ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Use of Artificial Neural Network to Predict the Various Impedances of K-Rated Transformer

Manish Kumar Verma[*1], Ruchika Kaushik[2], Swati Sharma[3] and Prof. Manab Kumar Sengupta[4]
[*1, 3, 4] Electrical Engineering Department, Institute of Engineering & Technology-Alwar, India
[2]Electronics & Communication Department, Institute of Engineering & Technology-Alwar, India
shilparamya18@gmail.com

Abstracts

In this research paper the detailed analysis and prediction of the influence of various harmonics on the supply transformer like power transformer and distribution transformer and changes in the reactance value of the same by using the approach of ANN. The harmonics created due to non-linear load such as solid state devices in the load side of the transformer. These harmonics travelled from load to source transformer and create various iron losses and other related losses which may eventually burn the source transformer. By this analysis a designer can predict by regression method the design parameters of the De-rated transformer which also called K-rated transformer.

Keywords: K-rated Transformer, Non-linear load, Harmonics, ANN, Back Propagation, De-rating of Transformer, Thermal Stress, Short Circuit test of Transformer.

Introduction

Artificial neural networks are evolved from the knowledge about biological neural cells (neurons) in the brain. ANNs can be described either as mathematical and computational models for non-linear function approximation, data classification, clustering and non-parametric regression or as simulations of the behavior biological neurons. They are used to model several aspects of the information combining and pattern recognition behavior of real neurons in a simple but meaningful way. A neural network can be considered as an intelligent box that is able to predict an output pattern when it recognizes a given set of input pattern (Simpson 1990; Jang et al. 1997). It requires initial training by processing a large number of input patterns and thus it gives the output results from each input pattern. A well trained neural network is able to recognize the similarities when presented with a new input pattern resulting from a predicted output pattern. ANN models are usually based on biological neural structures. The neuron has N input lines and a single output. Each input signal is weighted, and it is multiplied with the weight value of the corresponding input line. The neuron will combine these weighted inputs by forming their sum with reference to a threshold value and activation function, and then it will determine its output (Singh et al. 2004a; Kosko1994).

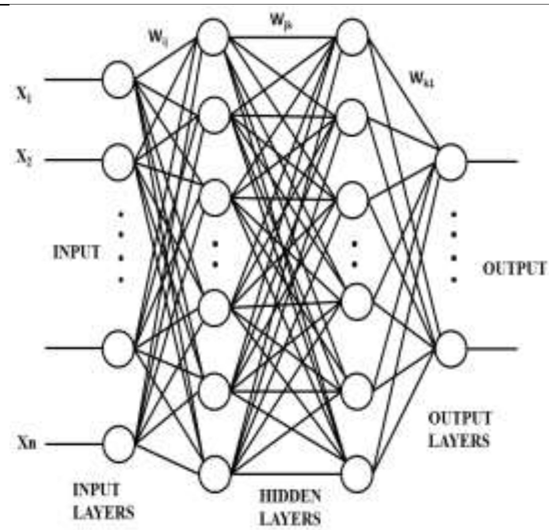

Figure 1: Back propagation neural network.

The neuron may be expressed mathematically as -

$$u = \sum_{i=1}^{N} w_i x_i$$

$$y = f(u - \theta)$$

where x1, x2, x3, ..., xn are the input signals, w1, w2, w3 ,...,wn are the synaptic weights, u is the activation potential of the neuron, h is the threshold, y is the output signal of the neuron, and f(h) is the activation function. The above equation can be rearranged in the form by substituting w0= h and x0=1.

Thus,

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

$$u = \sum_{i=1}^{N} w_i x_{i-\theta} \qquad \sum_{i} = 1^N w_i x_i$$

$$y = f(u = \sum_{i=1}^{N} w_i x_i)$$

The combination of a fixed input X0= -1and of an extra input weight W0=h is generally used for the adjustment of bias input. Note that the new notation has augmented any input vector X€RN to the vector (1, x) €RN+1, and also the weight vector W € RN of the neuron, to the vector (W0, W) €RN+1.

