in Figure 9. This testing is to remove the thin plain washer to lower the height. After putting the thick and thin plain washers under the gearbox, its rotor angle reaches the same level as that of the power generator. The height of the thick plain washer is 2.96 mm and the thin plain washer is 2.00 mm. The output current of the power generator is shown in Figure 7c.
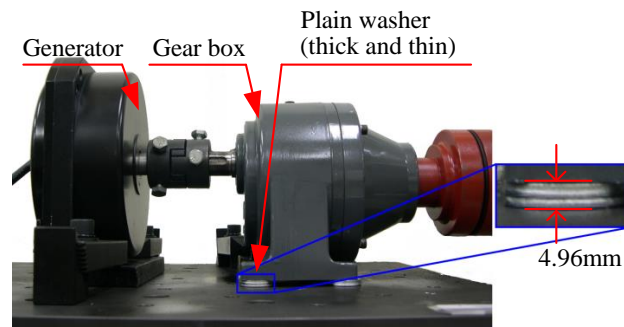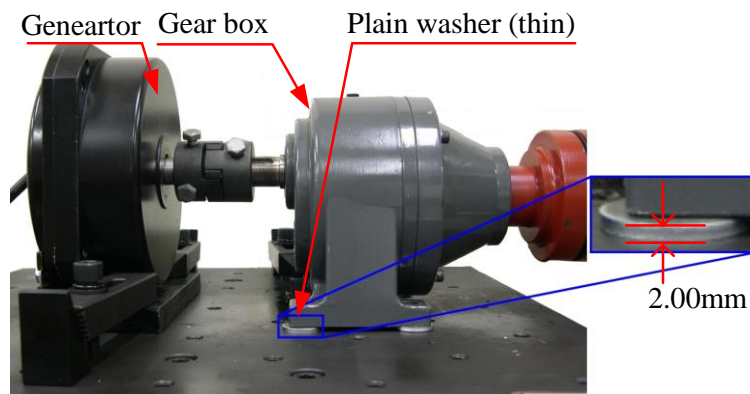
Figure 8. Operation of healthy condition.

Figure 9. Operation of rotor angular misalignment.

4.3. Bearing Fault

The ball bearing, shown in Figure 10a, involves an inner race, an outer race, steel balls, and a retainer. Normally, the steel ball rolls smoothly in the inner raceway and outer raceway. However, the external objects can damage the raceway or steel balls. In addition, the ball bearing can suffer abrasion by operating over a long period of time. This research suggests that the damage of the inner raceway and outer raceway is the malfunction situation. Since the bearing material is steel, electrical discharge machining (EDM) is employed to avoid the excessive abrasion. Figure 10b shows the outer raceway after damage. Figure 10c shows the damaged bearing and a sample of this testing. The apertures of the holes in the inner and outer raceway are 1 mm. The output current of the power generator is shown in Figure 7d.
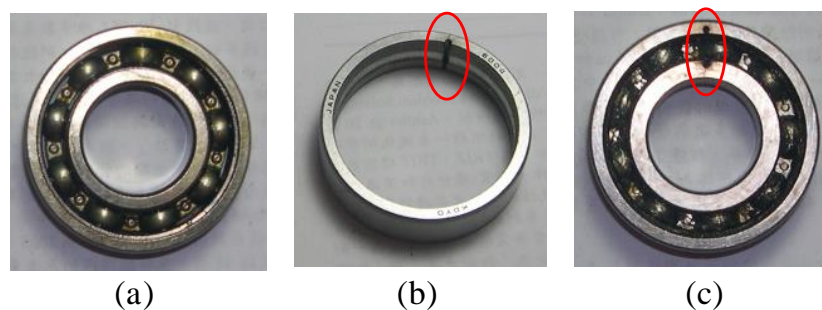

Figure 10. The bearing: (**a**) Healthy; (**b**) Outer raceway after damage; (**c**) Damaged bearing.

