Artificial neural network approach for fault detection in rotary system

S. Rajakarunakaran a,*, P. Venkumar a, D. Devaraj a, K. Surya Prakasa Rao b

a A.K College of Engineering, Anand Nagar, Krishnankoil 626190, Tamilnadu, India
b Department of Industrial Engineering, Anna University, Chennai 600025, Tamilnadu, India

Received 21 June 2006; received in revised form 6 June 2007; accepted 10 June 2007
Available online 14 June 2007

Abstract

The detection and diagnosis of faults in technical systems are of great practical significance and paramount importance for the safe operation of the plant. An early detection of faults may help to avoid product deterioration, performance degradation, major damage to the machinery itself and damage to human health or even loss of lives. The centrifugal pumping rotary system is considered for this research. This paper presents the development of artificial neural network-based model for the fault detection of centrifugal pumping system. The fault detection model is developed by using two different artificial neural network approaches, namely feed forward network with back propagation algorithm and binary adaptive resonance network (ART1). The training and testing data required are developed for the neural network model that were generated at different operating conditions, including fault condition of the system by real-time simulation through experimental model. The performance of the developed back propagation and ART1 model were tested for a total of seven categories of faults in the centrifugal pumping system. The results are compared and the conclusions are presented.

# 2007 Elsevier B.V. All rights reserved.

Keywords: Fault detection; Neural networks; Back propagation; Adaptive resonance theory; Rotary system

1. Introduction

The problem of detecting faults in complex real plants is strategically important for its various implications, e.g., avoiding major plant breakdowns and catastrophes, safety problems, fast and appropriate response to emergency situations and plant maintenance. For instance, the following systems represent only a small part of systems where fault detection is in general a very difficult, yet important task: chemical plants, refineries, power plants, airplanes, ships, submarines, space vehicles and space stations, automobiles and household appliances. Generally, in process industries, there is a crucial need for checking and monitoring the equipment condition precisely since they are mostly subject to hazardous environments, such as severe shocks, vibration, heat, friction, dust, etc. So fault detection, fault identification and diagnosis of equipments, machineries and systems have become a vigorous area of work. Due to the broad scope of the process fault diagnosis problem and the difficulties in its real-time solution, many analytical-based techniques [1,2] have been proposed during the past several years for the fault detection of technical plants. The important aspect of these approaches is the development of a model that describes the ‘cause and effect’ relationships between the system variables using state estimation or parameter estimation techniques. The problem with these mathematical model-based techniques is that under real conditions, no accurate models of the system of interest can be obtained. In that case, the better strategy is of using knowledge-based techniques where the knowledge is derived in terms of facts and rules from the description of system structure and behaviour. Classical expert systems were used for this purpose. The major weakness of this approach is that binary logical decisions with Boolean operators do not reflect the gradual nature of many real world problems. Recently, with the development of artificial intelligence, Computational Intelligence (CI) methods, Neural Networks (NN), Fuzzy Logic (FL), Evolutionary Algorithms (EA), etc., more and more fault diagnostic approaches have emerged as new techniques for fault diagnostic systems [3,4].

Any complex system is liable to faults or failures. A ‘fault’ is an unexpected change of the system functionality. It manifests as a deviation of at least one characteristic property or variable of a technical process. It may not, however, represent the failure of physical components. Such malfunctions may occur either in
the sensors (instruments) or actuators, or in the components of the process itself. In all but the most trivial cases, the existence of a fault may lead to situations related to safety, health, environmental, financial or legal implications. Although good design practice tries to minimize the occurrence of faults and failures, recognition that such events do occur, enables system designers to develop strategies by which the effect they exert is minimized. A system that includes the capability of detecting and diagnosing faults is called the ‘fault diagnosis system’ [5]. Such a system has to perform two tasks, namely fault detection and fault isolation. The purpose of the former is to recognize that a fault has occurred in the system. The latter has the purpose of locating the fault. The following are the set of desirable characteristics one would like the diagnostics system to possess: (a) Quick detection and diagnosis, (b) isolability, (c) robustness, (d) novelty identifiability, (e) classification error estimate, (f) adaptability, (g) explanation facility, (h) modeling requirements, (i) storage and computational requirements and (j) multiple fault identifiability.

