JST: Smart Systems and Devices
Volume 32, Issue 3, September 2022, 061-068

Dissolved Gas Analysis of Insulating Oil for Power Transformer Fault Diagnosis with Bayesian Neural Network

Son T. Nguyen[1*]***, Stefan Goetz***[2]
[1]*School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam*
[2]*Department of Electrical and Computer Engineering, Technical University of Kaiserslautern, Germany*
[*] *Corresponding author email: son.nguyenthanh@hust.edu.vn*

Abstract

Dissolved gas analysis is widely used for preventative maintenance techniques and fault diagnoses of oil-immersed power transformers. There are also various conventional methods of dissolved gas analysis for insulating oil in power transformers including methods of Doernenburg ratios, Rogers ratios and Duval's triangle. The Bayesian techniques have been developed over many years and applied to a range of different fields including the problem of training in artificial neural networks. In particular, the Bayesian approach can solve the problem of over-fitting of artificial neural networks after being trained. The Bayesian framework can be also utilised to compare and rank different architectures and types of artificial neural networks. This research aims at deploying a detailed procedure of training artificial neural networks with the Bayesian inference, also known as Bayesian neural networks, to classify power transformer faults based on Doernenburg and Rogers gas ratios. In this research, the IEC TC 10 databases were used to form training and test data sets. The results obtained from the performance of trained Bayesian neural networks show that despite the limitation of the available dissolved gas analysis data, Bayesian neural networks with an appropriate number of hidden units can successfully classify power transformer faults with accuracy rates greater than 80%.

Keywords: Power transformers, fault diagnosis, dissolved gas analysis, Bayesian neural networks.

1. Introduction

Power transformers are electrical equipment widely used in power production, transmission, and distribution systems. Incipient power transformer faults usually cause electrical and thermal stresses (arcing, corona discharges, sparking, and overheating) in insulating materials. Because of these stresses, insulating materials can degrade or breakdown and several gases are released. Therefore, the analysis of these dissolved gases can provide useful information about fault conditions and types of materials involved. Dissolved gas analysis (DGA) of power transformer insulating oil is a well-known technique in monitoring and diagnosing the power transformer health [1-3]. Conventional analysis techniques of dissolved gases can be performed by analysing different gas concentration ratios (Doernenburg ratios, Rogers ratios and Duval's triangle method) [4,5].

Artificial intelligence (AI) based methods have been introduced to improve the diagnosis accuracy and remove the inherent uncertainty in DGA. These methods were proposed with the use and exploration of artificial neural networks (ANNs) [6, 7], fuzzy logic (FL) [8,9], support vector machine (SVM) [10,11], decision tree (DT) [12, 13] and K-nearest neighbours (KNN) [14,15]. ANNs have been extensively used in applications of pattern recognitions as they are adaptive, capable of handling highly nonlinear relationships, and can generalise solutions for new sets of data (unseen data). As the development of ANNs does not require any physical models, the incipient fault detection in power transformers using ANNs can be reduced to an association process of inputs (patterns of gas concentration) and outputs (fault types). The use of ANNs and DGA samples for diagnosing incipient faults in power transformers have been reported in some related studies [6,7]. However, in these studies, ANNs were only trained by traditional neural network training methods, which could only minimise a defined data error function without the consideration of over-fitting and model complexity causing poor generalisation of ANNs trained on finite and uncertain data sets.

In this research, an improved version of ANNs, called Bayesian neural networks (BNNs) [16-18], have been proposed for diagnosing faults of oil-immersed power transformers. The main advantage of BNNs is that these neural networks can handle the uncertainty in parameters of ANNs and can be also trained with limited data. In addition, the training procedure of BNNs does not require a validation set separated from the available data. As a result, the entire available data can be only used to form training and test sets. The

ISSN: 2734-9373
https://doi.org/10.51316/jst.160.ssad.2022.32.3.8
Received: June 29, 2022; accepted: August 13, 2022

paper is organised as follows. Section 2 briefly describes conventional methods of DGA for power transformer fault diagnoses and the basic theory of BNNs including the suitable determination of regularisation parameters to prevent the over-fitting problem and the criterion to select the optimal number of hidden units. Results and discussions are presented in Section 3 based on the evaluation of the performance of trained BNNs used to classify power transformer faults. Finally, Section 4 is conclusion and future works for this research.

