

Bearing Fault Detection Using Artificial Neural Networks and Genetic Algorithm

B. Samanta

*Department of Mechanical and Industrial Engineering, College of Engineering, Sultan Qaboos University, P.O. Box 33, Muscat 123, Sultanate of Oman
Email: samantab@squ.edu.om*

Khamis R. Al-Balushi

*Department of Mechanical and Industrial Engineering, College of Engineering, Sultan Qaboos University, P.O. Box 33, Muscat 123, Sultanate of Oman
Email: kbalushi@squ.edu.om*

Saeed A. Al-Araimi

*Department of Mechanical and Industrial Engineering, College of Engineering, Sultan Qaboos University, P.O. Box 33, Muscat 123, Sultanate of Oman
Email: alaraimi@squ.edu.om*

Received 26 August 2002; Revised 22 July 2003; Recommended for Publication by Shigeru Katagiri

A study is presented to compare the performance of bearing fault detection using three types of artificial neural networks (ANNs), namely, multilayer perceptron (MLP), radial basis function (RBF) network, and probabilistic neural network (PNN). The time domain vibration signals of a rotating machine with normal and defective bearings are processed for feature extraction. The extracted features from original and preprocessed signals are used as inputs to all three ANN classifiers: MLP, RBF, and PNN for two-class (normal or fault) recognition. The characteristic parameters like number of nodes in the hidden layer of MLP and the width of RBF, in case of RBF and PNN along with the selection of input features, are optimized using genetic algorithms (GA). For each trial, the ANNs are trained with a subset of the experimental data for known machine conditions. The ANNs are tested using the remaining set of data. The procedure is illustrated using the experimental vibration data of a rotating machine with and without bearing faults. The results show the relative effectiveness of three classifiers in detection of the bearing condition.

Keywords and phrases: condition monitoring, genetic algorithm, probabilistic neural network, radial basis function, rotating machines, signal processing.

1. INTRODUCTION

Machine condition monitoring is gaining importance in industry because of the need to increase reliability and to decrease the possibility of production loss due to machine breakdown. The use of vibration and acoustic emission (AE) signals is quite common in the field of condition monitoring of rotating machinery. By comparing the signals of a machine running in normal and faulty conditions, detection of faults like mass unbalance, rotor rub, shaft misalignment, gear failures, and bearing defects is possible. These signals can also be used to detect the incipient failures of the machine components, through the online monitoring system, reducing the possibility of catastrophic damage and the downtime. Some of the recent works in the area are listed in [1, 2, 3, 4, 5, 6, 7, 8]. Although often the visual inspection of the frequency domain features of the measured signals is ad-

equated to identify the faults, there is a need for a reliable, fast, and automated procedure of diagnostics.

Artificial neural networks (ANNs) have potential applications in automated detection and diagnosis of machine conditions [3, 4, 7, 8, 9, 10]. Multilayer perceptrons (MLPs) and radial basis functions (RBFs) are the most commonly used ANNs [11, 12, 13, 14, 15], though interest in probabilistic neural networks (PNNs) is also increasing recently [16, 17]. The main difference among these methods lies in the ways of partitioning the data into different classes. The applications of ANNs are mainly in the areas of machine learning, computer vision, and pattern recognition because of their high accuracy and good generalization capability [11, 12, 13, 14, 15, 16, 17, 18]. Though in the area of machine condition monitoring MLPs are being used for quite some time, the applications of RBFs and PNNs are relatively recent

[3, 19, 20, 21]. In [19], a procedure was presented for condition monitoring of rolling element bearings comparing the performance of the classifiers MLPs and RBFs with all calculated signal features and fixed parameters for the classifiers. In this, vibration signals were acquired under different operating speeds and bearing conditions. The statistical features of the signals, both original and with some preprocessing like differentiation and integration, high- and lowpass filtering, and spectral data of the signals, were used for classification of bearing conditions.

However, there is a need to make the classification process faster and accurate using the minimum number of features which primarily characterize the system conditions with optimized structure or parameters of ANNs [3, 22]. Genetic algorithms (GAs) were used for automatic feature selection in machine condition monitoring [3, 21, 22, 23]. In [22], a GA-based approach was introduced for selection of input features and number of neurons in the hidden layer. The features were extracted from the entire signal under each condition and operating speed [19]. In [23], some preliminary results of MLPs and GAs were presented for fault detection of gears using only the time domain features of vibration signals. In this approach, the features were extracted from finite segments of two signals: one with normal condition and the other with defective gears.

In the present work, the procedure of [23] is extended to the diagnosis of bearing condition using vibration signals through three types of ANN classifiers. Comparisons are made between the performance of the three different types of ANNs, both with and without automatic selection of input features and classifier parameters. The classifier parameters are the number of hidden layer neurons in MLPs and the width of the radial basis function in RBFs and PNNs. Figure 1 shows a flow diagram of the proposed procedure. The selection of input features and the classifier parameters are optimized using a GA-based approach. These features, namely, mean, root mean square, variance, skewness, kurtosis, and normalized higher-order (up to ninth) central moments are used to distinguish between normal and defective bearings. Moments of order higher than nine are not considered in the present work to keep the input vector within a reasonable size without sacrificing the accuracy of the diagnosis. The roles of different vibration signals are investigated. The results show the effectiveness of the extracted features from the acquired and preprocessed signals in diagnosis of the machine condition. The procedure is illustrated using the vibration data of an experimental setup with normal and defective bearings.

2. VIBRATION DATA

Figure 2 shows the schematic diagram of the experimental test rig. The rotor is supported on two ball bearings MB 204 with eight rolling elements. The rotor was driven with a three-phase AC induction motor through a flexible coupling. The motor could be run in the speed range of 0–10,000 rpm using a variable frequency drive (VFD) controller. For the present experiment, the motor speed was

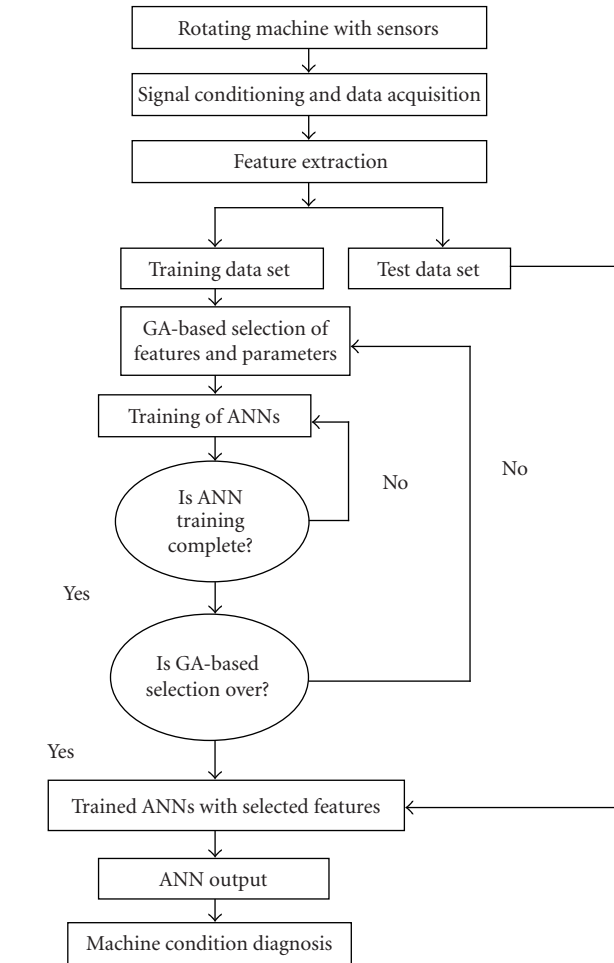


FIGURE 1: Flow chart of diagnostic procedure.

