Fault Prediction Method Combining Principal Component Analysis and Back Propagation Neural Network

Kaixuan Wang[1], Mei Li[1, *], Wanxin Shi[1], Xu Liu[1], Shiyao Qin[2], Xiaojing Ma[2]

[1]School of Information Engineering, China University of Geosciences, Beijing, China

[2]New Energy Institute, China Electric Power Research Institute, Beijing, China

Abstract

Fault prediction of wind turbine could greatly reduce its operation and maintenance cost. A fault prediction method combining Principal Component Analysis (PCA) and Back Propagation Neural Network (BPNN) is proposed. Using only BPNN to predict the turbine fault, one has to pick out proper attributes for training and that can be a troublesome work. With PCA revealing the hidden relationship between attributes, we no longer need to select attributes by experience thus saving cost. Also, PCA largely reduce the dimension of attributes so that the curse of dimensionality can be avoided. The proposed PCA-BPNN method can improve the accuracy by 3% compared BPNN method, so that abnormal state of the mechanical faults can be determined more precisely, again reduce the maintenance cost.

1. Introduction

The market forecasts that by 2018 the wind energy cumulative gigawatts (GW) will be 43% higher than of 2015's GW [1]. For wind turbines the maintenance costs are high because of their remote location, and this can amount to as much as 25 to 30% of the total energy production [2]. The most effective way to reduce maintenance costs is to continuously monitor generators' status and predict the malfunction of wind turbine. Then the system degradation problems can be found and responded in time. Maintenance can be carried out ahead of time before the system crash and excess maintenance is avoided. Therefore, fault prediction could maximize the normal production of wind power plant and greatly reduce maintenance costs.

The main methods of wind turbine fault diagnosis include these types such as the fault mode analysis based on statistical data, the fault diagnosis based on time series prediction, model-based fault diagnosis of control system, the fault diagnosis based on vibration analysis, and other auxiliary diagnosis methods like acoustic emission technique and ultrasonic capacitance liquid level detection [3, 4]. These methods did not combine different characteristic data such as vibration, power, start stop record, which could be more informative.

Up to now, as more and more monitoring data, e.g. the temperature data, the driving power, the swing in the direction of the tower, the average wind speed, the angle of blade and the average generator speed are available, many data mining methods are utilized in fault prediction, for instance, graphical model-based approach proposed by Aloraini A and Sayed-Mouchaweh M [5], SVM-based solution by Santos P et al. [6], GA optimization method by Odofin Sarah et al. [7] and Neural Network used by Lan Q et al. [8] probabilistic neural network by Malik H and Mishra S [9], k-means and neural

* Corresponding author

E-mail address: maggieli@cugb.edu.cn (Mei Li), qinshy@epri.sgcc.com.cn (Shiyao Qin)

network by Xu Liu et al. [10]. These methods mainly focus on predicting directly with original data or data selected by experience instead of preprocessing data. Take BPNN as a sample. The researchers always input original data or select the data by experience. It leads to an inefficient data analysis because of the redundancy of data or unrevealed internal relationship of the data.

The proposed method in this paper aims to reduce the redundancy of the attributes by utilizing PCA to preprocess the data. After that, a part of those data are used to train the BPNN and the rest of the data are predicted. The D-value of the actual value and the predicted value indicates the symptom of fault. The experiment led up to the fact that the prediction accuracy can be actually improved with the principal component analysis method.

2. Methods

2.1. The Principal Component Analysis (PCA)

PCA is a dimension reduction method based on statistical analysis [11, 12]. Assume that there are n samples and every sample has p attributes. They form a n×p data matrix as following.

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \quad (1)$$

We note a column vector of X as $\mathbf{X}_i = (x_{1i}, x_{2i}, \cdots, x_{pi})^T$ $i = 1, 2, \cdots, p$. When the original attributes are large, we have to replace them with a few mutually orthogonal comprehensive components. The new column vector is respectively called the first, second… m-th principal component, and represents new attributes that has the largest variance in order.

$$\begin{cases} \mathbf{Z}_1 = l_{11}\mathbf{X}_1 + l_{12}\mathbf{X}_2 + \cdots + l_{1p}\mathbf{X}_p \\ \mathbf{Z}_2 = l_{21}\mathbf{X}_1 + l_{22}\mathbf{X}_2 + \cdots + l_{2p}\mathbf{X}_p \\ \cdots\cdots \\ \mathbf{Z}_m = l_{m1}\mathbf{X}_1 + l_{m2}\mathbf{X}_2 + \cdots + l_{mp}\mathbf{X}_p \end{cases} \quad (2)$$

Where is the eigenvector corresponding to the i-th eigenvalue by quantity of matrix X's covariance matrix or correlation matrix. There are six steps to achieve principal components.