2. Material and Method

2.1. Conventional Methods of DGA for Power Transformer Insulating Oil

The main causes of gas formation within an operating power transformer are electrochemical and thermal decomposition, and evaporation. The basic chemical reactions involve the breaking of carbon–hydrogen and carbon–carbon bonds. This phenomenon can usually form active hydrogen atoms and hydrocarbon fragments that can combine with one another to make the following gases: hydrogen (H_2), methane (CH_4), acetylene (C_2H_2), ethylene (C_2H_4), and ethane (C_2H_6). With cellulose insulation, thermal decomposition or electric faults can produce methane (CH_4), hydrogen (H_2), monoxide (CO) and carbon dioxide (CO_2). These gases are generally called 'key gases'.

After samples of transformer insulating oil are taken, the first step in analysing DGA results is to measure the concentration level (in ppm) of each key gas. Once key gas concentrations are greater than normal limits, some analysis techniques should be used to determine the potential faults within the transformer. These techniques involve calculating key gas ratios and comparing these ratios to suggested limits. The most used techniques consist of Doernenburg ratios and Rogers ratios methods based on the following gas ratios: CH_4/H_2, C_2H_2/C_2H_4, C_2H_2/CH_4, C_2H_6/C_2H_2, and C_2H_4/C_2H_6. The suggested limits of Doernenburg ratios method and Rogers ratios method are shown in Tables 1 and 2, respectively.

In Duval's triangle method, the total accumulated amount of three key gases, methane (CH_4), acetylene (C_2H_2), and ethylene (C_2H_4), is calculated. Next, each gas concentration is divided by the total accumulated amount of three gases to find the percentage associated with each gas. These values are then plotted in Duval's triangle [6] as shown in Fig. 1 to derive a diagnosis. Sections within the triangle designate: partial discharge (PD), low-energy discharge (D1), high-energy discharge (D2), thermal fault below 300 °C (T1), thermal fault between 300 °C and 700 °C (T2), thermal fault above 700 °C (T3).

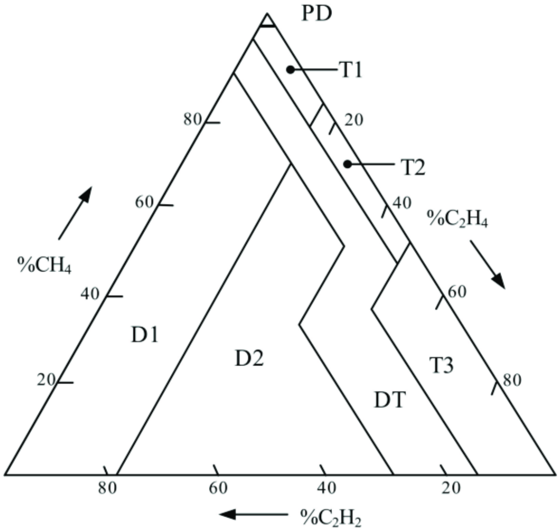

Fig. 1. Duval's triangle

2.2. Bayesian Neural Networks

2.2.1. Multi-layer perceptron neural networks

A) Feed-forward propagation

Multi-layer perceptron (MLP) neural networks are widely used in engineering applications. These networks take in a vector of real inputs, x_i, and from them compute one or more values of activation of the output layer, $a_k(x,w)$. For networks with a single layer of hidden nodes, as shown in Fig. 2, the activation of the output layer is computed as follows:

$$a_k(x) = b_k + \sum_{j=1}^{m} w_{kj} \tanh\left(\bar{b}_j + \sum_{i=1}^{d} \bar{w}_{ji} x_i\right) = b_k + \sum_{j=1}^{m} w_{kj} y_j \tag{1}$$

where, $\bar{w}_{ji}$ is the weight on the connection from input unit i to hidden unit j; similarly, w_{kj} is the weight on the connection from hidden unit j to output unit k. The $\bar{b}_j$ and b_k are the biases of the hidden and output units. These weights and biases are parameters of the MLP neural network.

