An Early Bearing Fault Diagnosis using Effective Feature Selection Methods and Data Mining Techniques

S.Devendiran [#1], K.Manivannan [*2], Soham Chetan Kamani[*3], & Razim Refai*[4]
[#1 *2,*3&*4] School of Mechanical and Building Sciences, VIT University, Vellore, India
[1] devendiran@vit.ac.in
[2] manivannan.k@vit.ac.in
[3] sohamkamani@gmail.com
[4] razim.refai@gmail.com

Abstract—This paper proposes a binary particle swarm optimization (BPSO) and binary Genetic Algorithm (BGA) in feature selection process using different fitness functions in the field of bearing fault diagnosis. The vibration data obtained by extracting vibrational signals, considering four cases such as Normal Bearing, Inner race fault, Outer race fault and Ball fault, at constant speed conditions. This paper proposes four fitness functions applied on BPSO and BGA that gives rise to a reduce feature set. Furthermore, standard classification algorithms such as KStar, Naïve Bayes, JRip, J48 are used.Furthermore, Neural Network classification algorithms such as Back Propagation Network (BPN), RBF network and Deep Neural network (DNN) apart from standard classifiers are used.The aim of the paper to identify an appropriate combination scheme contains feature selection method based on appropriate fitness function algorithm along with better classifier that maintains high accuracy with adequate computational time. It was observed that GA possess higher accuracy than BPSO, yet the computation time is high. It is observed that PSO based schema of feature selection are optimal with high levels of classification accuracy and optimum computational time.

Keyword-Fault Diagnosis, Statistical Features, Genetic Algorithm (GA), Particle Swarm Optimization(PSO), data mining techniques etc.,

I. INTRODUCTION

Rolling-element bearings have vital applications in various engineering fields. This is mainly due to its load bearing capacity, anti-friction property, low speed as well as high speed applications. However, the roller bearings yield to failure due to variation in axial and radial loads and also due to wear and prolonged usage. Due to its wide scale application, it is important to diagnose any type of fault within the shortest time possible so as to reduce the devastation caused due to the fault. These faults are diagnosed by 'Bearing conditioning methods'. Conditioning methods can include information derived from physical inspection, wear properties or analysis of working parameters.[1] Furthermore, it was found that vibration analysis was the most optimum technique used for fault diagnosis as it a reliable source due to the fact that it provides real-time data over a wide range of conditions giving rise to an unbiased dataset. Moreover, vibration analysis not only maintains correlation with the working conditions of the bearing, but also provides unique results for distinct kinds of faults. [2]This paper contains of 4 types of signals: 1) Normal (no fault); 2) Outer race fault; 3) Inner race fault; 4) Ball fault. Numerous data are collected under each category to make the result as comprehensive as possible. The real time data collection is in the time domain. Hence in order to get the Frequency domain, There is a need to perform Fast Fourier Transform (FFT)[3]. FFT is one of the most widely used signal processing techniques. However, the accuracy of FFT is questionable due to frequency resolution, steady state magnitude results and other data processing parameters. In order to counter this drawback, Short-Time Fourier Transform (STFT) is used. In STFT, the signal is processed by combining data from the time as well as frequency domain. This enables a clear-cut picture of when and at what frequencies a signal occurs. The variable for STFT is the size of the window used; this could also be considered as a drawback resulting in lesser precision.[4]Wavelet transformation (WT) is a more refined signal processing technique that discretizes the signal into wavelets depending on the frequency[5]. WT acquires data from both the frequency and time domain and hence can extract transitory features[6]. WT is categorized into Continuous Wavelet Transformation (CWT) and Discrete wavelet transformation (DWT)[6][7] .The next step that follows signal processing, is Statistical feature reduction. Statistical parameters like kurtosis, standard deviation, skewness, mean, variance, maximum, minimum, median, mean slope, maximum slope and minimum slope were calculated for time domain as well as frequency domain, from the dataset. One such technique is Principal component analysis (PCA). PCA is widely used method for data mining and data recollection applications. It transforms a set of correlated variables into a set of linearly uncorrelated variables.[8]Another technique used is Independent Component Analysis(ICA). ICA aims at decomposing a multivariate signal into independent signals. The distribution of these independent signals are

non-gaussian in nature. ICA is also used to separate two mixed signals. This is followed by running the processed data through classifiers. These classifiers, depending on their nature, train and test a particular percentage of data respectively in order to give results based on response time and accuracy, enabling the easy comparison and contrast the listed options. This paper aims at choosing an algorithm and the respective classifier that optimizes the result. Hence,PSO(Particle Swarm Optimization) technique has been used and compared it's results with Genetic Algorithm(GA). Although both of these have their disadvantages and advantages, an optimum balance must be obtained. PSO is said to have faster computational time; and in order to diagnose the fault rapidly, this is exactly what is needed. In order to justly compare PSO and GA, both these algorithms have been passed through the same set of 4 fitness functions.(that are described in detail later) The classification algorithms used in this paper are Kstar, Naïve bayes, Jrip, J48 and decision table. These results are contrasted with standard classifiers such as Deep Neural Network, RBF Network and Back Propagation Network. Yet again, the classifiers used in this paper have a faster computation time than RBF, BPN etc and thus plays a vital role in fault diagnosis.

II. EXPERIMENTAL SETUP

The proposed methodology is verified by performing tests on the designed experimental setup.The experimental set-up shown in Fig-1, comprises of components such as variable frequency drive (VFD), three phase 0.5 hp AC motor, bearing ,belt drive, gearbox and brake drum dynamometer with scale. This experiment utilizes a standard deep groove ball bearing (No. 6005). Measurements of the vibration acceleration signals are captured by a Triaxial type accelerometer (Vibration sensor) that is fixed over the bearing block. The data acquisition system used was 24 Bit, ATA0824DAQ51 and the signals were collected at a sampling frequency of 12800 Hz. The bearing was maintained at a constant rotating speed of 1700 r/min. The brake drum dynamometer applied a constant load and the speed was monitored by a tachometer. Fig-2 indicates normal, outer race, inner race and ball fault (1mm crack depth) conditions were formed using the EDM process. Table 1 depicts the number of sample data collected for each bearing condition. Each sample contains 6000 data points. Analysis of the signal is done to find out various conditions of bearing component using time domain and frequency domain by varying amplitudes. Time domain plots are illustrated in Fig.3.The frequency of the abnormal vibration is called fault frequency which is corresponds to the fault location.

