Chapter 8

Expert System for SFRA Based on ANN

8.1 Introduction

The experts system is based on ANN and Feed forward neural networks and can detect the multiple faults like incipient mechanical and electrical fault of transformer. Feed forward neural networks has been trained with various training algorithms and models of neurons in hidden layers and output layers for the fault classification in Transformer based on SFRA plots in frequency sub bands of Transformer. Design of network is adaptive in nature and it can predict fault in transformer by taking a base SFRA plot of the winding as input data to ANN. Also, Transformer is classified according to their rating, design and make . Once, the ANN is trained for a identical group of transformer for a particular rating, design and make it can predict the healthiness of other transformers of same group without the need of base data as the network design is adaptive in nature [17].

ANN has been trained for faults like open circuit, high impedance, short circuit, Tap changer related faults, overall axial, radial shift and core relatd faults in the transformer. The database used for the training is based on the real data collected form the field both for the healthy transformer and faulty transformer of various type and make worldwide.

In summary, this chapter discusses how the the knowledge base of SFRA interpretation is being utilized for developing expert system based on ANN to classify the fault in transformer. Similar kind of work based on the transformer modeling and simulated SFRA data had been discussed earlier in the reference literature [20].

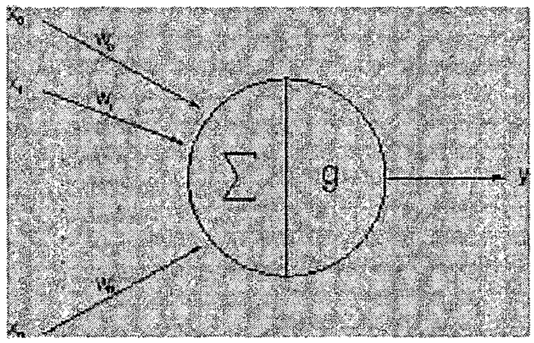

Figure 8.1: Structure of Neuron

8.2 Basic of ANN

8.2.1 The Artificial Neuron

A single artificial neuron can be implemented in many different ways. The general mathematic definition is shown in Figure 8.1 and described in equation below:

$$y(x) = g\left(\sum_{i=0}^{n} w_i x_i\right) \tag{8.1}$$

It represents a neuron with

Inputs :	$x_1, x_2, \ldots, x_n$
Output:	$y(x)$
Weights:	$w_1, w_2, \ldots, w_n$
Activation function:	$g()$

The output from the activation function $g()$ is either between 0 and 1, or between -1 and 1, depending on which activation function is used.

There are many different activation functions, some of the most commonly used are threshold , sigmoid and hyperbolic tangent as mentioned below.

$$g(x) = \frac{1}{1 + e^{2x(x+t)}}$$
$$g(x) = \tanh(s(x+t))$$

Sigmoid and hyperbolic tangent are both smooth differentiable functions, with very similar graphs, the only major difference is that hyperbolic tangent has output that ranges

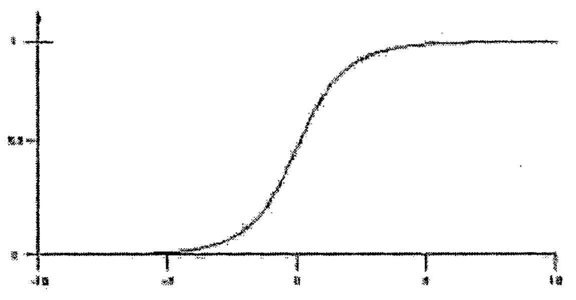

Figure 8.2: Sigmoid function

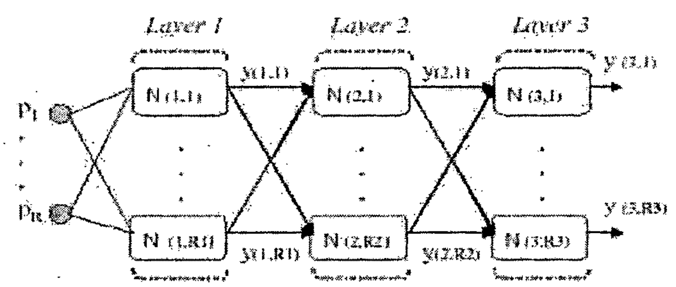

Figure 8.3: Multilayer Feed-forward network

from -1 to 1 and sigmoid has output that ranges from 0 to 1. A graph of a sigmoid function is given in Figure 8.2.

8.2.2 Feed-forward Neural Networks

Feed-forward neural networks (FF networks) are the most popular and most widely used models in many practical applications. Figure 8.3 illustrates a one-hidden-layer FF network with inputs $p_1, p_2, \ldots, p_n$ and output vector Y.

The input layer consists of the inputs to the network. Then follows a hidden layer, which consists of any number neurons, or hidden units placed in parallel. Each neuron performs a weighted summation of the inputs, which then passes through nonlinear function also called the neuron function.

8.2.3 Learning Rules and Training Algorithms

Training Algorithm is used to adjust the weights and biases for the ANN in order to perform the desired task. In training the network, its parameters are adjusted incrementally until the training data satisfy the desired mapping; that is,it matches the desired output y as closely as possible up to a maximum number of iterations.

The Neural Networks contains special initialization algorithms for the network parameters, or weights, that start the training with reasonably good performance. After this initialization, an iterative training algorithm is applied to the network and the parameters is optimized. The special initialization makes the training much faster than a completely random choice for the parameters.