In c-class classification problems, the target variables are discrete class labels indicating one of possible classes. The softmax (generalised logistic) model can be used to define the conditional probabilities of the various classes of a network with output units as follows:

$$z_k(x) = \frac{\exp(a_k(x))}{\sum_{k'=1}^{c} \exp(a_{k'}(x))} \tag{2}$$

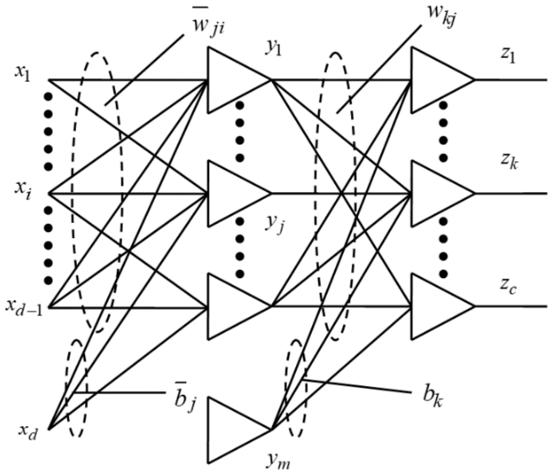

Fig. 2. Classification MLP neural network.

For c-classes ($c > 2$) classification problems, the data error function has the following form:

$$E_D = -\sum_{n=1}^{N}\sum_{k=1}^{c} t_k^n \ln z_k^n \tag{3}$$

where E_D is called the entropy function and N is the number of sample training patterns.

B) Regularisation

In MLP neural network training, the regularisation should be involved to prevent any weights and biases from becoming too large because large weights and biases can cause poor generalisation of the trained network for new test cases. Therefore, a weight decay penalty term is usually added to the data error function to penalise large weights and biases to obtain the following function:

$$S(w) = E_D + \sum_{g=1}^{G} \xi_g E_{W_g} \tag{4}$$

where $S(w)$ is known as the cost function, G is the number of groups of weights and biases in the network. The second term on the right-hand side of equation (4) is referred to as the weight decay term. ξ_g is the hyperparameter for the distribution of weights and biases in group g. E_{W_g} and w_g are the error and the vector of weights and biases in group g, respectively.

C) Updating weights and biases

The problem of neural network training has been formulated in terms of the minimisation of the cost function $S(w)$, which is a function of weights and biases in the network. We can also group the network weights and biases together into a single W-dimensional weight vector, denoted by w, with components $w_1 \ldots w_W$.

For MLP neural networks with a single layer of hidden units, the cost function is usually a highly non-linear function of weights and biases. Therefore, the cost function $S(w)$ can have many minima satisfying the following condition:

$$\nabla S(w) = 0 \tag{5}$$

Table 1. Suggested limits of Doernenburg ratios method

Suggested fault diagnosis	$R_1 = \frac{CH_4}{H_2}$	$R_2 = \frac{C_2H_2}{C_2H_4}$	$R_3 = \frac{C_2H_2}{CH_4}$	$R_4 = \frac{C_2H_6}{C_2H_2}$
Thermal decomposition	> 1.0	< 0.75	< 0.3	> 0.4
Partial discharge	< 0.1	-	< 0.3	> 0.4
Arcing	$> 0.1 - < 1.0$	> 0.75	> 0.3	< 0.4

Table 2. Suggested limits of Rogers ratios method

Suggested fault diagnosis	$R_1 = \frac{CH_4}{H_2}$	$R_2 = \frac{C_2H_2}{C_2H_4}$	$R_5 = \frac{C_2H_4}{C_2H_6}$
Unit normal	$> 0.1 - < 1.0$	< 0.1	< 1.0
Low-energy density arcing-PD	< 0.1	< 0.1	< 1.0
Arcing-high energy discharge	$0.1 - 1.0$	$0.1 - 3.0$	> 3.0
Low temperature thermal	$> 0.1 - < 1.0$	< 0.1	$1.0 - 3.0$
Thermal<700°C	> 1.0	< 0.1	$1.0 - 3.0$
Thermal>700°C	> 1.0	< 0.1	> 3.0