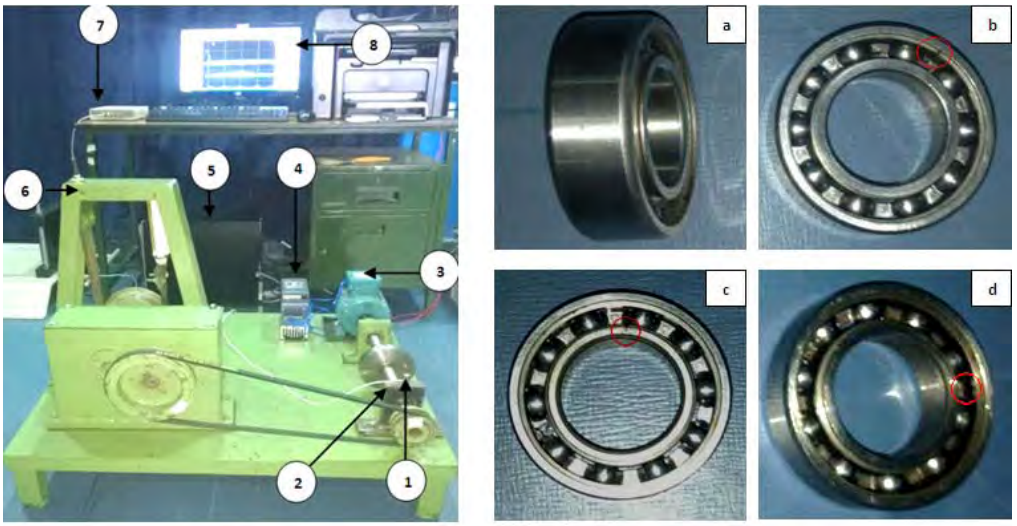

Fig 1. Experimental Setup

The following equations gives a detailed preview of fault characteristic frequencies for different parts of the bearing .The characteristic bearing frequencies are BPFO- Ball Pass Frequency Outer Race, BPFI- Ball Pass Frequency Inner Race, FTF- Fundamental Train Frequency and BSF- Ball Spin Frequency. One of the most basic approach for bearing conditioning monitoring is Frequency analysis. Fast Fourier transform (FFT), is used to transform the time series data to frequency domain, where the signal is used to deduce the sine and cosine waves from the sample. In practice, analysing those frequencies and measuring the amplitude variations in the particular frequency and its side bands as well the harmonics of those frequencies will provide information regarding the health of the bearing. The bearing conditions are difficult to be differentiated by their FFT spectral shown in Fig.4.

Shaft rotational frequency- Fs = shaft speed/60 (1)

$$(\text{BPFO}) = Fs\left(\frac{N_b}{2}\right)\left(1-\frac{B_d}{P_d}\cos\phi\right) \quad (2)$$

$$(\text{BPFI}) = Fs\left(\frac{N_b}{2}\right)\left(1-\frac{B_d}{P_d}\cos\phi\right) \quad (3)$$

$$(\text{FTF}) = Fs\left(\frac{1}{2}\right)\left(1-\frac{B_d}{P_d}\cos\phi\right) \quad (4)$$

$$(\text{BSF}) = Fs\left(\frac{P_d}{2B_d}\right)\left(1-\frac{{B_d}^2}{{P_d}^2}\cos^2\phi\right) \quad (5)$$

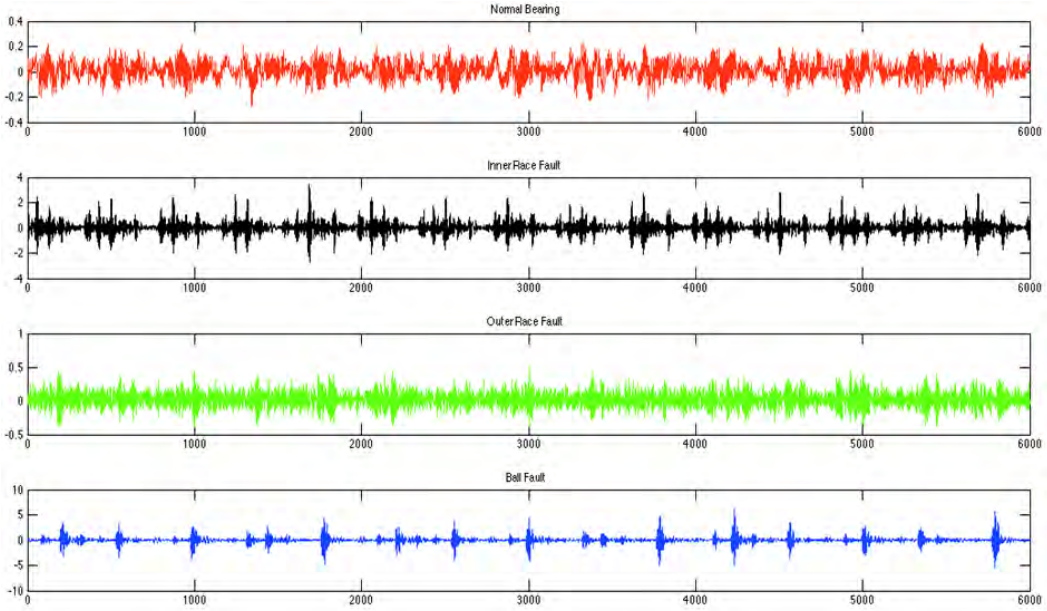

Fig.2. Time domain signals of various states of bearing(X axis–data points, Y axis-Amplitude in 'g')

After FFT, the bearing can be diagnosed by analysing the abnormal frequency-domain amplitude .After plotting time domain and frequency domain, the signals are decomposed using WPT. The decomposed signals are thenprocessed and the statistical features are derived using Table 2. Statistical features are extracted for both time domain as well as frequency domain and then given for feature reduction and other processes.

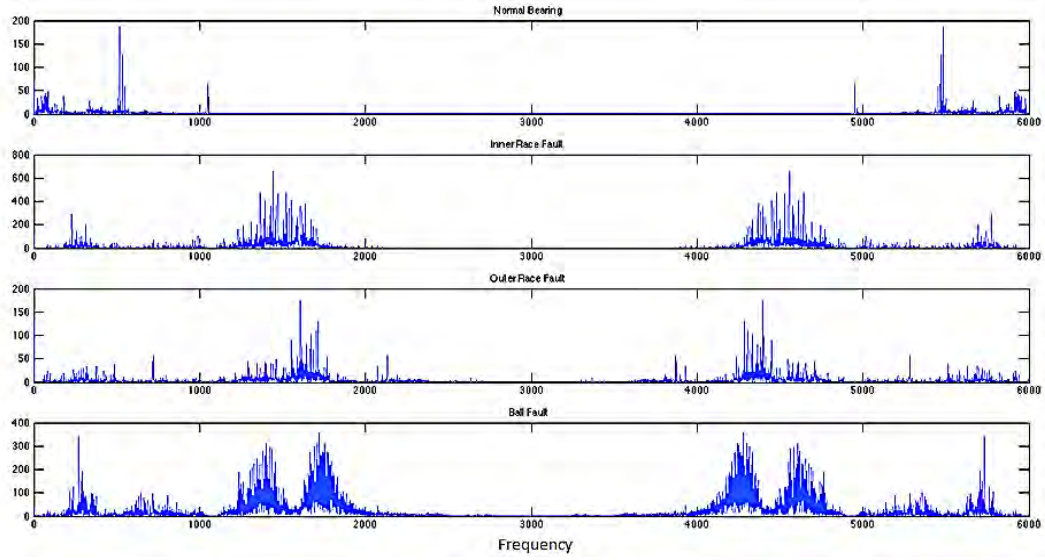

Fig.3. Frequency spectrum of various states of bearing (X axis–Frequency, Y axis-Amplitude in 'g')

TABLE I. Statistical features extracted

Feature	Equation	Definition
Mean	$k_{mean} = \frac{\sum_{i=1}^{n} k_i}{n}$	Average of all values in the population
Standard deviation	$k_{Sd} = \sqrt{\left(\frac{1}{n-1}\sum_{i=1}^{n}(k_i - \mu)^2\right)}$	Square root of an unbiased estimator of the variance of the population
Kurtosis	$k_{kur} = \frac{1}{n}\sum_{i=1}^{n}\left(k(t) - \bar{k}\right)^4$	Fourth central moment of X, divided by fourth power of its standard deviation
Root means square	$k_{rms} = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n} k^2\right)}$	Root of sum of squared values
Variance	$k_{\text{var}} = \sigma^2 = \frac{\sum_{n}(k_i - \bar{k})^2}{n-1}$	Measures how far a set of numbers is spread out
Peak to RMS	$k_{peak.to.rms} = \frac{\lvert k_h \rvert}{k_{rms}}$	Ratio of the largest absolute value and the root mean squared value
Peak to peak	$k_{p2p} = \lvert k_h - k_l \rvert$	Difference between largest and smallest values
Skewness	$k_{skew} = \frac{\sum_{i=1}^{n}\left(k_i - \bar{k}\right)^3}{n\sigma^3}$	Third central moment of the value, divided by the cube of its standard deviation
Minimum	$k_{\min} = \min(k_i)$	Minimum value in the set
Maximum	$k_{\max} = \max(k_i)$	Maximum value in the set

Where '*k*' is a signal series and 'n' is the total number of signal samples

III.THEORETICAL BACKGROUND OF CLASSIFIERS

Classification of data is the segregation of data based on decided attributes and qualities. Data mining essentially classifies the useful data into a sub population and separates them from the entire data set. An algorithm that carries out this classification process can be termed as a classifier. The term classifier at times is referred to as a mathematical function that processes the data fed in and outputs data that is distinctly categorised. Different classifiers have different algorithms, each having its own computing time and efficiency. In this paper, several classifiers have been used and compared their results with each other. The classifiers used are enlisted below with a brief related background.