5. Simulation Results

5.1. Classified Dataset

According to the four operating conditions of the generator, data is measured and recorded, sequentially. Each operating condition can generate 200 samples; the first 160 samples are taken for a training set and the latter 40 samples are taken as a testing set. The samples total 800, including the training set of 640 samples and testing data of 160 samples, shown in Table 1. Each data is a signal lasting for 0.5 s. According to Equation (5), E_f does not have unit, and the energy spectrums of four operational conditions of each feature are shown in Figure 11.

Table 1. The state distribution of training set and test set.

Operation Condition	Training Set	Test Set
Healthy	160	40
Gearbox of loss lubrication	160	40
Rotor angular misalignment	160	40
Rearing fault	160	40
Total	640	160
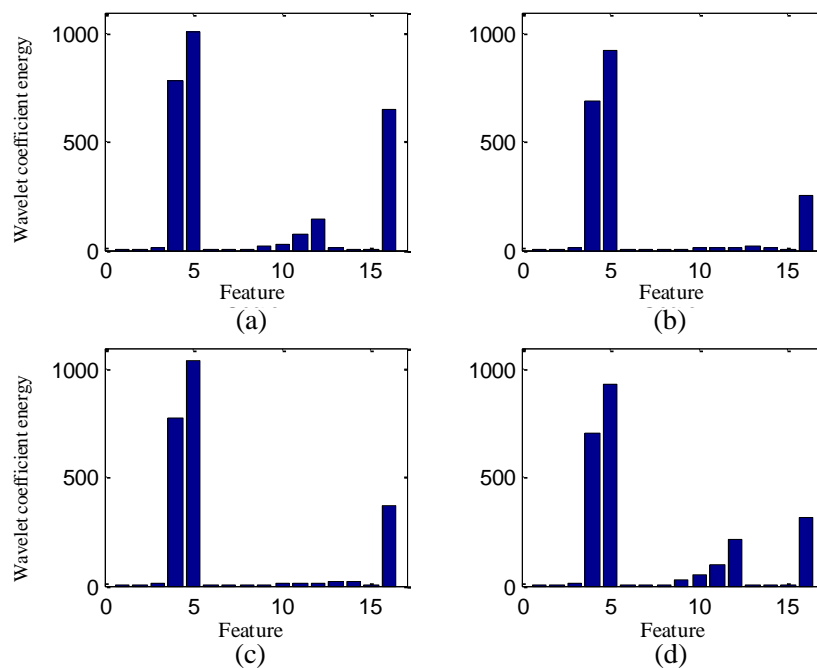
Figure 11. Energy spectrum: (**a**) Healthy; (**b**) Gearbox of loss lubrication; (**c**) Rotor angular misalignment; (**d**) Bearing fault.

Classification can be influenced by different levels of noise; therefore, various white Gaussian noises are involved in the testing. The signal-to-noise ratio (SNR) is used to measure the level of noise. SNRs of 40, 30, and 20 dB are included to compare the difference. The SNR is defined as follows:

$$\text{SNR} = 10 \, \log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \text{dB} \tag{17}$$

where P_{signal} is the power of original signal and P_{noise} is noise.

5.2. Simulation Results

This paper uses the PWKNN as a classifier of malfunction diagnosis and compares it with the back propagation neural network (BPNN), the *k*-NN, and the radial basis function network (RBFN). The iteration number of the PWKNN training is 400 times. The BPNN uses 40 neurons. The learning rate is 0.02; the iteration number is 400 times. The *k* parameter of *k*-NN is 3. The iteration number of RBFN training is 400 times; the spread is set as 0.1.