The minimum corresponding to the smallest value of the cost function is called the global minimum, while other minima are called local minima. In practice, it is impossible to find closed-form solutions for the minima. Instead, we consider algorithms that involve a search through the weight space with a succession of steps of the form:

$$w_{m+1} = w_m + \alpha_m d_m \tag{6}$$

where m labels the iteration step, w_m and w_{m+1} are the vectors of weights and biases at the m-th and $(m+1)$-th iteration steps, respectively. d_m and α_m are the search direction and step size at the m-th iteration step.

Different adaptive neural network training algorithms can automatically find the suitable search direction d_m and determine the optimal step size α_m. The advanced adaptive neural network training algorithms consist of Conjugate Gradient, Scaled Conjugate Gradient and Quasi-Newton methods [17].

2.2.2. Bayesian training for classification mlp neural networks

The Bayesian learning of MLP neural networks is performed by considering Gaussian probability distribution of weights and biases giving the best generalisation [16]. In particular, the weights and biases in the network are adjusted to their most probable values given the training data set-D. Specifically, the posterior distribution of weights and biases can be computed using Bayes' rule as follows:

$$p(w \mid D, X) = \frac{p(D \mid w, X)p(w \mid X)}{p(D \mid X)} \tag{7}$$

Given a set of candidate neural networks having different numbers of hidden nodes, the posterior probability of each network can be expressed as:

$$p(X_i \mid D) = \frac{p(D \mid X_i)p(X_i)}{p(D)} \tag{8}$$

If all the candidate neural networks can be seen to be equally probable before any data arrives, $p(X_i)$ are identical for all neural networks. As $p(D)$ does not depend on each neural network, the most probable network can be chosen corresponding to the highest value of $p(D \mid X)$. Therefore, the evidence can be utilized to rank different architectures of neural networks.

In neural network training, the hyperparameters are initialised to be arbitrary small values. Next, the cost function is minimised using an advanced optimisation technique. When the cost function has reached a local minimum, the hyperparameters can be re-estimated. This task requires the evaluation of the Hessian matrix of the cost function as follows:

$$A = H + \sum_{g=1}^{G} \xi_g I_g \tag{9}$$

where H is the Hessian matrix of E_D and I_g is the identity matrix selecting the weights and biases in the g-th group. The number of 'well-determined' weights γ_g in group g is calculated based on the old values of ξ_g as follows:

$$\gamma_g = W_g - \xi_g tr\left(A^{-1} I_g\right) \quad (g = 1, ..., G) \tag{10}$$

The new value of the hyperparameter ξ_g is then re-estimated as follows:

$$\xi_g = \frac{\gamma_g}{2E_{W_g}} \qquad (g = 1, ..., G) \tag{11}$$

The hyperparameters need to be re-estimated several times until the cost function value tends not to change significantly between consecutive re-estimation periods. After the network training is completed, the values of parameters γ_g and ξ_g are then used to compute the log evidence of network X_i having M hidden nodes as follows [18]:

$$\begin{aligned} \ln Ev(X_i) \equiv -S(w) + \sum_{g=1}^{G} \frac{W_g}{2} \ln \xi_g - \frac{1}{2} \ln |A| + \\ + \ln M! + M \ln 2 + \sum_{g=1}^{G} \frac{1}{2} \left(\frac{4\pi}{\gamma_g} \right) \end{aligned} \tag{12}$$

where W_g is the number of weights and biases in group g. Equation (12) is used to compare different neural networks having different numbers of hidden nodes. The best neural network will be selected with the highest value of the log evidence.

3. Results and Discussion

3.1. Input and Output Patterns

The IEC TC10 databases were used for training and testing BNNs [1]. For each input pattern, there is a corresponding output pattern describing the fault type for a given diagnosis criterion. Five key gasses, which are all combustible: hydrogen (H_2), methane (CH_4), ethylene (C_2H_4), ethane (C_2H_6), and acetylene (C_2H_2), are used in this study. The output vector contains codes of 0 and 1, which indicates five fault types as shown in Table 3. The training set was formed by taking 81 data samples and the test set consists of 36 data samples as shown in Table 4.