Before the trained network is accepted, it should be validated. Roughly, this means running a number of tests to determine whether the network model meets certain requirements. Probably the simplest way is to test the neural network on a data set that was not used for training, but which was generated under similar conditions. Sometime, Trained neural network fail this validation test, in which case the user will have to choose a better model. Sometimes, however, it might be enough to just repeat the training, starting from different initial parameters . Once the neural network is validated, it is ready to be used on new data.

The general purpose of the Neural Networks package can be described to be function approximation. However, depending on the origin of the data, and the intended use of the obtained neural network model, the function approximation problem may be subdivided into several types of problems.

1. Supervised Learning- For a given training set of pairs $\{p(1), t(1)\}, \ldots, \{p(n), t(n)\}$, where $p(i)$ is an instance of the input vector and $t(i)$ is the corresponding target value for the output y, the learning rule calculates the updated value of the neuron weights and bias.

2. Reinforcement Learning- Similar to supervised learning, instead of being provided with the correct output value for each given input, the algorithm is only provided with a given grade/score as a measure of ANN's performance.

3. Unsupervised Learning- The weight and unbiased are adjusted based on inputs only. Most algorithms of this type learn to cluster input patterns into a finite number of classes.

There are generally four steps in the training process:

1. Assemble the training data.
2. Create the network object.
3. Train the network.
4. Simulate the network response to new inputs.

8.3 Configuration of ANN for application in SFRA

The proposed ANNs are designed with aim to satisfy the following requirement:

1. It must possess a high level of ability to correctly discriminate input patterns with deformation from those without deformation.
2. For new input patterns obtained from recent measurement in the field, it can simulate the test data for different type of fault and pre-fault stage.

The results show that the algorithm is capable of distinguishing between normal and failed state quite satisfactorily and thus successfully establish the efficacy of the proposed method. Therefore, for modern transformer windings with hidden deformation, the ANN based SFRA method would be more effective for detection of such problem. The Architecture of an ANN is decided by following parameter:

1. Number of inputs and outputs of the network;
2. Number of layers;
3. Number of neurons in each layer
4. How the layers are connected to each other;
5. The transfer function of each layer;

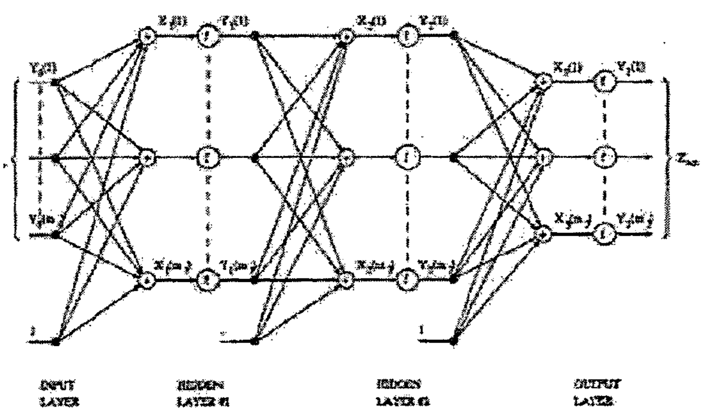

Figure 8.4: Feed-forward ANN structure

8.3.1 Input -Output Data Series

To train a network, we need a set of data containing N nos. of input-output pairs. The input and output data are each arranged in the form of a matrix. Each individual input vector,x_i , is a vector on row$i$ of the input data matrix, and y_i each is a vector on row$i$ of the output data matrix.

8.3.2 Multi Layer Feedforward Network with Back-propagation Training Algorithm

ANNs have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. For this application, Multiple layer Feedforward neural networks is used with hidden layers of sigmoid neurons followed by an output layer of linear neurons. In hidden layers of neurons nonlinear transfer functions sigmoid is used, which allows the network to learn nonlinear and linear relationships between input and output vectors.

In a multilayer feedforward ANN, the neurons are ordered in layers, starting with an input layer and ending with an output layer. Between these two layers are a number of hidden layers. Connections in these kinds of network is shown in figure8.4.

Multilayer feedforward ANNs have two different phases: A training phase and an execution phase. In the training phase the ANN is trained to return a specific output when given a specific input, this is done by continuous training on a set of training data. In the execution phase the ANN returns outputs on the basis of inputs. In this case for training of neurons the back-propagation algorithm is used . It works in much the same way as the name suggests that after propagating an input through the network, the error is calculated and the error is propagated back through the network while the weights are adjusted in order to make the error smaller. It is the most efficient way of to minimize the mean square error for all the training data .

Backpropagation is the generalization of the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by user . Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities.

Standard backpropagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function.

First the input is propagated through the ANN to the output. After this the error on a single output neuron is calculated as:

$$e_k = d_k - y_k \tag{8.2}$$

Where y_k is the calculated output and d_k is the desired output of neuron. This error e_k value is used to calculate a δ_k value, which is again used for adjusting the weights. The value is calculated by:

$$\delta_k = e_k g'(y_k) \tag{8.3}$$

Where is g' the derived activation function. When the δ_k value is calculated, we can calculate the δ_j values for preceding layers by the following equation:

$$\delta_j = \eta g'(y_j) \sum_{k=o}^{K} \delta_k w_{jk} \tag{8.4}$$

Where K is the number of neurons in this layer and η is the learning rate parameter; which determines how much the weight should be adjusted. Using these values, the values that the weights should be adjusted by, can be calculated by:

$$\Delta w_{jk} = \delta_j y_k \tag{8.5}$$

The value Δw_{jk} is used to adjust the weight

$$w_{jk} = w_{jk} + \Delta w_{jk} \tag{8.6}$$

and the back propagation algorithm moves on to the next input and adjusts the weights according to the output. This process goes on until a certain stop criteria is reached. The stop criteria is typically determined by measuring the mean square error of the training data while training with the data, when this mean square error reaches a certain limit, the training is stopped.

