Neural Network for Transformer Top-oil Temperature Prediction

R. Vilaithong[1*], S. Tenbohlen[1] and T. Stirl[2]
[1]Institute of Power Transmission and High Voltage Technology, University of Stuttgart,
Pfaffenwaldring 47, 70569 Stuttgart, Germany
[2] Areva Energietechnik GmbH, Activity Schorch Transformers
Rheinstrasse 73, 41065 Moenchengladbach, Germany
*Email: rummiya.vilaithong@ieh.uni-stuttgart.de

Abstract: In this paper, several simple Multilayer Feed-forward network structures for transformer top-oil temperature prediction are proposed and are trained with several Backpropagation training algorithms. Last-state of measured top-oil temperature, load current, ambient temperature and operating states of pumps and fans are applied as inputs of the networks. The neural network structures are later compiled to mathematical models for top-oil temperature calculation at varying load current and ambient temperature. The calculations are performed in long-term investigation for several transformer units with different modes of cooling. Performance of the networks is determined by a deviation between calculated and measured top-oil temperature. The performance from neural network model is also compared with the performance from semi-physical model. Investigations show that the top-oil temperature prediction from neural network model is sufficient for use in an on-line monitoring system and produces accurate results in case of rapid ambient temperature change.

1 INTRODUCTION

Aging of oil immersed cellulose insulation in power transformer depends mainly on its thermal load. In order to keep power transformers longer in service, it is worth to pay particular attention on their thermal behaviour. On-line transformer top-oil temperature prediction is an opportunity for diagnosis of the thermal behaviour of power transformers. Through an on-line comparison of the measured top-oil temperature values and its predicted values, some operational problems such as a malfunction of pumps or fans or a pollution of coolers can be detected.

A basic method for top-oil temperature calculation from the IEEE/ANSI C57-115 standard [1] has been accepted for decades. However, this fundamental model has a limitation of an accurately account due to an effect of variation in ambient temperature. B. C. Lesieutre [2] has later proposed a modified top-oil temperature model based on a concept originally developed from the IEEE top-oil rise temperature model by considering the ambient temperature at the first-order characterization as shown in (1).

$$t_{TO}\frac{dJ_{TO}}{dt} = -J_{TO} + J_{amb} + q_{TO,\infty} \qquad (1)$$

where t_{TO} is the top-oil time constant, J_{TO} is the top-oil temperature, J_{amb} is the ambient temperature and $q_{TO,\infty}$ is the ultimate top-oil rise temperature.

By using forward Euler approximation for a time derivative and by applying linear regression technique for the force cooling state n =1, the equation above is expressed again by D. J. Tylavsky [3] as a simplified semi-physical model in (2), where parameter J_1, J_2, J_3 and J_4 can be obtained from parameter estimation method.

$$J_{TO}[t] = J_1 J_{TO}[t-1] + J_4 J_{amb}[t] + J_2 K[t]^2 + J_3 \qquad (2)$$

The model from (2) and some other proposed top-oil temperature models were investigated in [4], [5] in long-term periods with varying load and ambient temperature. They were also applied in different cooling types of transformer units. Results from some models showed the deviations between measured and calculated top-oil temperature less than 2 K. However, since an artificial neural network presents a growing new technology as indicated by a wide range of applications, it has become an important tool in modern numerical calculation. Therefore, there is also an interest for transformer top-oil temperature prediction using neural networks. Several studies have been already presented, using neural networks as a tool to improve the accuracy of the top-oil temperature calculation [6], [7], [8]. Nevertheless, these investigations are presented only with short-term periods and their application to transformers in the field was not tested.

2 INTRODUCTION TO NEURAL NETWORKS

Neural network or artificial neural network (ANN) refers to an interconnecting group of artificial neurons that uses a mathematical model or a computational model designed to model some properties of biological neural networks. In more practical term, neural networks are non-linear statistical data modelling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

An attraction of neural networks is that they are best suited to solve the problems that are most difficult to solve by traditional computational methods.

