Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

Support Vector Machine for mechanical faults classification*

JIANG Zhi-qiang (蒋志强)[1], FU Han-guang (符寒光)†[2], LI Ling-jun (李凌君)[3]

([1]*Zhengzhou Aeronautical Institute of Industry Management, Zhengzhou 450015, China*)
([2]*Beijing Researching Institute for Metallurgical Equipment, Beijing 100029, China*)
([3]*School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China*)
†E-mail: fhg64@263.net

Received Oct. 3, 2003; revision accepted Feb. 27, 2004

Abstract: Support Vector Machine (SVM) is a machine learning algorithm based on the Statistical Learning Theory (SLT), which can get good classification effects with a few learning samples. SVM represents a new approach to pattern classification and has been shown to be particularly successful in many fields such as image identification and face recognition. It also provides us with a new method to develop intelligent fault diagnosis. This paper presents an SVM based approach for fault diagnosis of rolling bearings. Experimentation with vibration signals of bearing was conducted. The vibration signals acquired from the bearings were directly used in the calculating without the preprocessing of extracting its features. Compared with the Artificial Neural Network (ANN) based method, the SVM based method has desirable advantages. Also a multi-fault SVM classifier based on binary classifier is constructed for gear faults in this paper. Other experiments with gear fault samples showed that the multi-fault SVM classifier has good classification ability and high efficiency in mechanical system. It is suitable for on line diagnosis for mechanical system.

Key words: Support Vector Machine (SVM), Fault diagnosis, Multi-fault classification, Intelligent diagnosis
doi:10.1631/jzus.2005.A0433 **Document code:** A **CLC number:** TH17; TP18

INTRODUCTION

Support Vector Machine (SVM) is a relatively new soft computing method based on statistical learning theory presented by Vapnik (1995). In SVM, original input space is mapped into a high dimensional dot product space called feature space in which the optimal hyperplane is determined to maximize the generalization ability of the classifier. The optimal hyperplane is found by exploiting a branch of mathematics, called optimization theory, and respecting the insights provided by the Statistical Learning Theory (SLT) (Vapnik, 1995). SVM is widely used in recent years in many realms, such as face recognition (Guo *et al*., 2001), nonlinear equalization (Sebald and Bucklew, 2000) and spam categorization (Drucker *et al*., 1999). In fault diagnosis area some researches have also been done (Gao *et al*., 2002; Rychetsky *et al*., 1999). SLT is based on structure risk minimization and has good learning ability even though fewer learning samples are used. SVM based classifier is claimed to have better generalization properties than Artificial Neural Network (ANN) based classifiers in mechanical system fault diagnosis (Rychetsky *et al*., 1999). In addition to this, SVM based classification is interesting, because its efficiency does not depend on the number of features of classified entities. This property is very useful in fault diagnostics because the number of features to be chosen to be the base of fault classification is thus not limited, which make it possible to compute directly using original data without preprocessing them to extract their features.

A binary classifier was constructed in this paper and used to classify the bearing faults. The construction of the binary classifier only requires original

* Project (No. 0424260002) supported by the Natural Science Foundation of Henan Province, China

signals without preprocessing them for extracting their features. However, feature extraction is always done in ANN method, which increases the computation cost. The result of SVM method is more accurate than that of the ANN method and SVM's high computing efficiency indicates the potential of SVM techniques in machinery fault diagnosis.

SVM was originally designed for binary classification, which is not suitable for fault diagnosis, because it has several fault classes (not only one fault) in addition to health condition (normal class). So, it is necessary to develop a method to deal with a multi-class classification problem. In this paper, a multi-fault classifier based on binary classifier is proposed. This classifier is applied to classify 5 faults of a gearbox. The result showed that this classifier has good classification ability and high efficiency and is suitable for on-line fault diagnosis of mechanical system.

