Fault Diagnosis of Rotating Machinery Based on Multi-Class Support Vector Machines

Bo-Suk Yang*, Tian Han, Won-Woo Hwang
School of Mechanical Engineering, Pukyong National University, San 100, Yongdang-dong, Nam-gu, Busan 608-739, Korea

Support vector machines (SVMs) have become one of the most popular approaches to learning from examples and have many potential applications in science and engineering. However, their applications in fault diagnosis of rotating machinery are rather limited. Most of the published papers focus on some special fault diagnoses. This study covers the overall diagnosis procedures on most of the faults experienced in rotating machinery and examines the performance of different SVMs strategies. The excellent characteristics of SVMs are demonstrated by comparing the results obtained by artificial neural networks (ANNs) using vibration signals of a fault simulator.

Key Words : Fault Diagnosis, Support Vector Machine, Rotating Machinery, Multi-Class Classification

1. Introduction

Fault diagnosis of rotating machinery is increasingly becoming important in manufacturing industry due to the demand to keep up with production and the need to have highly reliable machinery. However, many of the techniques available presently require a great deal of expert knowledge to apply them successfully. Therefore simpler approaches are needed to allow relatively unskilled operators to make reliable decisions without the need of a specialist to examine the data and diagnose the problems. Hence, there is a demand to incorporate techniques that can make decisions on the health of the machine automatically and reliably. By learning from known problems, such as unbalance, shaft misalignment and bearing defects, fault diagnosis can be carried out. Artificial neural networks (ANNs) and support vector machines (SVMs) are popularly used as diagnostic tools in machine health condition monitoring.

ANNs have been applied in automated detection and diagnosis of machine conditions. The techniques can be treated as generalization/classification problems and are based on learning pattern from empirical data. However, traditional neural network approach has limitations on generalization and leads to models that can over fit the training data. This deficiency is partly due to the optimization algorithms used in the ANNs for the selection of parameters and the statistical measurements used to select the model. Many incremental and competitive learning networks were proposed to handle the problems mentioned above and to increase the classification performance. In the literature, self-organizing feature map (SOFM) (Kohonen, 1995), learning vector quantization (LVQ) (Kangas and Kohonen, 1996), radial basis function (RBF) (Sundararajan, 1999) and adaptive resonance theory (ART) (Capenter and Grossberg, 1988) networks can be seen as the most basic schemes in competitive learning network used in machine fault diagnosis.

SVMs are relatively new computing methods

* Corresponding Author,
E-mail : bsyang@pknu.ac.kr
TEL : +82-51-620-1604; **FAX** : +82-51-620-1405
School of Mechanical Engineering, Pukyong National University, San 100, Yongdang-dong, Nam-gu, Busan 608-739, South Korea. (Manuscript **Received** October 22, 2004; **Revised** January 25, 2005)

which are based on statistical learning theory presented by Vapnik (1999). SVMs have recently attracted a great deal of interest in the machine diagnostic community for their high accuracy and good generalization capability (Burges, 1998). The main difference between ANNs and SVMs is in the principle of risk minimization. ANNs incorporate recursive algorithms that adjust system parameters such as weights during the learning process. These algorithms adjust system parameters based on a risk function such as empirical risk minimization (ERM). During the learning process, the SVM uses a risk function known as structural risk minimization (SRM) which has been shown to be superior to ERM. The ERM is based just on minimizing the error of the training data itself. If the training data is sparse and/or not representative of the underlying distribution, then the system will be poorly trained and hence have limited classification performance (Vapnik, 1992). The SRM allows the algorithm designer to take into account the sparseness of the data and minimizes the error of the upper bound of an expected risk. The difference in risk minimization leads to better generalization performance for SVMs than ANNs.

SVM-based classification is a modern machine learning method that is rarely used in fault diagnosis even though it has given superior results in image identification and face recognition (Osuna et al., 1997; Burges, 1998). The possibilities of SVMs using binary classification in machine fault detection of damaged gears (Jack and Nandi, 2002), rolling element bearings (Samanta, 2004) and reciprocating compressors (Yang et al., 2005) are being attempted only recently. There are still limited applications in 'real' engineering situation using the technique. One of the reasons for the low popularity of SVM is essentially a two-class classifier, whereas formulations of other classification structures like neural network classifiers allow straightforward extension to multi-class classification problems which is often faced in fault diagnosis. A direct multi-class extension of SVM usually leads to a very complex optimization problem and tedious computations. Therefore, multi-class problems are often solved by training several binary SVM classifiers and fusing the outputs of the classifiers to find the global classification decision (Suykens et al., 2002).

The goal of this paper is to present a fault diagnosis scheme based on multi-class SVMs for a rotating machinery. This paper offers a comparison between two kinds of algorithms, the SVMs and ANNs such as the SOFM (Yang et al., 2000a), LVQ (Yang et al., 2000b) and RBF (Yang et al., 2002). Same data obtained from a fault simulator were used to train and test these algorithms.

2. Support Vector Machines (SVMs)

SVM is a relatively new computational learning method based on the statistical learning theory presented by Vapnik (1999). In SVM, original input space is mapped into a high-dimensional dot product space called a feature space, and in the feature space the optimal hyperplane is determined to maximize the generalization ability of the classifier. The optimal hyperplane is found by exploiting the optimization theory, and respecting insights provided by the statistical learning theory. For detailed tutorials on the subject the reader can refer to references (Vapnik, 1999; Burges, 1998; Muller, 2001) and references cited therein. In this section a brief outline of the method will be described.

2.1 Binary classification

The SVM attempts to create a line or hyperplane between two sets of data for classification. In a two-dimensional situation, the action of the SVM can be explained easily without any loss of generality. Figure 1 shows how to classify a series of points into two different classes of data, class A (circles) and class B (squares). The SVM attempts to place a linear boundary represented by a solid line between the two different classes and orients it in such a way that the margin represented by dotted lines is maximized. The SVM tries to orient the boundary such that the distance between the boundary and the nearest data point in each class is maximal. The boundary is then

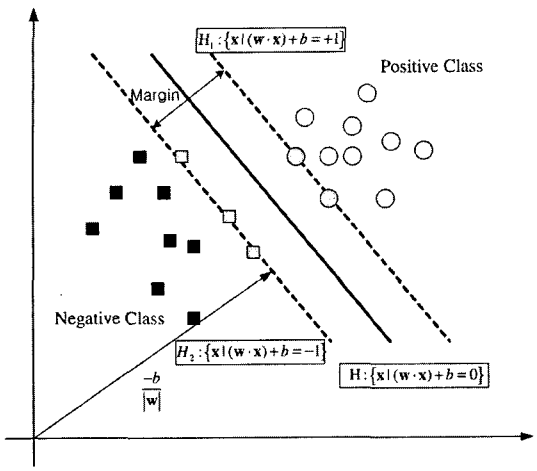


Fig. 1 An example of classification of two classes by SVM

placed in the middle of this margin between the two points. The nearest data points are used to define the margins and are known as support vectors (SVs) represented by gray circle and square. Once the SVs are selected, the rest of the feature sets can be discarded, since the SVs have all the necessary information for the classifier (Samanta, 2004).