Without feature selection and noise disturbance, PWKNN, BPNN, and RBFN show correct classified accuracy. However, *k*-NN has 97.5% classified accuracy, as shown in Table 2. However, SNRs of 40, 30, and 20 dB show different results. If the SNR is 30 dB, PWKNN has 94% classification accuracy. The classification accuracy of BPNN and *k*-NN is comparatively less, i.e., 79% and 71%, respectively. The RBFN shows 65% classification accuracy. Regarding the computing time, the training time of the PWKNN and the BPNN is 140 and 102 s, respectively. The *k*-NN does not need to train. The *k*-NN and RBFN take 1.67×10^4 and 1.78×10^4 s, respectively, to complete the output of classifiers, whereas PWKNN takes 2.19×10^4 s and BPNN takes the most time to complete. Although *k*-NN and RBFN are better than PWKNN in time, PWKNN and BPNN are superior to *k*-NN and RBFN in classification accuracy. Furthermore, PWKNN is better than BPNN in average computing time or classification accuracy. Although PWKNN with a SNR of 20 dB is inferior to BPNN, PWKNN with a SNR of 30 dB is better than BPNN in accuracy.

Table 2. The classification results with feature selection.

Without Feature Selection	Classification Accuracy (%)			
	PWKNN	**BPNN**	***k*-NN**	**RBFN**
No noise	100	100	97.5	100
40 dB	98.8	93.1	79.4	83.1
30 dB	93.5	79.4	71.3	65.0
20 dB	71.6	75.0	60.0	43.1
Training time (s)	104	120	–	36
Output time (10^{-4} s)	2.19	2.52	1.78	1.67
Feature number	16			

Correlation coefficients for the classification results of all features should be computed first in feature extraction. The features of smallest correlation coefficients are the first to be removed. Table 3 shows the correlation coefficients for the classification of all features; d_{15}, d_{14} and so forth are removed. PWKNN, *k*-NN and RBFN use the predictive classification accuracy as the criterion of feature selection; BPNN uses the mean square error as the criterion. The classification results and number after feature selection are shown in Table 4. Most classifiers enhance classification accuracy. When the SNR is 30 dB, RBFN raises the classification accuracy from 65% to 91%. The percentage of accuracy rises about 40%. The feature selection increases the training time but reduces the output time. In Table 4, while BPNN needs the longest training time, RBFN needs less training time than PWKNN, and *k*-NN needs the least training time. PWKNN, BPNN, *k*-NN, and RBFN enhance the output time by 61%, 6%, 42%, and 17%, respectively. The feature number is the average number of repetitive computations for five times. The average features of PWKNN, BPNN, *k*-NN, and RBFN are 2.8, 10.1, 2.1, and 6.5, respectively. PWKNN and *k*-NN have the highest efficiency of feature selection. The results show that the feature number has less influence on the output time because BPNN is limited to the network structure and neuron numbers. PWKNN has more efficient feature selection and can improve the output time.

Table 3. The correlation of each feature.

Correlation Coefficient							
d_1	d_5	d_4	d_3	d_8	d_2	a_{15}	d_7
4.53	3.18	3.16	3.08	1.03	0.89	0.47	0.42
d_6	d_{11}	d_9	d_{13}	d_{12}	d_{10}	d_{14}	d_{15}
0.33	0.32	0.32	0.28	0.27	0.21	0.20	0.20

Table 4. The classification results with feature selection.

With Feature Selection	Classification Accuracy (%)			
	PWKNN	BPNN	k-NN	RBFN
No noise	100	100	100	100
40 dB	99.4	90.8	81.9	100
30 dB	93.1	81.3	83.1	90.6
20 dB	82.5	75.5	80.6	55.0
Training time (s)	467	937	4	3046
Output time (10^{-4} s)	1.36	2.37	1.03	1.39
Feature number (Average)	2.8	10.1	2.5	6.5

Although PWKNN does not have the least training and output time, the average classification accuracy is higher than other classifiers. BPNN has the equivalent classification accuracy but needs longer training time. Nevertheless, the output time is the highest. k-NN needs the least time; however, its classification accuracy is not higher than that of PWKNN. In addition, features influence RBFN significantly. Therefore, feature selection is an important factor. Thus, PWKNN is efficient for recognizing the operating conditions of wind generators. With the feature selection, PWKNN can reduce the feature numbers effectively and shorten the output time of classifiers.