A. K Star Algorithm

The K start algorithm is an Instance based classifier. It compares problem instances to existing instances seen in training and is a representational form of lazy learning. K* is performed using the following values.P* is probability of all paths between the data points 'b' to 'a', where 'b' is to be classified. A K* function is computed using the P* value. What's to be noted is that K* is not directly a distance function - it is a symmetric non zero function. Since the distance function in K* is not conventional and depends on probability of the possible paths between the paths, it holds its distinction of being an entropy based instance classifier unlike many others. It utilizes all different transformation paths between the points in forming the entropy measure.

$$P*\left(\frac{b}{a}\right) = \sum_{t \in p:t(a)=b} P(t) \tag{6}$$

$$K*(b/a) = -\log_2 P*\left(\frac{b}{a}\right) \tag{7}$$

B.Deep Neural Network

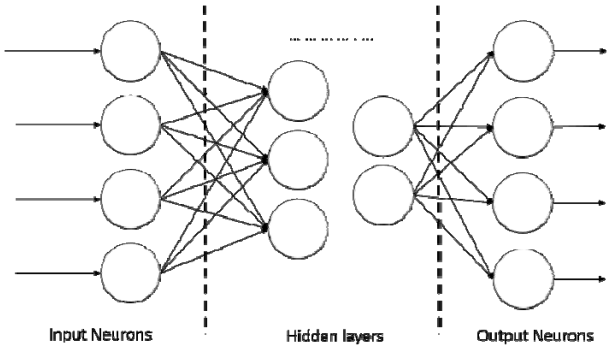

Fig. 4. Deep Neural Network

A deep neural network (DNN) is defined to be an artificial neural network with at least one hidden layer of units between the input and output layers. A DNN can be trained with the standard back propagation algorithm. The weight updates can be done via stochastic gradient descent using the following equation.

$$\Delta w_{ij}(t+1) = \Delta w_{ij}(t) + \eta \frac{\partial C}{\partial w_{ij}} \tag{8}$$

In the equation. η is the learning rate, and C is the cost function. The choice of the cost function depends on factors such as the learning type (supervised, unsupervised, reinforcement, etc.) and the activation function. For example, when performing supervised learning on a multiclass classification problem, common choices for the activation function and cost function are the softmax function and cross entropy function, respectively.

Soft max function is the following:

$$p_j = \frac{\exp(x_j)}{\sum_k \exp(x_k)} \tag{9}$$

Cross entropy is the following function

$$C = -\sum_j d_j \log(p_j) \tag{10}$$

In the above equations, p_j represents the class probability and x_j and x_k represent the total input to units j and k respectively. d_j represents the target probability for output unit j and p_j is the probability output for j after applying the activation function. Issues with DNNs are when they are naively trained. Issues including over fitting when training and the computation time to solve a DNN.

C.Naïve Bayes Algorithm

Uses bayes theorem from probability theory to form a classification algorithm. Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)} \tag{11}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c) \tag{12}$$

In the equation P(c|x) is the posterior probability of class (target) given predictor (attribute). P(c) is the prior probability of class. P(x|c) is the likelihood which is the probability of predictor given class. P(x) is the prior probability of predictor. The posterior probability can be calculated by first, constructing a frequency table for each attribute against the target. Then, transforming the frequency tables to likelihood tables and finally use the Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

D .The C4.5 Algorithm

The C4.5 algorithm consists of two phases, a building or growing phase and a pruning phase.

In the building phase, training sample data sets are used that have distinctive attributes. These training sets are partitioned in an iterative process until all the records have identical class labels. A decision tree is created

where new nodes are added to represent further partitioning. The formation of this decision tree is highly dependent on the test attribute that is selected. An information entropy evaluation function is used by C4.5 as the selection criteria. The entropy evaluation function first identifies the class of the training set, calculates the expected information values, obtain the information gain depending on the test attribute after partition, deduce the partition information value and finally calculate the gain ratio. The gain ratio values are help map the decision tree with the highest gain ratio value taken as the root of the tree.

The pruning phase involves removing branches of the decision tree that are less reliable such that a better classification performance is obtained over the either space despite possibly having a higher error over the training set. Pruning of decision trees can be done by pre-pruning (construction time pruning) or by post pruning. When it is required to stop the expansion of the decision tree, the pre-pruning methods are used. Post pruning occurs when the decision tree has been fully constructed. The C4.5 incorporates a predicted error rate during its classification process to deal with the problem of over training. This error based pruning technique helps reduce the total error rate of the root node.

The JRip classifier is an java instance of C4.5 and hence its description is not repeated.

E.Radial Basis Function Network (RBFN)

An RBFN performs classification by measuring the input's resemblance to instances from the training set. Each RBFN neuron stores a "model", which is just one of the instances from the training set. When a new input is to be classified, each neuron computes the Euclidean distance between the input and its model. Roughly speaking, if the input more closely resembles the class A models than the class B models, it is classified as class A.

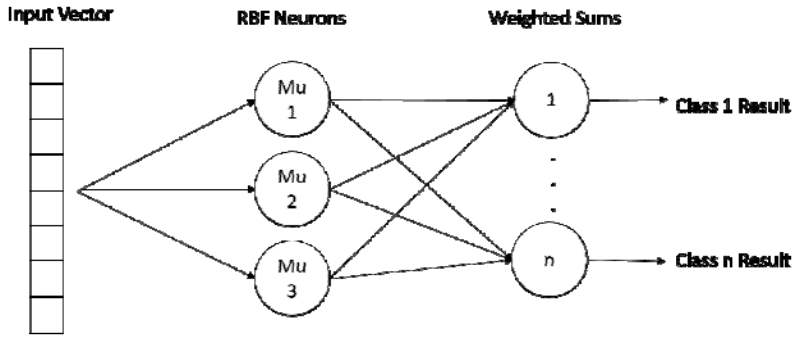

Fig. 5. Radial Basis Function Network

The input vector is the *n*-dimensional vector that you are trying to classify. The entire input vector is shown to each of the RBF neurons. Each RBF neuron stores a "model" vector which is just one of the vectors from the training set. Each RBF neuron compares the input vector to its model, and outputs a value between 0 and 1 which is a measure of resemblance. The neuron's response value is also called its "activation" value. n Figure 2, Mu1, Mu2 and Mu3 are the neurons that hold the reference or model vectors which are then compared to the input vectors. The output of the network consists of a set of nodes, one per category that are to be classified. Each output node computes a sort of score for the associated category. Typically, a classification decision is made by assigning the input to the category with the highest score. Each RBF neuron computes a measure of the similarity between the input and its model vector (taken from the training set). This resemblance is usually computed using a Gaussian function:

$$\phi(x) = e^{-\beta\|x-\mu\|^2} \tag{13}$$

In the Gaussian distribution, mu refers to the mean of the distribution. Here, it is the model vector which is at the centre of the bell curve.