The activation function, denoted by f (h), defines the output of the neuron in terms of the activity level at its input. The most common form of activation function used in the formulation of ANN model is the sigmoid function (Singh et al. 2003). An example of the sigmoid is the logistic function, which is defined by-

$$f(u) = \frac{1}{1+\exp(-u)}$$

Where, u is the slope parameter of the sigmoid function. By varying the slope parameter, sigmoid functions of different slopes can be obtained. As the slope parameter approaches infinity, the sigmoid function becomes simply a threshold function. The threshold function, however, can take only the value 0 or 1, whereas a sigmoid function assumes a continuous range of values from 0 to 1 (Nauck et al. 1997; Singh et al. 2004b). Also, the sigmoid function is differentiable, whereas the threshold function is not. Differentiability is an important feature of neural network simulation because it has a fundamental role in the learning process in ANN models.

A training of neural network is essential before interpreting new information. There are many algorithms that trains the neural networks but the back-propagation algorithm (Error-Back propagation) is the most versatile and robust techniques among them. It provides the most efficient learning procedure for multilayer neural networks. They are popular in solving the prediction problem in engineering (MacKay 1992; Singh et al. 2004c). The back-propagation algorithm is based on the selection of a suitable error function or cost function, whose values are determined by the actual and desired outputs of the network. It is also dependent on the network parameters such as the weights and the thresholds. The basic idea is that the cost function has a particular surface over the weight space, and an iterative process such as the gradient descent method can be used for its minimization.

The back-propagation training consists of a forward pass and a backward pass for computation. The signals from the input layer propagate to the units in the first layer and each unit produces an output. The outputs of these units are propagated to units in subsequent layers. This process continues until the signals reach the output layer where the actual response of the network to the input vector is obtained. During the forward pass, the synaptic weights of the network are fixed. During the backward pass, on the other hand, the synaptic weights are all adjusted in accordance with an error signal which is propagated backward through the network against the direction of synaptic connections. To differentiate between the different processing units, values called biases are introduced in the transfer functions. These biases are referred to as the temperature of a neuron. Except for the input layer, all neurons in the back propagation network are associated with a bias neuron and a transfer function. The bias is much like a weight, except that it has a constant input of 1.The transfer function filters the summed signals received from this neuron. These transfer functions are designed to map neurons or layers net output to its actual output. They are simple step functions either linear or non-linear. The application of these transfer functions depends on the purpose of the neural network. The output layer produces the computed output vectors corresponding to the solution.

During training of the network, data is processed through the network until it reaches the output layer (forward pass). In this layer, the output is compared to the measured values (the 'true' output). The difference or error between both is processed back through the network (backward pass), updating the individual weights of the connections and the biases of the individual neurons. The input and output data are mostly represented as vectors called training pairs. The process as mentioned above is repeated for all the training pairs in the data set, until the network error converges to a threshold minimum defined by a corresponding cost function usually based on the root mean squared error (RMS) or summed squared error (SSE) results. In Figure 1, the j^{th} neuron is connected with a number of inputs

$$X_i = (x_1, x_2, x_3 \ldots x_n)$$

The net input values in the hidden layer will be

$$Net_j = \sum_{i=1}^{n} X_i W_{ij} + \theta_j$$

Where, Xi is the input units, W_{ij} is the weight on the connection of i^{th} input and j^{th} neuron, hj is the Bias neuron (Optional), and n is the number of input units. So, the net output from hidden layer is calculated using a logarithmic sigmoid function

$$\theta_j = f(Net_j) = 1/1 + e^{-(Net_j + \theta_j)}$$

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

The total input to the l[th] unit is

$$Net_1 = \sum_{k=1}^{n} W_{kl} i_k + \theta_l$$

Where, hl is the Bias neuron, and Wkl is the Weight between k[th] neuron and l[th] output. So, the total output from l[th] unit will be,

$$\theta_l = f(Net_l)$$

In the learning process, the network is presented with a pair of patterns, an input pattern and a corresponding desired output pattern. The network computes its own output pattern using its (mostly incorrect) weights and thresholds. Now, the actual output is compared with the desired output. Hence, the error at any output in layer k is

$$e_l = t_l - O_l$$

Where, t_l is the desired output, and O_l is the actual output.