Step 1. The data which are n samples with attributes form a matrix with column vectors.

Step 2. Each element of matrix subtracts the mean of its column.

Step 3. Compute the covariance matrix of the above matrix.

Step 4. Calculate the eigenvalues and eigenvectors of the covariance matrix.

Step 5. Rank the eigenvalues from the highest to the lowest in order to choose the top corresponding m eigenvectors. The rest eigenvectors are dropped.

Step 6. Calculate the m principal components according to formula (2).

The typical goal of a PCA is to reduce the dimensionality of the original feature space by projecting it onto a smaller subspace, where the eigenvectors will form the axes. The eigenvectors with the lowest eigenvalues bear the least information about the distribution of the data.

2.2. The Error Back Propagation Neural Network Algorithm

The BPNN algorithm is extensively used in prediction. It is expert in approximating any nonlinear function, so it is especially suitable for the issues of complex internal mechanism. Its self-learning ability and parallel fast computation also make it practical. Generally, the BPNN has two stages, training and testing.

During the training phase, training data including the input and the expected output are given to the neutral network. For example, the input might be an encoded picture of an animal, and the output could be a code that represents the category of the animal. Figure 1 shows the topology of the BPNN that includes an input layer, one hidden layer and an output layer. It should be noted that it can have more than one hidden layer.

Suppose there are D inputs, K outputs, M hidden layer nodes. are the input variables, are the hidden layer's variables, and are the output variables, and are the weight parameters between and, and. and are the bias vectors of the hidden layer nodes and the output layer nodes.
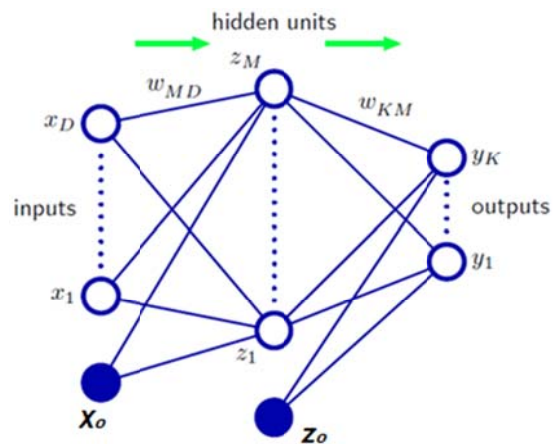

Figure 1. Three-layer neural network.

Take node as example. Its output is and its bias is, which is one conponent of bias vector. Its output is determined by its bias, the outputs of its former layer nodes and its activation function.

$$s_k = \sum_{i=1}^{M} w_{ik} z_i + z_{0k} \qquad (3)$$

$$y_k = f(s_k) \qquad (4)$$

where f is the activation function and it is usually the sigmoid function $f(x) = \dfrac{1}{1+e^{-x}}$.

The expected output is then compared with the actual output value, and an error signal is computed.

$$e = \frac{1}{2} \sum_{j=1}^{K} (y_k - d_k)^2 \qquad (5)$$

The training stage is a learning process and composed of two steps, which are input signal forward propagation and error signal back propagation. When the error exceeds its allowable scope, the BPNN will change over to the error signal back propagation step. The output error signals are transmitted backwards from the nodes of the output layer to each node in the immediate hidden layer to modify their connection weights and the bias of the nodes of the output layer proportion to the error signal, partial derivative of the output node activation function and the output of the connected node, where α is the learning rate.

$$w_{KM} = w_{KM} - \alpha \frac{\partial e}{\partial w_{KM}} \qquad (6)$$

The variation $\alpha \dfrac{\partial e}{\partial w_{KM}}$ is calculated as follows.