There are other similarity functions that can be used like a Thin plat simpline - $\varphi(z) = z^2\log z$, Quadratic - $\varphi(z) = (z^2 + r^2)^{1/2}$, Inverse Quadratic - $\varphi(z) = 1 / (z^2+r^2)^{1/2}$. In the mentioned similarity functions z is $x - \mu$.

F. Back Propagation Neural Network

The back propagation algorithm trains a given feed-forward multilayer neural network using a set of known output classifications for some standard inputs. When each entry of the known classifications is presented to the network, the network examines its output response to the sample input attributes. The output response is then compared to the known and desired output and the error value is calculated. The different weights of the neural network are then adjusted to adapt to the errors arising to the known classifications. The back propagation algorithm is based on *Widrow-Hoff delta learning rule* in which the weight adjustment is done through *mean square error* of the output response to the sample input. When there are inadequate known classifications the standard inputs are repeated multiple times in order to achieve the least error arising due to the weights of the neural network.

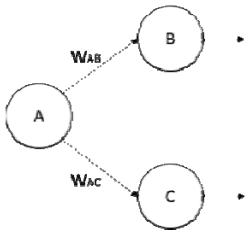

Fig. 6 Back Propagation Neural Network

Initially apply the inputs to the network and work out the output – The initial output could be anything, as the initial weights were random numbers. Next work out the error for neuron B.

$$\text{ErrorB} = \text{OutputB}\ (1\text{-OutputB})(\text{TargetB} - \text{OutputB}) \quad (14)$$

The "Output (1-Output)" term is necessary in the equation because of the Sigmoid Function – if there was only a threshold neuron it would just be (Target – Output). Change the weight. Let $W+_{AB}$ be the new (trained) weight and W_{AB} be the initial weight.

$$W+_{AB} = W_{AB} + (\text{ErrorB} \times \text{OutputA}) \quad (15)$$

Notice that it is the output of the connecting neuron (neuron A) used (not B). All the weights in the output layer are updated in this way. Calculate the Errors for the hidden layer neurons. Unlike the output layer these can't be calculated directly (due to lack of a Target), so they are Back Propagated from the output layer (hence the name of the algorithm). This is done by taking the Errors from the output neurons and running them back through the weights to get the hidden layer errors. For example if neuron A is connected as shown to B and C then the errors are taken from B and C to generate an error for A.

$$\text{ErrorA} = \text{Output A}\ (1 - \text{Output A})(\text{ErrorB}\ W_{AB} + \text{ErrorC}\ W_{AC}) \quad (16)$$

Again, the factor "Output (1 - Output)" is present because of the sigmoid squashing function. Having obtained the Error for the hidden layer neurons now proceed as in stage 3 to change the hidden layer weights. By repeating this method a network of any number of layers can be trained.

IV. FEATURE REDUCTION PROCESS

A. Using GA as a feature reduction algorithm

In genetic algorithms, it is believed that future generations adapt to surrounding nearby more than parent generations. Better Adaptability is achieved by two properties that are CrossOver and Mutation. CrossOver is defined as if two different set of values for same set is given. A part of set of values is mixed with the other part. Mutation is defined as some value in a set are changed. The best set of k values in a population is evaluated using a fitness function or an objective function. This is the metric used to evaluate a given instance of a population. The following constitutes the genetic algorithm procedure. Candidate solutions are initialized randomly. N different sets with k number of values. A few sets are selected by evaluating with objective function. Let's say x sets. Crossover and Mutation are performed. And generate xnew Sets. The above procedure is repeated for a specific number of generations which is computationally feasible. At the end of the iterations the sample with the best fitness metric constitutes the solution. The genetic algorithm functions in a manner that the values aforementioned in the set to be optimized consist of all binary elements i.e. each element in the set can have only a value of 1 | 0. The objective function is modified to calculate some metric over this given set during the iterative stage until convergence is reached. The binary genetic algorithm works in exactly the same manner as the regular genetic algorithm except for the fact that every individual element in the set is a binary 1 | 0 element.

B. Using PSO as a feature reduction algorithm

Particle swarm optimization mimics the way swarms work. Swarms, say ants for example, scatter all around the surroundings in search of food. On their way, they excrete a fluid called pheromones. These pheromones evaporate very quickly. Ants choose a path which has high pheromone content. Shortest path has highest pheromone level and all ants follow this path. Forming the solution incurs a cost called cost function. The aim of the optimization algorithm is to reduce the cost function. Formally, it is tried to find out 'a' for which $f(a) < f(b)$ for all b in search space where the search space refers to the area where the solution exists. The optimization is carried out in the following manner. Let's take a set of particles S. For every particle in S the following steps are done. Firstly, generate its position randomly. (They are in practice generated uniformly from lower boundary to upper boundary of the search space). Secondly, Initialize the particle's best known position to its initial position (after few iterations you might get performance of a particle at few position, choose the best one). Thirdly, if the particle best known position lets say $f(p) < f(g)$ where g is the swarm's best known position. Make swarm's best known position as p. The particle's velocity is initialized.

Now it is iterated and the following are performed until an acceptable solution is formed where f(a) is low enough or until a specific number of iterations are done. End the loop if there is an acceptable solution (f(a) is low enough to be used). While iterating, firstly, pick two random numbers R_p and R_g. Secondly, the particles velocity is updated using a mathematical velocity using R_p and R_g. Thirdly, update particle's positions if f(new position) is less than f(best position), best position is updated as new position. If f(new position) is less than f(best position of entire swarm of particles), best position of the swarm is updated to new position. Particle may be travelling in multiple dimensions. Velocity in each dimension might be different. Think metric is used as vectors but not scalars.The binary particle swarm optimization approach is to utilize the binary digits 1 | 0 to represent the individual particles' best known solution. This approach of using the binary digits to formulate the best known positions oversimplifies the problem to ensure that the iterations have to carry until the best known solution position i.e. can be reached. This makes working with the process to find convergence a little easier on a merely logistic degree.

C. Application of fitness function in GA &PSO

The solution of the feature selection algorithms such as GA and PSO is based on the fitness function used. Therefore it is essential that a variety of fitness functions are tested and chosen. For the purpose of feature selection, it is needed to select the most significant features present. Hence the fitness functions mainly rely on the similarity of two signals as the measure of fitness. The more similar two signals are, the lesser is the fitness. The aim is to obtain a feature set containing features that are as dissimilar from each other as possible.

D. Fitness function 1 (Multiplication and summation with normalization):

This fitness function multiplies all pairs of feature data and adds their summation products.

First, the data is loaded as x_{ij} where 'i' denotes the feature number and 'j' denotes each table value of the feature. Thus one requirement of this fitness function is that all the features have the same number of data points. The input matrix is given as the input parameter. It is a 1xT matrix which consists of either 0s or 1s, depending on whether the feature has to be considered or not.

First each feature is normalized as follows:

$$xn_{ij} = \frac{x_{ij} - \mu(x_i)}{\mu(x_i)} \tag{17}$$

Where,

$$\mu(x_i) = \sum j\, x_{ij} \tag{18}$$

Next, for each pair of normalized featuresthe summation of the products are found as,

$$s = \sum_j \sum_{l=0}^{T-1} \sum_{k=l+1}^{T} xn_{kj} xn_{lj} \quad , input(k) \neq 0, input(l) \neq 0 \tag{19}$$

Where T is the total number of features to be considered.