The total error function is given by

$$E = 0.5 \sum_{l=1}^{n} (t_l - O_l)^2$$

Training of the network is basically a process of arriving at an optimum weight space of the network. The descent down error surface is made using the following rule:

$$\nabla W_{jk} = -\eta(\delta E / \delta W_{jk})$$

Where, η is the learning rate parameter, and E the error function. The update of weights for the (n+1)th pattern is given as:

$$W_{jk}(n+1) = W_{jk}(n) + \nabla W_{jk}(n)$$

Similar logic applies to the connections between the hidden and the output layers. This procedure is repeated with each pattern pair of the training exemplar assigned for training the network. Each pass through all the training patterns is called a cycle or epoch. The process is then repeated as many epochs as needed until the error within the user specified goal is reached. This quantity is the measure of how the network has learned.

Network operation of back-propagation network

Back-propagation network is a way of setting up mapping input value and output value; it assembles simple and nonlinear function, and after many times of assemblies, a complicated function form is set up to solve the complicated mapping issue. Figure 2 below is the learning process flow of back propagation network.

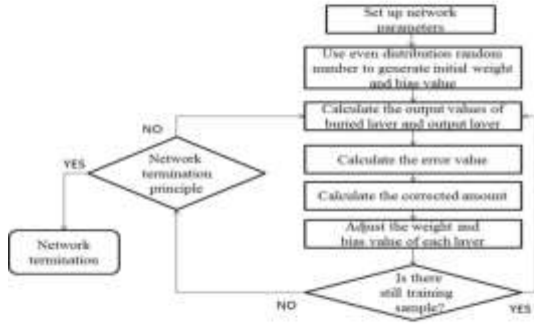

Figure 2: Learning process flow of back propagation network

Network architecture

Feed forward network is used in the study. This architecture is considered to be suitable for the problem based on the pattern identification. Pattern matching is basically an input/output mapping problem. The closer the mapping the better is the performance of the network. A cross-validation technique termed Leaving-One-Out is considered to be more appropriate when small data set are available for the analysis. We used feed forward network for 10 KVA and 2 KVA K-Rated Transformer respectively.

Data Sets for K-rated transformer

Table 1: Data Sets for 10 KVA K-Rated Transformer

Harmonic Order	Frequency (Hz)	R (Ohm)	X (Ohm)
1	50	68.56452	61.73578
3	150	77.59976	198.4374
5	250	86.48783	332.4495
7	350	101.6585	465.3463
9	450	117.7553	595.7743
11	550	136.5354	720.4357
13	650	153.2624	842.5648
15	750	175.1216	968.4564
17	850	191.5371	1090.759
19	950	211.8114	1193.684
21	1050	228.1785	1292.238
23	1150	249.1587	1436.538
25	1250	262.2435	1540.376
27	1350	284.145	1672.746
29	1450	294.9235	1795.227
31	1550	310.2358	1916.345
33	1650	329.2245	2032.243
35	1750	344.3583	2157.823
37	1850	360.5694	2273.235
39	1950	377.2145	2388.435

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

41	2050	395.1442	2527.287
43	2150	412.142	2652.545
45	2250	431.7852	2768.335
47	2350	450.2637	2887.541
49	2450	471.2152	3036.436

Note: Out of 25 data sets, 20 were taken to train the network, and they were tested and validated by the remaining 5 data set.