Artificial neural network-based methods for fault diagnosis have received considerable attention over the last few years. The advantage of the neural network approach is their generalization capability, which lets them deal with partial or noisy inputs. The neural networks are able to handle continuous input data and the learning must be supervised in order to solve the fault detection and diagnosis problem. The multilayer perceptron network is the most common network today. Due to their powerful non-linear function approximation and adaptive learning capabilities, neural networks have drawn great attention in the field of fault diagnosis. But the neural network approach needs lot of data to develop the network before being put to use for real-time applications.

Adaptive resonance theory (ART) [6] refers to a class of self-organizing neural architectures that cluster the pattern space and produce appropriate weight vector templates. Conventional artificial neural networks have failed to solve the stability–plasticity dilemma. A network remains open to new learning (remain plastic) without washing away previously learned codes. Too often, learning a new pattern erase or modifies previous training. If there is only a fixed set of training vectors, the network can be cycled through these repeatedly and may eventually learn all. In a real network, it will be exposed to a constantly changing environment; it may never see the same training vector twice. Under such conditions, back propagation will learn nothing. It will continuously modify its weights to no avail, never arriving at satisfactory settings. ART networks and algorithms maintain the plasticity required to learn new patterns, while preventing the modification of patterns that have been learned previously. The objective of the present work is as follows:

- To detect the fault based on the observed data.
- To alert the operating personnel about deviations of normal behaviour of the processes that can turn into failures.
- To monitor and predict the condition of the equipment/system and avoidance of failures and damage.

The paper is organized as follows: in the next section, the description of the experimental system for this study is outlined. Section 3 describes the review of artificial neural network. Sections 4 and 5 demonstrate the development of neural network model and detailed discussions on simulation results and finally, in Section 6, conclusions are drawn from the work.

2. System description

Centrifugal pumps are used in a variety of industrial applications, such as for lifting thin liquids as well as highly dense liquids, such as muddy and sewage water, paper pulp, sugar molasses, chemicals, etc. Centrifugal pumps are classified as rotodynamic type of pumps in which a dynamic pressure is developed which enables the lifting of liquids from a lower to a higher level. The basic principle on which a centrifugal pump works is that when a certain mass of liquid is made to rotate by an external force, it is thrown away from the central axis of rotation and a centrifugal head is imparted which enables it to rise to a higher level. In addition to the centrifugal action, as the liquid passes through the revolving wheel or impeller, its angular momentum changes, this also results in increasing the pressure of the liquid. Finally the diffuser casing increases the pressure at the expense of kinetic energy of the liquid.

In the experimental arrangement of the centrifugal pumping system, the pump is connected to the motor through shaft. The voltmeter (0–600 V) and ammeter (0–15 A) are serially connected with the autotransformer (2 Ω, 0–270 V) and in turn with energy meter. Autotransformer is used to vary the voltage and current supplied to the motor. Energy meter is used to measure the input energy applied to the pump. Vacuum gauge and pressure gauge are fitted in the suction and delivery side. By using this, suction pressure and delivery pressure can be measured. This delivery pressure is the measure of head developed in the system. Collecting tank with flow meter is provided for measuring volume discharge delivered in the system. The schematic layout of the centrifugal pumping system setup is shown in Fig. 1.

The first step in the operation of a centrifugal pumping system is priming. Priming is the operation in which the suction pipe, casing of the pump and the portion of the delivery pipe up to the delivery valve are completely filled with the liquid which is to be pumped, so that all the air (or gas or vapour) from this portion of the pump is driven out and no air pocket is left. It has been observed that even the presence of a small air pocket in any of the portion of pump may result in no delivery of liquid from the pump. After the pump is primed, the delivery valve is still kept closed and the electric motor is started to rotate the impeller. The delivery valve is kept closed in order to reduce the starting torque for the motor. The rotation of the impeller in the casing full of liquid produces a forced vortex, which imparts a centrifugal head to the liquid and thus results in an increase of pressure throughout the liquid mass. The increase of pressure at any point is proportional to the square of the product of angular velocity and the distance of the point from the axis of rotation.
When the delivery valve is opened, the liquid is made to flow in an outward radial direction thereby leaving the vanes of the impeller at the outer circumference with high velocity and pressure.