Finally, the sum is normalized by dividing it my the number of iterations

$$sn = \frac{s}{\sum_{l=0}^{T-1} \sum_{k=l+1}^{T} 1} \tag{20}$$

The output of the fitness function is 'sn'

E. Fitness function 2 (Correlation):

This fitness function performs correlation of all pairs of feature data and adds their summation products. First , the data is loaded as x_{ij} where 'i' denotes the feature number and 'j' denotes each table value of the feature. The input matrix is given as the input parameter. It is a 1xT matrix which consists of either 0s or 1s , depending on whether the feature has to be considered or not.

First each feature is normalized as follows:

$$xn_{ij} = \frac{x_{ij} - \mu(x_i)}{\mu(x_i)} \tag{21}$$

Where,

$$\mu(x_i) = \sum j\, x_{ij} \tag{22}$$

Next , for each pair of normalized features the sum of the correlation of the pairs is found as,

$$Cm = \sum_{l=0}^{T-1} \sum_{k=l+1}^{T} \sum_{\tau=-\infty}^{\infty} xn_{kj} xn_{l(\tau+j)} \quad , input(k) \neq 0, input(l) \neq 0 \tag{23}$$

Where T is the total number of features to be considered. And m is the iteration number

Next the summation of all the correlations is found as,

$$s = \sum_m Cm \tag{24}$$

Finally , the sum is normalized by dividing it my the number of iterations

$$sn = \frac{s}{\sum_{l=0}^{T-1}\sum_{k=l+1}^{T} 1} \quad (25)$$

The output of the fitness function is 'sn'

E.Fitness function 3 (Multiplication and summation without normalization):

This fitness function multiplies all pairs of feature data and adds their summation products , however , normalization is not used. This fitness function can be used for datasets , where the magnitude of certain feature datasets is required to be high , because when normalization is used , even minute changes can be detected as large ones if the overall magnitudes of the readings is small.

First , the data is loaded as x_{ij} where 'i' denotes the feature number and 'j' denotes each table value of the feature. Thus one requirement of this fitness function is that all the features have the same number of data points.The input matrix is given as the input parameter . It is a 1xT matrix which consists of either 0s or 1s , depending on whether the feature has to be considered or not.

First , the summation of the products is found as,

$$s = \sum_j \sum_{l=0}^{T-1}\sum_{k=l+1}^{T} x_{kj}x_{lj} \quad , input(k) \neq 0, input(l) \neq 0 \quad (26)$$

Where T is the total number of features to be considered.

Finally , the sum is normalized by dividing it my the number of iterations

$$s = \frac{s}{\sum_{l=0}^{T-1}\sum_{k=l+1}^{T} 1} \quad (27)$$

The output of the fitness function is 's'

F.Fitness function 4 (Based on Genetic Algorithm)

The GA is processed with two objectives. The first objective is to determine a minimum in the within class distance, and the second is to determine a maximum in the average between-class distance. For this purpose, the fitness function is defined as :

$$J = J_i + (\frac{1}{J_b}) \quad (28)$$

The within class distance is given by

$$J_i = (\frac{1}{n_i})\sum_{k=1}^{n}(x_{ki} - m_i)^T (x_{ki} - m_i) \quad (29)$$

Class i = 1, …, c; m_i is the mean vector of class i; ni is the number of samples in class i; pi is the number of samples in class i.

The between class distance is given by

$$J_b = \sum_{i=1}^{c}(m - m_i)^T (m_i - m) \quad (30)$$

m is the mean vector of all of the classes.

V. EXPERIMENTAL ANALYSIS

The experimental procedure consists of collecting data from the vibration of a motor under various fault conditions. After the data is collected, statistical features are extracted from it.

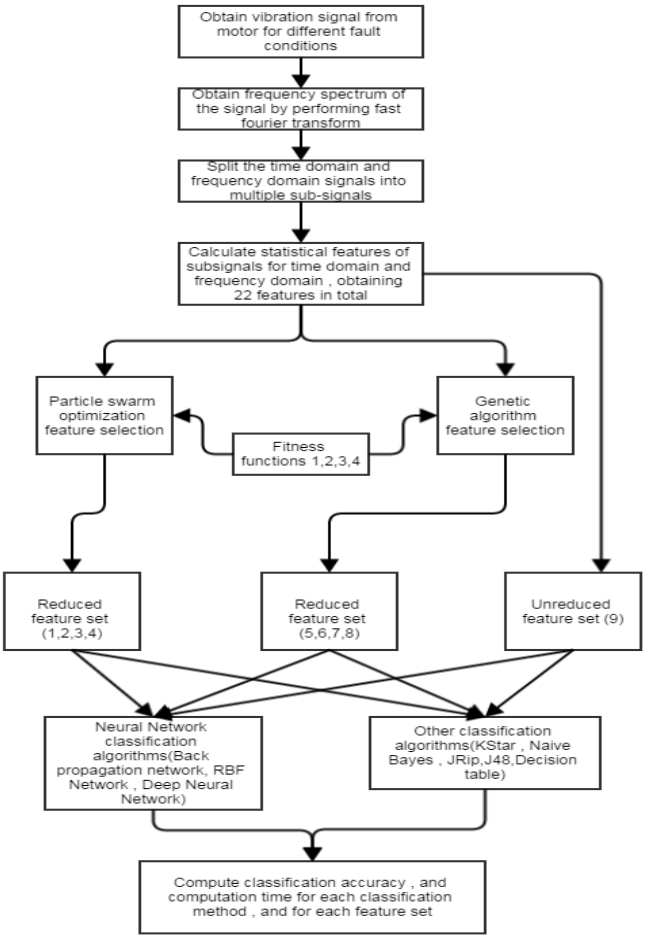

Fig.7. Feature selection and classification process

The data signal obtained is expressed in time domain and frequency domain. For each domain there are 11 statistical features as explained in the previous section, which make a total of 22 statistical features. The features selected are optimized through feature selection methods. The methods used are particle swarm optimization algorithm and genetic algorithm. 4 different fitness functions as mentioned in the previous section are used with each algorithm and thus 8 optimized feature sets are obtained which are selected by each feature selection algorithm and for each fitness function. Along with the original unreduced feature set a total of 9 feature sets are obtained for comparison. The accuracy of neural network classifiers and other classifiers is computed and the computation time is also calculated. From the results obtained, the best classification algorithm and feature selection method is determined.

A. Data collection

The data collected is obtained from the vibration of a motor under various conditions. Thus a signal is obtained for each condition of the motor such as normal, IR fault, OR fault, and ball fault. Each signal is originally obtained in the time domain. Therefore to obtain the frequency domain a discrete Fourier transform is applied to the signals.

B. statistical feature extraction

Each signal is broken into subsignals by considering a subset of the original number of data points. For this purpose a total of 40 subsignals is obtained for each signal. For each subsignal , the statistical features like local mean , standard deviation , etc. are considered thus obtaining 40 data points for each signal and for each feature.