$$
\begin{aligned}
\alpha \frac{\partial e}{\partial w_{KM}} &= \alpha \cdot \frac{\partial}{\partial w_{KM}} \cdot \frac{1}{2} \sum_{j=1}^{K} (y_k - d_k)^2 \\
&= \alpha \cdot (y_k - d_k) \cdot \frac{\partial y_k}{\partial w_{KM}} \\
&= \alpha \cdot (y_k - d_k) \cdot f'(s_k) \cdot \frac{\partial s_k}{\partial w_{KM}} \\
&= \alpha \cdot (y_k - d_k) \cdot f(s_k)\left[1 - f(s_k)\right] \cdot z_M
\end{aligned} \qquad (7)
$$

This mechanism is also applied to its bias and the weights and biases of hidden layer nodes. There are five steps in the training stage.

Step 1. An input sample is applied to the input layer.

Step 2. The signal propagates layer by layer through the hidden layers until an output is produced according to formula (3) and (3).

Step 3. For each of the output nodes, the actual output value is compared with the expected output according to formula (5).

Step 4. Modify the weights and the bias of each node according to formula (6) and (7).

Step 5. The learning process will keep working and the above steps repeat for every input samples until the total error of the output of the network gradually reduced to the degree that can be accepted or to the set number of learning.

3. Experiments and Result

The proposed method has been tested on the wind turbine gearbox bearing condition monitoring using MATLAB. Among all wind turbine faults, gearbox bearing faults are relatively common [1, 3]. When the gearbox is out of order, it's bearing temperature would arise. Thus among the numerous parameters contained by wind turbine status data, the gearbox bearing temperature is the predicted target. The data was provided by New Energy Institute of China Electric Power Research Institute, from March 17 to April 17 in 2015, taking one month every 10 minutes to log data for test, with a total of 4500 data points.
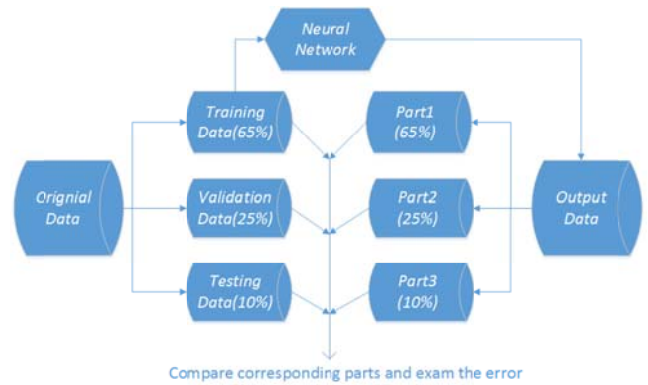

Figure 2. Data partion for training validation and testing.

The data was separated into 3 parts, first 65% of data for Neural Network training, next 25% for validation and last 10% for testing as figure 2 shows. The actual temperature will rise when wind turbine faults. We define D-value here to determine wind turbine fault (discussed in detail in part 4.1). Let T_a= actual gearbox oil temperature, T_b= predict gearbox oil temperature

$$\text{D-value} = T_a - T_b \qquad (8)$$

The fault happened after data point 4430, i.e. in the testing dataset. So unlike most of the prediction problem, we are expecting the predict value not fitting the actual ones in the testing set. The prediction accuracy we be represent by the mean square error of the validation set.

3.1. BPNN Method with 38 Attribute

Figure 3 shows the last 1000 point of actual and predicted value of gearbox bearing temperature applying BPNN method

with 38 attribute. The prediction result fit very well but the fault data was also tracked, the D-value is small all the time in this case we can't tell the turbine fault.