Back-propagation is an iterative steepest descent algorithm, in which the performance index is the mean square error E [e2] between the desired response and network's actual response as shown in Figure 8.5.

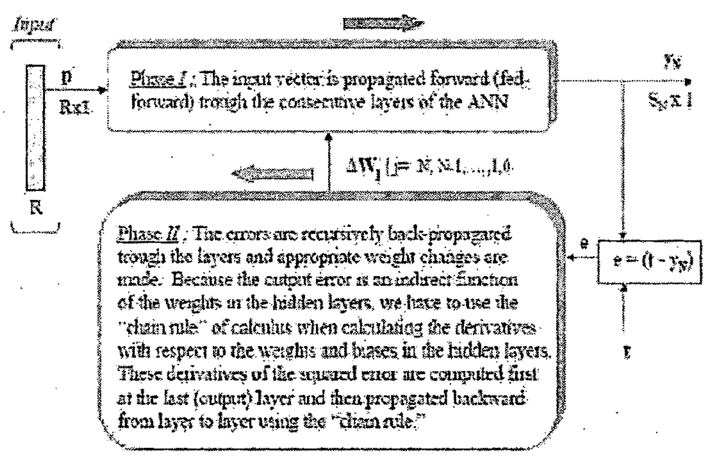

Figure 8.5: Algorithm for Back-propagation ANN

Properly trained backpropagation networks tend to give reasonable output when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for input vectors used in training which are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good output results without training the network on all possible input/output pairs.

Matlab code: The ANN algorithm for SFRA expert system, is designed in Matlab software [[57],[31]].

net=newff(minmax(p),[3,1],'tansig','purelin','traingd');

Training parameters are set as mentioned : net.trainParam.show = 50; net.trainParam.lr = 0.05; net.trainParam.epochs = 300; net.trainParam.goal = 1e-5

Here, the function minmax is used to determine the range of the inputs to be used in creating the network.

8.3.3 Feedforward Neural Networks

Feed forward adaptive linear network is used with an output layer of linear neurons as shown in Figure 8.6.The linear function for output layer allow the network to produce values outside the range -1 to +1 and provide the actual value.

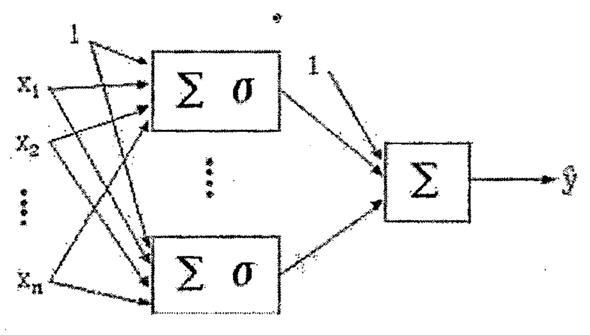

Figure 8.6: Feed forward adaptive linear network

Linear neurons in output layer have a linear transfer function that allows to use a Least Mean-Square (LMS) procedure. LMS algorithm adjusts weights and biases in such away to minimize the mean-square error E [e2] between all sets of the desired response and network's actual response as shown in Figure8.7.

$$E[(t-y)^2] = E[(t-(w_1 \dots w_R b).(p_1 \dots p_R 1)^T)^2] = E[(t-W.p)^2]$$

Matlab code:

The following code creates a training set of inputs p and targets t in Matlab .

net = newlind(p, t);

y = sim(net, p);

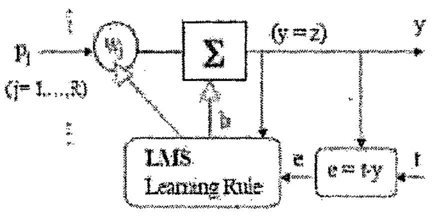

Figure 8.7: Least Mean-Square (LMS) algorithm

For batch training, all of the input vectors are placed in one matrix. $p = [p_1 p_2 p_3 p_4]$ and target vector in matrix, $t = [t_1 t_2 t_3 t_4]$ and network is created with adaptive training algorithm applying LMS learning rule.

8.4 Field data collection and classification of DATA

The Frequency Response Measurement has been carried out on more than 1000 EHV power transformers in order to record reference signatures. Some of the tests confirmed deformation of winding after transformer failures due to short circuit and inter-turn fault. Identical plots have been obtained from transformers of the same design, Therefore measurements on one unit also provide reference for other units of the same design. In most cases, reference or fingerprint results are not necessary, provided comparisons between phases of identical design transformers are made. Nevertheless, fingerprinting results of every design and each unit is desirable since this enables accurate assessment of even minor damage and provides conclusive results.

8.5 Classification of Fault by ANN

The neural network, when adequately designed and trained, can synthesize a useful non-linear mapping between input and output patterns. This is a key property for winding deformation detection. The measured frequency range is divided into three sections: high frequency section, middle frequency section and low frequency section. The correlation coefficient and the standard deviation in used for every one of three frequency sections to compare.