A pump is usually designed for one particular speed, flow rate and head, but in actual practice, the operation may be at some other condition of head or flow rate, and for the changed conditions, the behaviour of the pump may be quite different. In order to predict the behaviour and performance of a pump under varying conditions, real-time simulation tests are performed in the experimental setup.

In the experimental testing, the following parameters are measured under different operating conditions of the system (including faulty situations), such as voltage reading (V), ammeter reading (A), vacuum gauge reading \( h_1 \), pressure gauge reading \( h_2 \), speed \( N \), time \( t \) for \( h \) metre rise in the collecting tank and the time \( T \) for number of revolutions in energy meter \( N_e \). The total head is calculated by adding the suction head and delivery head. The discharge, input to the pump and output of the pump are calculated by using the formula.

\[
\text{Discharge} \quad Q = \frac{axh}{t}
\]

Input to the pump \( p_i = \frac{3600}{N_e} \frac{N_e}{T} (1000) \text{ watts} \)

Output of the pump \( p_o = \omega \lambda \)

where \( N_e \) is the energy meter constant in rev/kW/h, “a” is the area of the collecting tank, \( \lambda \) is the specific weight of water and \( \omega \) is the weight of water. The 20 numbers of probable faults under 7 categories are simulated in the system in real-time in the experimental setup. Table 1 shows the fault categories with its corresponding fault class. Table 2 shows the details of fault descriptions with its corresponding code. The operating characteristics under normal condition of a centrifugal pump system are shown in Fig. 2.

### 3. Review of artificial neural network

Neural networks have recently attracted much attention based on their ability to learn complex, non-linear functions. Artificial neural networks [7,8] can be viewed as parallel and distributed processing systems that consist of a huge number of simple and massively connected processors. These networks can be trained offline for complicated mapping, such as of determining the various faults and can then be used in an efficient way in the online environment.

#### 3.1. Back propagation

Neural networks [9] have a variety of architectures, but the most widely used is the feed forward network trained by back propagation. Back propagation [10] is a systematic method for training multilayer artificial neural network. Back propagation algorithm [11] has been applied to many pattern recognition problems. The neural network architecture in this class shares a common feature that all neurons in a layer are connected to all neurons in adjacent layers through unidirectional branches.

![Fig. 1. Schematic layout of the centrifugal pumping system.](image-url)
That is, the branches and links can only broadcast information in one direction, that is, the “forward direction”. The branches have associated weights that can be adjusted according to a defined learning rule. Fig. 3 shows a typical multilayer feed forward network.

Training the network with back propagation algorithm results in a non-linear mapping between the input and output variables. Thus, given the input/output pairs, the network can have its weights adjusted by the back propagation algorithm to capture the non-linear relationship. After training, the networks with fixed weights can provide the output for the given input.

The standard back propagation algorithm for training the network is based on the minimization of an energy function representing the instantaneous error. In other words, it is desired to minimize a function defined as:

$$E(m) = \frac{1}{2} \sum_{q} [d_q - y_q]^2$$

where $d_q$ represents the desired network output for the $q$th input pattern and $y_q$ is the actual output of the neural network. Each weight is changed according to the rule:

$$\Delta w_{ij} = -k \frac{dE}{dw_{ij}}$$

where $k$ is a constant of proportionality, $E$ is the error function and $w_{ij}$ represents the weights of the connection between neuron $j$ and neuron $i$. The weight adjustment process is repeated until the difference between the node output and actual output is within some acceptable tolerance.