3.2. BPNN Method with 9 Attribute

Because too much features can lead to data overfit when applying BPNN, we selected 9 attributes of wind turbines, part of them are listed in Table 1. And the BPNN prediction result is shown Figure 4 (a), with the blue line representing the actual gearbox bearing temperature and orange for prediction value. Compared to Figure 3, the prediction is less accurate but in the last about 80 point the prediction line deviate from the actual one obviously, also see figure 4 (b) the D-value rise indicates turbine fault.
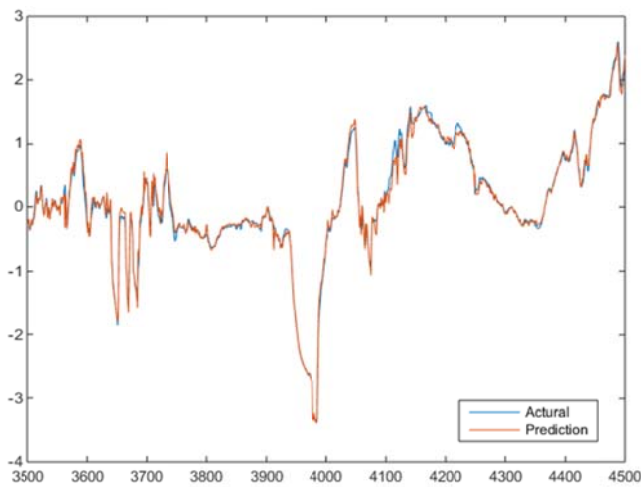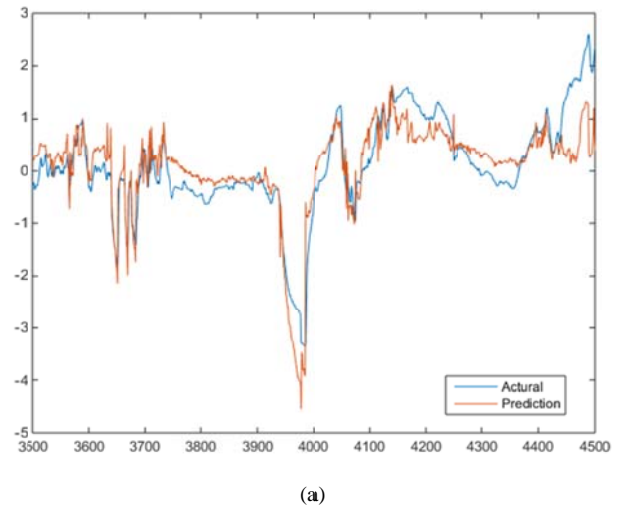

(a)

(b)

Figure 3. Actual and predicted value of gearbox bearing temperature applying BPNN method with 38 attribute.
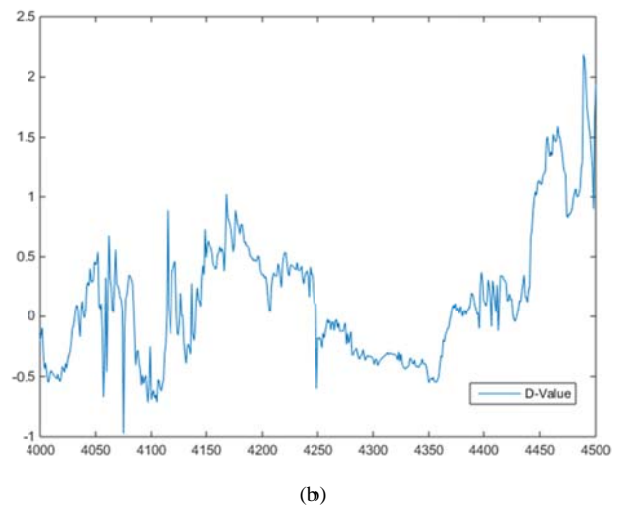
Figure 4. (a) Actual and predicted value of gearbox bearing temperature applying BPNN method with 38 attribute (b) the corresponding D-value.

Table 1. Part of normalized data of 9 attributes of wind turbines selected by experience*.

id	Att_1	Att_2	Att_3	Att_4	Att_5	Att_6	Att_7	Att_8	Att_9
1362	-0.2033	0.3450	0.8640	1.3042	0.5394	-0.2743	0.5775	-0.0836	0.7651
1363	-0.2033	0.3229	0.8651	1.3109	0.3073	-0.2743	0.0793	-0.2187	0.7651
1364	-0.2033	0.3110	0.8628	1.3325	0.3184	-0.2743	0.3515	-0.0913	0.7648
1365	-0.2033	0.3007	0.8616	1.3459	0.4038	-0.2743	0.4195	-0.0991	0.7654
1366	-0.2033	0.2803	0.8876	1.3230	0.5294	-0.2743	0.1643	-0.1791	0.7653
1367	-0.2033	0.2582	0.8640	1.3122	0.0476	-0.2743	0.2421	-0.0936	0.7655