Because of the diversity of structure of windings, the distribution of pole points of transfer function is different. And in order to reflect the state of winding, division of the frequency range must be reasonable.

Experience shows that differences in the lower frequency ranges relate to core changes, or shorted/open circuits. Medium frequencies show winding shifts, while more localized winding movement is seen at the higher frequencies [32].

ANN has been trained for faults like:

1. Shorted turns fault (NNET1).

2. Open circuit or high impedance winding fault (NNET2).

3. Overall radial and axial shift of the winding- Mid frequency variation (NNET3).

4. Core earthing related faults in the transformer , Tap changer leads and bushing leads related faults - High frequency variation (NNET 4).

After the ANNs are trained , for any base data, the output of NNET1, NNET2, NNET3, NNET4 is collected. Then the real SFRA plot is compared with the NNET1, NNET2, NNET3, NNET4 output by using auto-correlation function and the coefficient of auto-correlation function decide the type of fault or multiple faults or no faults in the winding . The database used for the training is based on the real data collected form the field both for the healthy transformer and faulty transformer of various type and make worldwide. Each fault mode and the designed ANN is discussed below.

8.5.1 Shorted turn or Core related Fault (NNET1)

For Frequency response of transformer in low frequency range of 10 Hz. to 1 kHz. , the core effect is dominant, when the SFRA measurement carried out on a winding with all the terminals of other winding left open-circuited due to effect of magnetizing inductance . Besides the magnetizing inductance, it also include the self inductance as well as resistance of winding, determined by the overall winding configuration and these winding parameters effects configuration of SFRA short circuit plot in the low frequency range. Any deformation in the core and winding fault like shorted turn or open circuit fault will be detected by the change in SFRA plot in this region.

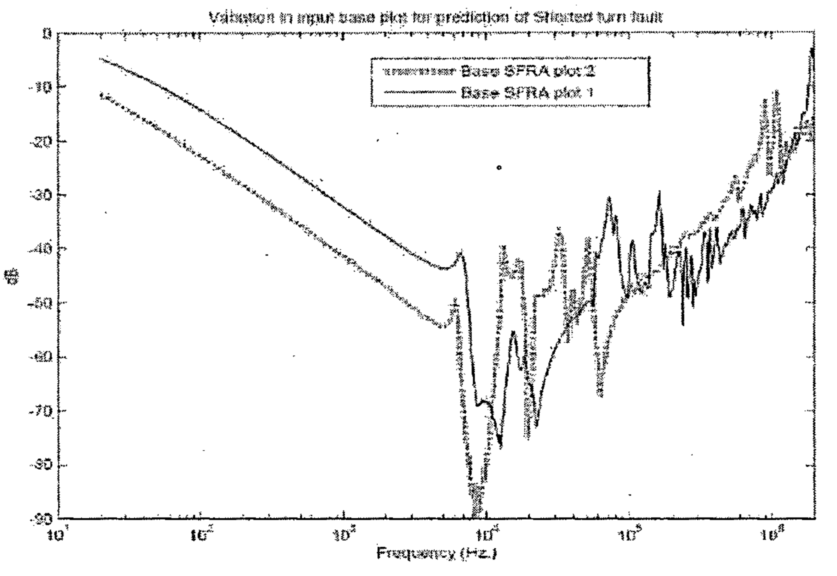

Figure 8.8: Variation in InputBase SFRA plot applied for prediction of shorted turn fault by NNET 1

The ANN learns deformation patterns by itself, using the back-propagation learning method and the details of the algorithm is provided in section 8.3.2 of this chapter. The back-propagation algorithm performs the input to output mapping by making weight connection adjustments following the discrepancy (error) between the computed output value and the desired output value (target signal).

After the inputs and outputs are defined, the next task is to incorporate hidden layers in the network. The selection of hidden layer is a matter of trial and error. In order to obtain the appropriate number of neurons in the hidden layer, networks with different number of hidden neurons are trained.

The ANN is designed in such a way that once it is trained for the shorted turn fault and it is stored as NNET1, it can predict this fault mode for different shape of base plot. Figure 8.8 shows the variation in two base plots which is the short circuit SFRA plot of the different winding and still with the same NNET1 shorted turn fault pattern is predicted for both the base plot 1 and base plot 2 as shown in Figure 8.9 and Figure8.10.