2.1 Feed-forward Architecture

Feed-forward neural networks are the most popular and most widely used models in many practical applications. They have been being applied successfully to solve some complex problems including nonlinear system identification and control, financial market analysis, signal modelling, power load forecasting, etc.

Feed-forward neural networks are composed of many computing elements, called neurons, working in parallel. The elements are connected by weights, which are allowed to be adapted through a learning process. The weights on these connections encode the knowledge of a network. An input unit represents raw information that is fed into a network and connected to an output layer through one or more layers, which called hidden layer (Fig. 1). Number of hidden layers and number of neurons in each hidden layer are user design parameters. The general rule is to choose these parameters so that the best possible model with as few parameters as possible is obtained.

Every unit in the layer is connected with all units in the next layer. Each connection may have different strength or weight. Data enter at the inputs and pass through the network, layer by layer to the next, until they arrive at the outputs. There is no feedback between layers. No unit is linked between the same layer, back to the previous layer or skipping the layer. This is why they are called feed-forward neural networks.

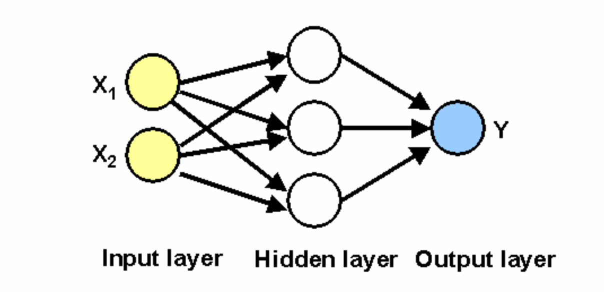

Fig. 1: Feed-forward neural network

The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden units and output units. The outputs can be obtained from taking a linear combination of input signals and their weights, then transform them with an activity function. Each neuron starting from the hidden layer is usually transformed with a nonlinear (sigmoid or hyperbolic tangent) activation function and the activation function in the output layer can be either nonlinear (a nonlinear-nonlinear network) or linear (a nonlinear-linear network). The output of the network can be written in mathematical form as presented in (3).

$$y = \sigma\left(\sum_{i}^{n} w_i x_i - \theta\right) \qquad (3)$$

where y is the output, x_i is the input, w_i is the neuron weight, q is the bias term (another neuron weight) and s is the activity function.

The network weights are adjusted by training the network. The training process involves adjusting the weights till an aim is obtained. The aim involves minimizing the sum of squares of the differences between desired and actual outputs. The network learning is carried out by repeatedly feeding the input-output patterns to the network. One complete entire training set is called an epoch. There are a number of such learning rules available for neural network models. The delta rule is one of the most commonly used learning rules. It is also called the Least Mean Square (LMS) method. For a given input vector, an output vector is compared to a correct answer. If the difference is zero, no learning takes place, otherwise, the weights are adjusted to reduce this difference. The change in weight w from output u_i to u_j is given by (4), where r is the learning rate, a_i represents the activation of u_i and e_j is the difference between the expected output and the actual output of u_j [9]. If the set of input patterns form a linearly independent set then arbitrary associations can be learned using the delta rule.

$$\Delta w_{ij} = r * a_i * e_j \qquad (4)$$

2.2 Backpropagation training algorithms

Backpropagation is an algorithm that extends the analysis of the delta rule to the networks with hidden nodes. It was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and non-linear differentiable transfer functions. Input vectors and corresponding target vectors are used to train a network unit. It can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by analyst. It computes the error term for the output units using the observed error. The model repeat propagating the error term back from output layer to the previous layer and updating the weights between the two layers until the earliest hidden layer is reached. Whereas, weights between neurons of successive layers are initially assigned in random. The speed and accuracy of the learning process, that is the process of updating the weights, also depends on a factor known as the learning rate.