* Att_1: Swing amplitude of driving chain [%]; Att_2: Ambient temperature [°C]; Att_3: Variable plasma motor temperature [°C] Att_4: Cabin temperature [°C]; Att_5: Active power [kW]; Att_6: Swing amplitude of drive direction tower [%]; Att_7: mean wind speed [m/s]; Att_8: angle of Blade 1 [°]; Att_9: Generator mean speed [rpm];

3.3. PCA-BPNN Method

Selecting attributes by experience is sometimes a troublesome work, but with PCA we can avoid selecting features by hand. The main difference between our proposed PCA-BPNN method and BPNN method is that we are now running PCA to all attribute to reduce the dimensions before sending data to the neural net. We employ PCA to reduce the dimension, but it is notable that generally we calculate the variance to decide

how many dimension we want. One commonly use value is variance=0.95 meaning that 95% of the information of the original data is retained. The PCA process tells that with 7 new features we achieve 0.95 variance as shown in figure 5.

By setting variance we can decide which dimension to reduce to. Setting variance to 0.95 in fact reduce the dimension to 7 as in figure 5, by engineer experience we choose 9 features to make prediction. The number of features are relatively close, showing the effectiveness of PCA process in another way.
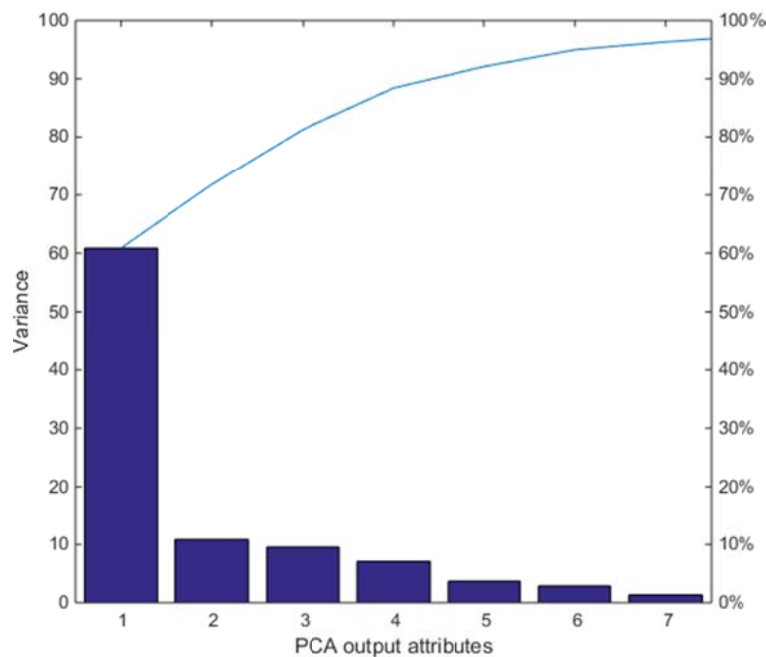
**Figure 5.** Variance of each new feature adding up to 0.95.

Table 2. Part of PCA process output data by setting variance=0.95.

id	Pca 1	Pca 2	Pca 3	Pca 4	Pca 5	Pca 6	Pca 7	Pca 8	Pca 9
1336	0.5155	-2.4869	0.9243	0.8254	0.6375	-0.2749	0.0447	-0.0952	0.5094
1337	0.0288	-2.8181	0.8864	0.9192	-0.3510	1.2242	0.1111	0.0726	0.0542
1338	0.3218	-2.5282	1.0117	0.7937	0.6525	-0.0845	0.0344	-0.1446	0.1896
1339	0.4693	-2.2445	1.0060	0.6798	0.7091	-0.2400	0.1860	-0.2416	0.3606
1340	1.5239	-1.1056	0.9546	-0.0980	0.8182	-0.5173	0.8853	-0.6393	0.6373
1341	2.4739	-0.8746	0.8512	-1.2886	1.1101	-0.5316	0.8119	-0.6925	0.6061
1342	3.7948	-0.4611	0.2985	-2.3830	0.1185	0.8588	1.0181	-0.6600	0.1866