6. Conclusions

This paper proposes the PWKNN to recognize the operational conditions of power generators. The PWKNN solves the identical probability of multi-label classification. The weight after adjustment can influence the features of classification through the parameter optimization of PSO and raise the classification accuracy effectively. Feature selection can eliminate the input time of classifiers and remove features effectively without lowering the classification accuracy. The research results show that whether there is noise or not, the average classification accuracy of the PWKNN is higher than that of other classifiers. If the SNR is 30 dB, the PWKNN has 93% classification accuracy. With the feature selection, the average feature numbers decrease from 16 to 2.8. The output time lowers 61%. If the SNR is 20 dB, the PWKNN increases classification accuracy from 72% to 82% through feature selection, which is better than other traditional classifiers. The PWKNN's ability to efficiently eliminate redundant features and strong classifications in noise conditions significantly improves the predictability of wind turbine failures as compared with other methods. In addition, the PWKNN is capable of diagnosing faults in wind power systems and can also consider applying in fault diagnosis of rotating machines.

Author Contributions: Conceptualization, C.-Y.L., K.-Y.H., Y.-X.S., Y.-C.L.; Methodology, C.-Y.L., K.-Y.H., Y.-X.S., Y.-C.L.; Software C.-Y.L., K.-Y.H., Y.-X.S., Y.-C.L.; Validation, C.-Y.L., K.-Y.H., Y.-X.S., Y.-C.L.; Formal Analysis, C.-Y.L., K.-Y.H., Y.-X.S., Y.-C.L.; Investigation, C.-Y.L., K.-Y.H., Y.-X.S., Y.-C.L.; Resources, C C.-Y.L., K.-Y.H., Y.-X.S., Y.-C.L.; Data Curation, C.-Y.L., K.-Y.H., Y.-X.S., Y.-C.L.; Writing—Original Draft Preparation, C.-Y.L., K.-Y.H., Y.-X.S., Y.-C.L.; Writing—Review & Editing, C.-Y.L., K.-Y.H., Y.-X.S., Y.-C.L.; Visualization, C.-Y.L., K.-Y.H., Y.-X.S., Y.-C.L.; Supervision, C.-Y.L.; Project Administration, C.-Y.L.; Funding Acquisition, C.-Y.L. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

a	The scale parameter
a_n	Approximation coefficient
b	Translation parameter
$CWT(a,b)$	Continuous wavelet transform
dB	Decibel
$d_1 \cdots d_n$	Detail coefficients
$d1 \ldots dn$	Euclidean distance between unknown point and class
$dist(X,Y)$	Euclidean distance computes of n-dimensions vectors
D	Decision result
E_f	Energy spectrum
E_{a_j}	Energy of approximation coefficient
E_{d_j}	Energy of detail coefficients
$f(t)$	Original signal
$Fit(\bullet)$	Fitness function
$gbest_i$	The best position of all particles
j	Scale
k	Nearest neighbor
l and z	Time
N	Number of all features
$N_{Correctly}$	Number of the correct classification
N_{Total}	Number of total training samples
P_{CV}	Predictive accuracy of PWKNN
pca_{CV}	Cross-validation predictive classification accuracy
pca_{CV}	Classification accuracy
$pca_{CV}^{\max,N}$	Classification accuracy of PWKNN
$pca_{CV}^{\max,N-1}$	classification accuracy through weight optimization
P_{signal}	Power of original signal
P_{noise}	Noise
$pbest_i$	The best position of particle i
R	Correlation coefficient,
r_1 and r_2	Random number
t_j	One of the neighbors in the training set
x	Input
x_i	The current position of particle i
$\bar{x}$	Mean of input,
X and Y	n-Dimensions vectors
y	Output
$\bar{y}$	Mean of output
$y(t_j, c_m)$	Indicates whether t_j belongs to class c_m
v_i	The velocity of particle i
w	Inertia weight factor
W	Weight
W_N	Weight after the PWKNN optimization
W_{N-1}	Features after removal can gain the optimum weight
$\psi(t)$	Mother wavelet
φ_1	Particle self-learning coefficient
φ_2	Particle swarm learning coefficient

