Chapter 6

Fault Diagnosis by ANN Approach

The concept of artificial neural networks, commonly referred to as neural networks, was developed based on the idea that the brain computes in an entirely different manner from a conventional digital computer. When the Artificial Neural Network carries out a task, it tries to emulate how the brain performs tasks. A neural network is a massively parallel distributed processor made up of simple processing units that can store experimental knowledge and make it available for use. It can be implemented using electronic components or simulated in software on a digital computer. The network resembles the human brain in the following two ways [59 – 60,67]:

- The ANN obtains the information from the environment through a learning process.
- Using Interneuron connection strengths usually called the Synaptic weights which are used to store knowledge.

The function of learning algorithm is to modify the synaptic weights of the network in an orderly fashion to attain a desired design objective.

6.1 Artificial neuron model

A neuron is a fundamental information-processing unit to the operation of a neural network. Figure 6.1 shows the model of a neuron, which forms the basis for designing (artificial) neural networks. The three basic elements of the neuronal model are briefed as follows:

- A set of synapses or connecting links, each of which is characterized by a weight or strength of its own. Specifically, a signal $X_j$ at the input of synapse $j$ connected to neuron $k$ is multiplied by the synaptic weight $W_{kj}$. The first subscript refers to the neuron being considered and the second subscript refers to the input end of the synapse to which the weight refers.
- An adder for summing the input signals, weighted by the respective synapses of the neuron.
- An activation function for limiting the amplitude of the output of a neuron. The activation function is also referred to as a squashing function in that it squashes (limits) the permissible amplitude range of the output signal to some finite value. Typically, the
The neuronal model of Figure 6.1 also includes an externally applied bias, denoted by $b_k$. The bias $b_k$ has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative, respectively [59 -60,67].

In mathematical terms, a neuron $k$ may be described by writing the following pair of equations:

$$V_k = \sum_{j=1}^{m} W_{kj} X_j \quad (6.1)$$

and

$$U_k = V_k + b_k \quad (6.2)$$

Therefore,

$$Y_k = f(U_k) \quad (6.3)$$

Where $X_1 , X_2 , \ldots X_m$ are the input signals, $W_{k1}, W_{k2}, \ldots W_{km}$ are the synaptic weights of neuron $k$, $U_k$ is the linear combiner output due to the input signals, $b_k$ is the bias, $f(\cdot)$ is the activation function and $Y_k$ is the output signal of the neuron.

6.2 Artificial Neural Network architectures

Generally, there are two main types of neural networks: Feed Forward Architecture and Recurrent Architecture. Feed Forward Architecture is further classified in three types: Single – Layered Artificial Neural Network, Multi – Layered Artificial Neural Network and Radial Basis Function Network (RBFN). A Multi – Layered Artificial Neural Network is the most popular neural network type.
The inner structure of the processing element (neuron) in each network is interconnected differently, and the configuration set-up is often referred to as Network Topology. The behavior of the network relies greatly on the network topology. Figure 6.2 shows the graphical representation of different Artificial Neural Network Topologies.

![Artificial Neural Network Diagram](image)

**Figure 6.2 Classification of artificial neural network architecture**

6.2.1 Single layered feed forward Artificial Neural Network

The Single Layer Feed-Forward Network Consists of a Single Layer of weights, where the inputs are directly connected to the outputs via a series of weights. The synaptic links carrying weights connect every input to every output, but not other way. The sum of the products of the weights and the inputs is calculated in each neuron node.

![Single Layer Feed Forward Network Diagram](image)

**Figure 6.3 Single layered feed forward artificial neural network.**
6.2.2 Multi layered feed forward Artificial Neural Network

The architecture of this class of network, besides having the input and the output layers, also have one or more intermediate layers called Hidden Layers. The computational units of the hidden layer are known as Hidden Neurons.

The function of hidden neuron is to intervene between the external input and the network output in some useful manner. By adding more hidden layers, the network is enabled to extract higher order statistics.

The hidden layer does intermediate computation before directing the input to output layer. The input layer neurons are linked to the hidden layer neurons; the weights on these links are referred to as Input Hidden Layer Weights. The hidden layer neurons and the corresponding weights are referred to as Output Hidden Layer Weights. A Multi – Layer Feed – Forward network with \( l \) Input Neurons, \( m_1 \) Neurons in the First Hidden Layers, \( m_2 \) Neurons in the Second Hidden Layers, and \( n \) Output Neurons in the Output Layers is written as \((l - m_1 - m_2 - n)\). Figure 6.4 shows a Multi – Layered Feed – Forward Network with a configuration \((l - m - n)\).