*2.2.1 Levenberg-Marquardt backpropagation **(trainlm)***

The Lavenberg-Marquardt algorithm uses an early stopping criterion to improve network training speed and efficiency. To determine the criterion, all the data are divided into three sets. The first set is the training set for determining the weights and biases of the network. The second set is the validation set for evaluating the weights and biases and for deciding when to stop training. The validation error normally decreases at the beginning of the training process. When the network starts to over-fit the data, the validation error begins to increase. The training is stopped when the validation error begins to increase and the weights and biases will then be derived at the minimum error. The last data set is for validating the weights and biases to verify the capability of the stopping criterion and to estimate the expected network operation on new data sets.

2.2.2 *Scaled conjugate gradient backpropagation **(trainscg)***

The Scaled conjugate Gradient, developed by Moller [10], is based on a well-known optimization technique in numerical analysis called the Conjugate Gradient Method. It was designed to avoid the line search per learning iteration by using a Levenberg-Marquardt approach in order to scale the step size. The basic idea is to combine the model-trust region approach with the conjugate gradient approach. Unlike many other standard backward propagation algorithms, this technique does not require any user-specified parameters and its computation is faster.

2.2.3 *Automated Bayesian Regularization **(trainbr)***

Bayesian regularization is a modification of the Levenberg-Marquardt training algorithm to improve the model's generalization. Over-fitting problem or poor generalization capability happens when a neural network over learns during a training period. As a result, such a too well trained model may not perform well on unseen data set due to its lack of generalization capability. This approach involves modifying the performance function, which is normally chosen to be the sum of squares of the network errors on the training set (MSE or E_d).

$$F = E_d = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 \qquad (5)$$

The objective function in (5) is possible to be generalization improved if it is added by a term E_w, that is the sum of squares of the network weights (6).

$$F = \mathrm{b}E_d + \mathrm{a}E_w \qquad (6)$$

The b and a parameter are optimized in Bayesian framework of MacKay [11], [12]. It is assumed that the weights and biases of the network are random variables following Gaussian distributions and the parameters are related to the unknown variances associated with these distributions. Using this performance function will cause the network to have smaller weights and biases, and this will force the network response to be smoother and less likely to over-fitting.

3 PREDICTION OF TOP-OIL TEMPERATURE

In present section, one-hidden layer and two-hidden layer feed-forward networks with various numbers of neurons were structured. These networks were trained with three different backpropagation training algorithms as mentioned in 2.2. The works were divided into two steps. First, the networks with one hidden layer were investigated with a number of neurons from 1-20 neurons. In the second step, the models were further constructed with two hidden layers and with a number of hidden neurons between 1-5 neurons in both hidden layers. Measured data of three transformer units provided by an on-line monitoring system MS 2000 [13] were applied as inputs to the models during training process. These inputs were last-state of measured top-oil temperature, load current, ambient temperature and operating states of pumps and fans. The interval time of each data set was vary. The specifications of these three transformers are shown in Tab. 1. It has been noted that there were different types of cooling systems. The operating states of the cooling units of Transformer 2 (Tr2) remained constant during the investigated period (number of fans operated = 2). Whereas, the cooling operating states of Transformer 3 (Tr3) were varied among five states as can be seen from Tab.2.

Tab. 1: Investigated power transformer main characteristics

Transformer	Tr1	Tr2	Tr3
Rated power [MVA]	40	150	850
Rated Voltage [kV]	110	245/36/6	21/220
Short-circuit loss [kW]	135	414.2	2060
No-load loss [kw]	20	67.52	370
Type of cooling	ONAN	ONAF	ODAF

The measured data from year 2004 of Tr1 were collected from a whole year. While, a whole-year measured data of Tr2 were collected from year 2003. The data from January until May of Tr1 and Tr2 were applied to be a train data set of the networks. Whereas, the data from June until December were applied to be a test data set of the models. The measured data for Tr3 were collected from both year 2003 and year 2004. Therefore, the data from year 2003 were used as train data and the data from year 2004 were used as test data. The calculation of the top-oil temperature is the testing process of the networks. It was performed at varying load current and ambient temperature in long-term investigation. The performances of the networks were determined by the average deviations between calculated and measured top-oil temperature.