3.2. Adaptive resonance theory (ART1)

The ART network [8] is a vector classifier. It accepts an input vector and classifies it into one of the categories depending upon which of the stored patterns it most resembles. If the input vector does not match any stored pattern, a new category is created by storing pattern that is like the input vector. Once a stored pattern is found that matches the input vector within a specified tolerance, that pattern is adjusted to make it still more like the input vector. No stored pattern is modified, if it does not match the current input pattern within the specified tolerance. By this, the problem of stability–plasticity dilemma is solved.

In ART, the changes in the activation of units and weights are governed by differential equations. The net is continuously changing dynamical system, but the complexity is reduced because the activations are assumed to change much more rapidly than the weights. Once an acceptable cluster unit is selected for learning, the weights may be maintained over an extended period. During that period, only weight changes should be done. This period is called ‘resonance period’. In ART, ART1 is designed for clustering binary vectors and ART2 can accept continuous valued vectors. These nets cluster the input by unsupervised learning. The training algorithm of ART1 network [12] is as follows:

Step 1: Initialize parameters: $L > 1$ and $0 < \rho \leq 1$. Initialize weights: $0 < b_{ij}(0) < \frac{L-1}{L} + n$, $t_{ij}(0) = 1$.

Step 2: While stopping condition is false, perform steps 3–14.

Step 3: For each training input, do steps 4–13.

Table 2

<table>
<thead>
<tr>
<th>Fault code</th>
<th>Description of faults</th>
<th>Fault code</th>
<th>Description of faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Shaft wear</td>
<td>F11</td>
<td>Discharge valve leakage</td>
</tr>
<tr>
<td>F2</td>
<td>Bearing worn-out condition</td>
<td>F12</td>
<td>Oil seal leakage</td>
</tr>
<tr>
<td>F3</td>
<td>Bearing outside diameter reduced</td>
<td>F13</td>
<td>Speed too high</td>
</tr>
<tr>
<td>F4</td>
<td>Shaft wear and bearing outside diameter reduced</td>
<td>F14</td>
<td>Speed too low</td>
</tr>
<tr>
<td>F5</td>
<td>Coupling washer failure</td>
<td>F15</td>
<td>Obstructions in line</td>
</tr>
<tr>
<td>F6</td>
<td>Shaft wear and bearing backlash</td>
<td>F16</td>
<td>Net positive suction head</td>
</tr>
<tr>
<td>F7</td>
<td>Bearing bracket oval size</td>
<td>F17</td>
<td>Vibration</td>
</tr>
<tr>
<td>F8</td>
<td>Flange leakage in discharge</td>
<td>F18</td>
<td>Wrong rotation of impeller</td>
</tr>
<tr>
<td>F9</td>
<td>Sensor leakage in discharge</td>
<td>F19</td>
<td>Wrong impeller</td>
</tr>
<tr>
<td>F10</td>
<td>Entrained air</td>
<td>F20</td>
<td>Defective impeller vanes</td>
</tr>
</tbody>
</table>

That is, the branches and links can only broadcast information in one direction, that is, the “forward direction”. The branches have associated weights that can be adjusted according to a defined learning rule. Fig. 3 shows a typical multilayer feed forward network.
Step 4: Set activations of all $F_2$ units to zero. Set activations of $F_1(a)$ units to input vector $s$.

Step 5: Compute the norm of $s$: $\|s\| = \sum_i s_i$.

Step 6: Send input signal from $F_1(a)$ to $F_1(b)$ layer $x_i = s_i$.

Step 7: For each $F_2$ node that is not inhibited, if $y_j \neq -1$, then, $y_j = \sum_j b_{ij} x_i$.

Step 8: While reset is true, perform steps 9–12.

Step 9: Fine $J$ such that $y_J \geq y_j$ for all nodes $j$. If $y_J = -1$, then all then odds are inhibited and this pattern cannot be clustered.

Step 10: Recompute activation $x$ of $F_1(b)$. $x_i = s_i t_{ji}$.

Step 11: Compute the norm of vector $x$: $\|x\| = \sum_i x_i$.

Step 12: Test for reset. If $|(\|x\|)/|s|) < \rho$, then $y_j = -1$ (inhibit node $J$) and continue executing step 8 again. If $(|(\|x\|)/|s|) \geq \rho$, then proceed to step 13.