BINARY SVM CLASSIFICATION

Binary SVM classification algorithm (Vapnik, 1995; Cristianini and Shawe-Taylor, 2000)

Suppose we have N given observations, each consisting of a pair of data: a data vector $\boldsymbol{x}_i \in R^d$, $i=1,\ldots, N$ and a class label $y_i \in \{+1,-1\}$ for each vector. We say $\boldsymbol{x}_i$ belongs to class I, if $y_i=+1$ and $\boldsymbol{x}_i$ belongs to class II, if $y_i=-1$. These data pairs build the training sets. For linearly separable data, we can determine a hyperplane $f(\boldsymbol{x})$ that separates the data. For a separating hyperplane $f(\boldsymbol{x})\geq 0$, if the input $\boldsymbol{x}$ belongs to a positive class, and $f(\boldsymbol{x})<0$, if $\boldsymbol{x}$ belong to a negative class.

$$f(\boldsymbol{x})=\boldsymbol{w}\cdot\boldsymbol{x}+b=\sum_{j=1}^{d}\boldsymbol{w}_j\boldsymbol{x}_j+b \tag{1}$$

$$y_i f(\boldsymbol{x}_i)=y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i+b)\geq 0,\ i=1,\ldots,N \tag{2}$$

where $\boldsymbol{w}$ is a d-dimensional vector and b is a scalar. $\boldsymbol{w}\cdot\boldsymbol{x}$ is inner product of $\boldsymbol{w}$ and $\boldsymbol{x}$.

If we additionally require that $\boldsymbol{w}$ and b be such that the point closest to the hyperplane has a distance of $1/|\boldsymbol{w}|$, then Eq.(2) should be rewritten as:

$$y_i(\boldsymbol{w}\cdot\boldsymbol{x}+b)\geq 1,\ i=1,\ \ldots,\ N \tag{3}$$

The separating hyperplane that has the maximum distance between the hyperplane and the nearest data, i.e. the maximum margin, is called the optimal separating hyperplane. The generalization ability is maximized with the optimal hyperplane. An example of optimal separating hyperplane of two datasets is presented in Fig.1, where $\boldsymbol{H}$ is the optimal separating hyperplane. The optimal hyperplane can be obtained by solving the following convex quadratic optimization problem (Vapnik, 1995):

$$\text{minimize } \frac{1}{2}\|\boldsymbol{w}\|^2 \tag{4}$$

subject to

$$y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i+b)\geq 1,\ i=1,\ \ldots,\ N \tag{5}$$

This problem can be transformed into the equivalent Lagrange dual problem as:

$$\text{maximize } Q(\boldsymbol{\alpha})=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i,k=1}^{N}\alpha_i\alpha_k y_i y_k \boldsymbol{x}_i\cdot\boldsymbol{x}_k \tag{6}$$

$$\text{subject to } \sum_{i=1}^{N}y_i\alpha_i=0,\ \alpha_i\geq 0,\ i=1,\ldots,N \tag{7}$$

where $\boldsymbol{\alpha}=(\alpha_1, \alpha_2, \ldots, \alpha_i)$ is the Lagrange multiplier. Each sample has corresponding α_i, $i=1, \ldots, N$. Those samples for which $\alpha_i>0$ are called support vectors, and are ones that the equality condition in Eq.(4) holds. All other training samples having $\alpha_i=0$ can be removed from the training set without affecting the optimal hyperplane.

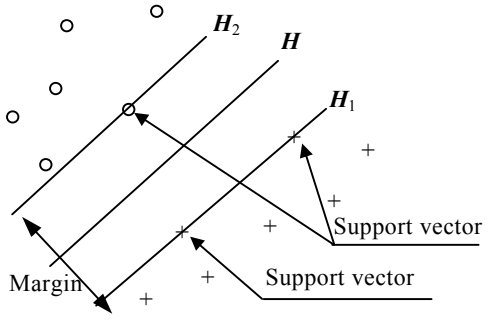

Fig.1 Optimal separating plane

Assume that optimal solution of $\boldsymbol{\alpha}$ for the dual problem is $\boldsymbol{\alpha}^*$. Then the solution of $\boldsymbol{w}$ and b are $\boldsymbol{w}^*$ and b^*, which can be given by:

$$\boldsymbol{w}^* = \sum_{i=1}^{N} \alpha_k^* y_i \cdot \boldsymbol{x}_i = \sum_{\text{support vectors}} \alpha_i^* y_i \cdot \boldsymbol{x}_i, \tag{8}$$

$$b^*=1-\boldsymbol{w}^* \cdot \boldsymbol{x}_i \text{ for } \boldsymbol{x}_i \text{ with } y_i=1 \tag{9}$$