maintained at 600 rpm. Two accelerometers were mounted at 90° on the right-hand side (RHS) bearing support to measure vibrations in vertical and horizontal directions (x and y). Separate measurements were obtained for two conditions, one with normal bearings and the other with an induced fault on the outer race of the RHS bearing. The outer race fault was created as a small line using electro-discharge machining (EDM) to simulate the initiation of a bearing defect. It should be mentioned that only one type of bearing fault has been considered in the present study to see the effectiveness of the proposed approach for two-class recognition. Diagnosis of different types and levels of bearing faults is important for optimal maintenance purposes and outside the scope of the present work. Each accelerometer signal was connected through a charge amplifier and an anti-aliasing filter to a channel of a PC-based data acquisition system. One pulse per revolution of the shaft was sensed by a proximity sensor and the signal was used as a trigger to start the sampling process. The vibration signals were sampled simultaneously at a rate of 49152 samples/s per channel. The lower and higher cutoff frequencies of each charge amplifier were set at 2 Hz and 100 kHz, respectively. The cutoff frequency

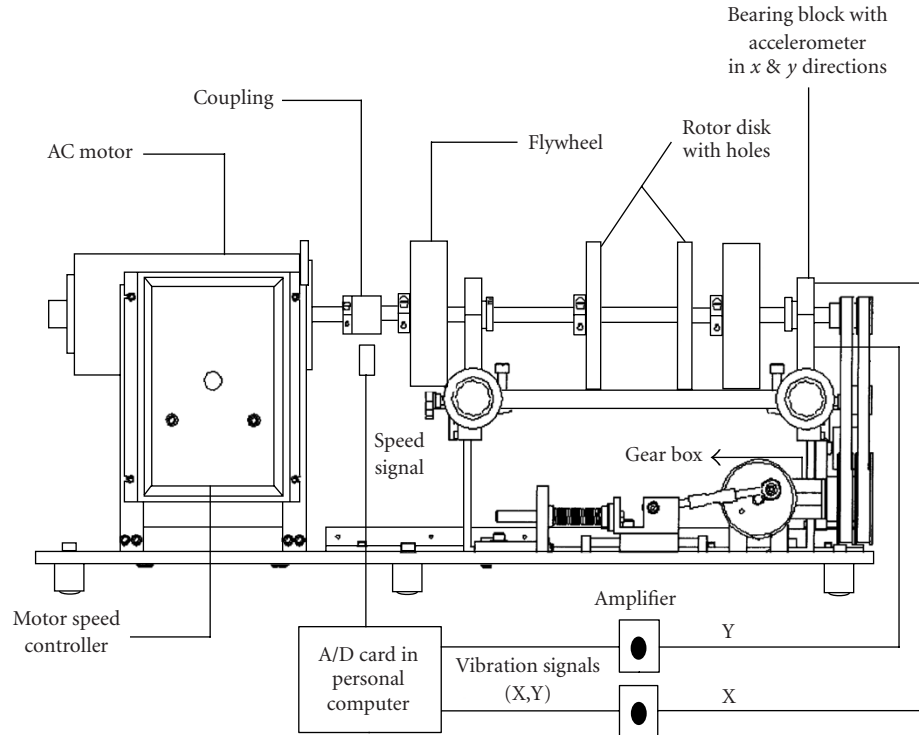


FIGURE 2: Experimental test rig.

of each anti-aliasing filter was set at 24 kHz, almost the half of the sampling rate. The number of samples collected for each channel was 24576 with each bearing condition: normal and faulty. The experiment was repeated under the same operating conditions and a further set of 24576 data points was acquired for each accelerometer signal and bearing condition. These time-domain data were preprocessed to extract the features, similar to [10], for using them as inputs to the ANNs. Half of the first data set was used for training and the other half for testing the ANNs, while the entire data of the second set were used for testing.

3. FEATURE EXTRACTION

3.1. Signal statistical characteristics

Two sets of experimental data, each with normal and defective bearings, were acquired. For each set, two vibration signals consisting of 24576 samples (q_i) were obtained using accelerometers in vertical and horizontal directions to monitor the machine condition. The magnitude of the vibration was constructed from the two component signals $z = \sqrt{x^2 + y^2}$. These signals were divided into 24 segments (bins) of 1024 (n) samples each. An alternative approach would have been to take 24 individual measurements from 24 different runs. However, the present approach was used, similar to [10], to see the effectiveness of the proposed procedure in situations where multiple runs of data may not be feasible, especially in actual industrial setting. Each of these data segments was further processed to extract the following features (1–9): mean (μ), root mean square (RMS), variance (σ^2), skew-

ness (normalized third central moment γ_3), kurtosis (normalized fourth central moment γ_4), and normalized fifth to ninth central moments (γ_5 – γ_9) as follows:

$$\gamma_n = \frac{E\{[q_i - \mu]^n\}}{\sigma^n}, \quad n = 3, 9, \quad (1)$$

where $E\{\cdot\}$ represents the expected value of the function. Figure 3 shows plots of some of these features extracted from the vibration signals (q_i) x , y , and z of the first set of data, each row representing the features for one signal. Only a few of the features are shown as representatives of the full feature set.

It is important to note that in the present work, only two (normal and faulty) conditions of bearings have been considered and the sample size for feature extraction was chosen as 1024 to keep the length of acquired data within a reasonable limit. The features were also calculated, doubling the number of samples with no significant difference. However, for consideration of multiple fault conditions, the data of longer duration (in terms of number of cycles or shaft revolutions) and larger sample size for feature extraction, especially for higher-order (fifth–ninth) moments, may be necessary.