Step 13: Update the weights for node $J$. $b_{ij(new)} = \frac{y_j}{L_{ji} - \sum_k i L_{jk}}$; $t_{ji(new)} = x_i$.

Step 14: Test for stopping condition.

The stopping condition may be no weight changes, no units reset or maximum number of epochs searched. In winner selection, if there is a tie, take $J$ to be the smallest such index. Also $t_{ji}$ is either 0 or 1, and once it is set to 0 during learning, it can never be set back to 1 because of stable learning method.

4. Model development

The proposed methodology for fault detection in centrifugal pumping system is based on using back propagation and adaptive resonance network in artificial neural network (ANN) for detecting the normal and abnormal conditions of the given parameters, which leads to various faults. The neural network approach for this purpose has two phases: training and testing. During the training phase, neural network is trained to capture the underlying relationship between the chosen inputs and outputs. After training, the networks are tested with a test data set, which was not used for training. Once the networks are trained and tested, they are ready for detecting the faults at different operating conditions. The following issues are to be addressed, while developing the model for fault detection in centrifugal pumping system: (a) selection of input and output variables, (b) data generation, (c) data normalization and (d) selection of network structure and network training. Fig. 4 shows the schematic representation of the issues to be addressed while developing an ANN model for fault detection in centrifugal pumping system.

4.1. Selection of input and output variables

For the application machine learning approaches, it is important to properly select the input variables, as ANNs are supposed to learn the relationships between input and output variables on the basis of input–output pairs provided during training. In neural network-based fault detection model, the input variables represent the operating state of the centrifugal pumping system, and the output will be the condition of normal or abnormal, which may in turn cause the faults. Normal and abnormal conditions are taken as the output of the ANN model.

4.2. Data generation

The generation of training data is an important step in the development of ANN models. To achieve a good performance of the neural network, the training data should represent the complete range of operating conditions of the centrifugal pumping system, which contains all possible fault occurrences. The procedure for generating the data at normal condition is given below:

(i) The internal plan dimensions of the collecting tank and the difference in level between the centers of vacuum and pressure gauges are measured.

(ii) The speed of the pump and the energy meter constant are noted.

(iii) The pump was primed with water.

(iv) With the delivery valve fully closed, the driving unit is started.

(v) By regulating the delivery valve, the discharge and the delivery head are varied. For each position of the delivery valve, from completely closed to maximum open, the following observations are made:

- vacuum gauge reading, pressure gauge reading, time taken for $N_r$ revolution of the energy meter and time taken for a particular rise of water level in the collecting tank.

(vi) The above observations for different delivery valve openings are tabulated.

The above steps are repeated for different faults. The ways of simulating the fault in the experimental setup of centrifugal pumping system are as follows: for injecting the fault in the experimental setup, for example, the pump setup has been dismantled and fitted with the respective faulty component.
(fault 1—shaft having reduced diameter due to wear) in its place and then assembled the setup for taking observations. Fault category viz., shaft problem are implemented in the system by replacing the normal components with faulty components, which are deliberately created in the test pieces. Fault category namely bearing problem and impeller problem are implemented in the system by replacing the normal components with faulty components, which are taken from the old similar capacity pumps. Fault category namely mechanical problem, leakage problem, operability problem and vibration problem are directly induced manually in the experimental setup itself.

For easy and convenient handling, only one fault could be injected at a time. After taking observations for different delivery valve openings for the case of fault 1, then go for fault 2. By adapting the same methodology, all the faults can be injected one by one in the experimental setup. The various faults simulated in the experimental setup for generating the data are given in Table 2. For the given input, the pumping system is run with different operating conditions, which include various faults and the combination, which gives the maximum possibility of fault occurrence, s found out. The binary value of normal and abnormal is taken as the output. The same procedure is repeated for different combinations of input features.