Table 2: Data Sets for 2 KVA K-Rated Transformer

Harmonics Order	Frequency (Hz)	Resistance (Ohm)
1	50	0.71
2	100	0.72
3	150	0.72
4	200	0.73
5	250	0.75
6	300	0.75
7	350	0.76
8	400	0.76
9	450	0.77
10	500	0.79
11	550	0.8
12	600	0.82
13	650	0.84
14	700	0.85
16	800	0.88
17	850	0.91
18	900	0.93
19	950	0.95
20	1000	0.98
21	1050	1.01
22	1100	1.02
23	1150	1.04
24	1200	1.06
25	1250	1.10
26	1300	1.11
27	1350	1.15
28	1400	1.17

Note: Out of 27 data sets, 22 were taken to train the network, and they were tested and validated by the remaining 5 data set.

The procedure is repeated 15 times, leaving one observation randomly chosen out at a time. The cross validation technique has been used for both the networks used for resistance and reactance. For each network, the input layer consists of two neurons, and the output layer consists of a two neurons. The number of hidden layers was decided by the training and predicting the 'training data' and 'testing data' by varying the number of hidden layers and neurons in the hidden layer. A suitable configuration is to be chosen for the best performance of the network. Out of the different configurations tested, a single hidden layer with four hidden neurons has shown better result for both resistance and reactance. Hence, the final configuration chosen for each network used for resistance and reactance consisted of two input neurons, one hidden layer with four hidden neurons, and two output neurons. A suitable number of epochs has to be assigned to overcome the problems of the over fitting and under fitting of the data. To deal with the above mentioned problem Bayesian regulation is used in the present study. Bayesian regulation is an automated regulation. It removes the chances of over fitting, because it never lets the data to suffer from over fitting. This eliminates the guesswork required in determining the optimum number of epochs of the network (MacKay 1992; Singh et al. 2004b). In the network, the learning rate assigned was 10, numbers of training epochs given were 500 for resistance and reactance with an error goal to be kept 0.0005.

Results and discussion

The results are presented in this section to demonstrate the performance of the network. The mean absolute percentage error (MAPE) and the coefficient of correlation between the predicted and observed values are taken as the performance measures. The prediction was based on the input data sets discussed earlier.

Training for 10 KVA K-Rated Data Sets

Out of 25 data sets, 20 were taken to train the network, and they were tested and validated by the remaining 5 data set.

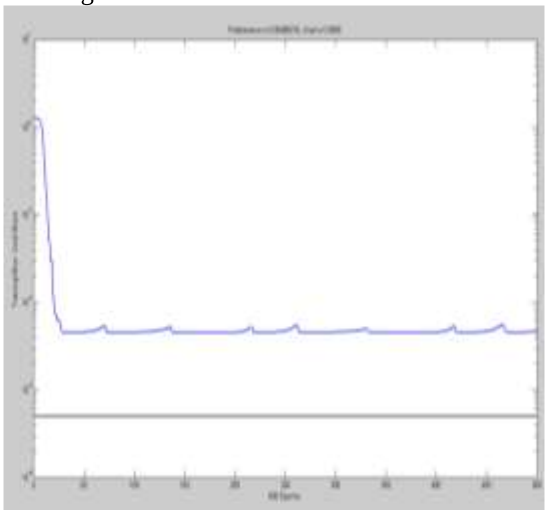

Figure 3: Training for 10 KVA K-Rated Data Sets

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Table 3: Percentage Error of Observed and Predicted Resistance of 10 KVA K-Rated Transformer

Observed Resistance	Predicted Resistance	% Error
395.1	399.4	-1.1
412.1	408.8	0.8
431.7	431.6	0
450.2	455	-1.1
471.2	466.3	1

Table 4: Percentage Error of Observed and Predicted Reactance of 10 KVA K-Rated Transformer

Observed Reactance	Predicted Reactance	% Error
2527.287	2571.6	-1.8
2652.545	2652.4	0
2768.335	2785.4	-0.6
2887.541	2918.4	-1.1
3036.436	2999.7	1.2