After training, the classifier can be used to classify an unknown data example $\boldsymbol{x}$ by the decision functions:

$$f(\boldsymbol{x})=\text{sign}\left(\sum_{j=1}^{N} \alpha_j^* y_j \boldsymbol{x}_j \cdot \boldsymbol{x} + b^*\right) \tag{10}$$

$$\boldsymbol{x} \in \begin{cases} \text{Class I, if } f(\boldsymbol{x}) = +1 \\ \text{Class II, if } f(\boldsymbol{x}) = -1 \end{cases} \tag{11}$$

SVM is a non-linear kernel-based classifier, which maps the data to be classified from original data space $\boldsymbol{X}$, onto a space, where the data can be linearly classified. The space is called a feature space $\boldsymbol{F}$. This is depicted in Fig.2. Now using the non-linear vector function $\boldsymbol{\Phi}(\boldsymbol{x})=(\Phi_1(\boldsymbol{x}), \ldots, \Phi_l(\boldsymbol{x}))$, that maps the d-dimensional input vector $\boldsymbol{x}$ into the l-dimensional feature space, the linear decision function in dual form is given by:

$$f(\boldsymbol{x})=\text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i \boldsymbol{\Phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\Phi}(\boldsymbol{x}) + b^*\right) \tag{12}$$

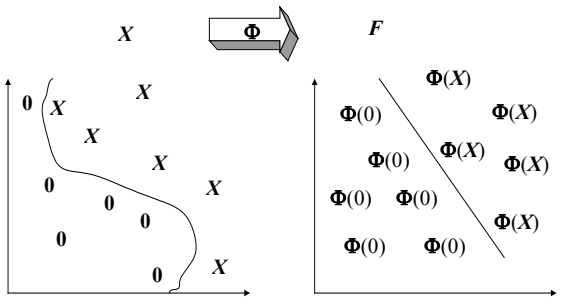

Fig.2 Mapping from the data space X to the feature space F

Notice that in Eq.(11) as well in the optimization problem (5), the data occur only in inner products. So in the higher dimensional space (feature space) we only deal with the data in the form of inner product $\boldsymbol{\Phi}(\boldsymbol{x}) \cdot \boldsymbol{\Phi}(\boldsymbol{z})$. If the dimension of $\boldsymbol{F}$ is very large, then this could be difficult, or very expensive computationally. However, if it is possible to find a kind of function to calculate inner products of feature space in original data space, this function is called a kernel function, $\boldsymbol{K}(\boldsymbol{x},\boldsymbol{z})=\boldsymbol{\Phi}(\boldsymbol{x}) \cdot \boldsymbol{\Phi}(\boldsymbol{z})$. Then we can use this kernel function in place of $\boldsymbol{\Phi}(\boldsymbol{x}) \cdot \boldsymbol{\Phi}(\boldsymbol{z})$ everywhere in the optimization problem, and never need to know explicitly what $\boldsymbol{\Phi}$ is. Using a kernel function, the decision function will be:

$$f(\boldsymbol{x})=\text{sign}\left[\sum_{\text{support vectors}} \alpha_i^* y_i \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}) + b^*\right], \tag{13}$$

$$\boldsymbol{x} \in \begin{cases} \text{Class I, if } f(\boldsymbol{x}) = +1 \\ \text{Class II, if } f(\boldsymbol{x}) = -1 \end{cases} \tag{14}$$

However, not all kernels correspond to inner products in some feature space $\boldsymbol{F}$. With a so-called Mercer's theorem it is possible to find out, whether a kernel $\boldsymbol{K}$ depicts an inner product in that space where $\boldsymbol{\Phi}$ is mapped.