4.3. Data normalization

If the generated data are directly fed to the network as training patterns, higher valued input variables may tend to suppress the influence of smaller ones. Also, if the raw data is directly applied to the network, there is a risk of the simulated neurons reaching the saturated conditions. If the neurons get saturated, then the changes in the input value will produce a very small change or no change in the output value. This affects the network training to a great extent. So the data are normalized before being presented to the neural network such that ANN will give equal priority to all the inputs. Data normalization compresses the range of training data between 0 and 1 depending on the type of transfer function. The input and output data are normalized using the expression,

\[ x_n = \frac{(x - x_{min}) \times \text{range}}{x_{max} - x_{min}} + \text{starting value} \quad (3) \]

where \( x_n \) is the normalized value of the data and \( x_{min} \) and \( x_{max} \) are the minimum and maximum values among all the values of the data.

4.4. Selection of network structure

To make a neural network to perform some specific task, one must choose number of input neurons, output neurons and hidden neurons. For the best network performance, an optimal number of hidden units must be properly determined using the trial and error procedure. The hidden layer neurons have tangent hyperbolic function as the activation function and the output have linear activation function. Once the appropriate structures of the network are selected, the ANN model is trained to capture the underlying relationship between the input and output using the training data. The network structures selected for this problem are multilayer feed forward network and adaptive resonance network. In this work, back propagation algorithm is used to train the network, which propagates the error from the output layer to the hidden layer to update the weight matrix. After training, the networks are tested with the test data set to assess the generalization capability of the developed network.

In the adaptive resonance network, binary input vector is presented to \( F_1(a) \) layer and is then received by \( F_1(b) \). The \( F_1(b) \) layer sends the activation signal to \( F_2 \) layer over weighted interconnection path. Each \( F_2 \) unit will calculate the net input. The unit with the largest net input will be the winner that will have activation \( d = 1 \). All the other units will have activation as zero. That winning unit alone will learn the current input pattern. The signal sent from \( F_2 \) to \( F_1(b) \) through weighted interconnections is called as top-down weights. The ‘X’ units remain ‘on’ only if they receive non-zero weights from both the \( F_1(a) \) and \( F_2 \) units.

The norm of the vector \(||x|||\) will give the number of components in which top-down weight vector for the winning unit \((t_i)\) and the input vector \( S \) are both ‘1’. Depending upon the ratio of norm of \( x \) to norm of \( S(||x||/||S||) \), the weights of the winning cluster unit are adjusted. The whole process may be repeated until either a match is found or all neurons are inhibited. The ratio \((||x||/||S||)\) is called match ratio. At the end of each presentation of a pattern, all cluster units are returned to inactive states but are available for further participation. The architecture of ART1 is shown in Fig. 5.

5. Simulation results

This section presents the details of the development and testing of ANN models for fault detection on centrifugal pumping system. Two different ANN models were developed for fault detection, one with back propagation algorithm and the other with adaptive resonance theory (ART1). The neural network model is developed using MATLAB 6.5 Neural Network Toolbox in Pentium 4 with 2.40 GHz processor with 512 MB of RAM. Using real-time simulation on laboratory
5.1. Case I: ANN model with back propagation algorithm

Initially all the 11 input features are given as input to the neural network. The ANN model used here has two hidden layers of tansigmoidal neurons, which receives the inputs, then broadcast their outputs to an output layer of linear neurons, which compute the corresponding values. The generated training data are normalized and applied to neural network with corresponding output, to learn the input–output relationship. The neural network model was trained using the back propagation algorithm, which propagates the error from the output layer to the hidden layer to update the weight matrix, is most commonly used for feed forward neural networks. At the end of the training process, the model obtained consists of the optimal weight and the bias vector. After training the network with least error rate, the testing data was fed as input to the network. Trail and error procedure was followed to identify the optimal number of hidden nodes. There are about 11 neurons in the input layer that corresponds to all the 11 input features and 20 neurons in the output layer in which all neurons set to 0 corresponds to normal and 1 in each neuron corresponds to any one of the 20 faults. The number of hidden units is directly related to the capabilities of the network.