Tab. 2: States of pumps and fans in operation of Tr3

State	Number of pumps	Number of fans
1	8	0
2	8	2
3	8	4
4	8	6
5	8	8

4 RESULTS AND DISCUSSION

After the training process, all investigated neural network structures were interpreted to the mathematical models for the top-oil temperature calculation. The weights and biases were transformed to be the coefficients of the models. The standard sigmoid nonlinear activation function was chosen to be a smooth step function. The expression of sigmoid function can be seen from (7), where x is the weighted sum of all inputs and the biases of neurons [9].

$$Sigmoid(x) = \frac{1}{1+e^{-x}} \qquad (7)$$

Fig. 2 is an example for algorithm structure of feed-forward neural network model with one hidden layer. These algorithms can be written in a general mathematical form as shown in (8). This mathematical model, along with optimized weights and biases obtained from training process, was later applied to the testing data for calculating the top-oil temperature.

$$y = \sum_{h=1}^{H} w_{ik} \left\{ f_k^{sig} \left(\sum_{j=1}^{J} w_{kj} i_j + b_k \right) \right\} + b_h \qquad (8)$$

where i_j is input of the network, j is input number, w_{kj} is weight connection from input to the neuron in hidden layer, k is neuron number in hidden layer, J is total number of inputs, h is neuron number in output layer and H is total number of neurons in output layer.

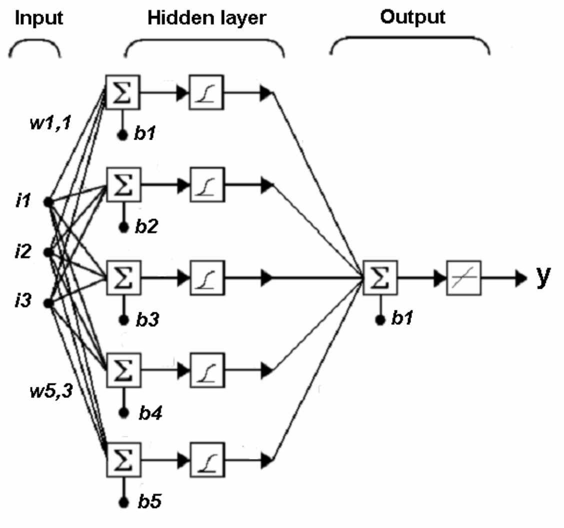

Fig. 2: Example of one hidden layer feed-forward structure with weights and biases.

4.1 Comparison of training time among different backpropagation training algorithms

Speed of the network training process can range from a few seconds to many hours, depending on the factors such as the number of weights in the networks, the number of training examples considered and the setting of various learning algorithm parameters. In this paper, the training time of each training function was also investigated. The one-hidden layer networks with number of hidden neurons 1, 10 and 20 are examined. As seen from Tab. 3, the Levenberg-Marquardt training algorithm shows the shortest training time for all results of investigated transformers. Whereas, the Bayesian regularization presents the longest time. This training time is proportional to the number of neurons in hidden layer and to the data size of investigated transformers.

Tab. 3: Training time for each training algorithm

Tr	Data Size	Number of Neurons	Training time (s)		
			trainbr	trainlm	trainscg
Tr1	27343	1	2	0.4	3
		10	57	3	16
		20	78	11	33
Tr2	34753	1	3	2	6
		10	58	20	29
		20	123	40	56
Tr3	118453	1	8	2	31
		10	323	41	146
		20	739	66	258

4.2 Comparison among different learning algorithms for one-hidden layer networks

The average top-oil temperature deviations calculated from the one-hidden layer feed-forward network trained from three different training functions are shown in Fig. 3 - Fig. 5. The results present the deviations from different networks with various numbers of hidden neurons. The Bayesian regularization training algorithm has a better performance than others. The lowest temperature deviations are found from the network with eighteen neurons, thirteen neurons and seventeen neurons for Tr1, Tr2 and Tr3 respectively.