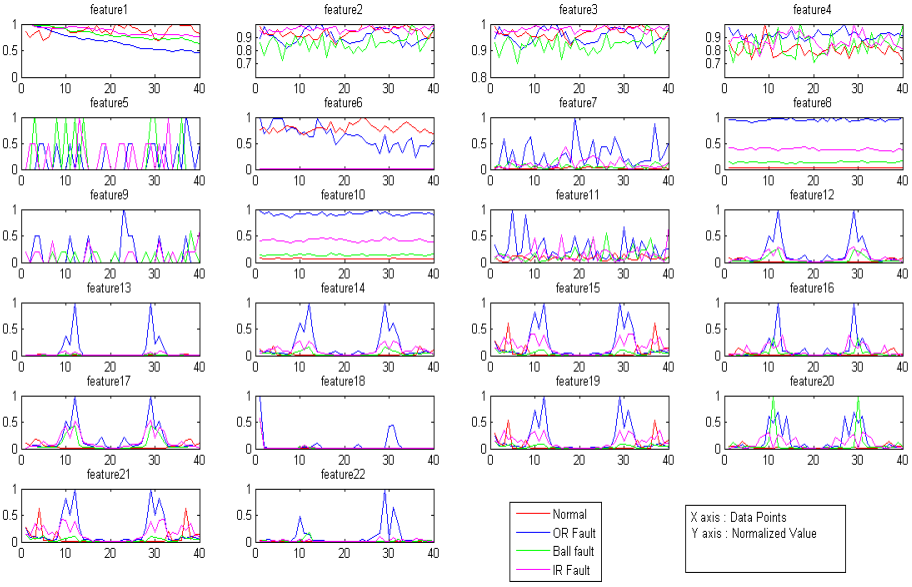

Fig 8. Graphical representation of features selected for each fault case

Feature 1 to 11 are in the time domain and features 12 to 22 are in the frequency domain. It is advantageous to analyse statistical features in the frequency domain as well because cases with faults show different frequency peaks and peaks which are higher than that without a fault.

C. Feature reduction process

For feature reduction process Particle swarm optimization (PSO) algorithm and genetic algorithm (GA) are used. These algorithms select optimum features according to the fitness function provided. Four different fitness functions are proposed and each algorithm selects the best features based on what gives the best results for each fitness function. Thus 8 feature sets are obtained, that is, 4 feature sets for each feature selection algorithm for the four fitness functions. The unreduced feature set of the original 22 features is considered as well and therefore there are 9 total feature sets taken into consideration. The computation time for each feature selection algorithm is calculated.

D. Classification process

The classification accuracy is computed for each feature set using different classifiers and neural nets. For this purpose, back propagation network (BPN), Radial basis function network (RBFN) and deep neural network (DNN) are used and other classifiers such as KStar, JRip, Decision tree, naïve bayes, and J48 algorithms are used. The classification accuracy and computation time for each classifier are calculated.

VI. DIAGNOSIS RESULTS

The original signal obtained from the motor was processed to give 22 statistical features, out of which 11 features were of the time domain and 11 of the frequency domain. The feature selection algorithms used were particle swarm optimization (PSO) and genetic algorithm (GA) each algorithm requiring a fitness function. There were 4 fitness functions tested and hence, a total of 8 reduced feature sets were determined, using each algorithm with the 4 different fitness functions. Each point on the evolution diagram for a particular fitness function indicates the fitness value calculated using that fitness function for each generation. The blue mark indicates mean fitness value while the black mark indicates the best fitness value. The computation time of each feature selection algorithm was compared. It is found that the time taken by PSO is far less than that of genetic algorithm. It is apparent that the PSO algorithm is superior to the genetic algorithm with respect to computation times, and fitness function 3 takes the least computation time for both GA and PSO algorithms. The figure (Fig.11) shows the classification accuracy of various classifiers. It is seen that for any type of classifier, there exists a feature selection method which has an accuracy that is greater than the original unreduced set of features.

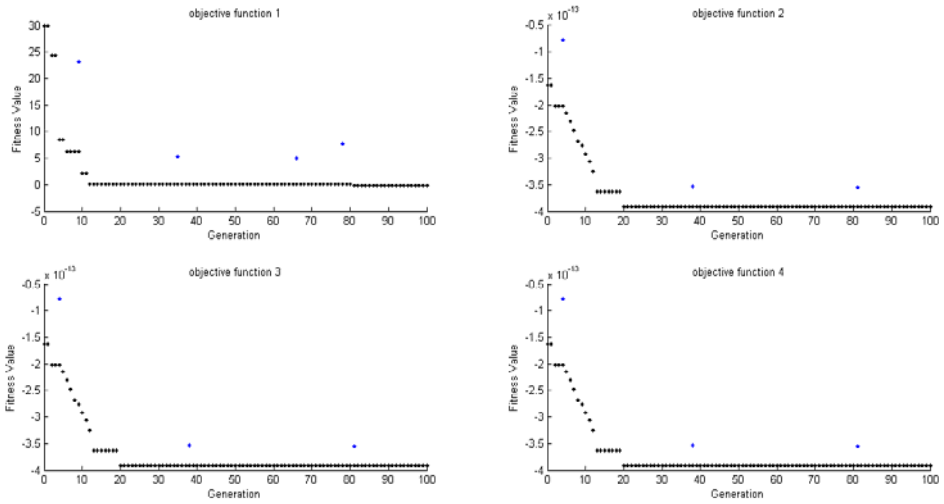

Fig 9. Plot of fitness value at each iteration (generation) for different fitness functions in genetic algorithm (Evolution diagram)

TABLE .II. Computation time for feature selection algorithms

Feature Selection Algorithm	Obj.Func	Features Selected(index)	Comp. time
PSO	1	1,2,3,5,6,11,18,19	2.065s
	2	4,5,7,8,9,12,13,18	5.272s
	3	5,7,9,13,14,18,19,20	0.527s
	4	5,6,8,10,15,16,20,21	3.141s
GA	1	1, 2,3,4,6,9,11,16	40.992s
	2	8, 9,12,14,15,18,20,22	100.817s
	3	1,3,4,5,8,9,13,18	11.055s
	4	5,6,8,10,15,19,20,21	43.832s
Unreduced	N/A	All	N/A