Most power transformers have low dissolved gas concentrations of a few ppm (part per million). However, faulty power transformers can often cause thousands or tens of thousands of ppm. This problem usually gives a difficulty to visualise the dissolved gas data. Therefore, the most informative features of DGA data can be obtained by using the order of magnitude of DGA concentrations, rather than their absolute values. An effective way to take these changes into account is to rescale DGA data using the logarithmic transform. For an easy interpretation, the $\log_{10}$ is used.

Table 3. Fault types and corresponding output vectors.

Fault type	Output vector
PD	$[1\ \ 0\ \ 0\ \ 0\ \ 0]^T$
D1	$[0\ \ 1\ \ 0\ \ 0\ \ 0]^T$
D2	$[0\ \ 0\ \ 1\ \ 0\ \ 0]^T$
T1 & T2	$[0\ \ 0\ \ 0\ \ 1\ \ 0]^T$
T3	$[0\ \ 0\ \ 0\ \ 0\ \ 1]^T$

Table 4. Datasets from the IEC TC 10 database.

	Numbers of data samples	
Fault type	Training set	Test set
PD	5	4
D1	18	8
D2	36	12
T1 & T2	10	6
T3	12	6
Total	81	36

Data normalisation: is a rescaling of the input data from the original range so that all values are within the range of 0 and 1:

$$y_i = \frac{x_i - \min(X)}{\max(X) - \min(X)} \tag{13}$$

3.2. The Network Training Procedure

To determine the optimal number of hidden nodes (number of nodes in the hidden layer) of a BNN, different BNNs with varied numbers of hidden nodes were trained and they have the following specifications:

1) Four hyperparameters ξ_1, ξ_2, ξ_3, and ξ_4 to constrain the magnitudes of the weights on the connection from the input nodes to the hidden nodes, the biases of the hidden nodes, the weights on the connection from the hidden nodes to the output nodes, and the biased of the output nodes.
2) The number of inputs depends on the number of gas ratios of a specific diagnosis method and one augmented input with a constant value of 1.
3) Five outputs, each corresponding to a specific class of faults as shown in Table 3. For a given number of hidden nodes, ten neural networks with different initial conditions were trained.

The training procedure was implemented as follows:

1) The weights and biases in four different groups were initialized by random selections from zero-mean, unit variance Gaussians and initial hyperparameters were chosen to be small values.
2) The network was trained to minimise the cost function using the scaled conjugate gradient algorithm.
3) When the network training had reached a local minimum, the values of the hyperparameters were re-estimated according to equations (10) and (11).
4) Steps 2 and 3 were repeated until the cost function value was smaller than a pre-determined value or the maximum number of training iterations has reached.

3.3. Power Transformer Fault Classification

Power transformer faults can be classified by using DGA and BNNs. Firstly, the inputs of BNNs must be formed based on Doernenburg and Rogers ratios.

3.3.1. Doernenburg ratios

The input vector in this case is a vector with four elements as follows:

$$[x] = \left[\frac{CH_4}{H_2},\ \frac{C_2H_2}{C_2H_4},\ \frac{C_2H_2}{CH_4},\ \frac{C_2H_6}{C_2H_2}\right]^T$$

Different classification BNNs with different numbers of hidden nodes were trained using the training set. For a given number of hidden nodes, ten BNNs with different randomly initial weights and biases were trained and the log evidence was then evaluated. As shown in Fig. 3, the networks with two hidden nodes have the highest log evidence. Simultaneously, Fig. 4 also shows the highest overall accuracy of fault classification, which is equivalent to the corresponding highest log evidence in Fig. 3.

Table 5 shows the change of four hyper-parameters and the number of well-determined parameters. Table 6 is the confusion matrix of the optimised BNN for classifying the unknown input vectors and the overall accuracy of fault classification is 83.33%.

Table 5. The change of four hyper-parameters and the number of well-determined parameters according to hyper-parameter re-estimation periods (Doernenburg ratios).