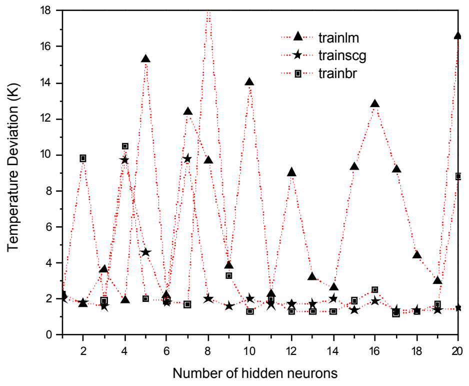

Fig. 3: Temperature deviation compared among different training algorithms of Tr1

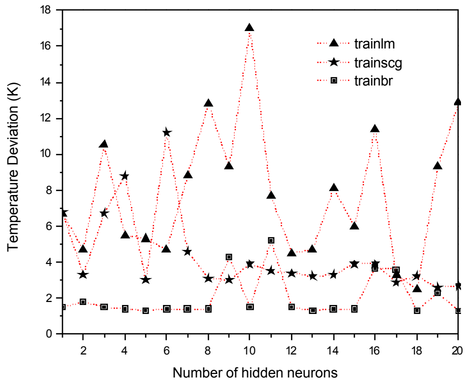

Fig. 4: Temperature deviation compared among different training algorithms of Tr2

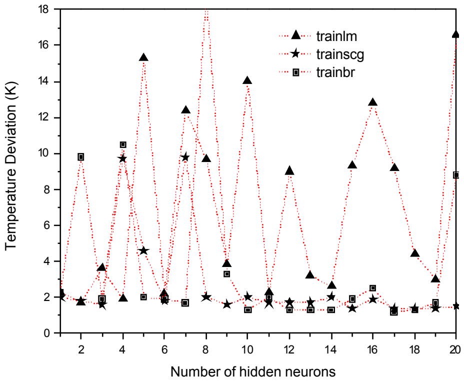

Fig. 5: Temperature deviation compared among different training algorithms of Tr3

4.3 Comparison among different learning algorithms for two-hidden layer networks

The Bayesian regularization training algorithm was applied as the training function for the models with two hidden layers. The sigmoid transfer function was applied in the first hidden layer and the linear transfer function was applied in second hidden layer. As seen from the results in 4.2 that the networks structured with the number of hidden neurons from 1 to 5 also perform the satisfy results and the average temperature deviations from all three transformers are less than 2 K when the models are trained with the Bayesian regularization training algorithm. Thus, the further investigations were done with the networks with the number of hidden neurons from 1 to 5 in both hidden layers. Consequently, in this section, there were 25 models to be investigated.

Tab. 4: Results of different networks comparison of Tr1

Selected Network model	Number of hidden neurons		Temperature deviation (K)
	First layer	Second layer	
1	2	2	4.1
2	2	2	4.9
3	3	3	4.0
4	3	3	5.3
5	4	4	5.3

Tab. 5: Results of different networks comparison of Tr2

Selected Network model	Number of hidden neurons		Temperature deviation (K)
	First layer	Second layer	
1	1	2	6.6
2	1	3	6.6
3	1	5	6.6
4	2	5	6.7
5	2	1	6.7

Tab. 6: Results of different networks comparison of Tr3

Selected Network model	Number of hidden neurons		Temperature deviation (K)
	First layer	Second layer	
1	2	1	2.2
2	2	2	2.2
3	3	1	2.2
4	3	5	2.1
5	5	1	2.2

Tab. 4 – Tab. 6 show the average temperature deviations between measured top-oil temperature and the top-oil temperature calculated from these different neural network models. The tables present the results from five models, which show the best results. It is found that the average temperature deviations calculated of various two-hidden layer neural network models are in the same range. Besides, it is found that the results from the networks with one-hidden layer show the better performance than the results from two-hidden layer networks.

4.4 Comparison between Network Model and Semi-physical Model

The average results of some periods from feed-forward networks trained by Bayesian regularization training algorithm with one-hidden layer and eighteen hidden neurons, thirteen hidden neurons and seventeen hidden neurons for Tr1, Tr2 and Tr3 respectively are compared with the results from semi-physical model as presented in [5]. Tab. 7 presents that the neural network models also show the accuracy in top-oil temperature calculating almost in the same level as the results from semi-physical models.