Some typical kernel functions are:

Polynomial function

$$K(\boldsymbol{x},\boldsymbol{y})=(\boldsymbol{x} \cdot \boldsymbol{y}+1)^d, \; d=1,2, \ldots \tag{15}$$

Radial basis function

$$K(\boldsymbol{x},\boldsymbol{y})=\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right) \tag{16}$$

Sigmoid function

$$K(\boldsymbol{x},\boldsymbol{y})=\tanh[b(\boldsymbol{x} \cdot \boldsymbol{y})-c] \tag{17}$$

Application of binary SVM classification

Some vibration signals of roll bearings have been obtained from the test-bed. The experiment includes the following steps:

1. The classifier is trained by 20 samples of the data, in which there are 10 fault samples and 10 healthy samples. After being trained, the classifier is used to classify the 6 test samples, which include 3 fault samples (numbers: 1–3) and 3 healthy samples (numbers: 4–6). We use the original data and linear kernel function $K(\boldsymbol{x},\boldsymbol{y})=\boldsymbol{x} \cdot \boldsymbol{y}$ to construct a classifier. Define: y=+1 for fault case and y=−1 for healthy case. The classed results by this classifier are shown in Table 1. It is clear that the result is correct. Where the distance to $\boldsymbol{H}$ is calculate by:

$$\sum_{i=1}^{N} \alpha_i^* y_i K(\boldsymbol{x}_i \cdot \boldsymbol{y}_i) + b^* \tag{18}$$

2. Compare the SVM based classifier with neural network based classifier. We use the ANN based method to classify the bearing faults for the same samples as above in Step (1). Here we employ the function in Matlab ANN toolbox to construct a 3 layers BP network. According to (Yuan, 2000), the input layer node number is the same as the training samples dimensions d; the hidden node number is $2d$+1, and the output node number is one. The output is nearly 1, if the bearing is in the fault state. On the other hand, the output is nearly –1, if the object is in the healthy state. The transfer function used in the network is:

$$f(\boldsymbol{x})=\frac{2}{1+\exp(-2\boldsymbol{x})}-1 \tag{19}$$

The variable learning rate method is used to train the network.

According to (Yuan, 2000), the net structure will be very complex if we directly use the original data in the calculation without preprocessing the sample for extracting its features. Because of the long calculating time, this network model will not be suitable for the actual engineering, even not suitable for our computing in this paper. So it is necessary to calculate with the features of the sample. It should be pointed out that this comparison is more rigorous for SVM.

We use the following features as input vector:

$$\boldsymbol{x}_i=[MAX(x_i)\ STD(x_i)\ RMS(x_i)\ PRO(x_i)\ SKE(x_i)\ KUR(x_i)] \tag{20}$$

where $MAX(x_i)$ is the peak value of x_i, $STD(x_i)$ is the standard deviation, $RMS(x_i)$ is the virtual value, $PRO(x_i)$ is the absolute mean value, $SKE(x_i)$ is skewness, $KUR(x_i)$ is the kurtosis.

Therefore, the input layer has 6 units, the hidden layer has 13 units, and the output layer has 1 unit. The result classed by ANN is also shown in Table 1, from which it is clear that both SVM and ANN can give good classification results in this condition (20 training samples). But the computing efficiencies are different. As for the same computer (CPU: PIII800, RAM: 128 M) the training and classing time used by ANN is 4.677 s, whereas that by SVM is 0.340 s. ANN method will cost long time if the preprocessing time is considered.

3. Keep all the conditions except the training samples number unchangeableness. A new classifier is reconstructed by decreasing the training sample numbers from 20 to 6. There are 3 fault samples and 3 healthy samples in these 6 training samples. The 6 test samples are also the same as the samples in Step (1). In these 6 test samples, the results classed by SVM and ANN are shown in Table 2.