Let (x_i, y_i), with $i=1, \cdots, N$; be a training set S; each $x_i \in R^N$ belongs to a class by $y_i \in \{-1, 1\}$. The goal is to define a hyperplane which divides S, such that all the points with the same label are on the same side of the hyperplane while maximizing the distance between the two classes A, B and the hyperplane. The boundary can be expressed as follows:

$$\mathbf{w}\cdot\mathbf{x}+b=0,\ \mathbf{w}\in R^N,\ b\in R \tag{1}$$

where the vector $\mathbf{w}$ defines the boundary, $\mathbf{x}$ is the input vector of dimension N and b is a scalar threshold. At the margins, where the SVs are located, the equations for classes A and B, respectively, are as follows:

$$\mathbf{w}\cdot\mathbf{x}+b=1,\ \mathbf{w}\cdot\mathbf{x}+b=-1 \tag{2}$$

As SVs correspond to the extremities of the data for a given class, the following decision function can be used to classify any data point in either class A or B:

$$f(\boldsymbol{x})=\text{sign}(\mathbf{w}\cdot\mathbf{x}+b) \tag{3}$$

For Gaussian kernels every finite training set is linearly separable in feature space (Burges, 1998). Then the optimal hyperplane separating the data can be obtained as a solution to the following optimization problem (Scholkopf, 1997):

find $\mathbf{w}\in R^N$ to minimize

$$\tau(\mathbf{w})=1/2\|\mathbf{w}\|^2 \tag{4}$$

subject to

$$y(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 1\ \ (i=1, 2, \cdots, N) \tag{5}$$

where N is the number of training sets.

However, if the only possibility to access the feature space is via dot products computed by the kernel, we cannot solve Eq. (4) directly since $\mathbf{w}$ lies in that feature space. But it turns out that we can get rid of the explicit usage of $\mathbf{w}$ by forming the dual optimization problem (Scholkopf, 1997). Introducing Lagrange multipliers $\alpha_i \geq 0$, $i=1, 2, \cdots, N$, one for each of the constraints in Eq. (5), we obtain the following Lagrangian:

$$L(\mathbf{w}, b, \boldsymbol{\alpha})=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{N}\alpha_i y_i(\mathbf{w}\cdot\mathbf{x}_i-b)+\sum_{i=1}^{N}\alpha_i \tag{6}$$

The task is to minimize Eq. (6) with respect to $\mathbf{w}$ and b, and to maximize it with respect to α_i. At the optimal point, we have the following saddle point equations:

$$\frac{\partial L}{\partial \mathbf{w}}=0,\ \frac{\partial L}{\partial b}=0 \tag{7}$$

which translate into

$$\mathbf{w}=\sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i,\ \sum_{i=1}^{N}\alpha_i y_i=0 \tag{8}$$

From the first equation of Eq. (8), we find that $\mathbf{w}$ is contained in the subspace spanned by $\mathbf{x}_i$. By substituting Eq. (8) into Eq. (6), we get the dual quadratic optimization problem:

Maximize

$$L_D(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{ij}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j \tag{9}$$

subject to

$$\alpha_i\geq 0 (i=1, 2, \cdots, N),\ \sum_{i=1}^{N}\alpha_i y_i=0 \tag{10}$$

Thus, by solving the dual optimization problem, one obtains the coefficient α_i which is required to express the $\mathbf{w}$ to solve Eq. (4). This leads to the nonlinear decision function

$$f(\mathbf{x})=\text{sign}\left(\sum_{i=1}^{N}\alpha_i y_i(\mathbf{x}_i\cdot\mathbf{x})+b\right) \tag{11}$$

In cases where the linear boundary in the input spaces are not enough to separate the two classes properly, it is possible to create a hyperplane that allows a linear separation in the higher dimension. In SVMs, this is achieved through the use of a transformation $\Phi(\mathbf{x})$ that converts the data from an N-dimensional input space to Q-dimensional feature space:

$$\mathbf{s}=\Phi(\mathbf{x}) \tag{12}$$

where $\mathbf{x}\in R^N$ and $\mathbf{s}\in R^Q$.

The SVM classifier is to take the input feature set and map it into a higher dimensional space using a non-linear function called a kernel. The reasoning for mapping into a higher dimensional space is based on a theory developed by Cover known as the Cover theorem (Cover, 1965). This theorem basically states that if a pattern recognition problem is mapped into a high enough dimensional space, then the classes will be linearly separable and will hence allow a simple linear discriminate technique to separate the classes. Figure 2 shows the transformation from input space to feature space where the nonlinear boundary has been transformed into a linear boundary in the feature space.

Substituting the transformation Eq. (12) into Eq. (3) gives the decision function as,

$$f(\mathbf{x})=\text{sign}\left(\sum_{i=1}^{N}\alpha_i y_i(\Phi(\mathbf{x})\cdot\Phi(\mathbf{x}_i))+b\right) \tag{13}$$

The kernel function $K(\mathbf{x},\ \mathbf{y})=\Phi(\mathbf{x})\cdot\Phi(\mathbf{y})$ is used to perform the transformation into higher-dimensional feature space. The basic form of SVM is obtained after substituting the kernel function in the decision function Eq. (13) as follows:

$$f(\mathbf{x})=\text{sign}\left(\sum_{i=1}^{N}\alpha_i y_i K(\mathbf{x},\ \mathbf{x}_i)+b\right) \tag{14}$$

Any function that satisfies Mercer's theorem (Osuna et al., 1997) can be used as a kernel function to compute a dot product in feature space. There are different kernel functions used in SVMs, such as linear, polynomial, Laplacian RBF, chi-square and Gaussian RBF, which avoid the computational burden of explicitly representing the feature vectors. The selection of an appropriate kernel function is important, since the kernel function defines the feature space in which the training set examples will be classified. As long as the kernel function is legitimate, an SVM will operate correctly even if the designer does not know exactly what features of the training data are being used in the kernel-induced feature space. The definition of legitimate kernel function is given by Mercer's theorem: the function must be continuous and positive definite. Human experts often find it easier to specify a kernel function than to specify explicitly the training set features for being used by the classifier. The kernel expresses prior knowledge about the phenomenon being modeled and encoded as a similarity measure between two vectors. In this work, linear, polynomial and Gaussian RBF kernel functions were evaluated and formulated as shown in Table 1.