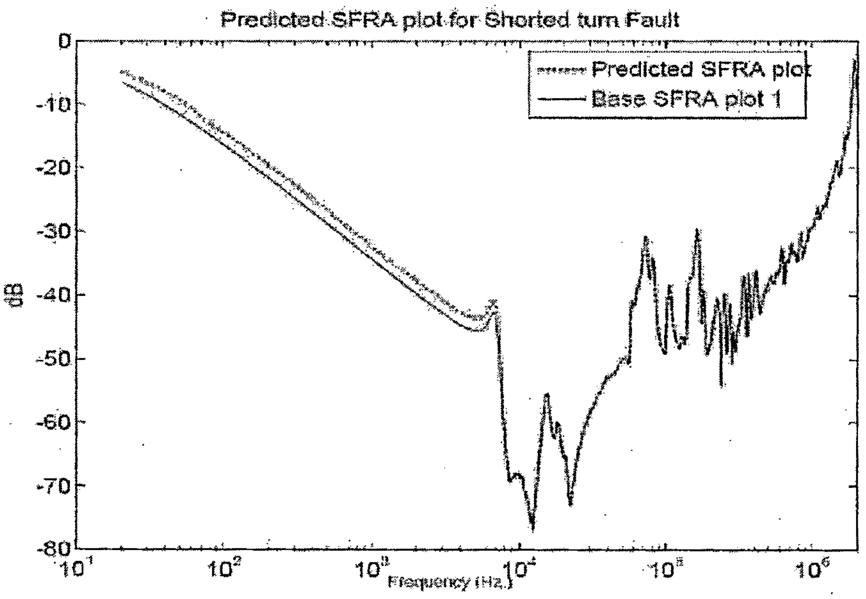

Figure 8.9: Predicted shorted turn plot for Base plot 1

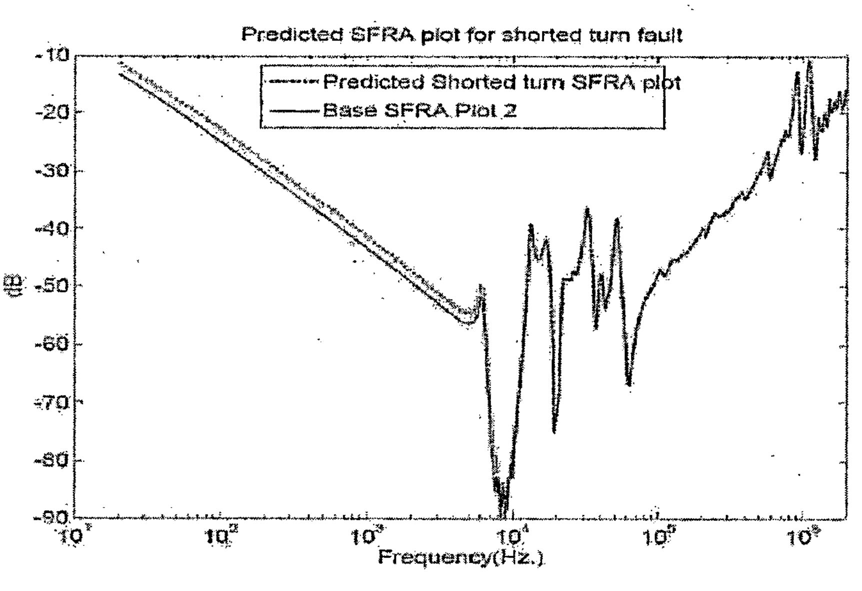

Figure 8.10: Predicted shorted turn plot for Base plot 2

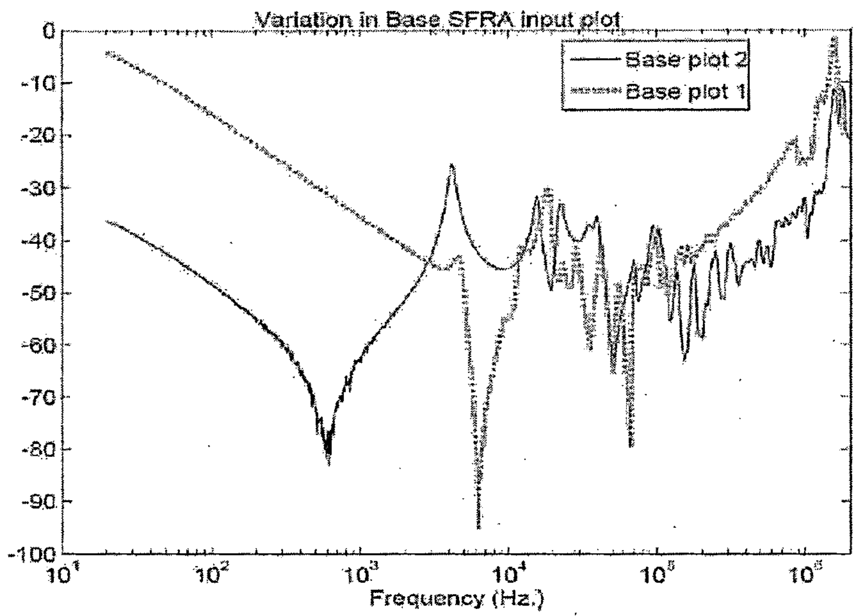

Figure 8.11: Variation in Input Base SFRA plot applied for prediction of high impedance fault by NNET 2

8.5.2 High winding Impedance Fault ($NNET2$)

High impedance fault can be faults like opening or loosening of the leads or turns due to mechanical type of fault in wining or tap changer or leads of the bushings. This may develop into incipient electrical fault like arcing across the two joints which become loose and develop high impedance at that point.

In this case the ANN learns deformation patterns using the back-propagation learning method and the details of the algorithm is provided in section 8.3.2 of this chapter. The ANN is designed in such a way that once it is trained for the high impedance fault and it is stored as $NNET2$, it can predict this fault mode for different shape of base plot. Figure 8.11 shows the variation in two base plots and with the $NNET2$ high impedance fault pattern is predicted for both the base plot 1 and base plot 2 as shown in Figure8.12 and Figure8.13.