The prediction alone with the actual value are shown in figure 6 (a), if no fault occurs i.e. before about point 4430 that prediction is closer to the actual, and when fault happens the two lines obviously separated. Also see figure 6 (b) the overall fluctuation is smaller than that in figure 4 (b) which mean less prediction error.
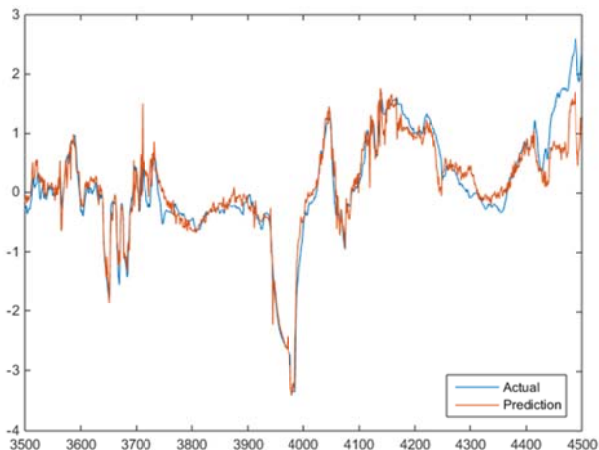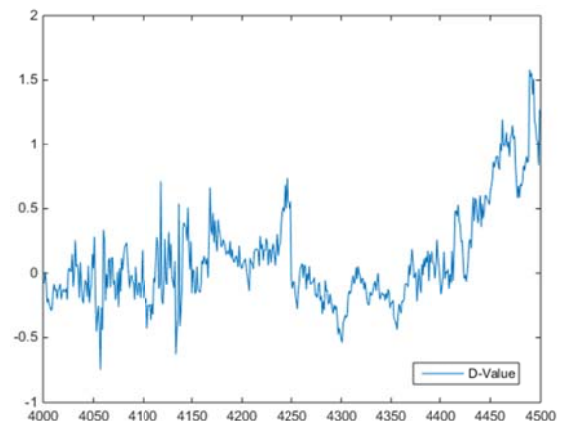
(b)

Figure 6. (a) Actual and predicted value of gearbox bearing temperature applying PCA- BPNN method (b) the corresponding D-value.

4. Discussion

4.1. D-value and Turbine Fault

Using steady state working D-Value of proposed

(a)

PCA-BPNN method, we set four warning lines as in figure 7. The red line and purple line stands for upper and lower bound of temperature respectively, if the D-value keeps exceeding for a set time, the machine will halt. The orange line and light blue green are used for warning, notify people earlier before the turbines have fault. We can find the D-value goes beyond the up or down alarm line for several times, this are some impulses which disappear quickly so that the alarm won't be triggered.
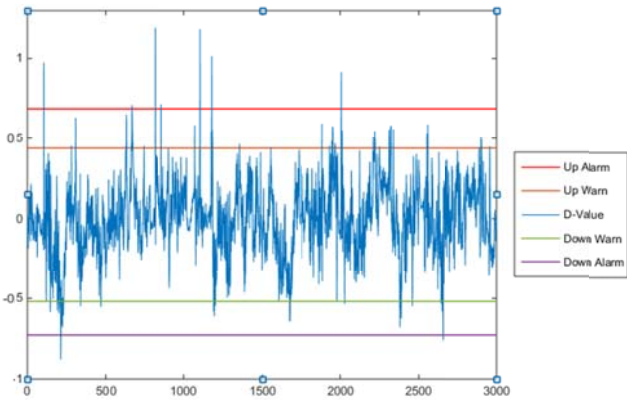

Figure 7. Using steady state working D-value to set warning lines.

Put the four warning together with the last 500 point in figure 8, we can see that the D-value keeps exceeding the up alarm line after point 4430, thus the fault can successfully be predicted with PCA-BPNN method.
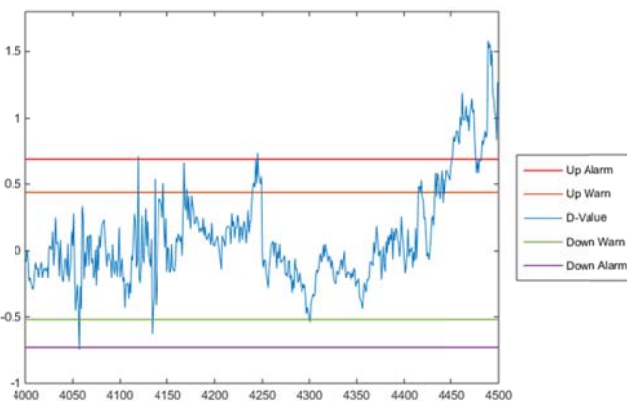

Figure 8. D-value and four critical line for fault warning.