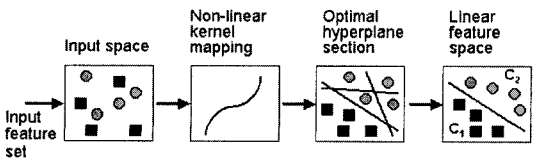


Fig. 2 Transformation to linear feature space from nonlinear input space

2.2 Multi-class classification

The above discussion deals with binary classification where the class labels can take only two values: ± 1. Many real-world problems, however, have more than two classes. For example, in fault diagnosis of rotating machinery there are

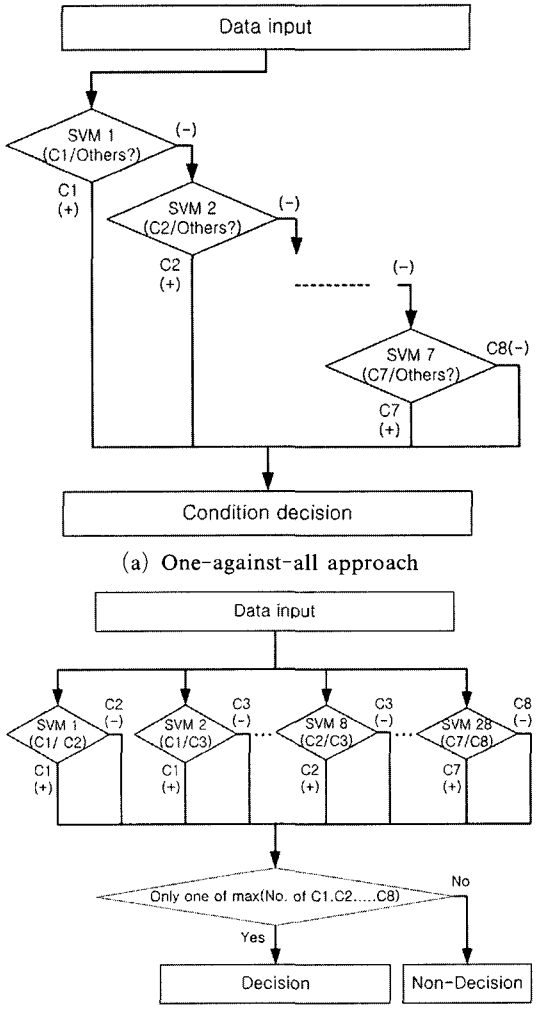
Table 1 Formulation for used in kernel functions

Kernel	$K(x,\ y)$
Linear	$\mathbf{x}\cdot\mathbf{y}$
Polynomial	$(\mathbf{x}\cdot\mathbf{y}+1)^d$
Gaussian RBF	$\exp\{-(\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2)\}$

several fault classes, such as mechanical unbalance, misalignment and bearing faults. Multi-class classification problems can be solved using one of the voting schemes, which are based on combining binary classification decision functions. Various approaches, such as one-against-all (Bottou et al., 1994; Hsu and Lin, 2002), one-against-one (Knerr et al., 1990; Friedman, 2003; Kreβel, 1999), directed acyclic graph (Platt et al., 2000) and binary tree (Schwenker, 2000) have been developed to decompose a multi-class problem into a number of binary classification problems.

The earliest usage of SVM multi-class classification is probably the one-against-all (rest) method (Knerr et al., 1990; Friedman, 2003). To obtain k-class classifiers, it is common to construct a set of binary classifiers $f_1, \cdots, f_k$, with each trained to separate one class from the rest and combine them by performing the multi-class classification according to the maximal output before applying the sign function. The flow chart of the working process is shown in Fig. 3(a). Here the ith SVM is trained with all of the data set in the ith class with positive labels and all other examples with negative labels. In the classification phase, the classifier with the maximal output defines the estimated class label of the current input vector.

Another frequently used method is the one-against-one method. In this method, for k-classes, will results in $k(k-1)/2$ binary classifiers as shown in Fig. 3(b). The number of classifiers is usually larger than the number of one-against-all classifiers. For instance, if $k=10$, one needs to train 45 binary classifiers rather than 10 classifiers as in the method above. Although this requires a larger training time, the individual problems that need to be trained are significantly smaller. Furthermore, if the training algorithm scales superlinearly with the training set size, it is possible to save processing time. This is related to the runtime execution speed. To classify a test pattern in this work, we need to evaluate all 45 binary classifiers and classify them according to the classes which get the highest number of votes. A vote for a given class is defined as a classifier



(a) One-against-all approach

(b) One-against-one approach

Fig. 3 Classification strategy of multi-class SVM

putting the pattern into that class. The individual classifiers, however, are usually smaller in size (they have fewer SVs) than they would be in the one-against-all approach. This is because, (i) the training sets are smaller and (ii) the problems to be learned are usually easier, since the classes have less overlap. If k is large and we need to evaluate the $k(k-1)/2$ classifiers, then the resulting system may be slower than the corresponding one-against-all SVMs.

To solve the SVM problem one has to solve the quadratic programming (QP) problem of Eq. (9) under the constraints of Eqs. (10) and (11). Vapnik (1982) describes a method which used

the projected conjugate gradient algorithm to solve the SVM-QP problem. Sequential minimal optimization (SMO) proposed by Platt (1998) is a simple algorithm that can be used to solve the SVM-QP problem without any additional matrix storage and without using the numerical QP optimization steps. This method decomposes the overall QP problem into QP sub-problems using the Osuna's theorem to ensure convergence. In this paper the SMO is used as a solver and detailed descriptions can be found in Platt (1998), Smola and Scholkopf (1998), Burges (1998) and Keerthi and Shevade (2002).

3. SVM-based Diagnosis System

3.1 System structure

The block diagram of a multi-class SVM based fault diagnosis system is shown in Fig. 4. The system consists of three sections: data acquisition, feature extraction and selection, and training and testing for fault diagnosis. The raw time signal is obtained from the Machinery Fault Simulator shown in Fig. 5. The features of the data are extracted through the discrete wavelet transform and feature extraction algorithms (Yang et al., 2004a). Wavelet transform is more effective than FFT in terms of data compression and is highly tolerant to the presence of additive noise and drift in the sensor responses. Feature selection technique is applied to rank the importance of input features from the extracted features. Finally, the SVMs are trained and used to classify the machinery faults.