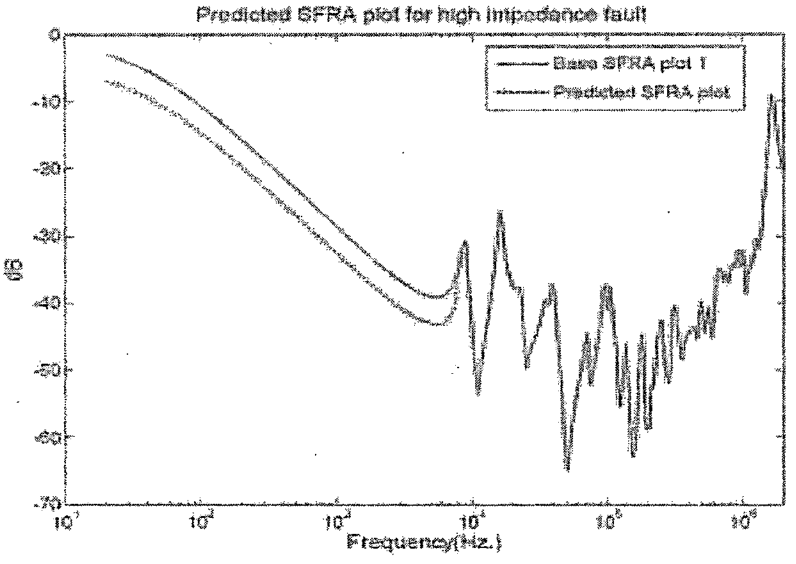

Figure 8.12: Predicted High impedance fault for a base SFRA plot 1 by NNET2

8.5.3 Winding deformation in mid frequency range ($NNET3$)

:

As the frequency increases, the core effect will become less significant as the flux penetration depth in the core is frequency dependent and the core will eventually behaves as an earth plate. However, the winding structure, especially for the winding under test, become dominant factor of the frequency responses and to represent a winding accurately in the medium frequency range. The frequency response in the range from $1kHz$ to $200kHz$ represent the overall winding structure of transformer. In this region hoop-buckling of the LV winding due to high short circuit system fault is normally detected well.

Also the frequency response in the range from $200kHz$ to $1MHz$ represent the main winding structure of transformer. In this region overall axial deformation of the HV, LV winding due to high short circuit system fault is detected well. The winding structure, especially for the winding under test, become dominant factor of the frequency responses and to represent a winding accurately in this frequency range series and shunt capacitance,

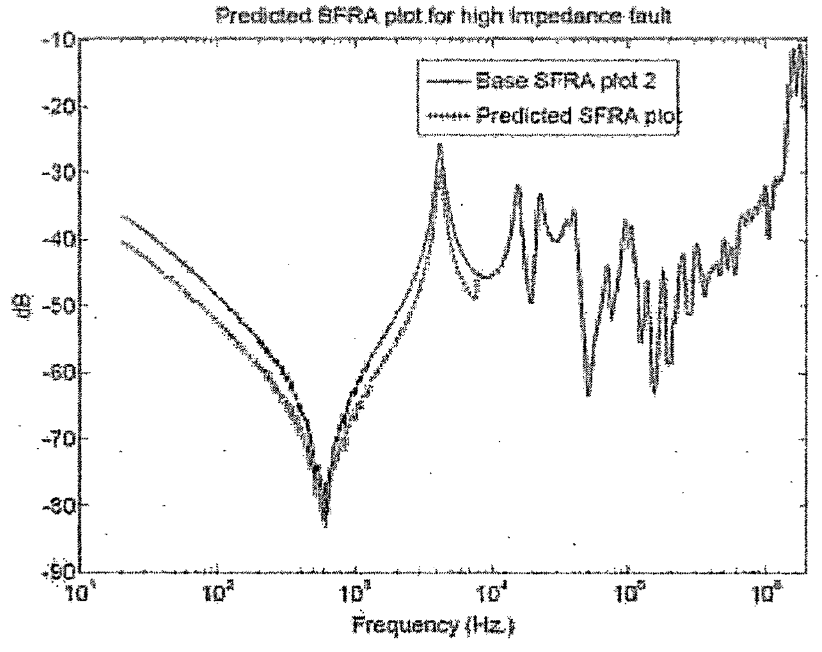

Figure 8.13: Predicted High impedance fault for a base SFRA plot 2 by NNET2

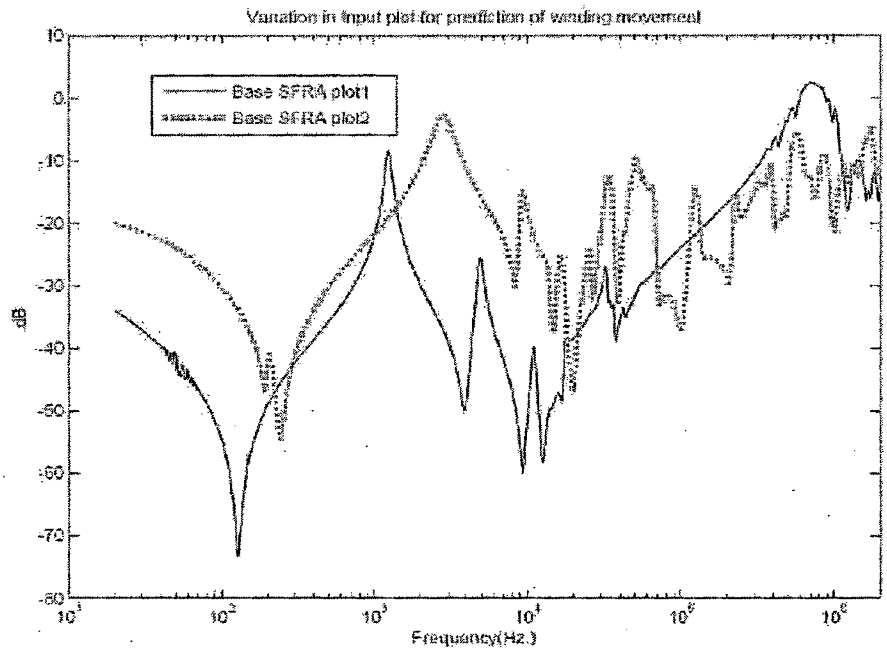

Figure 8.14: Variation in Base SFRA plot applied for prediction of winding movement fault by NNET 3

self-inductance and mutual inductance are the relevant circuit parameters.