3.2. Time derivative and integral of signals

The high- and low-frequency content of the raw signals can be obtained from the corresponding time derivatives and the integrals. In this work, the first time derivative (dq) and the integral (iq) have been defined, using sampling time as a factor, as follows:

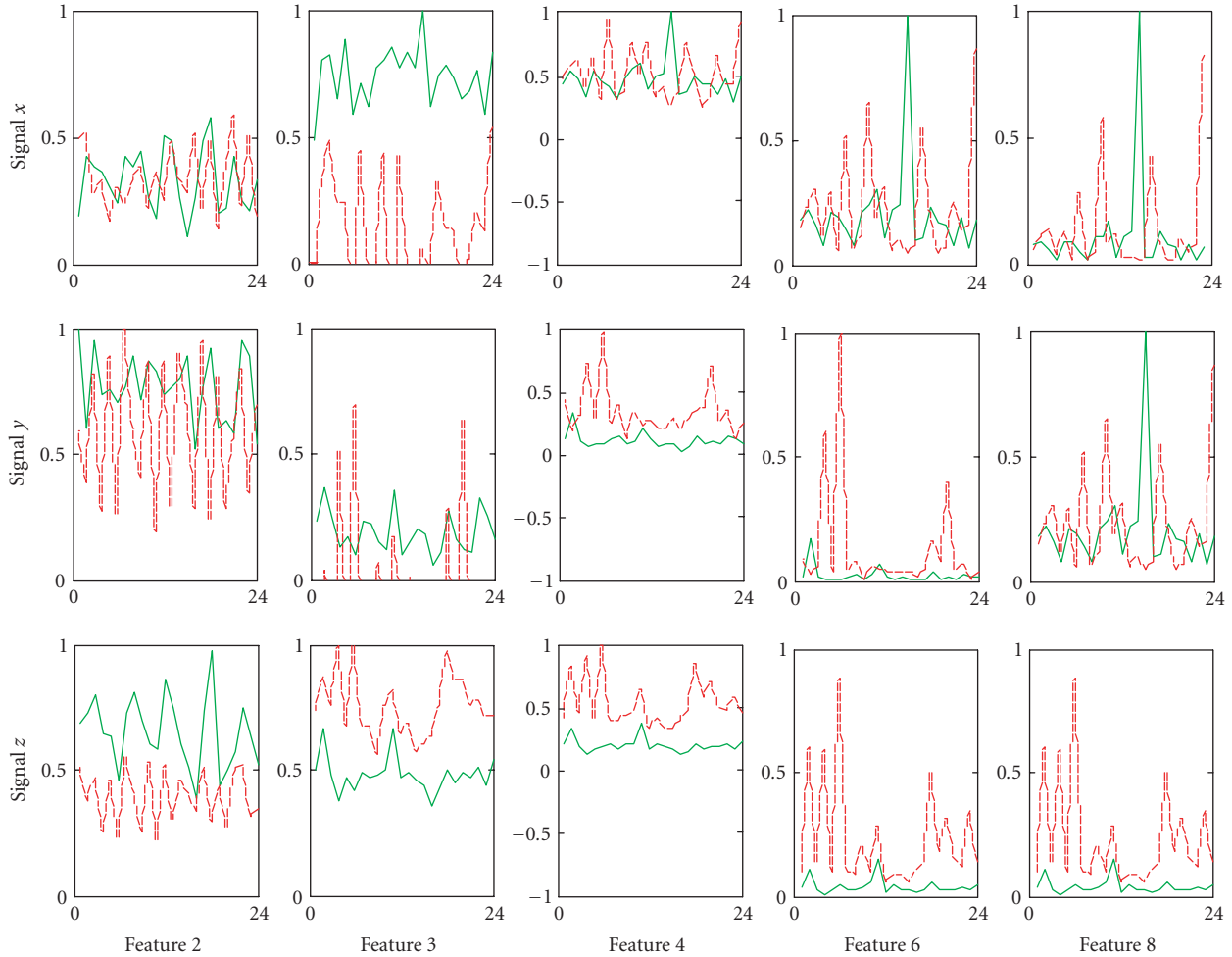


FIGURE 3: Time-domain features of acquired signals: (—) normal, (---) defective.

$$\begin{aligned} dq(k) &= q(k) - q(k - 1), \\ iq(k) &= q(k) + q(k - 1). \end{aligned} \tag{2}$$

The derivative and the integral of each signal were processed to extract an additional set of 18 features (10–27).

3.3. High- and lowpass filtering

The raw signals were also processed through low- and high-pass filters with a cutoff frequency as one-tenth ($f/10$) of the sampling rate ($f = 49152$ Hz). The cutoff frequency was chosen to minimize the effect of sampling on the low- and high-frequency characteristics of the signals. These filtered signals were processed to obtain a set of another 18 features (28–45) leading to a total of 45 features.

3.4. Normalization

The total set of features consists of $45 \times 144 \times 2$ array, where each row represents a feature and the columns denote the number of signals (three), segments per signal (24), bearing conditions (two), and sets of run (two). Each of the features was normalized, dividing each row by its absolute maximum

value and keeping it within ± 1 for better speed and success of the network training. A second scheme of normalization with zero mean and a standard deviation of 1 for each feature set was attempted. Another normalization scheme was also examined by making the features zero mean and then normalizing by the absolute maximum value. The results comparing the effectiveness of these normalization schemes are discussed in Section 6.5. However, it is to be mentioned that the use of absolute maximum in magnitude normalization scheme exploits the large peaks present in the fault signal lowering the normal rotational components. This changes the relative statistics of the signals with and without faults, leading to better classification success.

4. ARTIFICIAL NEURAL NETWORKS

In this section, three types of ANNs are briefly discussed with reference to the structures and the parameters. The main differences among these are also briefly discussed. Readers are referred to [13, 17, 24] for further details. Data from two different sets of run were used in the present work. For the first

set of run, half of the data were used for training the ANNs and the rest were used for testing. Entire data from the second set of run were used for testing.

4.1. Multilayer perceptron

The feed-forward MLP network, used in this work, consists of three layers: input, hidden, and output. The input layer has nodes representing the normalized features extracted from the measured vibration signals. There are various methods, both heuristic and systematic, to select the neural network structure and activation functions [24]. The number of input nodes was varied from 2 to 45 and that of the output nodes was 2. The target values of two output nodes can have only binary levels representing “normal” (N) and “failed” (F) bearings. In the MLPs, the sigmoidal activation functions were used in the hidden and output layers to maintain the outputs close to 0 and 1. The outputs were rounded to binary levels (0 and 1). The MLP was created, trained, and implemented using Matlab neural network toolbox with back-propagation (BPN) and the training algorithm of Levenberg-Marquardt. The ANN was trained iteratively using the training data set to minimize the performance function of mean square error (MSE) between the network outputs and the corresponding target values. No validation data were used in the present work. The classification performance of the MLPs was assessed using the test data set which had no part in training. The gradient of the performance function (MSE) was used to adjust the network weights and biases. In this work, an MSE of 10^{-6} , a minimum gradient of 10^{-10} , and a maximum iteration number (epoch) of 500 were used. The training process would stop if any of these conditions were met. The initial weights and biases of the network were generated automatically by the program.