6.2.3 Back propagation algorithm

A learning algorithm refers to the way in which the learning rules are used to adjust the weights. Error back-propagation learning consists of two passes through the different layers of the network: a forward pass and a backward pass. In the forward pass, an input vector is applied
to the nodes of the network, and its effect propagates through the network layer by layer. Finally, a set of outputs is produced as the actual response of the network. During the forward pass the weights of the networks are all fixed. During the backward pass, the weights are all adjusted in accordance with an error correction rule. The actual response of the network is subtracted from a desired response to produce an error signal. This error signal is then propagated backward through the network, against the direction of synaptic connections. The weights are adjusted to make the actual response of the network move closer to the desired response.

6.3 Simulation results obtained using ANN

For using Artificial Neural Network (ANN), the rms & average values of different parameters are used and motor health diagnosis is carried out. For this purpose, last 500 values for each of the parameters, from the recorded database are used. The neural network is trained in different ways to discriminate healthy/faulty condition as well as classify the different fault conditions. The variation in training is achieved by the change in number of input parameters and also the change in number of neurons in hidden layer.

The Table 6.1 represents the number of samples used for training & testing of each of the different conditions.

<table>
<thead>
<tr>
<th>Motor Condition</th>
<th>Type of Fault</th>
<th>Number of Data captured</th>
<th>Number of data used for training</th>
<th>Number of data used for testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Healthy</td>
<td>---</td>
<td>1500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty</td>
<td>Stator (3 Turns Shorted)</td>
<td>1500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty</td>
<td>Stator (5 Turns Shorted)</td>
<td>1500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty</td>
<td>Stator (7 Turns Shorted)</td>
<td>1500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty</td>
<td>Stator (10 Turns Shorted)</td>
<td>1500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty</td>
<td>Rotor (2 Bars Broken)</td>
<td>1500</td>
<td>8100</td>
<td>13500-8100=5400</td>
</tr>
<tr>
<td>Faulty</td>
<td>Rotor (3 Bars Broken)</td>
<td>1500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty</td>
<td>Rotor (4 Bars Broken)</td>
<td>1500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty</td>
<td>Rotor (5 Bars Broken)</td>
<td>1500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Data Generated</td>
<td>13500</td>
<td>8100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
As depicted in the above table, out of 13,500 samples, 60% i.e. 8,100 samples are used for training while rest 40% i.e. 5,400 samples are used for testing. MATLAB function is used for random selection of training & testing samples. Mean squared error is selected for the performance evaluation of proposed technique.

The feed forward neural network is used for training with variations in number of input parameters and number of neurons in hidden layer. The following sub-sections describe the different conditions of said variations.

### 6.3.1 Training & testing of ANN with 16 input parameters by simulation data

First the analysis is carried out by taking 16 parameters for training with different number of neurons in hidden layer. The parameters selected for training are three voltages, three currents, total active, reactive, apparent power, three voltage THDs, three current THDs and power factor.

The training & testing is carried out for motor health discrimination as well as fault classification. For discrimination, the number of neurons in the hidden layer is selected as 7, 8, 9, 10, 12, 13, 15 & 20. The best performance is observed for the number of neurons equal to 7 and 9, giving the 100% accuracy while for the fault classification the network with 13 hidden neurons gives 99.63% accuracy.

Figure 6.5 and Figure 6.6 show the performance curve and the target & actual output of the ANN for fault discrimination. The number of parameters is 16 and the number of neurons in hidden layer is 7. During the training, it is set that the healthy condition indicates 0 whereas the faulty (stator fault with 3, 5, 7 & 10 number of turns shorted and rotor fault with 2, 3, 4 & 5 numbers of broken bars) condition indicates 1; hence the target & actual output plot shows the two steps of 0 and 1 to indicate the discrimination by ANN between healthy and faulty conditions of motor.
As mentioned above, the ANN has been trained & tested for different number of neurons in hidden layer, giving different levels of accuracy. Figure 6.7 shows the accuracy percentage achieved with different numbers of hidden neurons, with 16 input parameters.
Figure 6.6 Fault discrimination accuracy with 16 parameters obtained by simulation

Similar to the above training & testing, now the ANN is trained & tested for fault classification. Here, during the training, it is set that healthy condition is indicated by 0 whereas the digits 1 & 2 indicate the stator and rotor fault conditions respectively.

Figure 6.7 Performance for fault classification with 16 parameters obtained by simulation
In the figure above, it can be seen that the red line indicates the target output while the blue line indicates the actual output. Those points which are seen between 0 & 1 or between 1 & 2 are off target outputs.