ANN learns mid frequency deformation patterns using Feed forward adaptive Neural Networks and the details of the algorithm is provided in section 8.3.3 of this chapter. Figure 8.14 shows the variation in two base plots which is fed as input to $NNET3$ and with the same $NNET3$ mid frequency fault pattern is predicted for both the base plot 1 and base plot 2 as shown in Figure 8.15 and Figure 8.16.

8.5.4 Winding deformation in high frequency range(NNET4)

:

Any change in the configuration or arrangement of bushing or tap changer leads or in the connecting leads of the core to earth (Tank) will change high frequency response between $1MHz$ to $2MHz$. ANN learns high frequency deformation patterns using Feed forward adaptive Neural Networks and the details of the algorithm is provided in section

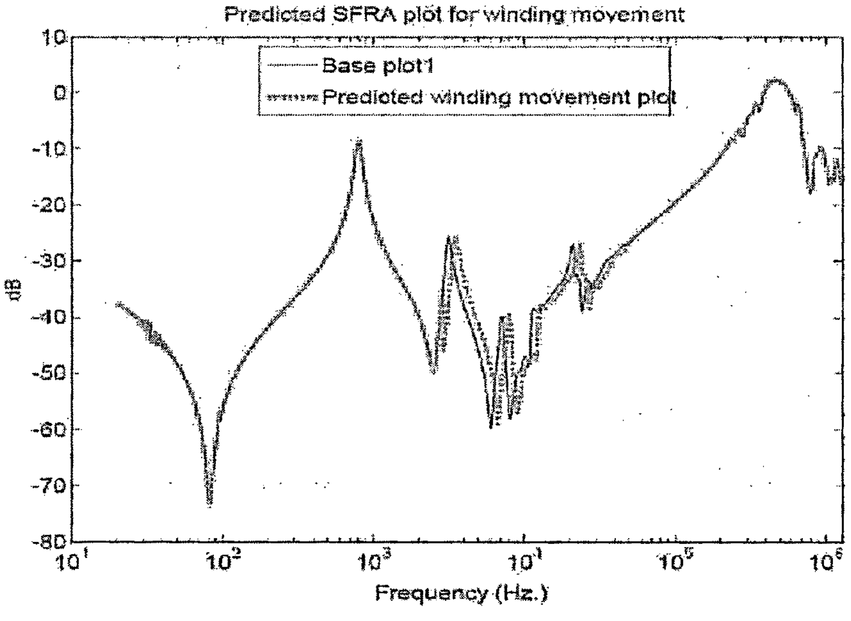

Figure 8.15: Predicted SFRA plot of Winding movement fault for Base SFRA plot 1

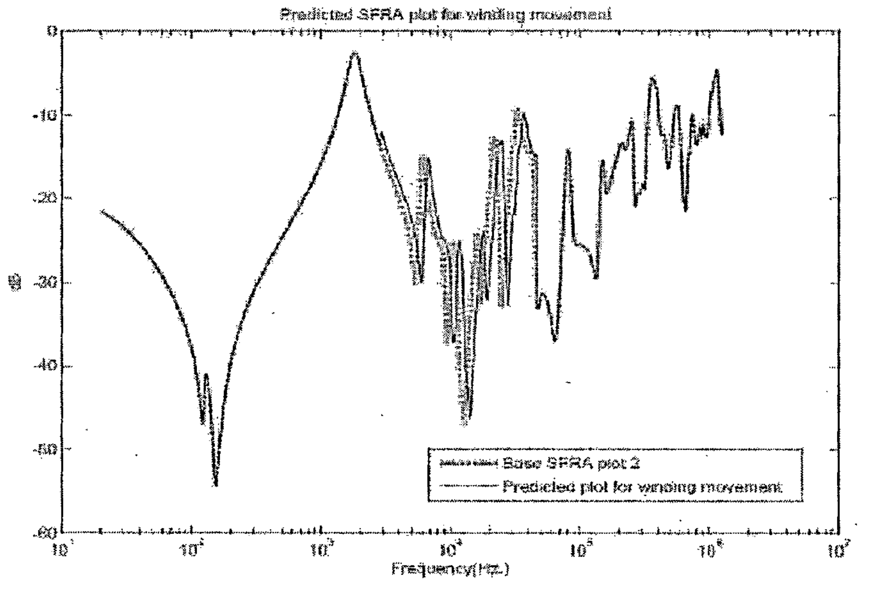

Figure 8.16: Predicted SFRA plot of Winding movement fault for Base SFRA plot 2

8.3.3 of this Chapter. Figure 8.17 shows the variation in two base plots and high frequency fault pattern is predicted for both the base plot 1 and base plot 2 as shown in Figure 8.18 and Figure 8.19.