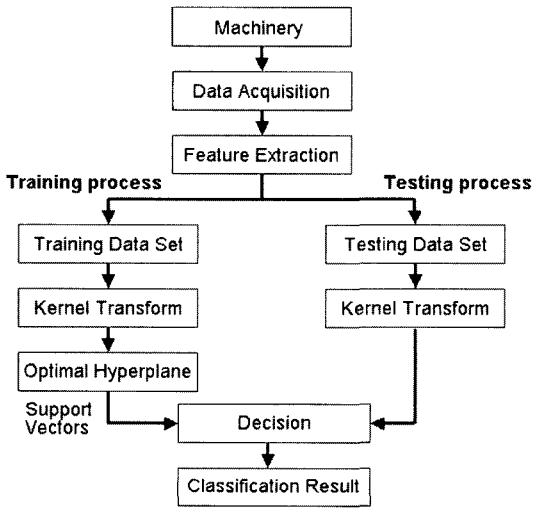


Fig. 4 Block diagram of a multi-class SVMs classifier system

3.2 Data acquisition

Experiments were performed on a small test rig (Machinery Fault Simulator) shown in Fig. 5 which can simulate most of faults that can commonly occur in a rotating machinery, such as misalignment, unbalance, resonance, ball bearing faults and so on. The machine has a range of operating speeds up to 6000 rpm. The fault simulator has a motor, a coupling, bearings, discs and a shaft. In this work the faults to be analyzed are the bearing faults and structural faults such as unbalance and misalignment. The faulty bearings used in the experiments were rolling element bearings with a damage on the inner race, the outer race, a ball and the combination of these faults, respectively. The parallel misalignment and angular misalignment were simulated by adjusting the height and degree of the simulator base plate using thin shims, respectively. Adding an unbalance mass on the disc leads to mechanical unbalance. A total 8 classes were analyzed in this experiment and detailed descriptions of the faults are shown in Table 2.

Acceleration in the radial direction was measured by an accelerometer located on top of the

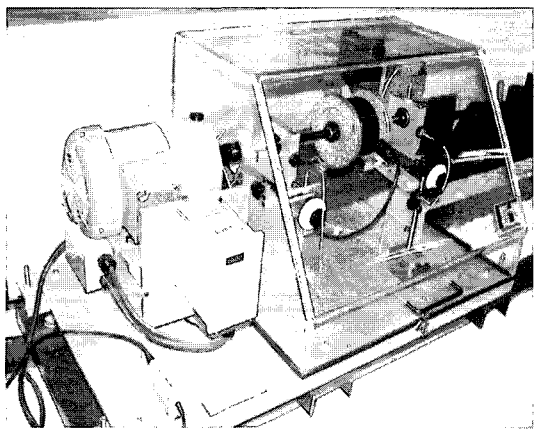
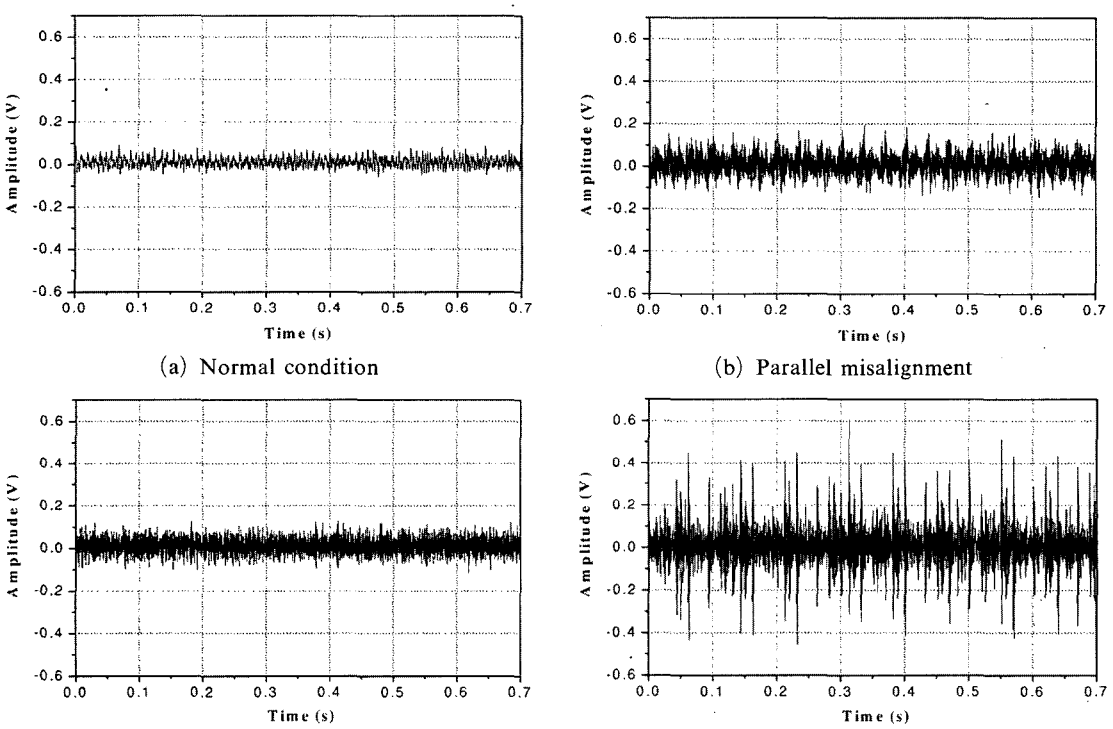


Fig. 5 Machinery fault simulator

Table 2 Description of each fault condition

Fault type	Label	Description
Normal	C1	No fault
Outer race defect	C2	A spalling on the outer raceway surface
Inner race defect	C3	A spalling on the inner raceway surface
Ball defect	C4	A spalling on the ball surface
Complex bearing defect	C5	Multiple defects with an inner, outer race and ball defect
Angular misalignment	C6	Angular eccentricity : 0.7°
Parallel misalignment	C7	Parallel eccentricity : 2 mm
Unbalance	C8	Mechanical unbalance : 578 g·mm



(a) Normal condition (b) Parallel misalignment

(c) Unbalance (d) Bearing inner race fault

Fig. 6 The vibration signals from the machinery fault simulator

right bearing housing. The shaft speed was obtained by a laser speedometer. Twenty continuous measurements were recorded for each condition. The maximum acquisition frequency rate was 5 kHz and the sampling number was 16384. A mobile DSP analyzer was used to perform data acquisition and the data was stored in a notebook computer. Samples of the raw vibration signals are shown in Fig. 6. The waveform of normal condition is quite clear about the period of running speed. In the faulty bearing waveform, there are many impulses related to the inner race defect.