4.2. Radial basis function networks

The structure of an RBF network is similar to that of an MLP. The activation function of the hidden layer is Gaussian spheroid function as follows:

$$y(x) = e^{-(\|x-c\|^2/2\sigma^2)}. \quad (3)$$

The output of the hidden neuron gives a measure of distance between the input vector x and the centroid c of the data cluster. The parameter σ , representing the radius of the hypersphere, is generally determined using iterative process selecting an optimum width on the basis of the full data sets. However, in the present work the width is selected along with the relevant input features using a GA-based approach. In the present work, the RBFs were created, trained, and tested using Matlab through a simple iterative algorithm of adding more neurons in the hidden layer till the performance goal is reached.

4.3. Probabilistic neural networks

The structure of a PNN is similar to that of an RBF, both having a Gaussian spheroid activation function in the first of the two layers. The linear output layer of the RBF is replaced with a competitive layer in PNN which allows only one neuron

to fire with all others in the layer returning zero. The major drawback of using PNNs was computational cost for the potentially large size of the hidden layer which could be equal to the size of the input vector. The PNN can be Bayesian classifier, approximating the probability density function (PDF) of a class using Parzen windows [17]. The generalized expression for calculating the value of Parzen approximated PDF at a given point x in feature space is given as follows:

$$f_A(x) = \frac{1}{(2\pi)^{p/2} \sigma^p N_A} \sum_{i=1}^{N_A} e^{-(\|x-c_i\|^2/2\sigma^2)}, \quad (4)$$

where p is the dimensionality of the feature vector and N_A is the number of examples of class A used for training the network. The parameter σ represents the spread of the Gaussian function and has significant effects on the generalization of a PNN.

One of the problems with the PNN is handling the skewed training data, where the data from one class are significantly more than the other class. The presence of skewed data is more likely in a real environment as the number of data for normal machine condition would, in general, be much larger than the machine fault data. The basic assumption in the PNN approach is the so-called prior probabilities, that is, the proportional representation of classes in training data should match, to some degree, the actual representation in the population being modeled [16, 17]. If the prior probability is different from the level of representation in the training cases, then the accuracy of classification is reduced. To compensate for this mismatch, the a priori probabilities can be given as input to the network and the class weightings are adjusted accordingly at the binary output nodes of the PNN [16, 17]. If the a priori probabilities are not known, then training data set should be large enough for the PDF estimators to asymptotically approach the underlying probability density.

In the present work, the data sets have equal number of samples from normal and faulty bearing conditions. The PNNs were created, trained, and tested using Matlab. The width parameter is generally determined using iterative process, selecting an optimum value on the basis of the full data sets. However, in the present work, the width is selected along with the relevant input features using the GA-based approach, as in case of RBFs.

5. GENETIC ALGORITHMS

GAs have been considered with increasing interest in a wide variety of applications [25, 26, 27]. These algorithms are used to search the solution space through simulated evolution of “survival of the fittest.” These are used to solve linear and nonlinear problems by exploring all regions of state space and exploiting potential areas through mutation, crossover, and selection operations applied to individuals in the population [25, 26]. The use of GA needs consideration of six basic issues: chromosome (genome) representation, selection function, genetic operators like mutation and crossover for reproduction function, creation of initial population,

termination criteria, and the evaluation (fitness) function. In the GA, a population size of ten individuals was used starting with randomly generated genomes. This size of population was chosen to ensure relatively high interchange among different genomes within the population and to reduce the likelihood of convergence within the population.

5.1. Genome representation

In the present work, GA is used to select the most suitable features and one variable parameter related to the particular classifier: the number of neurons in the hidden layer for MLPs and the width (σ) for RBFs and PNNs. Different mutation, crossover, and selection routines have been proposed for optimization [25]. In the present work, a GA-based optimization routine [28] was used.

5.1.1. MLP training

For MLPs, the genome X contains the row numbers of the selected features from the total set and the number of hidden neurons. For a training run needing N different inputs to be selected from a set of Q possible inputs, the genome string would consist of $N + 1$ real numbers. The first N numbers (x_i , $i = 1, N$) in the genome are constrained to be in the range $1 \leq x_i \leq Q$, whereas the last number x_{N+1} has to be within the range $S_{\min} \leq x_{N+1} \leq S_{\max}$. The parameters S_{\min} and S_{\max} represent, respectively, the lower and upper bounds on the number of neurons in the hidden layer of the MLP:

$$X = \{x_1, x_2, \dots, x_N, x_{N+1}\}^T. \quad (5)$$

5.1.2. RBF and PNN training

For RBFs and PNNs, the first N entries of the $(N+1)$ -element genome represent the row numbers of the selected features as in case of MLPs. However, the last element x_{N+1} represents the spread (σ) of the Gaussian function of (3) and (4) for RBFs and PNNs, respectively. For the present work, this was taken between 0.1 and 1.0 with a step size of 0.1.

5.2. Selection function

In a GA, the selection of individuals to produce successive generations plays a vital role. A probabilistic selection is used based on the individual's fitness such that the better individuals have higher chances of being selected. There are various schemes for selection process [25, 26]. In this work, normalized geometric ranking method was used because of better performance [26, 29]. In this method, the probability P_i for i th individual being selected is given as follows:

$$P_i = \frac{q}{1 - (1 - q)^P} (1 - q)^{r-1}, \quad (6)$$

where q represents the probability of selecting the best individual, r is the rank of the individual, and P denotes the population size. The parameter q is to be provided by the user. The best individual is represented by a rank of 1 and the worst having a rank of P . In the present work, a value of 0.08 was used for q .

5.3. Genetic operators

Genetic operators are the basic search mechanisms of the GA for creating new solutions based on the existing population. The operators are of two basic types: mutation and crossover. Mutation alters one individual to produce a single new solution, whereas crossover produces two new individuals (offsprings) from two existing individuals (parents). Let \bar{X} and \bar{Y} denote two individuals (parents) from the population and \bar{X}' and \bar{Y}' denote the new individuals (offsprings).

5.3.1. Mutation

In this work, nonuniform-mutation function [26] was used. It randomly selects one element x_i of the parent \bar{X} and modifies it as $\bar{X}' = \{x_1, x_2, \dots, x'_i, \dots, x_N, x_{N+1}\}^T$ after setting the element x'_i equal to a nonuniform random number in the following manner:

$$x'_i = \begin{cases} x_i + (b_i - x_i)f(G) & \text{if } r_1 < 0.5, \\ x_i - (x_i - a_i)f(G) & \text{if } r_1 \geq 0.5, \\ x_i & \text{otherwise,} \end{cases} \quad (7)$$

$$f(G) = \left(r_2 \left(1 - \frac{G}{G_{\max}} \right) \right)^s,$$

where r_1 and r_2 denote uniformly distributed random numbers between $(0, 1)$; G is the current generation number; G_{\max} denotes the maximum number of generations; s is a shape function used in the function $f(G)$; and a_i and b_i represent, respectively, the lower and upper bounds for each variable i .