Period	ξ_1	ξ_2	ξ_3	ξ_4	γ
1	0.022	0.044	0.008	0.409	18.555
2	0.039	0.083	0.006	0.753	15.803
3	0.061	0.134	0.005	0.865	15.451

Table 6. Confusion matrix of the BNN for classifying unknown input vectors (Doernenburg ratios).

		Predicted classification				
	Fault	PD	D1	D2	T1&T2	T3
Actual classification	PD	2	0	0	2	0
	D1	0	5	3	0	0
	D2	0	0	12	0	0
	T1&T2	0	0	0	5	1
	T3	0	0	0	0	6
Accuracy (%)			83.33			

3.3.2. Rogers ratios

The input vector in this case is a vector with four elements as follows:

$$[x]=\left[\frac{CH_4}{H_2},\ \frac{C_2H_2}{C_2H_4},\ \frac{C_2H_4}{C_2H_6},\ \frac{C_2H_6}{CH_4}\right]^T$$

Different BNN classifiers having different numbers of hidden nodes were trained using the training set. For a given number of hidden nodes, ten networks with different randomly initial weights and biases were trained and the log evidence was evaluated. As illustrated in Fig. 5, the networks with two hidden nodes can result in the highest log evidence. This network architecture can also give the highest overall accuracy of fault classification as shown in Fig. 6.

Table 7 shows the change of four hyper-parameters and the number of well-determined parameters. Table 8 is the confusion matrix of the optimised BNN for classifying the unknown input vectors and the overall accuracy of fault classification is 80.56%.

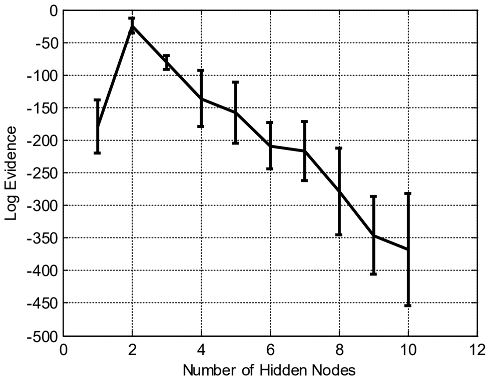

Fig. 3. Log evidence vs number of hidden nodes (Doernenburg ratios).

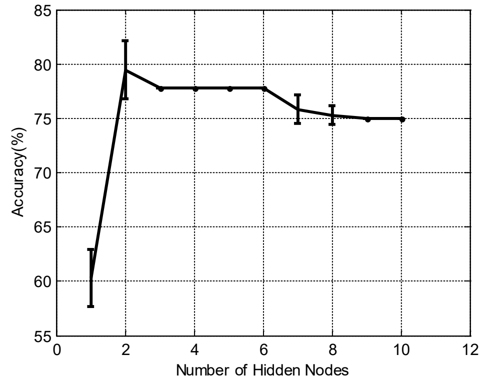

Fig. 4. Overall accuracy vs number of hidden nodes (Doernenburg ratios).

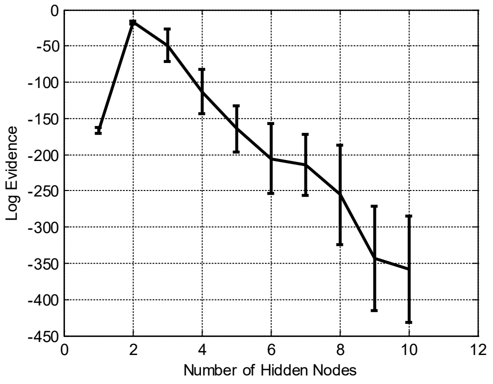

Fig. 5. Log evidence vs number of hidden nodes (Rogers ratios).

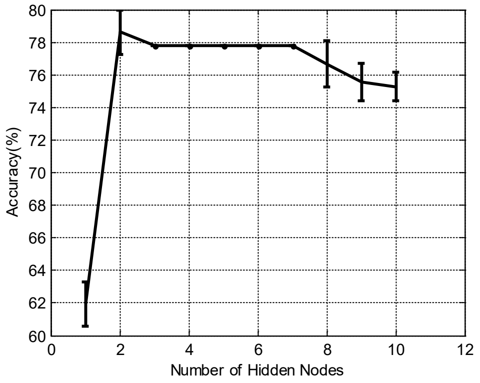

Fig. 6. Overall accuracy vs number of hidden nodes (Rogers ratios).