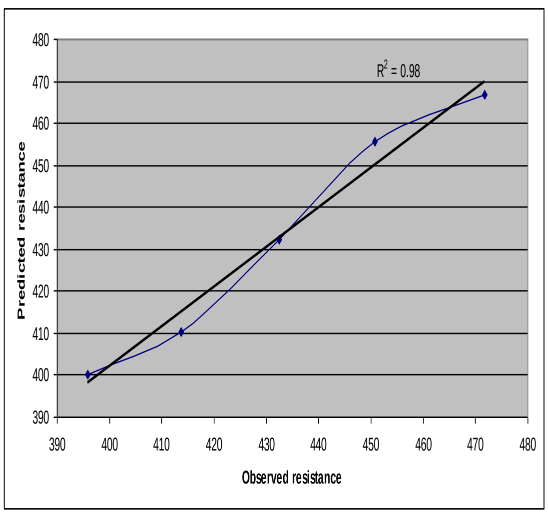

Figure 4: Predicted Resistance Vs Observed Resistance of 10 KVA K-Rated Transformers

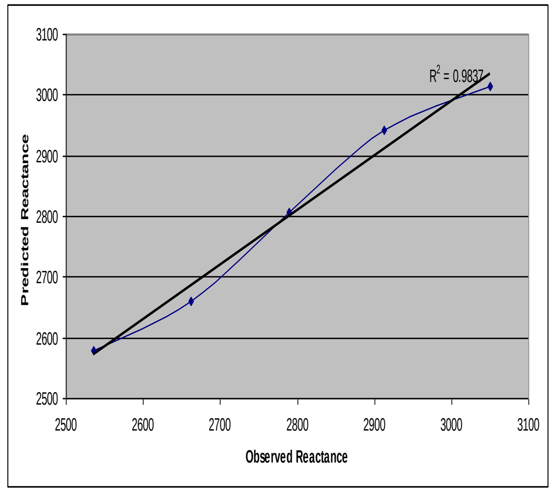

Figure 5: Predicted Reactance Vs Observed Reactance of 10 KVA K-Rated Transformers

Training for 2KVA K-Rated Data Sets

Out of 27 data sets, 22 were taken to train the network, and they were tested and validated by the remaining 5 data set.

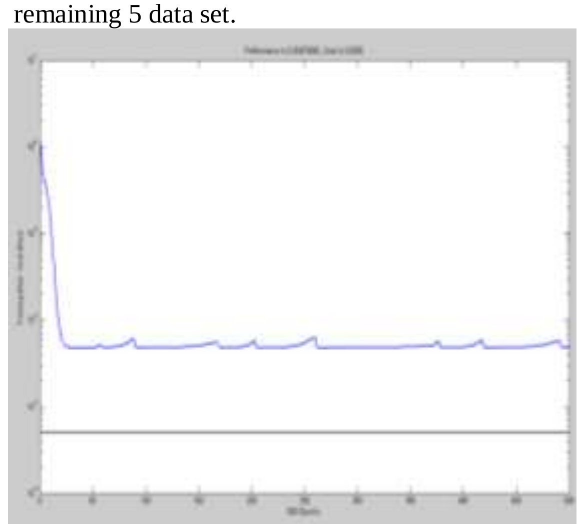

Figure 6: Training for 2 KVA K-Rated Data Sets

Table 5: Percentage Error of Observed and Predicted Resistance of 2 KVA K-Rated Transformer

Observed Resistance	Predicted Resistance	% Error
1.06	1.0627	-0.2499
1.10	1.0808	1.727
1.11	1.0954	1.2935
1.15	1.134	1.364
1.17	1.059	0.9299

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

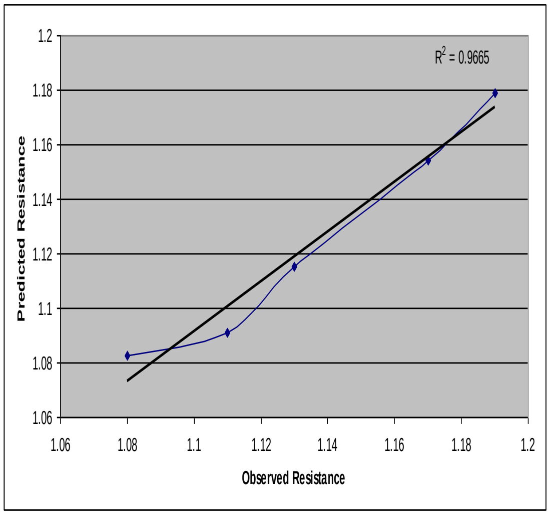

Figure 7: Predicted Resistance Vs Observed Resistance of 2 KVA K-Rated Transformers