The training function used was scaled conjugate gradient back propagation and gradient descent with momentum weight/bias learning function. The transfer function used in the input was tansigmoidal and in the output was linear with learning rate (0.01) and threshold (0.5). The mode of training used here is batch type. Table 5 shows the network performance for each fault category. Table 6 shows the

### Table 3

Name of the Input features in the experimental pumping system

<table>
<thead>
<tr>
<th>Label</th>
<th>Feature name</th>
<th>Label</th>
<th>Feature name</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Voltage</td>
<td>F7</td>
<td>Actual discharge</td>
</tr>
<tr>
<td>F2</td>
<td>Current</td>
<td>F8</td>
<td>Input power to motor</td>
</tr>
<tr>
<td>F3</td>
<td>Suction pressure</td>
<td>F9</td>
<td>Output power from pump</td>
</tr>
<tr>
<td>F4</td>
<td>Discharge pressure</td>
<td>F10</td>
<td>Time for number of revolutions in energy meter</td>
</tr>
<tr>
<td>F5</td>
<td>Total head</td>
<td>F11</td>
<td>Time for h meter rise in collecting tank</td>
</tr>
<tr>
<td>F6</td>
<td>Speed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 4

Training simulation steps

<table>
<thead>
<tr>
<th>ART1</th>
<th>Back propagation network (BPN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:- Initialize the parameters.</td>
<td>1:- Load the data in a file.</td>
</tr>
<tr>
<td>2:- When stopping conditions is false, perform steps 3–10.</td>
<td>2:- Separate the input and output data.</td>
</tr>
<tr>
<td>3:- For each input vector, do steps 4–9.</td>
<td>3:- Separate the training and test data.</td>
</tr>
<tr>
<td>4:- F1 layer process starts.</td>
<td>4:- Normalize all the input and output values.</td>
</tr>
<tr>
<td>5:- While reset condition is true, do steps 6–8.</td>
<td>5:- Define the network structure.</td>
</tr>
<tr>
<td>6:- Find unit to learn current input pattern, i.e., F5 unit (not inhibited) with largest input.</td>
<td>6:- Initialize the weight matrix and biases.</td>
</tr>
<tr>
<td>7:- F((b) units combine their inputs from F((a) and F2)</td>
<td>7:- Specify the number of epochs.</td>
</tr>
<tr>
<td>8:- Test for reset condition. If reset is true, then current candidate unit is rejected (inhibited).</td>
<td>6:- Train the network with the train data.</td>
</tr>
<tr>
<td>Return to step 5. If reset is false, then the current candidate unit is accepted for learning, then do step 9.</td>
<td></td>
</tr>
<tr>
<td>9:- Learning starts. Weight updation occurs as per the differential equations.</td>
<td>7:- Test the network with the test data.</td>
</tr>
<tr>
<td>10:- Test for stopping condition.</td>
<td>8:- Re-normalize the results.</td>
</tr>
</tbody>
</table>

### Table 5

Network performance for each fault category

<table>
<thead>
<tr>
<th>S. no.</th>
<th>Fault class</th>
<th>Fault categories</th>
<th>Fault code</th>
<th>Training pattern (%)</th>
<th>Testing pattern (%)</th>
<th>Efficiency (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FC-A</td>
<td>Shaft problem</td>
<td>F1, F4, F5</td>
<td>80</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>FC-B</td>
<td>Bearing problem</td>
<td>F2, F5, F6, F7</td>
<td>80</td>
<td>20</td>
<td>97</td>
</tr>
<tr>
<td>3</td>
<td>FC-C</td>
<td>Leakage problem</td>
<td>F4, F6, F11, F12</td>
<td>80</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>FC-D</td>
<td>Impeller problem</td>
<td>F16, F19, F20</td>
<td>80</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>FC-E</td>
<td>Mechanical problem</td>
<td>F16, F16</td>
<td>80</td>
<td>20</td>
<td>99.3</td>
</tr>
<tr>
<td>6</td>
<td>FC-F</td>
<td>Operability problem</td>
<td>F10, F13, F14</td>
<td>80</td>
<td>20</td>
<td>99.5</td>
</tr>
<tr>
<td>7</td>
<td>FC-G</td>
<td>Vibration problem</td>
<td>F17</td>
<td>80</td>
<td>20</td>
<td>99</td>
</tr>
</tbody>
</table>