3.3 Feature extraction

Features describing various attributes of the fault condition were extracted and a classifier used these attributes to assign a label to each fault. Therefore, the classification performance depends heavily on the quality of the feature extracted (Ob et al., 2004).

In order to improve signal to noise ratio, 1-D discrete wavelet transform was used to decompose the time signal. The discrete wavelet transform (DWT) permits a systematic decomposition of a signal into its sub-band levels. The analysis of the

data was performed using the MATLAB 5.1 Wavelet Toolbox (Misiti et al., 1996). Twenty time-waveform signals for each class were processed using the Daubeches-10 (db-10) wavelet (Daubeches, 1992) to estimate the condition. The sub-band (level) or the multi-resolution analysis (MRA) was performed by dividing them into ten sub-bands in the frequency range from 0-5 kHz. Levels 1 to 3 (0.625-5 kHz) in MRA are the most dominant band and other sub-bands cannot differentiate the difference between normal and faulty conditions. Hence, the feature extraction from levels 1 to 3 (D1-D3) could be very effectively realized. In Fig. 7, levels 1, 2 and 3 of wavelet coefficients for different conditions (C1-C8) under consideration correspond to 2.5-5 kHz, 1.25-2.5 kHz and 0.625-1.25 kHz frequency bands, respectively.

The wavelet transformed signal and the original signal were then estimated by eight feature parameters such as mean, standard deviation, RMS, shape factor, skewness, kurtosis, crest factor and entropy estimation. Figure 8 shows typical results of feature extraction of the time signals. Finally, a total 32 feature parameters (four kinds

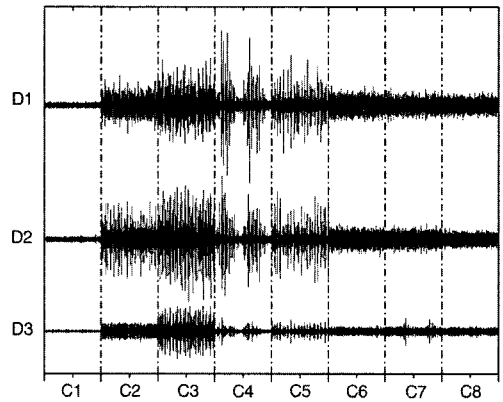
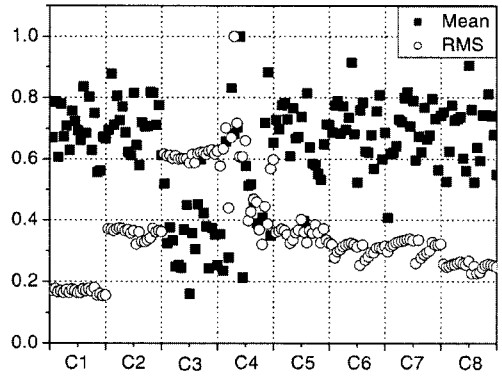
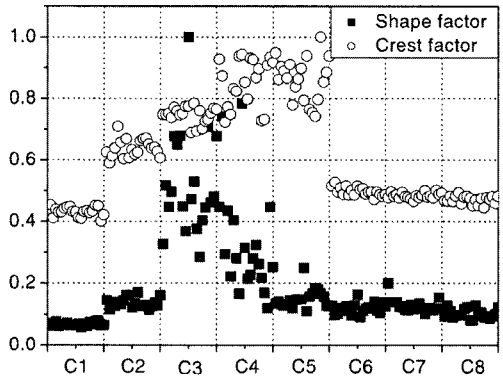


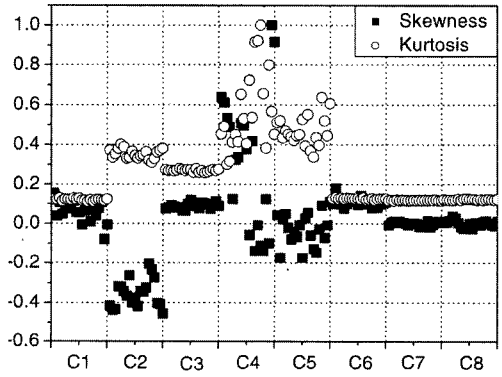
Fig. 7 Wavelet transform of vibration signal under different conditions



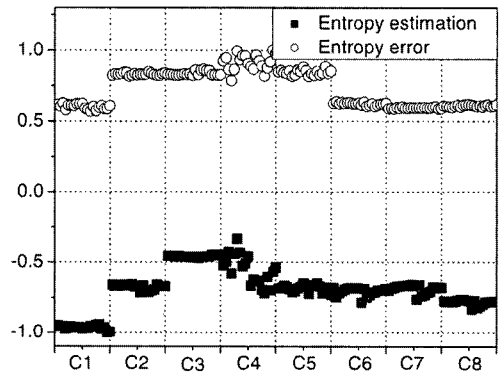
(a) Mean and RMS



(b) Shape factor and crest factor



(c) Skewness and kurtosis



(d) Entropy estimation and entropy error

Fig. 8 Feature extraction of the time waveform signal

Table 3 Attribute label of each input feature

Feature	Attribute label			
	Time waveform	Wavelet level 1	Wavelet level 2	Wavelet level 3
Mean	1	9	17	25
RMS	2	10	18	26
Shape factor	3	11	19	27
Skewness	4	12	20	28
Kurtosis	5	13	21	29
Crest factor	6	14	22	30
Entropy estimation	7	15	23	31
Entropy error	8	16	24	32

of signals, eight parameters) were obtained as shown in Table 3.