Figure 6.10 shows the variation in accuracy level with different number of neurons in hidden layer, for the fault classification.
6.3.2 Training & testing of ANN with 12 input parameters by simulation data

The above exercise is repeated now by reducing the number of input parameters to 12. The parameters chosen in this case are three voltages, three currents, three voltage THDs and three current THDs. Again the training & testing for fault discrimination and classification is carried out with different numbers of neurons in hidden layer. In this case network with 6 hidden neurons gives best performance at 100% accuracy for fault discrimination while for fault classification, 8 hidden neurons give best performance with 97.46% accuracy.

Figure 6.10 and Figure 6.12 show the performance curve and plot of target & actual output for the health discrimination, with number of input parameters taken as 12.

![Figure 6.10](image1.png)

Figure 6.10 Performance for fault discrimination with 12 parameters obtained by simulation

![Figure 6.11](image2.png)

Figure 6.11 Target & actual output for fault discrimination with 12 parameters obtained by simulation
Figure 6.13 shows the plot of accuracy with different numbers of hidden neurons, for health discrimination, with number of input parameters taken as 12.

![Accuracy vs. No. of Hidden Neurons](image)

**Figure 6.12 Fault discrimination accuracy with 12 parameters obtained by simulation**

Similarly the Figure 6.14 and Figure 6.15 show the performance curve and plot of target & actual output for fault classification with 12 input parameters.

![Plot of target & actual output](image)

**Figure 6.13 Performance for fault classification with 12 parameters obtained by simulation**
Figure 6.14 Target & actual output for fault classification with 12 parameters obtained by simulation

The accuracy variation with change in number of hidden neurons for fault classification with 12 input parameters is shown in Figure 6.16.

Figure 6.15 Fault classification accuracy with 12 parameters obtained by simulation
6.3.3 Training & testing of ANN with 6 input parameters by simulation data

The training and testing of ANN is again carried out by further reducing the number of input parameters; this time taking only the three voltage THDs and three current THDs. Again the results are observed for different numbers of neurons in hidden layer. This time the network with 3 hidden neurons gives highest accuracy of 100% in fault discrimination and 89.2% in fault classification.

The performance curve and the plot of target & actual output for health discrimination with 6 input parameters are shown in Figure 6.17 and Figure 6.18, while the accuracy variation with change in number of hidden neurons is shown in Figure 6.19.
Figure 6.18 Fault discrimination accuracy with 6 parameters obtained by simulation

Again, the Figure 6.20 and Figure 6.21 show the performance curve and plot of target & actual output for fault classification, with 6 input parameters, while the Figure 6.22 shows the variation of accuracy with change in number of hidden neurons for the same situation.

Figure 6.19 Performance for fault classification with 6 parameters obtained by simulation
Figure 6.20 Target & actual output for fault classification with 6 parameters obtained by simulation

Figure 6.21 Fault classification accuracy with 6 parameters obtained by simulation
6.4 Experimental results obtained using ANN

As discussed in simulation section, ANN technique is used to discriminate the healthy-faulty condition of motor as well as to classify the different fault conditions based on the data acquired. Here the data collected from experiments are utilized for the same purposes. It has already been described that a lot of parameters have been acquired with the help of power logger and vibration analyzer during the experiments. Selective parameters from among the collected data base are used to train the ANN for the motor health diagnosis.

Similar to analysis done with simulation, total 16 parameters namely, three voltages, three currents, total active, reactive, apparent power, three voltage THDs, three current THDs and power factor, are collected for training & testing purpose of ANN. Table 6.2 shows the different cases generated from the experiments with variation in faulty and healthy condition of motor having different load conditions.

<table>
<thead>
<tr>
<th>Motor Condition</th>
<th>Type of Fault</th>
<th>Total Samples collected</th>
<th>No. of samples used for training</th>
<th>No. of samples used for testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Healthy</td>
<td>---</td>
<td>606</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty</td>
<td>Stator (3 Turns Shorted)</td>
<td>596</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty</td>
<td>Stator (5 Turns Shorted)</td>
<td>558</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty</td>
<td>Stator (10 Turns Shorted)</td>
<td>336</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty</td>
<td>Rotor (3 Bars Broken)</td>
<td>738</td>
<td>2475</td>
<td>4125 – 2475 =1650</td>
</tr>
<tr>
<td>Faulty</td>
<td>Rotor (5 Bars Broken)</td>
<td>712</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faulty</td>
<td>Bearing</td>
<td>579</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total data generated</td>
<td></td>
<td>4125</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From the total sample collected, 60% data i.e. 4125 samples are used for training while the remaining 40% i.e. 1650 samples are used for testing purpose.
6.4.1 Training & testing of ANN with 16 input parameters by experiment data

As in case of simulation, here also the ANN is trained with different number of input parameters as well as varying number of hidden neurons. First, the network is trained with all the 16 input parameters mentioned in main section. The number of hidden neurons is varied as 7, 8, 9, 10, 12, 13 & 15. Fault discrimination accuracy in all the cases is found to be nearly 100%. For the function of fault classification, the network with 8 hidden neurons gives best accuracy of 99.82%.