5.3.2. Crossover

In this work, heuristic crossover [26] was used. This operator produces a linear extrapolation of two individuals using the fitness information. A new individual \bar{X}' is created as per (8) with r being a random number following uniform distribution $U(0, 1)$, and \bar{X}' is better than \bar{Y}' in terms of fitness. If \bar{X}' is infeasible, given as $\eta = 0$ in (10), then a new random number r is generated and a new solution is created using (8):

$$\bar{X}' = \bar{X} + r(\bar{X} - \bar{Y}), \quad (8)$$

$$\bar{Y}' = \bar{X}, \quad (9)$$

$$\eta = \begin{cases} 1 & \text{if } x'_i \geq a_i, x'_i \leq b_i \forall i, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

The choice of heuristic crossover was based on its main characteristics of utilizing the fitness function to determine the search direction for better performance [26].

5.4. Initialization, termination, and evaluation functions

To start the solution process, the GA has to be provided with an initial population. The most commonly used method is the random generation of initial solutions for the population.

TABLE 1: Performance comparison of classifiers without feature selection for different sensor locations.

Data sets	Input features	Test success (%)		
		MLP ($N = 24$)	RBF ($\sigma = 1.0$)	PNN ($\sigma = 0.1$)
Signal x	1–45	87.50	50.00	83.33
Signal y	1–45	95.83	50.00	83.33
Signal z	1–45	87.50	95.83	83.33

The solution process continues from one generation to another, selecting and reproducing parents until a termination criterion is satisfied. The most commonly used terminating criterion is the maximum number of generations.

The creation of an evaluation function to rank the performance of a particular genome is very important for the success of the training process. The GA will rate its own performance around that of the evaluation (fitness) function. The fitness function used in the present work returns the number of correct classification of the test data. The better classification results give rise to higher fitness index.

6. SIMULATION RESULTS

The data set $45 \times 144 \times 2$ consisted of 45 normalized features for each of the three signals split in form of 24 segments of 1024 samples each, with two bearing conditions and two sets of run. Two cases were studied. In the first case (Case A), data of the first set of run were further divided into two equal subsets. The first 12 bins of each signal were used for training the ANNs giving a training set of 45×72 and the rest (45×72) were used for testing. In the second case (Case B), the complete data of the first set of run were used for training the ANNs and the data of the second set of run were used for testing. In both cases, the testing data sets had no part in the training of ANNs. In each case, the training was based on the training data sets only. No validation set was used for early stopping of the training process because of the limited size of the available data sets. However, for a larger data set, it would be preferred to have separate sets for training, validation, and testing.

For each of the MLPs and RBFs, two output nodes were used, whereas for PNNs only one output node was used. The use of one output node for all classifiers would have been enough. However, the classification success was not satisfactory with one output node in case of MLPs and RBFs for the present data sets with the particular choice of network structure and activation functions. The target value of the first output node was set as 1 and as 0 for normal and failed bearings, respectively, and the values were interchanged (0 and 1) for the second output node. For PNNs, the target values were specified as 1 and 2, respectively, representing normal and faulty conditions. Results are presented to see the effects of accelerometer location (direction) and signal processing for diagnosis of machine condition using ANNs with and without feature selection based on GA. The training success for each case was 100 percent.

6.1. Performance comparison of ANNs without feature selection

In this section, classification results are presented for straight ANNs without feature selection for the data of the first set of run (Case A). For each straight MLP, number of neurons in the hidden layer was kept at 24, and for straight RBFs and PNNs, widths (σ) were kept constant at 1.00 and 0.10, respectively. These values were found on the basis of several trials of training the ANNs.

6.1.1. Effect of sensor location

Table 1 shows the classification results for each of the signals x , y , and the resultant z using all input features (1–45). For all classifiers, test success was mostly unsatisfactory. The test success was in the range of 87.50%–95.83% for MLPs, 50.00%–95.83% for RBFs, and 83.33% for PNNs. The classification error was in the failure to recognize a fault, termed as fault-not-recognized (FNR) which may suggest the overlap of the features of faulty bearings to that of normal bearings. The performance of MLPs and PNNs is reasonably consistent for all signals; however, for RBF, the signal z gives a classification success around 45% higher than the signals in other two directions (x and y). This may be attributed to the better classification capability of RBF using features extracted from the combined signal z .

6.1.2. Effect of signal preprocessing

Table 2 shows the effects of signal processing on the classification results for straight ANNs with all three signals. In each case, all the features from the signals with and without signal processing were used. To see the relative effectiveness of the lower- and the higher-order features of the original signals, results were obtained for the feature ranges separately (1–4 and 5–9) and together (1–9). The use of the three signals x , y , and z gave rise to better classification success than using individual signals. This may be due to the fact that the feature sets extracted from the three signals gave better representation of the bearing conditions than the individual signals. The classification performance of using only lower-order moments (1st–4th) was better than using the higher-order moments (5th–9th). The use of all nine features gave classification success better than higher-order features only, but slightly worse than the lower-order features.

The test success, based on the last four rows of data sets, was in the range of 90.97%–95.83% for MLPs, 98.61% for RBFs, and 94.44% for PNNs. Here again, the classification error was of type FNR for all cases, except for PNN, it was

TABLE 2: Performance comparison of classifiers without feature selection for different signal preprocessing.

Data sets	Input features	Test success (%)		
		MLP ($N = 24$)	RBF ($\sigma = 1.0$)	PNN ($\sigma = 0.1$)
Signals x, y, z	1–4	97.22	100.0	97.22
Signals x, y, z	5–9	90.28	50.00	75.00
Signals x, y, z	1–9	95.83	98.61	94.44
Derivative/integral	10–27	95.83	98.61	94.44
High-/lowpass filtering	28–45	90.97	98.61	94.44
All features	1–45	95.83	98.61	94.44

TABLE 3: Performance comparison of classifiers with feature selection for different sensor locations.

Data sets	GA with MLP		GA with RBF			GA with PNN		
	Features N	Test success (%)	Features	σ	Test success (%)	Features	σ	Test success (%)
Signal x	5, 21, 42 17	95.83	8, 13, 41	0.90	100	3, 10, 13	0.60	100
Signal y	4, 14, 26 28	100	3, 4, 29	0.50	95.83	6, 14, 32	0.30	100
Signal z	9, 21, 41 23	95.83	3, 12, 21	0.80	87.50	19, 42, 44	0.50	100

4.17% FNR and 1.39% false alarm (FA). The misclassification suggests the inadequacy of separation of the data sets (normal and faulty) for all three classifiers. From examination of the data sets, no particular explanation for the difference in misclassification type (FNR or FA) for PNNs could be put forward since for each case, the data sets included equal number of samples from normal and faulty classes.