3.4 Feature selection

Too many features can cause cures of dimensionality and peaking phenomenon (Bishop, 1995; Raudys et al., 1991) that greatly degrade classification accuracy since some features are essential, some are less important, some of them may not be mutually independent and some may be useless. Also too many features can be a burden, as it requires a large amount of time to calculate. Thus feature selection is necessary to remove garbage features and pick up the significant ones for fault diagnosis. Usually 5 to 12 parameters are sufficient to perform the calculation and provide sufficient accuracy (Yang et al., 2000a; 2000b). In order to remove the redundant and irrelevant features from the feature set, a careful analysis of the feature set must be carried out. The objective is to identify the features that show high variability between different classes and thus help in distinguishing between them. In order to solve this problem, an evaluation technique (Yang et al., 2004) is used to select feature parameters that can represent the fault features from using all parameters and is described as follows:

Step 1. Calculate the relative average value of the sampling data for the same class d_{ij} and then obtain the average distance of 8 classes d_{ai}. The equation can be defined as follows:

$$d_{ij}=\frac{1}{N\times(N-1)}\sum_{m,n=1}^{N}|p_{i,j}(m)-p_{ij}(n)| \quad (m,\ n=1,\ 2,\ \cdots,\ N,\ m\neq n) \tag{16}$$

where N is the sampling number of each class ($N=20$), p_{ij} is the value of ith feature under jth class.

$$d_{ai}=\frac{1}{M}\sum_{j=1}^{M}d_{ij} \tag{17}$$

where M is the number of class ($M=8$).

Step 2. Calculate the average distance of inter-class d'_{ai}

$$d'_{ai}=\frac{1}{M\times(M-1)}\sum_{m,n=1}^{M}|p_{ai,m}-p_{ai,n}| \quad (m,\ n=1,\ 2,\ \cdots,\ M\ ;\ m\neq n) \tag{18}$$

where $p_{ai,m}$ and $p_{ai,n}$ are the average values of the sampling data under different class.

$$p_{ai,j}=\frac{1}{N}\sum_{n=1}^{N}p_{ij}(n) \quad (n=1,\ 2,\ \cdots,\ N) \tag{19}$$

Step 3. Calculate the ratio d_{ai}/d'_{ai}

Step 4. Select the eight largest feature parameters α_i, $i=1$ to 8. Bigger α_i represents a well selected feature. This requires a small d_{ai} and a large d'_{ai}.

$$\alpha_i=d'_{ai}/d_{ai} \tag{20}$$

where α_i ($i=1,\ \cdots,\ k$) is the effectiveness factor of the features and k is the number of selected features.

Given α_i, one can now establish a raking methodology among the individual feature components. The useful features are expected to show high values of α_i, indicating a good inter-class spread in the classifier.

3.5 Fault diagnosis

The four classifiers used in this work were SVMs, SOFM, LVQ and RBF networks (Yang et al., 2004b). Same examples were used to compare the effectiveness among these networks. The features selected from feature selection algorithm were used as input vectors. The breakdown of the classification process consisted of 80 samples for the training set and 80 samples for the testing set (ten samples for each class). In the training process, the networks were trained until the mean square error is below 0.01 or the maximum epochs (=10000) were reached.

4. Simulation Results

4.1 Effect of kernel functions

The performance of a SVM depends to a great extent on the choice of the kernel function to transform a data from input space to a higher dimensional feature space (Smola et al., 1998). The choice of kernel function is data dependent and there are no definite rules governing its choice that might yield a satisfactory performance. Table 4 presents results of SVM with the three kernel functions defined in Table 1 and used the same eight selected feature examples. In Table 4, d is the degree of the polynomial. The width of the RBF kernel parameter is given by σ and can be determined in general by an iterative process selecting an optimum value based on the full feature set (Scholkopf, 1997). These kernels are also well accepted for constructing SVM and provide excellent results for real-world applications (Strauss and Steidel, 2002). We have investigated the construction of multi-class classifiers using the one-against-one method and the one-against-all method. The most important criterion for evaluating the performance of these methods is their classification success rate. The results in Table 4 show that the performance of one-against-one classifiers is better than that of one-against-all classifiers from the view of classification accuracy and training time. The overall success ratio of class classification ranged from 98.125 to 100% for training and 88.75 to 98.75% for testing. Among these classifiers, Gaussian RBF is the best with high training and testing accuracy. The detailed process of one against all method is illustrated in Table 5. Some SVs are used many times for different classes. Thus the total SVs are not equal to the summation of each class of SVs.

4.2 Effect of feature selection

Figure 9 shows the computation results of effectiveness factor α_i of 32 feature parameters. From the magnitude of the effectiveness factor, some of the feature parameters were selected. They were entropy error of the time waveform signal and

Table 4 Fault classification results due to kernel and multi-class classification strategy

Kernel	Multi-class approach	Classification rate (%)		Number of SVs	Training time (s)
		Training	Testing		
Linear	One vs. one	100	93.75	44	1.25
	One vs. all	98.125	90.00	55	1210
Polynomial (d=1)	One vs. one	100	93.75	41	0.93
	One vs. all	98.125	90.00	55	20.56
Polynomial (d=2)	One vs. one	100	92.5	37	0.94
	One vs. all	100	90.00	38	28.31
Polynomial (d=3)	One vs. one	100	93.75	37	0.94
	One vs. all	100	88.75	32	22.45
Polynomial (d=4)	One vs. one	100	93.75	36	0.98
	One vs. all	100	91.25	36	62.66
Gaussian RBF (σ=0.168)	One vs. one	100	98.75	43	3.37
	One vs. all	100	92.50	44	9.90

Table 5 Performance comparisons for one-against-all method

Kernel		Linear	Polynomial				Gaussian RBF σ=0.168
			d=1	d=2	d=3	d=4	
Number of SVs	SVM1	4	4	4	4	5	21
	SVM2	10	10	7	7	7	20
	SVM3	4	4	3	3	3	21
	SVM4	6	6	6	4	5	21
	SVM5	17	17	9	7	9	10
	SVM6	18	18	10	8	7	6
	SVM7	7	7	5	5	6	22
	Total	55	55	38	32	36	44
Number of error	C1	0	0	0	0	0	0
	C2	2	2	2	2	2	1
	C3	1	1	1	2	1	0
	C4	2	2	1	3	1	4
	C5	0	0	0	0	0	1
	C6	3	3	4	2	3	0
	C7	0	0	0	0	0	0
	C8	0	0	0	0	0	0
Success rate (%)		90.0	90.0	90.0	88.75	91.25	92.50

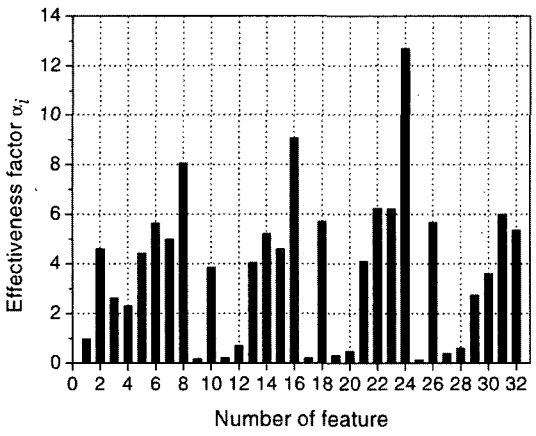


Fig. 9 Effectiveness factor of features

wavelet transform level 1; RMS, crest factor, entropy estimation and entropy error of wavelet transform level 2; RMS and entropy estimation of wavelet transform level 3. The selected features were used as the input vectors of the classifiers for fault diagnosis.