Conclusions

ANN can be useful for prediction of resistance and reactance for 10 KVA & resistance for 2 KVA. The corresponding coefficients of correlation are 0.98 and 0.9837 respectively for resistance and reactance of 10 KVA Transformer. Coefficient of correlation is 0.9665 for resistance of 2 KVA Transformer. Because of the complexity of the relationship between the inputs and outputs, the results obtained are encouraging and satisfactory.

Since neural network can learn new patterns that are not previously available on the training data sets, and as they can update knowledge over time as long as more training data sets are presented, and information in a parallel way, they resulted in a greater degree of accuracy, robust and fault tolerance than any other used analysis techniques. Perhaps the most interesting feature of this approach is that it can cope scientifically with subjectivity and uncertainty in the engineering process, rather than blindly avoiding them.

References

[1] *Tony Hoevenaars, P.Eng., Vice President, MIRUS International Inc. "How the Harmonic Mitigating Transformer Outperforms the K-Rated Transformer", Sept. 10, 1999.*

[2] *S. V. Kulkami, S. A. Khaparde., "Stray Loss Evaluation in Power Transformers – A Review", IEEE 2000.*

[3] *N.R Jayasinghe, J.R Lucas, K.B.I.M. Perera. "Power System Harmonic Effects on Distribution Transformers and New Design Considerations for K Factor Transformers", IEE Sri Lanka Annual Sessions – September 2003*

[4] *Bora ACARKAN and Osman KILIC, "Electrical Harmonics Modeling of Office Equipments Using Matlab and Simulink". Kiran Deshpande, Prof.Rajesh Holmukhe, Yogesh Angal, "K-Factor Transformers and Non-linear Loads", 2010.*

[5] *Yang Y & Zhang Q, Analysis for the results of point load testing with artificial neural network, Comput Methods Adv Geomech, China, (1997) 607-612.*

[6] *Singh T.N., Kanchan R & Verma A K, "Prediction of p-wave velocity and anisotropic property of rock using artificial neural network technique", Jol. Of Scientific & Industrial Research, Vol. 63, Jan 2004, pp 32-38.*

[7] *Tai-Ken Lu, Chien-Ta Yeh, and Min-Doon Dirn, "Estimation model of transformer iron Loss using neural network", Journal of Marine Science and Technology, Vol. 18, No. 1, pp. 47-55 (2010).*

[8] *D. J. C. Mackay, "A Practical Bayesian Framework for Back-propagation Networks", Neural Computation, vol. 4(3), pp. 448-472, 1992.*

Author Bibliography

Manish Kumar Verma
M.Tech Scholar
(Department of Electrical Engineering) at IET-Alwar.
Worked in Power Transformer manufacturing company. Have filed 07 Patents, published 05 papers in international journals and presented 06 papers in national and international

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

	conferences. **Email:** ee.manishverma@gmail.com
	**Ruchika Kaushik M.Tech Scholar** (Department of Electronics & Communication Engineering) at IET-Alwar. **Email:** ietruchika@gmail.com
	**Prof. Manab Kumar Sengupta** Graduated with Honours in Physics awarded gold medal then passed B. Tech, M.Tech from University of Calcutta and PhD work in MM Wave technology. Worked in various MNC as R&D engineer and executives. Senior faculty of Calcutta University worked in DRDO-Hyderabad & GMRT-Pune through C.U. Developed many products which are patented- with industry experience of 20 years and teaching in PG and UG level for 17 years with research of 13 years in electrical engineering, automotive electrical and MM Wave technology. Published more than 20 papers in various national and international journals. **Email:** manabksg@yahoo.co.in