6.2. Performance comparison of ANNs with feature selection

In this section, classification results are presented for ANNs with feature selection based on GA for the Case A. Only three features were selected from the corresponding ranges. In case of MLPs, the number of neurons in the hidden layer was selected in the range of 10 and 30, whereas for RBFs and PNNs, the Gaussian spread was selected in the range of 0.1 and 1.0 with a step size of 0.1.

6.2.1. Effect of sensor location

Table 3 shows the classification results along with the selected parameters for each of the signals x , y , and the resultant z . In all cases, the input features were selected by GA from the entire range (1–45). The test success improved substantially in each case with feature selection, compared with the results of Table 1. The test success was 95.83%–100% for MLPs, 87.50%–100% for RBFs, and 100% for PNNs. The classification error was of type FNR with MLPs and RBFs. Features selected for different schemes are also shown for comparison. Though some of the features were selected by two of the three schemes, there was no apparent fixed combination of features. However, it should be noted that features from higher-order moments (features 5–9, 14–18, 23–27, 32–36, and 41–45) were selected by GAs quite often, justifying their inclusion in the feature sets.

6.2.2. Effect of signal preprocessing

Table 4 shows the effects of signal processing on the classification results for the signals x , y , and z with GA. In all cases, only three features from the signals with and without signal preprocessing were used from each of these ranges. The effectiveness of the lower-order moments (1st–4th) was found to be better than the higher-order moments (5th–9th). In case of PNN, the higher-order moment (5th) improved the classification success more than using only the lower-order features. Here again, the selection of features from higher-order moments was evident. The groupings of the features selected for different cases showed no apparent bias or preference. From the results of last four rows, the test success was 97.22%–100% for MLPs, 88.89%–100% for RBFs, and 94.44%–98.61% for PNNs. For PNNs, the classification errors were as follows: 1.39%–4.17% FNR and 0%–1.39% FA.

6.3. Performance of PNNs with selection of six features

In this section, results are presented for PNNs with six features from the corresponding ranges as shown in Tables 5 and 6. The test success was 100% for all cases with individual signals (Table 5) and also for all signals and features taken together (Table 6). Here again, the features from higher-order moments were selected by GAs. The computation time (on a PC with Pentium III processor of 533 MHz and 64 MB RAM) for training the PNNs is shown for each case. These values (36.893–41.130 seconds) are not much different from PNNs with three features (36.232–40.819 seconds) but are higher than without feature selection (0.250–0.761 seconds). These values are substantially lower than RBFs and MLPs, however, direct comparison is not made among the ANNs due to difference in code efficiency. It should also be mentioned that the difference in computation time should not be very important if the training is done offline.

TABLE 4: Performance comparison of classifiers with feature selection for different signal preprocessing.

Data sets (input feature range)	GA with MLP			GA with RBF			GA with PNN		
	Features	N	Test success (%)	Features	σ	Test success (%)	Features	σ	Test success (%)
Signals x, y, z (1–4)	1, 2, 3	21	100	1, 2, 4	0.90	100	1, 2, 3	0.10	87.50
Signals x, y, z (5–9)	5, 6, 8	17	95.83	5, 6, 7	0.80	80.56	5, 6, 7	0.10	76.39
Signals x, y, z (1–9)	2, 3, 5	27	100	1, 2, 3	0.50	100	1, 4, 5	0.10	94.44
Derivative/integral (10–27)	10, 12, 13	19	98.61	11, 12, 13	0.10	94.22	10, 12, 14	0.10	97.22
High-/lowpass filtering (28–45)	32, 35, 42	19	97.22	30, 38, 39	0.60	88.89	28, 33, 37	0.10	94.44
All features (1–45)	4, 5, 41	23	100	11, 13, 27	0.10	93.06	11, 12, 14	0.10	98.61

TABLE 5: PNN performance with six selected features for different sensor locations.

Data set	GA with PNN (six features)			
	Input features	Width (σ)	Training time (s)	Test success (%)
Signal x	1, 2, 2, 13, 24, 33	0.50	37.274	100
Signal y	4, 11, 13, 15, 17, 37	0.40	38.866	100
Signal z	12, 13, 31, 32, 37, 42	0.50	39.597	100

TABLE 6: PNN performance with six selected features for different signal preprocessing.

Data set	GA with PNN (six features)			
	Input features	Width (σ)	Training time (s)	Test success (%)
Signals x, y, z	1, 2, 2, 3, 4, 6	0.20	36.893	97.22
Derivative/integral	10, 11, 14, 21, 22, 25	0.10	39.797	94.44
High-/low-pass filtering	28, 29, 33, 37, 39, 43	0.10	41.130	95.83
All features	1, 5, 10, 20, 29, 32	0.10	37.664	100

6.4. Results with second test data set

In the previous sections, both training and test feature sets were derived from the same vibration signals of the first set of run (Case A) although the test data were not used in training. In this section, simulation results are presented for Case B using the entire data of the first set of run for training of ANNs and the data of the second set of run for testing. The size of training and test data was 24576 each. The normalization was carried out using maximum values of the particular feature set [10]. Table 7 shows the results of different generation numbers on the classification performance of ANNs with six features. Training time for each number of generation is also shown for comparison. Training time, as expected, increases with the generation number. From the results, a generation number of 30 would be adequate for six features. However, to account for lower number of features, a generation number of 40 was used for subsequent results (Tables 8 and 9).

Table 8 shows the effect of number of input features on the ANN classification performance with a generation number of 40. In general, the test success improved with higher number of input features, it was 100% for all classifiers with 8 features. The test success with six features was 100% for MLP

and PNN, and 99.31% for RBF. Though the performance of MLP was better than the other two classifiers with lower number of features, the training time for MLP was much higher.

6.5. Results with second test data set using statistical normalization

The data sets discussed so far were normalized in magnitude to keep the features within ± 1 . In this section, results are presented using the statistical normalization scheme with zero mean and unit standard deviation, see Table 9. The performance of PNNs for two normalization schemes can be compared from the results presented in last columns of Tables 7 and 9. The classification success of the statistical normalization scheme (with zero mean and standard deviation of 1) is slightly better than the magnitude normalization scheme for lower number of features (up to 3). However, the test success deteriorated with the scheme of statistical normalization for higher number of features. Training time increased somewhat with higher number of features but not in direct proportion.

To investigate the separability of the data sets with and without bearing fault, three features selected by GA were

TABLE 7: PNN performance with six selected features for different generation numbers.

Number of generations	GA with PNN (six features)			
	Input features	Width (σ)	Training time (s)	Test success (%)
15	12, 23, 37, 39, 40, 41	0.30	21.125	93.06
20	3, 19, 25, 36, 37, 39	0.60	29.031	95.83
30	5, 10, 27, 37, 39, 42	0.60	43.938	100
40	3, 5, 10, 32, 39, 42	0.20	58.312	100

TABLE 8: ANN performance with magnitude normalized data for different number of features selected.