Table 6 shows the classification results for SVM using the RBF kernels and the one-against-one method with the selected features. In each case, the test success, number of SVs and training time for the selected features were compared with the results used in all features without feature selection. In Table 6, the change of the test success and training time are listed against the number of retained features. When the features are discarded, the training performance monotonically decreases, while the test performance increases slightly at the beginning. This can be explained by the reduced over-fitting effects due to smaller number of features. A drastic reduction of features, however, can lead to a decrease in the test performance (Hermes and Buhmann, 2000). Figure 10 shows the influences of the number of selected features on the test success and training time. It can be seen that when the number of selected features takes a small value (e.g., 2), the test success rate is very low (62.5%). The success rate increases with increment of the number of selected features and remains at a maximum value of 100% in a certain range (i.e., 12-24). It tends to decease as the number of features continues to increase. On the other hand, the training time increased almost linearly with increase in the number of selected features. The results are very encouraging as the technique shows a significant reduction in size of the feature vector in the feature extraction process. It is particularly useful

Table 6 Performance comparisons of SVMs with feature selection by using RBF kernels and one-against-one method

No. of features	Input features	Kernel width σ	Test success (%)	No. of SVs	Training time (s)
2	16,24	0.142	62.50	54	1.87
4	8,16,22,24	0.50	96.25	49	2.09
6	8,16,22,23,24,31	0.152	96.25	41	2.21
8	8,16,18,22,23,24,26,31	0.168	98.75	43	3.37
10	6,8,16,18,22,23,24,26,31,32	0.162	98.75	48	2.85
12	6,7,8,14,16,18,22,23,24,26,31,32	0.145	100	49	3.91
16	2,5,6,7,8,14,15,16,18,21,22,23,24, 26,31,32	0.10	100	53	4.62
24	1,2,3,4,5,6,7,8,10,12,13,14,15,16, 18,21,22,23,24,26,29,30,31,32	0.28	100	56	7.07
8	Time waveform (1-8)	0.25	98.75	51	3.23
32	All (1-32)	0.60	97.50	67	7.49

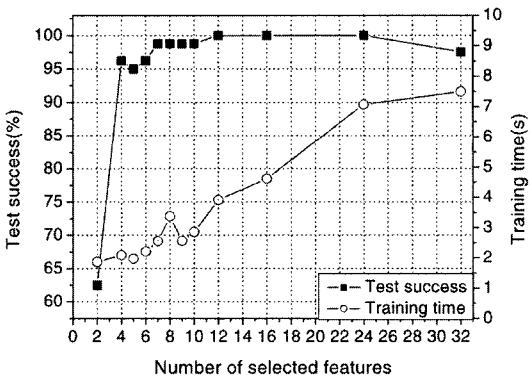


Fig. 10 Performance of SVMs for different number of selected features

to reduce the training time in order to improve the classification performance of the SVM classifier.

4.3 Performance comparison of SVMs and ANNs

In order to verify the effectiveness and robustness of the proposed classification approach, the authors compared the classification results between the SVMs and other traditional neural networks, such as the SOFM, LVQ and RBF networks. The above results were obtained from multi-class SVMs using the one-against-one classifier and the one-against-all classifier with the linear, polynomial and Gaussian kernels. The classification results of the SVMs, SOFM, LVQ and RBF networks are shown in Table 7. The

Table 7 Classification results of SVMs, SOFM, LVQ and RBF networks

Classifier	SOFM	LVQ	RBF	SVMs
Success rate (%)	93	93	89	100

maximum classification success rate for the SVMs was 100% and for the SOFM, LVQ and RBF networks were 93%, 93% and 89%, respectively. It can be concluded from Table 7 that the SVMs perform significantly better than the SOFM, LVQ and RBF networks.

5. Conclusions

This paper shows that the proposed SVMs based fault diagnosis approach for rotating machinery is superior to many traditional intelligent networks. The experiments have demonstrated that this approach can successfully diagnose any condition and the average fault diagnosis accuracy is above 90%. Although the mechanical behavior of the each fault results were complex and non-stationary, the wavelet transform has been demonstrated to be a useful signal processor to extract different time-frequency features of symptoms of various fault conditions. It is shown that using only the important features for classification can obtain high success rate and reduces the training time of the SVMs classifier. When the same class on fault diagnosis of rota-

ting machinery, the success rate of SVMs can reach 100%, while the SOFM, LVQ and RBF networks were 93%, 93% and 89%, respectively. The one-against-one SVM classifier using a Gaussian RBF kernel shows superior performance in comparison with the previously published classifiers and the one-against-all SVM classifier. The high performance of the SVMs is attributed primarily to its inherent generalization capability. This allows the SVMs to be optimized based on the amount of training data. SVMs hold significant promise in the diagnosis of rotating machinery due to their ability to give optimal performance with a limited training data for application in real industry.

Acknowledgments

This work was supported by the Center for Advanced Environmentally Friendly Energy Systems, Pukyong National University, Korea (Project number : R12-2003-001-00018-0).

References

Bishop C. M., 1995, *Neural Networks for Pattern Recognition*, Oxford Clarendon Press.

Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Jackel, L., LeCun, Y., Muller, U., Sackinger, E., Simard, P. and Vapnik, V., 1994, "Comparison of Classifier Methods : A Case Study in Handwriting Digit Recognition," *Proc. International Conference on Pattern Recognition*, pp. 77~87.

Burges, C. J. C., 1998, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, Vol. 2, No. 2, pp. 955~974.

Capenter, G. A. and Grossberg, S., 1988, "The ART of Adaptive Pattern Recognition by a Self-organizing Neural Network," *IEEE Trans. on Computers*, Vol. 21, No. 3, pp. 77~88.

Cover, T. M., 1965, "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition," *IEEE Trans. on Electronic Computers*, Vol. 14, pp. 326~334.

Daubechies, I., 1992, *Ten Lectures on Wavelets*, SIAM, Pennsylvania.

Friedman, J., 2003, *Another Approach to Polychotomous Classification*, Department of Statistics, Stanford Univ., CA. http://www-stst.stanford.edu/report/friedman/poly.ps.Z.