4.2. PCA-BPNN Method and BPNN Method Comparison

For X_i represent the actual temperature of i-th point and Y_i for the prediction, the MSE is calculated by the formula (8). Note again that we do not use the MSE of prediction result because our test data set contains the fault data and that will affect the result. The MSE and corresponding accuracy are listed in table 3, compared to BPNN method, PCA-BPNN increase the accuracy by 3%.

Table 3. CV MSE of two different experiment.

Method	BPNN Method	PCA-BPNN Method
MSE	0.0687	0.0368
Accuracy	93.13%	96.32%

Also we can see in table 4 that the prediction accuracy increased by about 3% from 96.15% to 99.08% with respect to the mean predictive value and mean actual value, just like that of MSE value.

Table 4. Mean predictive value of two different experiment.

Method	Mean predictive value	Mean actual value	Accuracy
PCA-BPNN	60.37	60.93	99.08%
BPNN	58.59	60.93	96.15%

5. Conclusions

According to the results of data comparison, it can be seen that the prediction given by PCA-BPNN method is closer to the actual values than that without using PCA. The forecast error is smaller and most important, no need to select attributes by experience. The PCA-BPNN method can improve the learning accuracy at about 3% and promote the timeliness of the machine fault diagnosis. Combining PCA and neural network is efficient to pick up representative attributes of samples thus realize the redundancy reduction of attributes. That is, to achieve better operation and maintenance effects, the prediction accuracy of the proposed method is better than that of the neural network algorithm, it is more timely and accurate to determine the abnormal condition of the gearbox. The method can also be extended to the operation and maintenance of other machine faults, which has good practical value.

Acknowledgement

References

[1]   Azevedo D; Machado H D; Araújo, et al. A review of wind turbine bearing condition monitoring: State of the art and challenges. *RSER*, 2016, 56: 368-379.

[2]   Aziz, M. A.; Noura, H., & Fardoun, A. General review of fault diagnostic in wind turbines. *18th Mediterranean Conf. on Con. & Aut.*, 2010, *MED'10*, pp. 1302 - 1307 doi: 10.1109/med.2010.5547870.

[3]   Amirat Y, Benbouzid MEH, Al-Ahmar E, Bensaker B and Turri S. A brief status on condition monitoring and fault diagnosis in wind energy conversion systems. *RSER*, 2009, *13*, pp. 2629-36.

[4] Schlechtingen M, Santos IF and Achiche S. Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 1: System description. *App. Soft Com.* 2013, *vol. 14*, pp. 447–460.

[5] Aloraini, A., & Sayed-Mouchaweh, M. Graphical Model Based Approach for Fault Diagnosis of Wind Turbines. *International Conference on Machine Learning and Applications,* 2014, pp. 614 - 619.

[6] Santos P; Villa LF; Reñones A; Bustillo A and Maudes J. An SVM-Based Solution for Fault Detection in Wind Turbines. *Sensors*, 2015, *vol. 15*, 5627-48.

[7] Odofin S, Gao Z and Sun K. Robust fault estimation in wind turbine systems using GA optimisation. Industrial Informatics (INDIN), *2015 IEEE 13th International Conference on. IEEE*, 2015, pp. 580-5.

[8] Lan Q; Zhao PD; Wang ML. Fault Diagnosis of Wind Turbines Based on Improved Neural Network. *Advanced Materials Research, Trans Tech Publ*, 2014, pp. 78-83.

[9] Malik H; Mishra S. Application of Probabilistic Neural Network in Fault Diagnosis of Wind Turbine Using FAST, TurbSim and Simulink. *Procedia Computer Science*. 2015; *vol. 58*, pp. 186-93.

[10] Xu Liu, Mei Li et al, A Predictive Fault Diagnose Method of Wind Turbine Based on K-means Clustering and Neural Networks. *JIT*, 2016, *vol. 17* (online), doi: 10.6138/JIT.2016.17.7.20151027i.

[11] Jolliffe I. Principal component analysis. *Wiley Online Library*, 2002.

[12] Mcmurtrey SD. Training and optimizing distributed neural networks using a genetic algorithm. *Nova Southeastern University*, 2010.