Number of selected features	Test success (%) (40 generations)		
	MLP	RBF	PNN
2	100	83.33	83.33
3	98.61	96.53	88.19
4	94.79	98.61	95.83
5	97.22	97.92	97.22
6	100	99.31	100
8	100	100	100

TABLE 9: PNN performance with statistically normalized data for different number of selected features.

Number of selected features	GA with PNN (40 generations)			
	Input features	Width (σ)	Training time (s)	Test success (%)
2	9, 26	0.10	51.938	86.11
3	9, 31, 40	0.10	53.859	91.67
4	1, 17, 28, 37	0.10	52.578	82.64
5	20, 21, 28, 37, 38	0.60	53.348	84.72
6	1, 2, 28, 30, 37, 38	0.50	54.750	90.97
8	3, 8, 20, 26, 29, 34, 37, 39	0.80	59.156	84.72

plotted, as shown in Figures 4a and 4b. In Figure 4a, the magnitude normalized features are shown, whereas in Figure 4b, the statistically normalized features are shown. In both cases, the data clusters are not well separated and have considerable overlap. This can explain the unsatisfactory classification success with three features only. The smaller width selected by GA for lower number of features (up to 3) may be attributed to the closeness of the data clusters. However, the separation of classes is slightly better for the statistically normalized data than the magnitude normalized data. Another normalization scheme was also examined by making the features zero mean and then normalizing by the absolute maximum value. However, no significant difference in classification performance of the magnitude normalized data (with and without zero mean) was noticed.

7. CONCLUSIONS

A procedure is presented for the diagnosis of bearing condition using three classifiers, namely, MLP, RBF, and PNN

with GA-based feature selection from time-domain vibration signals. The selection of input features and the appropriate classifier parameters have been optimized using a GA-based approach. The roles of different vibration signals and preprocessing techniques have been investigated. The effects of number of features and generations on the classification success have been studied. The use of six selected features gave 100% test success for most of the cases considered in this work. Though the classification performance of MLP was comparable with that of PNN with six features, the training time of MLP was much higher than PNN. The false classification with lower number of features may be attributed to the overlap of data sets with and without bearing faults. The effectiveness of the features from lower-order statistics was better than the higher-order moments. However, the selection of features from higher-order moments using GAs justified the inclusion of these moments in the feature sets. The results show the potential application of GAs for selection of input features and classifier parameters in ANN-based condition monitoring systems.

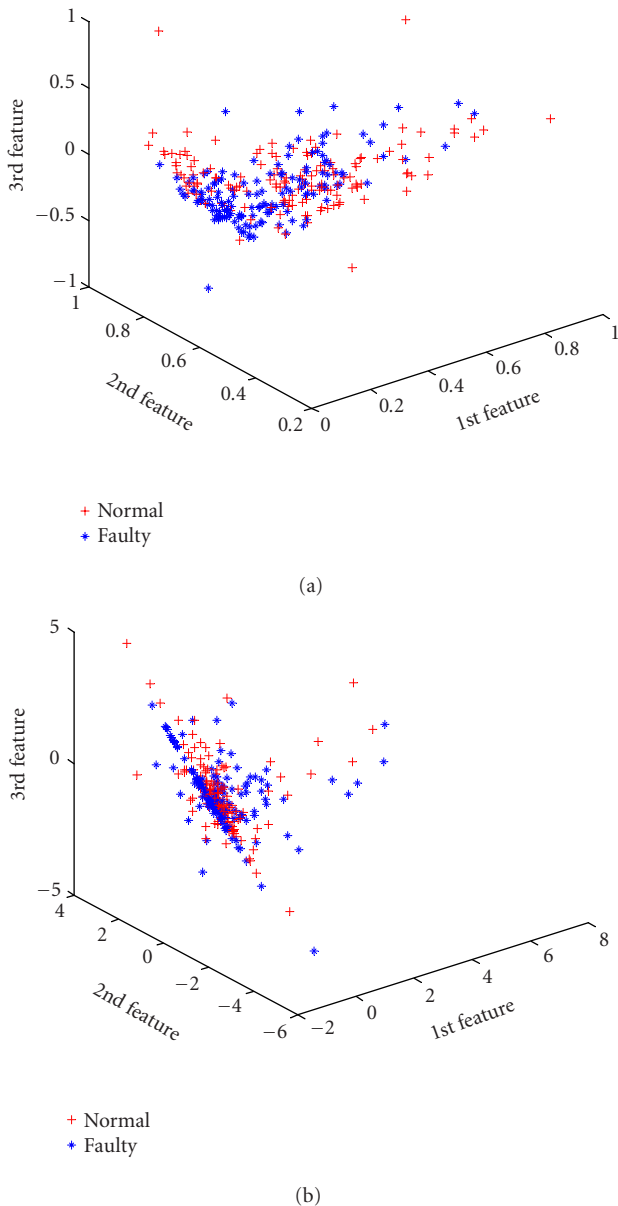


FIGURE 4: (a) Scatter plot of features with magnitude normalization. (b) Scatter plot of features with statistical normalization.

However, in the present study, the data sets include equal representation from normal and faulty bearings under similar operating conditions. All the features have been considered from time-domain vibration signals. The sample size used for extraction of features is kept relatively small for the two-class (normal and faulty) problem considered in this work. For multiple fault conditions (multiclass problems), the issue of suitable sample size for feature extraction needs to be examined. This leaves a scope for future work including consideration of skewed data sets, incorporation of frequency-domain data, studying the effects of varying machine conditions, and extension to multiclass problems covering different types and levels of bearing faults.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support from Sultan Qaboos University Grant IG/ENG/MIED/01/01 to carry out the research. The authors would also like to thank the reviewers for their suggestions that helped revising the paper to its present form.