Tab. 7: Average top-oil Temperature deviations compared between neural network models and the semi-physical models

Transformer	Neural Network (K)	Semi-physical (K)
Tr1	1.8	1.9
Tr2	2.2	1.2
Tr3	1.1	1.5

The top-oil temperature calculated by means of neural network models and semi-physical models from the periods as mentioned above are depicted in Fig. 6 – Fig. 8 for Tr1, Tr2 and Tr3 respectively. It can be seen that the neural network models could also give accurate calculated results as the results calculated from the semi-physical models.

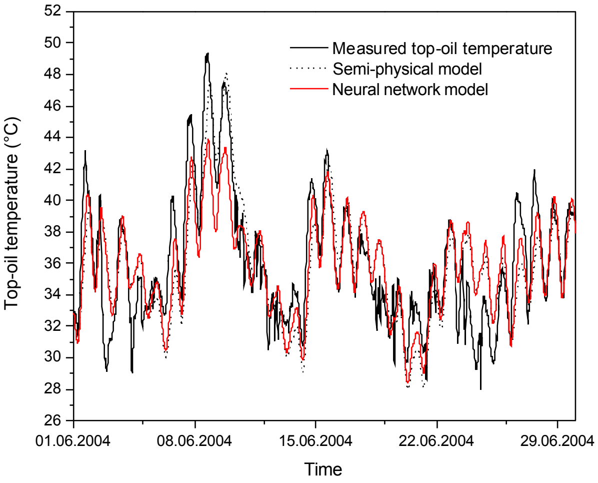

Fig. 6: Measured and calculated top-oil temperature results from neural network model and semi-physical model of Tr1

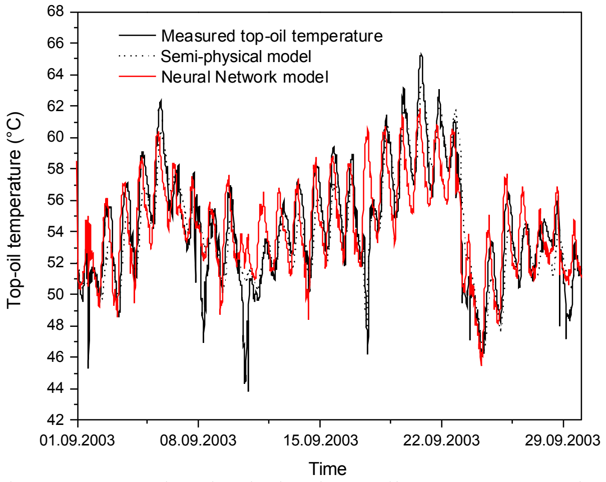

Fig. 7: Measured and calculated top-oil temperature results from neural network model and semi-physical model of Tr2

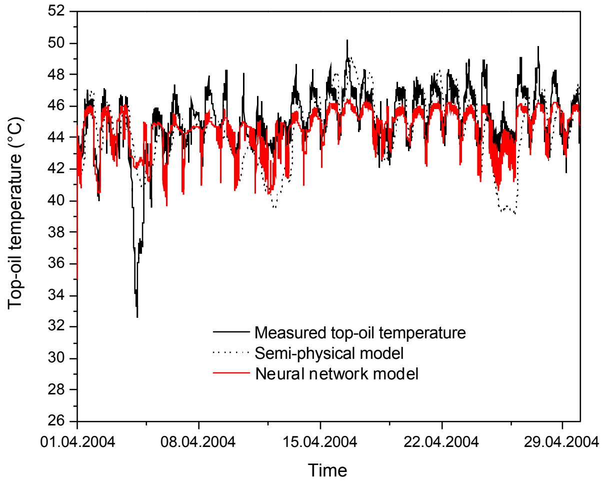

Fig. 8: Measured and calculated top-oil temperature results from neural network model and semi-physical model of Tr3