Table 2 clearly shows that, SVM method can give us the right class result in this case (with only 6 training samples), whereas the ANN method cannot, as shown in Table 2 the No.1 sample was classed failure by ANN method. It proves that the SVM method has better classification ability than ANN method in the case of fewer training samples. Furthermore, it is not necessary for SVM to preprocess the signal for extracting its features, which provide us more advantages in actual practice.

Table 1 Compare result of SVM with ANN in the case of more samples

Samples number	1	2	3	4	5	6
Fault type	Failure	Failure	Failure	Healthy	Healthy	Healthy
Classed by SVM $f(x_i)$	+1	+1	+1	–1	–1	–1
SVM: distance to H	5.1864	2.7701	1.9680	–1.2874	–1.3125	–1.3243
Classed by ANN	0.4288	0.7950	0.8486	–1.1609	–1.1054	–1.0686

Table 2 Compare result of SVM with NNT in the case of fewer samples

Samples number	1	2	3	4	5	6
Fault type	Failure	Failure	Failure	Healthy	Healthy	Healthy
Classed by SVM $f(x_i)$	+1	+1	+1	–1	–1	–1
SVM: distance to H	3.1722	1.5323	0.6496	–1.2645	–1.1340	–1.2765
Classed by ANN	–0.1702	1.4774	0.8464	–1.0150	–1.0917	–1.1180

MULTI-CLASS SVM CLASSIFICATION ALGORITHM

Multi-class SVM classification algorithm

SVM was originally designed for binary classification. Multi-class (K-class, k>2) classification can be obtained by the combination of binary classification. There is a relationship between binary classification and multi-classification. Suppose we have a classable K-class problem; then it must be separated from each other by binary classification; On the contrary, in a K-class event, it must be K-class classable if any two classes of it is separable. We can construct a multi-class classifier by combining several binary classifiers. Several methods have been proposed (Hsu and Lin, 2002). Some of them are: "one-against-one", "one-against-all", and directed acyclic graph SVM (DAGSVM). In literature (Hsu and Lin, 2002), Hsu and Lin gave a comparison of these methods and pointed out that the "one-against-one" method is more suitable for practical use than other methods. In this paper we use "one-against-one" method to classify the 5 faults of the gears.

For K-class event, the "one-against-one" method, construct $M=C_k^2=k(k-1)/2$ classifiers, where each is trained on data from the ith and the jth class, we solve the following binary classification problem:

$$\min_{w^{ij},b^{ij},\xi^{ij}} \frac{1}{2}(\boldsymbol{w}^{ij})^{\mathrm{T}}\boldsymbol{w}^{ij}+C\sum_t \xi_n^{ij}(\boldsymbol{w}^{ij})^{\mathrm{T}}$$
$$(\boldsymbol{w}^{ij})^{\mathrm{T}}K(x_n)+b^{ij}\geq 1-\xi_n^{ij},\ \text{if } y_n=i$$
$$(\boldsymbol{w}^{ij})^{\mathrm{T}}K(x_n)+b^{ij}\leq -1=\xi_n^{ij},\ \text{if } y_n=j$$
$$\xi_n^{ij}\geq 0 \qquad (21)$$

where $K(x_n)$ is kernel function, (x_n, y_n) is a ith or jth training sample.

There are different methods for doing the future testing after all M classifiers are constructed. After some tests, we decided to use the following voting strategy: if $\mathrm{sign}((\boldsymbol{w}^{ij})^{\mathrm{T}}\phi(x)+b^{ij})$ decides x is in the ith class, then the vote for the ith class is increased by one. Otherwise, the jth class is increased by one. Then we predict x is in the class with the largest vote. The voting approach described above is also called the "Max Wins" strategy, Fig.3 is its flow chart.