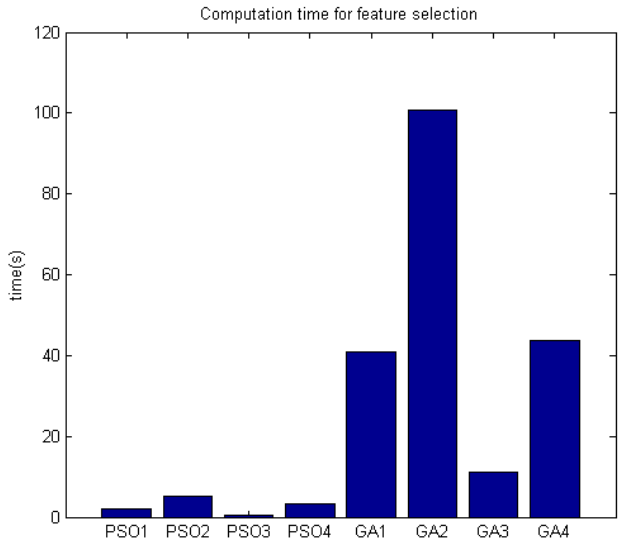

Fig 10. Bar graph of computation times of feature selection algorithms

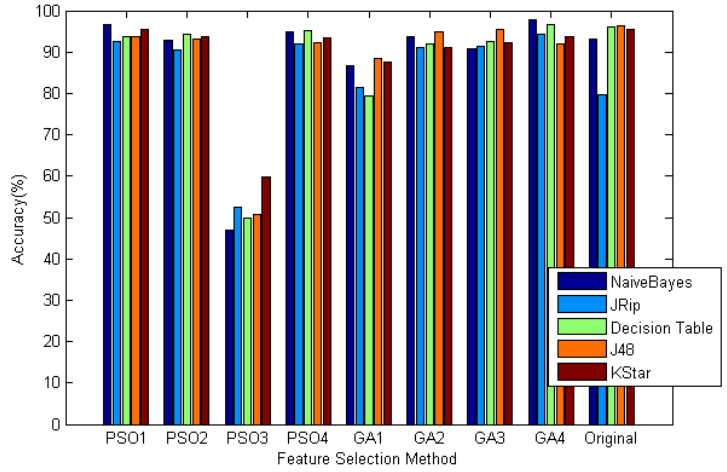

Fig 11. Accuracy of classifiers

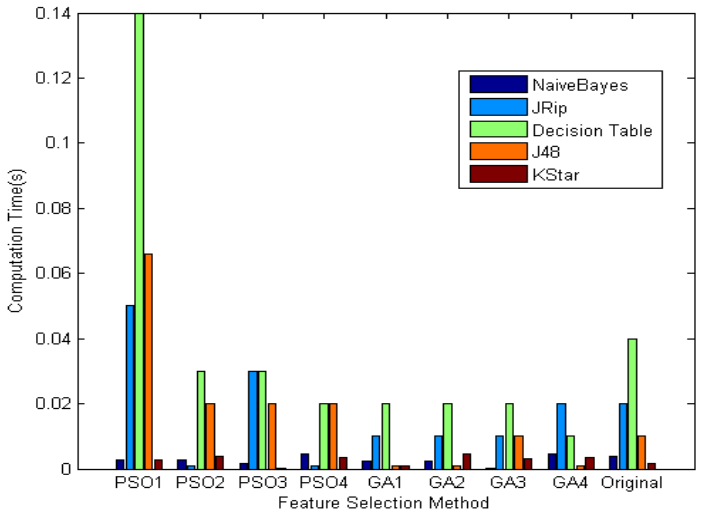

Fig 12. Computation time of classifiers

The above figure shows the computation times for various classifiers, for each feature selection method. The computation time in general for the unreduced feature set is seen to be greater than the feature set selected by the feature selection methods, with the exception of PSO algorithm with fitness function 1.

TABLE III. Accuracy and computation time of classifiers

Feature Selection	Obj.Func	Features Selected(index)	Niave bayes		Jrip		Decision table		J48		Kstar	
Algorithm			Accuracy(%)	time(s)	Accuracy(%)	time(s)	Accuracy(%)	time(s)	Accuracy(%)	time(s)	Accuracy(%)	time(s)
PSO	1	1,2,3,5,6,11,18,19	96.7081933	0.0028	92.6444003	0.05	93.725493	0.14	93.630813	0.066	95.44866305	0.0026
	2	4,5,7,8,9,12,13,18	92.7529035	0.0027	90.4169973	0.001	94.3616453	0.03	93.074742	0.02	93.67983527	0.0038
	3	5,7,9,13,14,18,19,20	46.7842392	0.0016	52.4452407	0.03	49.7271363	0.03	50.7444065	0.02	59.8061361	0.0002
	4	5,6,8,10,15,16,20,21	94.9518704	0.0045	92.0067335	0.001	95.0951282	0.02	92.3543694	0.02	93.3857902	0.0033
GA	1	1, 2,3,4,6,9,11,16	86.8254053	0.0025	81.3346713	0.01	79.3014332	0.02	88.4787602	0.001	87.71653918	0.001
	2	8, 9,12,14,15,18,20,22	93.7034319	0.0025	91.1102736	0.01	92.0214541	0.02	94.7783268	0.001	91.07702663	0.0045
	3	1,3,4,5,8,9,13,18	90.8552958	0.0002	91.3993602	0.01	92.5786133	0.02	95.3382535	0.01	92.29873652	0.0032
	4	5,6,8,10,15,19,20,21	97.8390614	0.0046	94.2269981	0.02	96.7536103	0.01	92.0266779	0.001	93.74804376	0.0036
Unreduced	N/A	all	93.1050255	0.0038	79.618817	0.02	95.9288172	0.04	96.375168	0.01	95.37140513	0.0017

Besides the above classifiers, different neural nets were also used for classification such as BPN (back propagation network) , RBFNN (Radial basis function neural network) and DNN(deep neural network).For neural networks, the classification error is also a function of the number of hidden layer neurons. Hence the

number of hidden layer neurons is chosen so as to minimize the classification error. The classification error of the different neural networks as a function of hidden layer neurons is shown below.

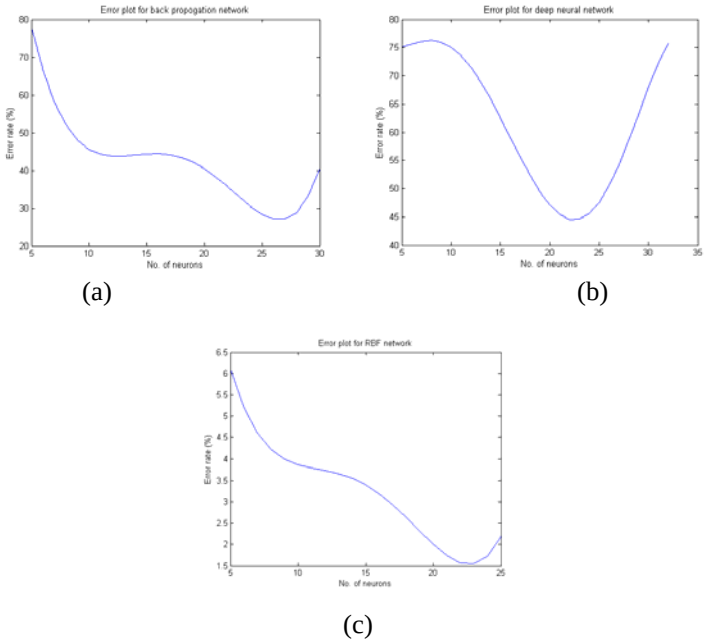

Fig 13. Accuracy of neural networks vs number of hidden layer neurons

It is observed that for each type of neural net, the optimal number of hidden layer neurons is different hence, the number corresponding to the minima is chosen. Thus, for each neural network and feature set, the classification accuracy and computation time were computed. The optimal amount of hidden layer neurons were found by testing each network with different number of hidden layer neurons by trial and error and selecting the one with higher accuracy and lower computation time. For BPN and DNN the number of hidden layer neurons varies significantly for the different feature sets, however for RBFN the optimal hidden layer neurons that satisfy both accuracy and computation time is found to be at a constant 20. Deep neural network is seen to have the least classification accuracy out of the neural networks, and BPN for most feature sets has a slightly better accuracy than RBF network.

TABLE IV. Accuracy, computation time and optimal hidden layer neurons for neural network classifiers

Feature Selection	Obj.Func	Features Selected(index)	BPN			DNN			RBFN		
Algorithm			Accuracy(%)	Neurons	time(s)	Accuracy(%)	Neurons	time(s)	Accuracy(%	Neurons	time(s)
PSO	1	1,2,3,5,6,11,18,19	94.2969342	8	0.041853	48.24578025	32	0.04035111	92.45455	20	0.042077927
	2	4,5,7,8,9,12,13,18	93.6594564	5	0.040942	51.89670843	29	0.04842387	91.08355	20	0.040405076
	3	5,7,9,13,14,18,19,20	71.0110923	25	0.04873	26.20585053	29	0.04029407	87.90982	20	0.043442593
	4	5,6,8,10,15,16,20,21	93.606369	5	0.040866	47.29983136	20	0.04042833	98.75002	20	0.049642883
GA	1	1, 2,3,4,6,9,11,16	94.3734853	15	0.043676	48.60237046	26	0.04514624	90.75609	20	0.041508707
	2	8, 9,12,14,15,18,20,22	99.3172172	5	0.049025	39.39803672	17	0.0419972	91.26205	20	0.040660068
	3	1,3,4,5,8,9,13,18	98.0505125	8	0.047215	50.00679563	20	0.04858114	92.64885	20	0.043212648
	4	5,6,8,10,15,19,20,21	96.1718294	5	0.044531	73.04767102	29	0.04578239	93.29582	20	0.042422599
Unreduced	N/A	all	93.2974522	25	0.066078	43.58985132	23	0.06344522	93.79807	20	0.068288133