REFERENCES

- [1] J. Shiroishi, Y. Li, S. Liang, T. Kurfess, and S. Danyluk, "Bearing condition diagnostics via vibration and acoustic emission measurements," *Mechanical Systems and Signal Processing*, vol. 11, no. 5, pp. 693–705, 1997.
- [2] P. D. McFadden, "Detection of gear faults by decomposition of matched differences of vibration signals," *Mechanical Systems and Signal Processing*, vol. 14, no. 5, pp. 805–817, 2000.
- [3] A. K. Nandi, "Advanced digital vibration signal processing for condition monitoring," in *Proc. 13th International Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management (COMADEM' 00)*, pp. 129–143, Houston, Tex, USA, December 2000.
- [4] R. B. Randall, Ed., "Special issue on gear and bearing diagnostics," *Mechanical Systems and Signal Processing*, vol. 15, no. 5, pp. 827–1029, 2001.
- [5] K. R. Al-Balushi and B. Samanta, "Gear fault diagnosis using energy-based features of acoustic emission signals," *Proceedings of the I MECH E Part I Journal of Systems and Control Engineering*, vol. 216, no. 3, pp. 249–263, 2002.
- [6] J. Antoni and R. B. Randall, "Differential diagnosis of gear and bearing faults," *Transactions of the ASME: Journal of Vibration and Acoustics*, vol. 124, no. 2, pp. 165–171, 2002.
- [7] A. C. McCormick and A. K. Nandi, "Classification of the rotating machine condition using artificial neural networks," *Proceedings of the I MECH E Part C Journal of Mechanical Engineering Science*, vol. 211, no. 6, pp. 439–450, 1997.
- [8] M. R. Dellomo, "Helicopter gearbox fault detection: a neural network based approach," *Transactions of the ASME: Journal of Vibration and Acoustics*, vol. 121, no. 3, pp. 265–272, 1999.
- [9] B. Samanta and K. R. Al-Balushi, "Use of time domain features for the neural network based fault diagnosis of a machine tool coolant system," *Proceedings of the I MECH E Part I Journal of Systems and Control Engineering*, vol. 215, no. 3, pp. 199–207, 2001.
- [10] B. Samanta and K. R. Al-Balushi, "Artificial neural network based fault diagnostics of rolling element bearings using time-domain features," *Mechanical Systems and Signal Processing*, vol. 17, no. 2, pp. 317–328, 2003.
- [11] A. K. Jain and J. Mao, Eds., "Special issue on artificial neural networks and statistical pattern recognition," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, 1997.
- [12] A. Baraldi and N. A. Borghese, "Learning from data: general issues and special applications of radial basis function networks," Tech. Rep. TR-98-028, International Computer Science Institute, Berkeley, Calif, USA, 1998.
- [13] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, England, UK, 1995.
- [14] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feed-forward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [15] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 5, no. 2, pp. 305–316, 1993.
- [16] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, no. 1, pp. 109–118, 1990.

- [17] P. D. Wasserman, *Advanced Methods in Neural Computing*, Van Nostrand Reinhold, New York, NY, USA, 1995.
- [18] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.
- [19] L. B. Jack, A. K. Nandi, and A. C. McCormick, "Diagnosis of rolling element bearing faults using radial basis functions," *EURASIP Journal on Applied Signal Processing*, vol. 6, pp. 25–32, 1999.
- [20] L. B. Jack and A. K. Nandi, "Comparison of neural networks and support vector machines in condition monitoring applications," in *Proc. 13th International Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management (COMADEM' 00)*, pp. 721–730, Houston, Tex, USA, December 2000.
- [21] L. B. Jack, *Applications of artificial intelligence in machine condition monitoring*, Ph.D. thesis, Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, England, UK, 2000.
- [22] L. B. Jack and A. K. Nandi, "Genetic algorithms for feature extraction in machine condition monitoring with vibration signals," *IEE Proceedings Vision, Image and Signal Processing*, vol. 147, no. 3, pp. 205–212, 2000.
- [23] B. Samanta, K. R. Al-Balushi, and S. A. Al-Araimi, "Use of genetic algorithm and artificial neural network for gear condition diagnostics," in *Proc. 14th International Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management (COMADEM' 01)*, pp. 449–456, Manchester, England, UK, September 2001.
- [24] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2nd edition, 1999.
- [25] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, Mass, USA, 1989.
- [26] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, New York, NY, USA, 3rd edition, 1996.
- [27] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 22–37, 1996.
- [28] C. R. Houck, J. A. Joines, and M. Kay, "A genetic algorithm for function optimization: a Matlab implementation," Tech. Rep. NCSU_IE_TR_95_09, North Carolina State University, Raleigh, NC, USA, 1995.
- [29] J. A. Joines and C. R. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's," in *Proc. 1st IEEE Conference on Evolutionary Computation (ICEC' 94)*, pp. 579–584, Orlando, Fla, USA, June 1994.

B. Samanta received his B.Tech. (Honours) and Ph.D. degrees in mechanical engineering from Indian Institute of Technology (IIT), Kharagpur. He is currently an Associate Professor in the Department of Mechanical and Industrial Engineering at Sultan Qaboos University (SQU), Muscat, Sultanate of Oman. Prior to joining SQU, Dr. Samanta was an Assistant Professor at IIT. His major research interests include system dynamics and control, machine condition monitoring and diagnostics, rotordynamics, vibration, smart structures, applications of artificial intelligence (AI) techniques, and soft computing. He has over fifty research publications including articles in the journals of professional bodies like American Society of Mechanical



Engineers (ASME), Institution of Mechanical Engineers (IMechE), American Institute of Aeronautics and Astronautics (AIAA), and Institute of Electrical and Electronics Engineers (IEEE). He is a member of ASME, IEEE, and Control System Society (CSS-IEEE). Dr. Samanta and the coauthors represent a research group engaged in a number of funded research projects in the area of machine condition monitoring and diagnostics.

Khamis R. Al-Balushi started his professional career in 1971 working for Petroleum Development Oman (PDO) in various jobs related to oil and gas production. During his employment with PDO, he went to UK and obtained his B.S. (Honours.) degree in mechanical engineering from the University of Wales, Swansea, in 1981. After obtaining his B.S. degree, he worked for PDO and progressed in his career to a position of Senior Production Supervisor. In 1986, he joined Sultan Qaboos University (SQU). During his career at SQU, he obtained his M.S. degree in mechanical engineering from Texas Tech University, USA, in 1990 and his Ph.D. degree in mechanical engineering from Cranfield University, UK, in 1996. He is now an Assistant Professor in the Department of Mechanical and Industrial Engineering. He was the Assistant Dean for Student Academic Affairs in the College of Engineering at SQU for two years (2000–2002). His research interest is in machinery condition monitoring and diagnostics. He has published in a number of international journals and conference proceedings. Dr. Al-Balushi is a member of the research group engaged in a number of funded research projects in the area of machine condition monitoring and diagnostics.



Saeed A. Al-Araimi received his Ph.D. degree in engineering management from the University of Missouri-Rolla in 1993, his M.S. degree in engineering management from Northwestern University in 1988, and his B.S. degree in industrial engineering from the University of New Haven in 1981. Prior to joining Sultan Qaboos University (SQU) in 1986, Dr. Al-Araimi was the Deputy Director General of Industry, Ministry of Commerce and Industry (MCI), Sultanate of Oman. He was the Assistant Dean for Graduate Studies and Research in the College of Engineering at SQU (2000–2002). He is currently an Assistant Professor in the Department of Mechanical and Industrial Engineering at SQU. His areas of interest are in project and operations management, management of technology, total quality management, statistical process control, multicriteria decision making, and engineering education. He has publications in a number of international journals and conference proceedings. Dr. Al-Araimi is a member of the research group engaged in a number of funded research projects in the area of machine condition monitoring and diagnostics.