Hermes, L. and Buhmann, J. M., 2000, "Feature Selection for Support Vector Machines," *Proc. 15th International Conference on Pattern Recognition*, pp. 712~715.

Hsu, C. W. and Lin, C. J., 2002, "A Comparison of Methods for Multiclass Support Vector Machines," *IEEE Trans. on Neural Networks*, Vol. 13, No. 2, pp. 415~425.

Jack, L. B. and Nandi, A. K., 2002, "Fault Detection Using Support Vector Machines and Artificial Neural Networks, Augmented by Genetic Algorithms," *Mechanical Systems and Signal Processing*, Vol. 16, No. 2-3, pp. 373~390.

Kangas, J. and Kohonen, T., 1996, "Developments and Applications of the Self-organizing Map and Related Algorithms," *Mathematics and Computers in Simulation*, Vol. 41, pp. 3~12.

Keerthi, S. S. and Shevade, S. K., 2002, *SOM Algorithm for Least Squares SVM Formulations*, Technical Report CD-02-8. http://guppy.mpe.nus.edu.sg/~mpessk.

Knerr, S., Personnaz, L. and Dreyfus, G., 1990, "Single-layer Learning Revisited : A Stepwise Procedure for Building and Training a Neural Network," in *Neurocomputing : Algorithms, Achitectures and Applications*, J. Fogelman, Ed. Springer-Verlag, New York.

Kohonen, T., 1995, *Self-Organizing Maps*, Springer-Verlag, New York.

Kreβel, U., 1999, "Pairwise Classification and Support Vector Machines," in *Advances in Kernel Methods-Support Vector Learning*, B. Scholkopf, C. J. C. Burges, A. J. Smola, Eds. MIT Press, Cambridge, pp. 255~268.

Misiti, M., Misiti, Y., Oppenheim, G. and Poggi, J. M., 1996, *Wavelet Toolbox for Use with MATLAB*, The Math Works Inc.

Muller, K. R., Mika, S., Ratsch, G., Tsuda, K. and Scholkopf, B., 2001, "An Introduction to Kernel-based Learning Algorithm," *IEEE Trans. on Neural Network*, Vol. 12, No. 2, pp. 181~201.

Ob, J. H., Kint, C. G. and Cho, Y. M., 2004, "Diagnostics and Prognostics Based on Adaptive Time-Frequency Feature Discrimination," *KSME International Journal*, Vol. 18, No. 9, pp. 1537~1548.

Osuna, E., Freund, R. and Girosi, F., 1997, "Training Support Vector Machines : An Applications to Face Detection," *Proc. CVPR*, pp. 1~6

Platt, J. C., 1998, *Sequential Minimal Optimization : A Fast Algorithm for Training Support Vector Machines*, Technical Report 98-14, Microsoft Research, Redmond, Washington. http://www. research. microsoft.com/~jplatt/smo.ttml.

Platt, J. C., Cristianini, N. and Shawe-Talyor, J., 2000, "Large Margin DAG's for Multiclass Classification," *Advances in Neural Information Processing Systems*, Vol. 12, pp. 547~553.

Raudys S. J. and Jain A. K., 1991, "Small Sample Size Effects in Statistical Pattern Recognition : Recommendations for Practitioners," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 3, pp. 252~264.

Samanta, B., 2004, "Gear Fault Detection using Artificial Neural Networks and Support Vector Machines with Genetic Algorithms," *Mechanical Systems and Signal Processing*, Vol. 18, No. 3, pp. 625~644.

Scholkopf, B., 1997, *Support Vector Learning*, Oldenbourg-Verlag, Germany.

Schwenker, F., 2000, "Hierarchical Support Vector Machines for Multi-class Pattern Recognition," *Proc. 4th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, pp. 561~565.

Smola, A. J., Scholkopf, B. and Muller, K. R., 1998, "The Connection between Regularization Operators and Support Vector Kernels," *Neural Networks*, Vol. 11, pp. 637~649.

Smola, A. J. and Scholkopf, B., 1998, *A Tutorial on Support Vector Regression*, Technical Report NC2-TR-1998-030. http://www.neurocolt.com.

Strauss, D. J. and Steidel, G., 2002, "Hybrid Wavelet-Support Vector Classification of Waveforms," *Journal of Computational and Applied Mathematics*, Vol. 148, pp. 375~400.

Sundararajan, N., Saratchandran, P. and Wei, L. Y., 1999, *Radial Basis Function Neural Networks with Sequential Learning*, World Scientific, Singapore.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B. and Vandewalle, J., 2002, *Least Squares Support Vector Machines*, World Scientific, Singapore.

Vapnik, V. N., 1982, *Estimation of Dependences Based on Empirical Data*, Springer-Verlag.

Vapnik, V. N., 1992, *Principles of Risk Minimization for Learning Theory*, pp. 831~838 in J. E. Moody et al.(Eds.) *Advances in Neural Information Processing Systems 4*, Morgan Kaufmann Publishers, San Mateo, CA.

Vapnik, V. N., 1999, *The Nature of Statistical Learning Theory*, Springer, New York.

Yang, B. S., Lim, D. S., Seo, S. Y. and Kim, M. H., 2000a, "Defect Diagnostics of Rotating Machinery using SOFM and LVQ," *Proc. 7th International Congress on Sound and Vibration*, pp. 567~574.

Yang, B. S., Lim, D. S. and An, J. L., 2000b, "Vibration Diagnostic System of Rotating Machinery using Artificial Neural Network and Wavelet Transform," *Proc. 13th International Congress on COMADEM*, pp. 923~932.

Yang, B. S., Kim, K. and Rao, Raj B. K. N., 2002, "Condition Classification of Reciprocating Compressors using RBF Neural Network," *International Journal of COMADEM*, Vol. 5, No. 4, pp. 12~20.

Yang, B. S., Han, T. and An, J. L., 2004a, "ART-Kohonen Neural Network for Fault Diagnosis of Rotating Machinery," *Mechanical Systems and Signal Processing*, Vol. 18, No. 3, pp. 645~657.

Yang, B. S., Han, T., An, J. L., Kim, H. C. and Ahn, B. H., 2004b, "A Condition Classification System for Reciprocating Compressors," *International Journal of Structural Health Monitoring*, Vol. 3, No. 3, pp. 277~284.

Yang, B. S., Hwang, W. W., Kim, D. J. and Tan, A., 2005, "Condition Classification of Small Reciprocating Compressor for Refrigerators using Artificial Neural Networks and Support Vector Machines," *Mechanical Systems and Signal Processing*, Vol. 19, No. 2, pp. 371~390.