Table 7. The change of four hyper-parameters and the number of well-determined parameters according to hyper-parameter re-estimation periods (Rogers ratios).

Period	ξ_1	ξ_2	ξ_3	ξ_4	γ
1	0.026	0.012	0.009	0.268	18.645
2	0.039	0.015	0.007	0.353	16.315
3	0.053	0.02	0.005	0.333	15.801

Table. 8. Confusion matrix of the trained BNN for classifying unknown input vectors (Rogers ratios).

		Predicted classification				
	Fault	PD	D1	D2	T1&T2	T3
Actual classification	PD	2	0	0	2	0
	D1	0	4	4	0	0
	D2	0	0	12	0	0
	T1& T2	0	0	0	5	1
	T3	0	0	0	0	6
Accuracy (%)		80.56				

Table. 9. Accuracy comparison between suggested gas ratio limit and BNN based classification methods.

Doernenburg ratios method with suggested gas ratio limits	79.48 (%)
Doernenburg ratios method with BNN	83.33 (%)
Rogers ratios method with suggested gas ratio limits	40.17 (%)
Rogers ratios method with BNN	80.56 (%)

Table 9 is a comparison between suggested limit and BNN based methods in DGA with the same training data set. Obviously, the BNN based methods can significantly dominate over the suggested limit-based methods.

4. Conclusion

This paper presents the key steps in developing BNNs used for classifying oil-immersed power transformer faults using DGA. Based on the exploration of the Bayesian inference framework for MLP neural network training, the regularisation parameters (hyperparameters) and the appropriate number of hidden nodes in the network can be conveniently obtained. Specifically, the BNNs were trained on two common criteria of Doernenburg and Rogers gas ratios. It is shown that a BNN configuration based on a few nodes in the hidden layer is suitable for the incipient faut detection in power transformers. The number of hidden units mainly depends on the diagnosis criterion under consideration. When the BNNs with two hidden units were trained using the DGA data from the IEC TC 10 database, they can classify power transformer faults with overall accuracies greater than 80%. This research also performs a comparison between suggested gas ratio limit-based methods and BNN based methods for power transformer fault diagnoses. It is obvious that the BNN based method clearly dominates over the suggested gas ratio limit-based methods. The future work of this study is to perform a comparison between the BNNs and other machine learning classifiers for DGA of power transformers. In addition, various training algorithms for the BNN should be also investigated.

Acknowledgments

The authors would like to express very great appreciation to Professor Ian Nabney (University of Bristol, United Kingdom) for his valuable assistance during the exploration of the open-source Netlab software used for this research work. His willingness to give his time so generously has been also very much appreciated.

References

[1] M.Duval, A.dePabla, Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC 10 databases, IEEE Electrical Insulation Magazine, vol. 17, no. 2, pp. 31–41, Apr. 2001. https://doi.org/10.1109/57.917529

[2] Sung-wook Kim, Sung-jik Kim, Hwang-dong Seo, Jae-ryong Jung, Hang-jun Yang, Michel Duval, New methods of DGA diagnosis using IEC TC 10 and related databases Part 1: application of gas-ratio combinations, IEEE Trans. Dielectrics and Electrical Insulation, vol. 20, no. 2, pp. 685–690, May. 2013. https://doi.org/10.1109/TDEI.2013.6508773

[3] Osama E. Gouda, Salah Hamdy El-Hoshy, Sherif S. M. Ghoneim, Enhancing the diagnostic accuracy of DGA techniques based on IEC-TC10 and related databases, IEEE Access, vol. 9, pp. 118031–118041, Aug. 2021. https://doi.org/10.1109/ACCESS.2021.3107332

[4] Ibrahim B. M. Taha, Hatim G. Zaini, Sherif. S. M. Ghoneim, Comparative study between Dorneneburg and Rogers methods for transformer fault diagnosis based on dissolved gas analysis using Matlab Simulink Tools, 2015 IEEE Conference on Energy Conversion (CENCON), 2015, pp. 363–367.