This method needs $k(k-1)/2$ classifiers. Although the number is more than that of "one-against-all", which needs k classifiers (as we know: $k(k-1)/2\geq k$ when $k\geq 3$), as each problem is smaller (only data from two classes). For the "one-against-all", each classifier is trained with all the data, so the total training time of "one-against-one" may not be more than that of "one-against-all" method (Hsu and Lin, 2002). In case that two classes have identical votes and both are the max, how can we make the decision? In this case it may not be a good strategy. To deal with this problem we simply selected the one with the small index. So it is necessary to number the important or more often appearing fault with small index.

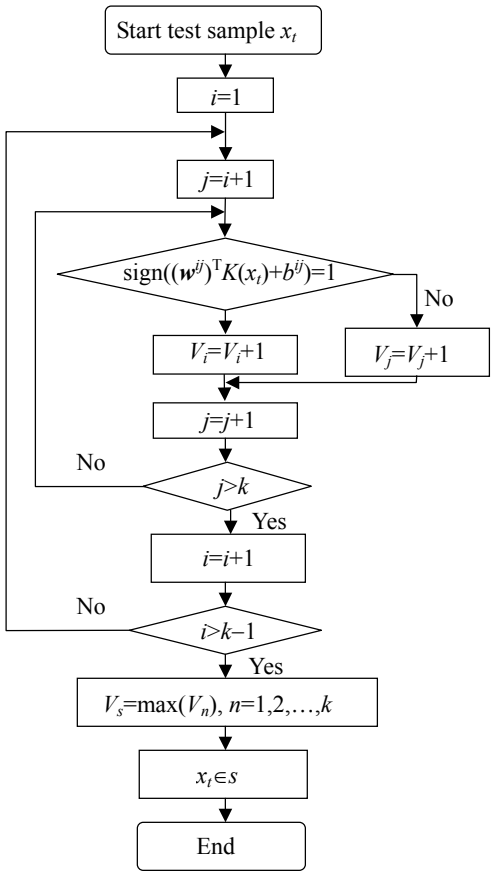

Fig.3 Flow chart of voting strategy

Application of multi-class SVM classification

In this section, the multi-class SVM classification discussed above is applied to diagnosis of the gear faults. The fault types and numbers are shown in Table 3. Here we consider healthy as a special fault

class. As the principle pointed out in Section 3.1, we make the number small for the fault which occurs more often and the number large for the fault which seldom occurs. It is considered that misdiagnose of a healthy machine as a failure one will result in lesser loss than otherwise. So the healthy is arranged at last with largest number.

Table 3 Number of faults

Fault type	Fault number
Eccentricity	1
Misalignment	2
Wear out	3
Pitting	4
Healthy	5

Some waveforms of the fault gears are shown in Fig.4 (only 2 types of the faults are shown for lack of space).

The multi-fault classifier is trained by 20 training samples, which include 5 fault classes, with 4 samples for each fault class. The faults and fault number are shown in Table 3. We compute directly using original data without preprocessing the signal to extract its features. There are 5 fault classes (include healthy class), so n=5, and we must design $k(k-1)/2$=10 binary classifiers. Choosing a kernel function for classifiers has considerable impact on classification results. However, there does not exist general rules for choosing the kernel function, the best kernel function depends on the classification problem considered. In this experiment, several kernel functions were compared, as a result, the linear kernel function was found to be better. So we used linear kernel function in this paper. The upper bound C for the Lagrange multipliers is chosen to be 10.

We used the other 30 samples to test the classifier. The 30 samples included 5 fault classes, with each class having 6 samples. The sample number and its fault type are shown in Table 4. The result classed by this classifier is also shown in Table 4. Table 3 and Table 4 show that the class results were all correct. This proved that this classifier has good classification ability for multi-fault classification in mechanical system. The computer used in this experiment was PIII800 CPU, 128 M RAM. The training time in training the 20 samples was 0.191 s. The classing time for 30 samples was 0.220 s. It was obvious that the computing efficiency was high. So the classifier is suitable for mechanical system on line fault diagnosis. And this method can get good result without preprocessing the signals to extract their features, which makes the fault diagnosis very simple.