References

1. Turnbull, A.; Carroll, J.; Koukoura, S.; McDonald, A. Prediction of wind turbine generator bearing failure through analysis of high-frequency vibration data and the application of support vector machine algorithms. *J. Eng.* **2019**, 4965–4969. [CrossRef]

2. Artigao, E.; Sapena-Bano, A.; Honrubia-Escribano, A.; Martinez-Roman, J.; Puche-Panadero, R.; Gómez-Lázaro, E. Long-term operational data analysis of an in-service wind turbine DFIG. *IEEE Access* **2019**, *7*, 17896–17906. [CrossRef]

3. Kabir, M.J.; Oo, A.M.T.; Rabbani, M. A brief review on offshore wind turbine fault detection and recent development in condition monitoring based maintenance system. In Proceedings of the 2015 Australasian Universities Power Engineering Conference (AUPEC), Wollongong, Australia, 27–30 September 2015; pp. 1–7.

4. Rezamand, M.; Carriveau, R.; Ting, D.S.; Davison, M.; Davis, J.J. Aggregate reliability analysis of wind turbine generators. *IET Renew. Power Gener.* **2019**, *13*, 1902–1910. [CrossRef]

5. Tobon-Mejia, D.A.; Medjaher, K.; Zerhouni, N. CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks. *Mech. Syst. Signal Process.* **2012**, *28*, 167–182. [CrossRef]

6. Lee, C.Y.; Lin, W.C. Induction motor fault classification based on FCBF-PSO feature selection method. *Appl. Sci.* **2020**, *10*, 5383. [CrossRef]

7. Watson, S.J.; Xiang, B.J.; Yang, W.; Tavner, P.J.; Crabtree, C.J. Condition monitoring of the power output of wind turbine generators using wavelets. *IEEE Trans. Energy Convers.* **2010**, *25*, 715–721. [CrossRef]

8. Tavner, P.J. Review of condition monitoring of rotating electrical machines. *IET Electr. Power Appl.* **2008**, *2*, 215–247. [CrossRef]

9. Lee, C.Y.; Shen, Y.X. Optimal feature selection for power-quality disturbances classification. *IEEE Trans. Power Deliv.* **2011**, *26*, 2342–2351. [CrossRef]

10. Jin, N.; Liu, D.-R. Wavelet basis function neural networks for sequential learning. *IEEE Trans. Neural Netw.* **2008**, *19*, 523–528. [CrossRef]

11. Perera, N.; Rajapakse, A.D. Recognition of fault transients using a probabilistic neural-network classifier. *IEEE Trans. Power Deliv.* **2010**, *26*, 410–419. [CrossRef]

12. Tripathy, M.; Maheshwari, R.; Verma, H. Power transformer differential protection based on optimal probabilistic neural network. *IEEE Trans. Power Deliv.* **2009**, *25*, 102–112. [CrossRef]

13. Ying, S.; Jianguo, Q. A method of arc priority determination Based on Back-Propagation Neural Network. In Proceedings of the 2017 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, China, 21–23 July 2017; pp. 38–41.