5 CONCLUSION

A brief introduction of feed-forward neural network together with some backpropagation learning algorithms was given in this paper. Several multilayer feed-forward neural network models for top-oil temperature long-term prediction were examined. The models were different in a number of hidden layers, a number of neurons and learning algorithms. Measured data of three transformer units provided by an on-line monitoring system MS 2000 were applied to the models. Last-state of measured top-oil temperature, load current, ambient temperature and operating states of pumps and fans were selected as inputs of the networks. Results show that the Bayesian regularization-training algorithm provides best performance in term of temperature deviation between measured and calculated top-oil temperature. Whereas, the Levenberg-Marquardt backpropagation training algorithm provides best performance in training time investigation. It is found in all investigated transformers that the average temperature deviations calculated from the models with one hidden layer are lower than the models with two hidden layers. The good performance in top-oil temperature calculation from each transformer can be found in the models with different number of neurons. When the results from neural network models are compared with the results from semi-physical models, investigations show that the performances from both neural network models and semi-physical models are nearly similar, the average temperature deviations are less than 2 K. Therefore, the predictions from both models are sufficient for use in an on-line monitoring system. Furthermore, the neural network models also produce accurate results in the case of rapid ambient temperature change.

6 REFERENCES

[1] *IEEE Guide for Loading Mineral-Oil-Immersed Transformers*, IEEE Standard C57.91-1995.

[2] B. C. Lesieutre, W. H. Hagman, and J. L. Kirtley Jr., "An Improved Transformer Top Oil Temperature Model for Use in An On-Line Monitoring and Diagnostic System," *IEEE Trans. Power Delivery*, vol. 12, pp. 249-256, Jan. 1997.

[3] D. J. Tylavsky, Q. He, and J. R. Hunt, "Transformer Top-oil Temperature Modeling and Simulation," *IEEE Trans. Industry Application* vol. 36, pp. 1219-1225, Sep. 2000.

[4] R. Vilaithong, S. Tenbohlen and T. Stirl, "Improved top-oil temperature model for unsteady-state conditions of power transformer," in *Proc. 2005 ISH Internal Symposium on High Voltage Engineering*, paper no. F-42.

[5] R. Vilaithong, S. Tenbohlen and T. Stirl, "Investigation of different top-oil temperature models for on-line monitoring system of power transformer," in *Proc. 2006 CMD Internal Conference on Condition Monitoring and Diagnosis*, paper no. 661

[6] Q. He, J. Si, and D. J. Tylavsky "Prediction of Top-Oil Temperature for Transformers Using Neural Networks," *IEEE Trans. Power Delivery*, vol. 15, pp. 1205-1211, Jan. 2000.

[7] J. K. Pylvänäinen, K. Nousiainen, and P. Verho, "Studies to Utilize Loading Guides and ANN for Oil-Immersed Distribution Transformer Condition Monitoring," *IEEE Trans. Power Delivery*, vol. 22, pp. 201-207, Jan. 2007.

[8] M. K. Pradhan and T.S. Ramu, "On-line Monitoring of Temperature in Power Transformers using Optimal Linear Combination of ANNs," in *Proc. 2004 IEEE International Symposium on Electrical Insulation*, pp. 70-73.

[9] H. Demuth, M. Beale, *Neural Network Toolbox: For Use with MATLAB*, USA: The Math Works, Inc., 1992.

[10] M. F. Moller, “A scaled conjugate gradient algorithm for fast supervised learning” *Neural Networks*, vol. 6, pp. 525-533, 1993.

[11] D. J. C. Mackay, “A Practical Bayesian Framework for Backpropagation Networks,” *Neural Computation*, vol. 4(3), pp. 448-472, 1992.

[12] D. J. C. Mackay, “Bayesian Interpolation,” *Neural Computation*, vol. 4(3), pp. 415-447, 1992.

[13] S. Tenbohlen, T. Stirl, G. Bastos, J. Baldauf, P. Mayer, M. Stach, B. Breitenbauch and R. Huber, “Experience-based Evaluation of Economic Benefits of On-line Monitoring Systems for Power Transformers,” *CIGRE Session 2002*, pp. 12-110.