Considering all fault class (total classification accuracy) | 80 | 20 | 99.3
After training, the generalization performance of the network is evaluated with the 150 test data that contain the combination of both normal as well as all types of fault categories. The trained neural network classified 149 data correctly, which shows an overall detection rate of 99.3%. The network is trained with least mean square algorithm until it reaches the mean square error of 0.01. The mean square error achieved during training is 0.0121. During testing, the mean square error achieved by the network is 0.121. With 14 hidden nodes, the network took 81.8130 s to reach the error goal.

5.2. Case II: ANN model with ART1 algorithm

In this case, the network is trained using adaptive resonance theory (ART1). The adaptive resonance theory classification process consists of three major phases: recognition, comparison and the search phase.

Phase 1: Initially no input vector is applied; hence all the components of input vector $X$ are zero. This sets $F_2$ to zero, thereby disabling all recognition layer neurons and causing their outputs to be zero. Because all the recognition layer neurons start out in the same state, all have an equal chance to win the subsequent competition. The pattern to be classified is now applied for each neuron in the recognition layer. A dot product is formed between its associated weight vector and the input vector. The neuron with largest dot product has weights that best match the input vector. It wins the competition and fires, inhibiting all other outputs from this layer, i.e., the ART1 network stores a set of patterns in the weights associated with the recognition layer neurons one for each classification category.

Phase 2: In comparison phase, the single neuron firing in the recognition layer passes a ‘1’ back to the comparison layer on its output signal. In accordance with the two-third rule, the only comparison layer neurons that will fire are those that receive simultaneous ‘1’ from the input vector and the comparison layer excitation vector. A comparison of the input vector to the inner layer vector is done and if degree of similarity is less than the vigilance parameter, it causes reset. The effect of the reset is to force the output of the firing neuron in the recognition layer to zero, disabling it for the duration of the current classification.

Phase 3: In search phase, if no reset signal is generated, the match is adequate and the classification is finished. Otherwise, the other stored pattern must be sent to seek a correct match. The process is repeated for neurons, until a stored pattern is found that matches the specified tolerance or all the stored patterns have been tried and all are in mismatch with the input vector.

The description of data used in this network is as follows: the number of instances is 600, number of attributes is 11 plus the class attributes. Each instance has one of two possible classes as normal or fault. The performance of network based on different train–test ratios with different vigilance parameters is shown in Table 7. The maximum number of epochs is 300.

Table 8 shows comparative performance of various parameters of the network model. From the simulation results, it is found that the trained neural network is able to produce the correct output even for the new input with minimum time.

6. Conclusion

This paper has presented a neural network-based approach for fault diagnosis in centrifugal pumping system. The data required for the development of neural network model have been obtained through the real-time operational data of the system considered. Totally 7 categories of faults including 20 faults from the centrifugal pumping system were considered in the development of the neural network model.
the developed model. For the ANN model the testing data are fed to the designed model to check the accuracy. The testing samples are different from the training samples and they are new to the trained network. The developed neural network model is simulated for pattern classification of fault data using the ART1 learning and testing algorithm, and about 100% accuracy is obtained. Simulation of ART network has achieved the objective of classification of faults in a given dataset of centrifugal pumping system according to a high level of accuracy based on training–testing ratio. Thus for unsupervised input pattern, the output is obtained with a good level of accuracy. Simulation results show that this neural network approach is very much effective in detecting the various faults in the system. The effectiveness of the proposed neural network approach has been demonstrated through different fault detection analysis in the centrifugal pumping system. The proposed system shows promising results in fault detection for centrifugal pumping system. The same models can be extended to any technical systems by considering appropriate parameters and the faults. Industrial applications of the proposed system will pave path for wide implementation because of its simplicity and efficiency. To further improve the performance of the model is to identify a representative set of features from which to develop the network for a particular task by using suitable dimensionality reduction techniques.

References