14. Wu, W.; Feng, G.; Li, Z.; Xu, Y. Deterministic convergence of an online gradient method for BP neural networks. *IEEE Trans. Neural Netw.* **2005**, *16*, 533–540. [CrossRef] [PubMed]

15. Wang, D.; Tsui, K.; Miao, Q. Prognostics and health management: A review of vibration based bearing and gear health indicators. *IEEE Access* **2017**, *6*, 665–676. [CrossRef]

16. Wang, J.; Peng, Y.; Qiao, W. Current-aided order tracking of vibration signals for bearing fault diagnosis of direct-drive wind turbines. *IEEE Trans. Ind. Electron.* **2016**, *63*, 6336–6346. [CrossRef]

17. Gong, X.; Qiao, W. Current-based mechanical fault detection for direct-drive wind turbines via synchronous sampling and impulse detection. *IEEE Trans. Ind. Electron.* **2015**, *62*, 1693–1702. [CrossRef]

18. Cheng, F.; Qu, L.; Qiao, W.; Wei, C.; Hao, L. Fault diagnosis of wind turbine gearboxes based on DIFG stator current envelope analysis. *IEEE Trans. Sustain. Energy* **2019**, *10*, 1044–1053. [CrossRef]

19. Sapena-Bañó, A.; Pineda-Sanchez, M.; Puche-Panadero, R.; Martinez-Roman, J.; Matić, D. Fault diagnosis of rotating electrical machines in transient regime using a single stator current's FFT. *IEEE Trans. Instrum. Meas.* **2015**, *64*, 3137–3146. [CrossRef]

20. Bayram, D.; Şeker, S. Redundancy-based predictive fault detection on electric motors by stationary wavelet transform. *IEEE Trans. Ind. App.* **2017**, *53*, 2997–3004. [CrossRef]

21. Ambusaidi, M.A.; He, X.; Nanda, P.; Tan, Z. Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE Trans. Comp.* **2016**, *65*, 2986–2998. [CrossRef]

22. Chkirbene, Z.; Erbad, A.; Hamila, R.; Mohamed, A.; Guizani, M.; Hamdi, M. TIDCS: A dynamic intrusion detection and classification system based feature selection. *IEEE Access* **2020**, *8*, 95864–95877. [CrossRef]

23. Alhakami, W.; Alharbi, A.; Bourouis, S.; Alroobaea, R.; Bouguila, N. Network anomaly intrusion detection using a nonparametric bayesian approach and feature selection. *IEEE Access* **2019**, *7*, 52181–52190. [CrossRef]

24. Sarvari, S.; Mohd Sani, N.F.; Mohd Hanapi, Z.; Abdullah, M.T. An efficient anomaly intrusion detection method with feature selection and evolutionary neural network. *IEEE Access* **2020**, *8*, 70651–70663. [CrossRef]

25. Han, E.H.; Karypis, G.; Kumar, V. Text categorization using weight adjusted k-Nearest Neighbor classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 53–65.

26. Mullick, S.S.; Datta, S.; Das, S. Adaptive learning-based k-nearest neighbor classifiers with resilience to class imbalance. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 5713–5725. [CrossRef] [PubMed]

27. Mallat, S. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1989**, *11*, 674–693. [CrossRef]

28. Zhang, S.; Li, X.; Zong, M.; Zhu, X.; Wang, R. Efficient kNN classification with different numbers of nearest neighbors. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 1774–1785. [CrossRef]

29. Kennedy, J.; Eberhart, R. Particle swarm optimization. *Proc. IEEE Int. Conf. Neural Netw.* **1995**, *6*, 1942–1948.

30. Lee, C.Y.; Tuegeh, M. Optimal optimisation-based microgrid scheduling considering impacts of unexpected forecast errors due to the uncertainty of renewable generation and loads fluctuation. *IET Renew. Power Gen.* **2020**, *14*, 321–331. [CrossRef]

31. Lee, C.Y.; Tuegeh, M. An Optimal solution for smooth and non-smooth cost functions-based economic dispatch problem. *Energies* **2020**, *13*, 3721. [CrossRef]

32. Shi, Y.; Eberhart, R. Parameter selection in particle swarm optimization. In Proceedings of the Seventh Annual Conference on Evolutionary Programming, San Diego, CA, USA, 25–27 March 1998; pp. 591–600.

33. Shi, Y.; Eberhart, R. Empirical study of particle swarm optimization. *Proc. IEEE Int. Congr. Evol. Comp.* **1999**, *3*, 101–106.