[5] Jawad Faiz, Milad Soleimani, Assessment of computational intelligence and conventional dissolved gas analysis methods for transformer fault diagnosis, IEEE Trans. Dielectrics and Electrical Insulation, vol. 25, no. 5, pp. 1798–1806, Oct.2018. https://doi.org/10.1109/TDEI.2018.007191

[6] J.L. Guardado, J.L. Naredo, P. Moreno, C.R. Fuerte, A comparative study of neural network efficiency in power transformers diagnosis using dissolved gas

analysis, IEEE Transactions on Power Delivery, vol. 16, no. 4, pp. 643–647, Oct. 2001. https://doi.org/10.1109/61.956751

[7] Jiejie Dai, Hui Song, Gehao Sheng, Xiuchen Jiang, Dissolved gas analysis of insulating oil for power transformer fault diagnosis with deep belief network, IEEE Trans. Dielectrics and Electrical Insulation, vol. 24, no. 5, pp. 2828–2835, Oct. 2017. https://doi.org/10.1109/TDEI.2017.006727

[8] Q. Su, C. Mi, L.L. Lai, P. Austin, A fuzzy dissolved gas analysis method for the diagnosis of multiple incipient faults in a transformer, IEEE Trans. Power Systems, vol. 15, no. 2, pp. 593 – 598, May. 2000 https://doi.org/10.1109/59.867146

[9] Secil Genc, Serap Karagol, Fuzzy logic application in DGA methods to classify fault type in power transformer, 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2020. https://doi.org/10.1109/HORA49412.2020.9152896

[10] Seifeddine Souahlia, Khmais Bacha, Abdelkader Chaari, SVM-based decision for power transformers fault diagnosis using Rogers and Doernenburg ratios DGA, 10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13), 2013, pp. 1–6. https://doi.org/10.1109/SSD.2013.6564073

[11] Yuhan Wu, Xianbo Sun, Yi Zhang, Xianjing Zhong, Lei Cheng, A power transformer fault diagnosis method-based hybrid improved seagull optimization algorithm and support vector machine, IEEE Access, vol. 10, pp. 17268–17286, Nov. 2021. https://doi.org/10.1109/ACCESS.2021.3127164

[12] Arief Basuki, Suwarno, Online dissolved gas analysis of power transformers based on decision tree model, 2018 Conference on Power Engineering and Renewable Energy (ICPERE), 2018. https://doi.org/10.1109/ICPERE.2018.8739761

[13] Omar Kherif, Youcef Benmahamed, Madjid Teguar, Ahmed Boubakeur, Sherif S. M. Ghoneim, Accuracy improvement of power transformer faults diagnostic using KNN classifier with decision tree principle, IEEE Access, 2021, pp. 81693–81701. https://doi.org/10.1109/ACCESS.2021.3086135

[14] Y. Benmahamed, Y. Kemari, M. Teguar, A. Boubakeur, Diagnosis of power transformer oil using KNN and naive Bayes classifiers, 2018 IEEE 2nd International Conference on Dielectrics (ICD), 2018. https://doi.org/10.1109/ICD.2018.8514789

[15] Wenxiong Mo, Tusongjiang Kari, Hongbing Wang, Le Luan, Wensheng Gao, Fault diagnosis of power transformer using feature selection techniques and KNN, 2017 3rd IEEE International Conference on Computer and Communications (ICCC), 2017, pp. 2827–2831.

[16] D. Mackay, A practical Bayesian framework for backpropagation networks, Computation and Neural Systems, vol. 4, pp. 448–472, 1992. https://doi.org/10.1162/neco.1992.4.3.448

[17] Ian T. Nabney, Netlab: Algorithms for pattern recognition, Advances in Pattern Recognition, Springer, 2001.

[18] W.D. Penny, S.J. Robert, Bayesian neural networks for Classification: how useful is the evidence framework, Neural Networks, vol. 12, pp. 877–892, 1999. https://doi.org/10.1016/S0893-6080(99)00040-4