Table 4 Classification of gear faults by SVM multi-fault classifier

Samples numbers	Fault type	Classification result
1~6	Eccentricity	1
7~12	Misalignment	2
13~18	Wear out	3
19~24	Pitting	4
25~30	Healthy	5

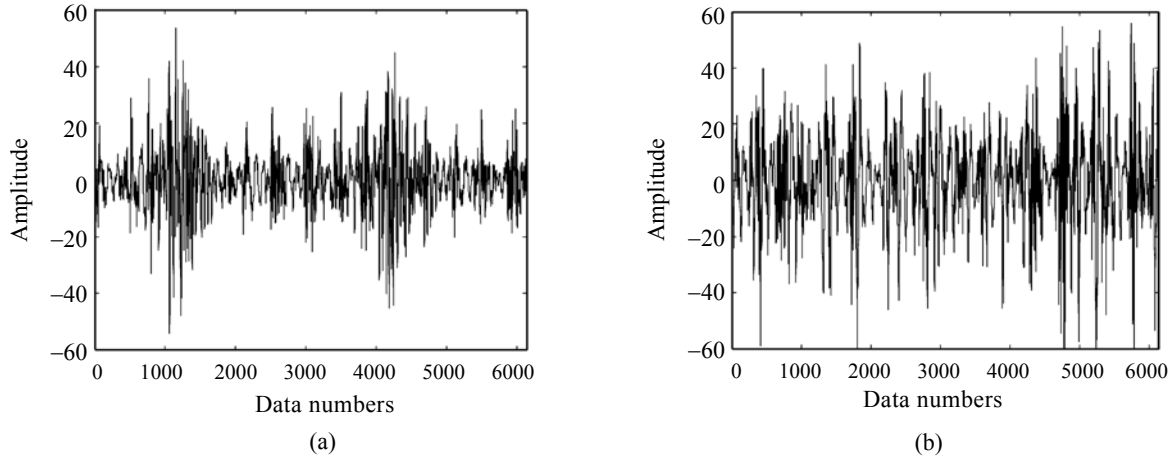

Fig.4 Signal waveforms of fault gears. (a) Eccentricity; (b) Wear out

CONCLUSION

SVM is a new machine learning method developed in recent years, which requires relatively fewer learning samples. The experiment results in this work indicated that the SVM method has better effectiveness than traditional artificial neural network. The multi-class faults classifier designed in this study has many advantages: simple algorithm, good classification and high efficiency. It is very suitable for online monitoring and diagnosis. SVM provides us a new and useful method for developing intelligent diagnosis.

References

Cristianini, N., Shawe-Taylor, J., 2000. Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, p.254-269.

Gao, J.F., Shi, W.G., Tan, J.X., Zhong, F.J., 2002. Support Vector Machines Based Approach for Fault Diagnosis of Valves in Reciprocating Pumps. IEEE Canadian Conference on Electrical & Computer Engineering, p.1622-1628.

Guo, G.D., Li, S.Z., Chan, K.L., 2001. Support Vector Machine for face recognition. *Image and Vision Computing*, **19**:631-638.

Drucker, H., Wu, D., Joksons, D.W., 1999. Support Vector Machine for spam categorization. *IEEE Trans on Neural Networks*, **10**:1048-1054.

Hsu, C.W., Lin, C.J., 2002. A comparison of methods for multiclass Support Vector Machines. *IEEE Trans on Neural Networks*, **13**:415-425.

Rychetsky, M., Ortmann, S., Glesner, M., 1999. Support Vector Approaches for Engine Knock Detection. International Joint Conference on Neural Networks. IJCNN 99. Washington, USA, p.969-974.

Sebald, D.J., Bucklew, J.A., 2000. Support Vector Machine techniques for nonlinear equalization. *IEEE Trans on Signal Processing*, **48**:3217-3226.

Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, New York, p.157-173.

Yuan, Z.G., 2000. Artificial Neural Network and Its Application. Tsinghua University Press, Beijing, p.177-205 (in Chinese).