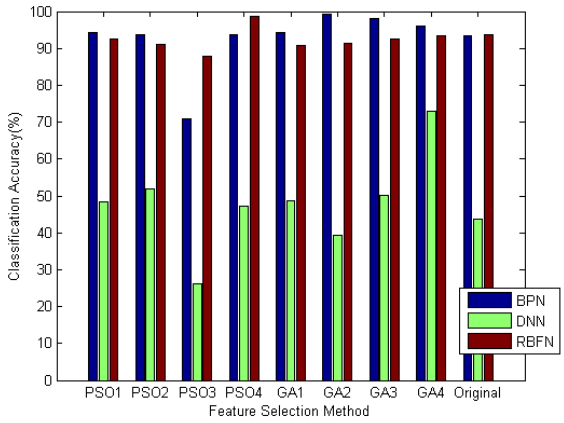

Fig 14. Accuracy of neural networks

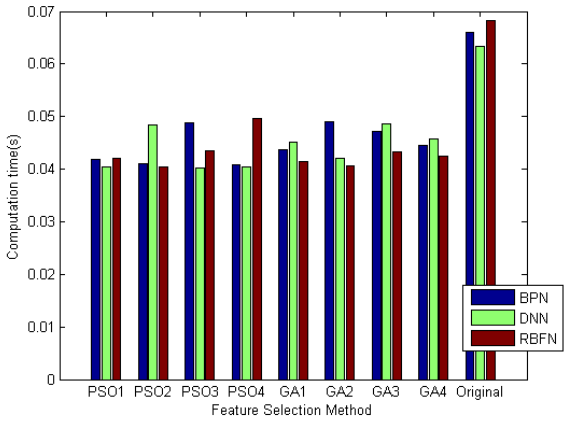

Fig 15. Computation time of neural networks

The computation times for the neural networks are approximately the same , except for the unreduced feature set , which is seen to have a greater computation time than the reduced feature sets.

VII. CONCULSIONS

From the above results, it is seen that each method of feature selection and classification comes with its own advantages and disadvantages. Particularly, there is a trade-off between accuracy and computation time. Particularly, it is seen that the feature selection method with the least computation time, (PSO3) contains the highest error rate, with a substantially higher error rate than any of the other feature selection methods. Conversely, all the feature sets selected by genetic algorithm seem to have a higher accuracy rate than that of PSO, but the computation time taken by GA for feature selection is much higher than any of the PSO algorithms. Considering that with an appropriate fitness function the accuracy of the PSO feature set classification can be improved, PSO is seen to have an advantage over GA in terms of quality of features selected.Overall , PSO2, and PSO4 can be considered optimal for feature selection , as they have a low computation time for feature selection , and also for classification , as well as a high classification accuracy , higher than or comparable to the original feature set.

REFERENCES

[1] HocineBendjama, Salah bouhouche, Mohamed SeghirBoucherit, “Application of wavelet transform for fault diagnosis in rotating machinery”, International Journal of machine learning and Computing, Vol.2, No.1 , Feb 2012

[2] SaravanaBharathi K, Shajeev M, Nair Pravin R, Prasath Kumar P, Santhosh kumar C, “Application of Multi-Wavelet Denoising and Support Vector Classifier in Induction Motor Fault Conditioning”, International Journal of Computer Applications, Vol.8, Article 1, 2011

[3] Milind Natu, “Bearing Fault Analysis Using Frequency Analysis and Wavelet Analysis”, International Journal of Innovation, Management and Technology, Vol. 4, No. 1, February 2013

[4] NeelamMehala, RatnaDahiya, “Condition monitoring methods, failure identification and analysis for Induction machines”, International journal of circuits, systems and signal processing, Issue 1, Vol. 3, 2009

[5] KhalafSalloumGaeid and Hew WooiPing, “Wavelet Fault Diagnosis of Induction Motor” , MATAB for Engineers- Applications in Control , Electrical Engineering, IT and Robotics, Oct 2011

[6] Hui lui, Lihui Fu, Haiqi Zheng, “Bearing Fault diagnosis based on amplitude and phase map of Hermitian wavelet transform” , Journal of Mechanical Science and Technology, Vol. 25, Issue 11, P 2731-2740, Nov 2011

[7] N.S. Swansson and S.C Favaloro, “Applications of vibration analysis to the condition monitoring of rolling element bearings”, Defence science and technology organisation, Aeronautical research laboratories, Melbourne, Aero propulsion report 163, January 1984

[8] Nathan Halko, Per-Gunnar Martinsson, YoelShkolnisky, Mark Tygert , “An algorithm for the principal component analysis of large data sets”, SIAM Journal on Scientific computing, Vol 33, P 2580-2594 2011

[9] D S Broomnhead and D Lowe, “Radial basis functions, multi-variable functional interpolation and adaptive networks”, Royal Signals and Radar Establishment

[10] H. Hannah Inbarani, Ahmad Taher Azar, G. Jothi, “Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis”, Computational Methods and Programs in Biomedicine, Vol. 113, Issue 1, P 175-185, Jan 2014

[11] Aria E. H., Amini J., and Saradjian M.R.., 2003, “Back propagation neural network for classification of IRS -1D satellite images,” Joint Workshop of High Resolution Mapping from Space, Tehran University , Vol.1, pp.100 -104.

[12] ChulminYun, ByonghwaOh, JihoonYang and JonghoNang, “Feature subset selection based on bio-inspired algorithms”, Journal of information science and engineering, Vol 27, Pg 1667-1686, 2011

[13] YuanningLiu, Gang Wang, HuilingChen, HaoDong, XiaodongZhu, SujingWang, “An improved particle swarm optimization for feature selection”, Journal of bionic engineering, Vol 8, Issue 2, Pages 191-200, June 2011

[14] Ngoc-TuNguyen, Hong-HeeLe and Jeong-min Kwon, “Optimal feature selection using genetic algorithm for mechanical fault detection of induction motor”, Journal of mechanical science and technology, Vol 22, Page 490-496, 2008

AUTHOR PROFILE

S.Devendiran received his B.E. degree in Mechanical engineering from Anna University Chennai, India in 2006 and M.E. Degree in Engineering Design from Anna University, Chennai, India in 2009.He is currently Assistant Professorin the School of Mechanical and Building Sciences, VIT University, Vellore, India.His research interests are condition monitoring and fault diagnosis, Soft Computing Techniques, Finite element analysis and Non-destructive testing. Email: devendiran@vit.ac.in

K. Manivannan received his B.E. degree in Automobile Engineering from Bharathiar University,Coimbatore, M.S.degree from BITS,Pilani and Ph.D degree in Industrial Management from Pondicherry University, India in the year 2011.He is currently Professor in School of Mechanical and Building Sciences ,VIT University,Vellore, India.His research interests are Six sigma in service sector, Industrial Design and Simulation, Auto components design and simulation; condition monitoring and fault diagnosis Email: manivannan.k@vit.ac.in

Soham Chetan Kamani is B.Tech student of the School of Electronics and Communication Engineering, VIT University, Vellore. His research interests are intelligent fault diagnosis and prognostics and signal processing. Email: sohamchetan.kamani2011@vit.ac.in

Razim Refai is B.Tech student of the School of Mechanical and Building Sciences ,VIT University, Vellore.His research interests are intelligent fault diagnosis and prognostics, composite material design and heat transfer in combustion. Email: razim.refai2011@vit.ac.in