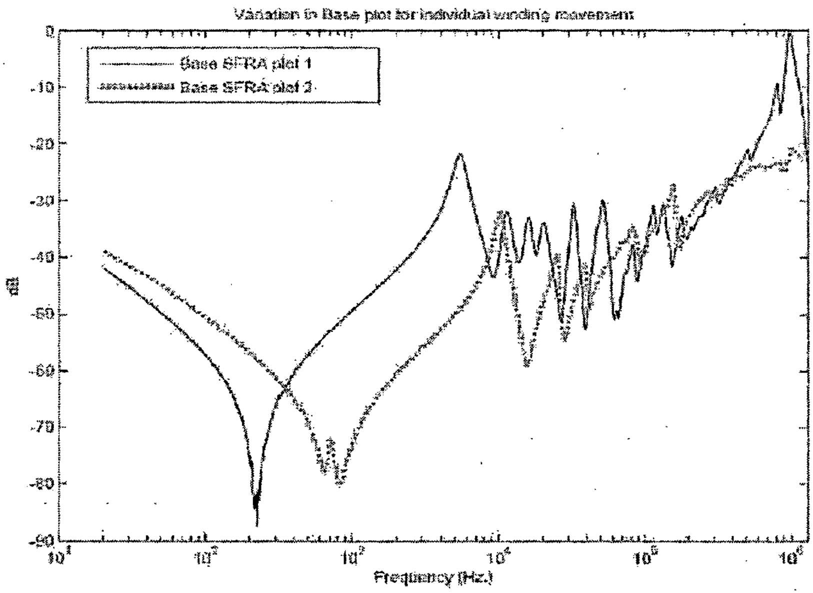

Figure 8.17: Variation in Input Base SFRA plot applied for prediction of connecting leads fault by NNET 4

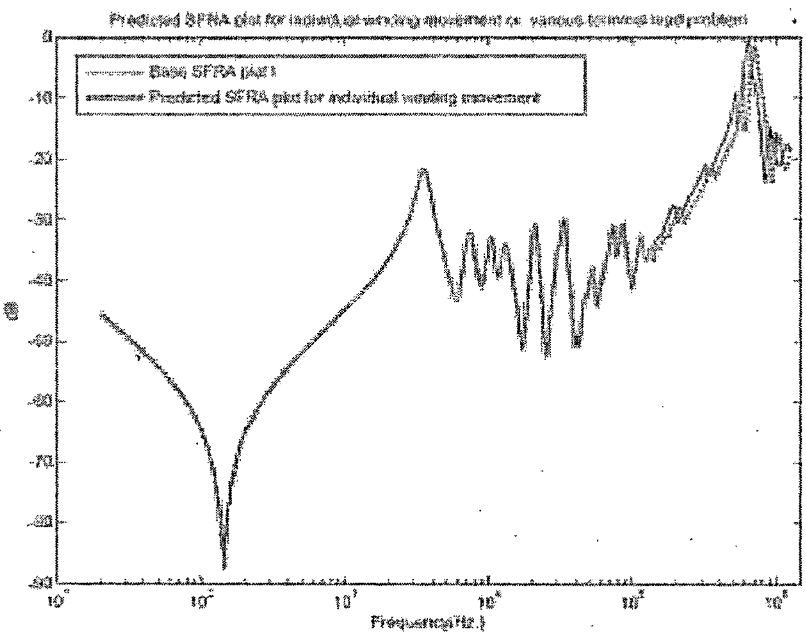

Figure 8.18: Predicted plot of lead fault for Base plot 1

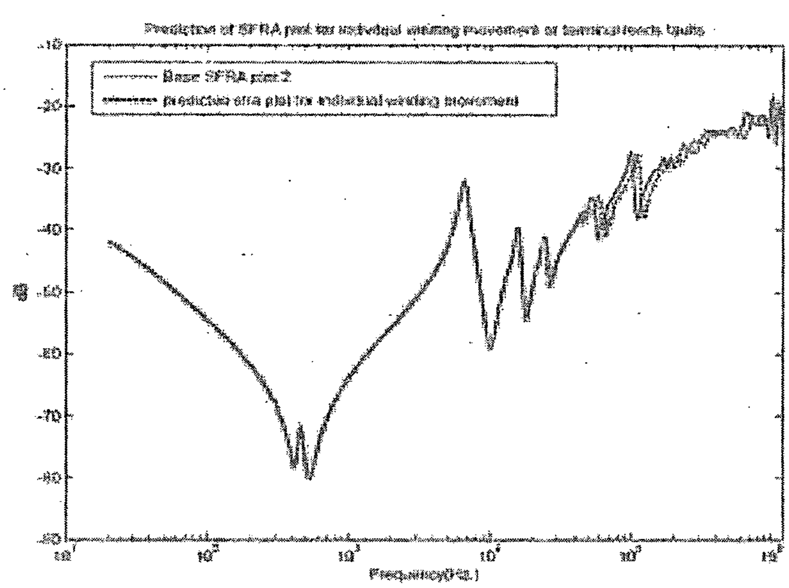

Figure 8.19: Predicted plot of lead fault for Base plot 2

8.6 Conclusion

Hence, based on practical experience of years, in SFRA analysis of the field data, Artificial Neural Network (ANN) has been designed and applied to provide the means to comparatively analyze SFRA plots in the three frequency sub-bands for various type of Transformer fault and classify the fault.

The results show that the algorithm is capable of distinguishing between normal and failed state quite satisfactorily and thus successfully establish the efficacy of the proposed method. Therefore, for modern transformer windings with hidden deformation, the ANN aided SFRA method would be more effective. The method is simple and easy to implement as described for different fault.