In the case of experiments, the ANN is trained for identification of bearing fault also apart from the healthy, stator fault and rotor fault conditions. Hence, in the plot of target & actual output, 0 indicates the healthy condition while 1 indicates the faulty condition which includes stator fault, rotor fault and bearing fault. Similarly in the case of fault classification the digits 0, 1, 2 & 3 indicate the healthy, stator fault, rotor fault and bearing fault respectively.

Figure 6.23 and Figure 6.24 show the performance curve and plot of target & actual output for the fault discrimination, while the Figure 6.26 and Figure 6.27 show the same for fault classification. Similarly, Figures 6.25 and Figure 6.28 show the variation of accuracy with change in number of hidden neurons for fault discrimination and fault classification respectively.

![Performance for fault discrimination with 16 parameters obtained by experiment](image)

Figure 6.22 Performance for fault discrimination with 16 parameters obtained by experiment
Figure 6.23 Target & actual output for fault discrimination, 16 parameters obtained by experiment

Figure 6.24 Fault discrimination accuracy with 16 parameters obtained by experiment
Figure 6.25 Performance for fault classification with 16 parameters obtained by experiment

Figure 6.26 Target & actual output for fault classification with 16 parameters obtained by experiment
6.4.2 Training & testing of ANN with 12 input parameters by experiment data

Now the input parameters are reduced from 16 to 12 by removing power and power factor components. This time the network is tested with 5, 8 & 14 numbers of hidden neurons. Again the fault discrimination performance of 100% is achieved in all the cases. For fault classification, network with 5 hidden neurons gives best accuracy of 99.82%.

Here the performance curve and plot of target & actual outputs for fault discrimination and classification are shown in Figures 6.29, 6.30, 6.32 and 6.33 respectively. Also, the variation of accuracy with change in number of hidden neurons, for fault discrimination and classification are shown in Figure 6.31 and 6.34 respectively.
Figure 6.28 Performance for fault discrimination with 12 parameters obtained by experiment

Figure 6.29 Target & actual output for fault discrimination, 12 parameters obtained by experiment
Figure 6.30 Fault discrimination accuracy with 12 parameters obtained by experiment

Figure 6.31 Performance for fault classification with 12 parameters obtained by experiment
Finally the exercise is repeated by taking only THDs of voltages and currents i.e. total 6 input parameters. This time the number of hidden layers is varied as 2, 3, 4, 5 & 6. For health
discrimination this procedure gives best performance of 83.7% with 6 hidden neurons whereas for fault classification the best performance of 67% is achieved with 3 hidden neurons.

Finally, the Figures 6.35, 6.36, 6.38 and 6.39 show the performance curve and plot of target & actual output for fault discrimination and classification respectively. Similarly, the Figures 6.37 and Figure 6.40 show the variation of accuracy with number of hidden neurons for fault discrimination and classification respectively.

![Figure 6.34 Performance for fault discrimination with 6 parameters obtained by experiment](image)

![Figure 6.35 Target & actual output for fault discrimination with 6 parameters obtained by experiment](image)
Figure 6.36 Fault discrimination accuracy with 6 parameters obtained by experiment

Figure 6.37 Performance for fault classification with 6 parameters obtained by experiment

Figure 6.38 Target & actual output for fault classification with 6 parameters obtained by experiment
From the above descriptions, it may be concluded that ANN is a very successful technique for the health discrimination as well as fault classification. However, the number of input parameters must be sufficient in numbers to get high accuracies in results. Here it can be seen that while the accuracy is nearly 100% with 16 input parameters, it reduces to about 66% when only 6 input parameters are taken. Further, it is seen that the training process of ANN is somewhat cumbersome.

6.5 Conclusion

This chapter presented one of the intelligent approaches used in the research work i.e. the Artificial Neural Network (ANN). MATLAB function has been used for the training & testing of ANN. The network was trained and tested with different number of input parameters and varying number of neurons in hidden layer. The training & testing has been performed both from simulation and experimental data. Sample plots of performance curve and target & actual outputs have also been presented. The accuracy of ANN is found to be quite good for both the health discrimination and fault classification. It is also noted that as number of input parameters decreases, the accuracy of ANN also decreases.