Electrical Power and Energy Systems 67 (2015) 324–335

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Design and evaluation of a hybrid system for detection and prediction of faults in electrical transformers

Samaher Al-Janabi [a,*], Sarvesh Rawat [b], Ahmed Patel [c,d], Ibrahim Al-Shourbaji [e]

[a] Department of Information Networks, Faculty of Information Technology, University of Babylon, Babylon 00964, Iraq
[b] School of Electronics and Electrical Engineering (SELECT), VIT University, Vellore 632014, India
[c] School of Computer Science, Centre of Software Technology and Management (SOFTAM), Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor Darul Ehsan, Malaysia
[d] School of Computing and Information Systems, Faculty of Science, Engineering and Computing, Kingston University, Kingston Upon Thames KT1 2EE, United Kingdom
[e] Computer Network Department, Computer Science and Information System College, Jazan University, Jazan, Saudi Arabia

ARTICLE INFO

Article history:
Received 17 May 2014
Received in revised form 4 November 2014
Accepted 1 December 2014

Keywords:
Dissolved Gas-in-oil Analysis (DGA)
Electrical transformer
Fault detection
Fault prediction
Genetic algorithm
Neural network

ABSTRACT

Transformers are the vital parts of an electrical grid system. A faulty transformer can destabilize the electrical supply along with the other devices of the transmission system. Due to its significant role in the system, a transformer has to be free from faults and irregularities. Dissolved Gas-in-oil Analysis (DGA) is a method that helps in diagnosing the faults present in an electrical transformer. This paper proposes a hybrid system based on Genetic Neural Computing (GNC) for analyzing and interpreting the data derived from the concentration of the dissolved gases. It is further analyzed and clustered into four subsets according to the standard C57.104 defined by IEEE using genetic algorithm (GA). The clustered data is fed to the neural network that is used to predict the different types of faults present in the transformers. The hybrid system generates the necessary decision rules to assist the system's operator in identifying the exact fault in the transformer and its fault status. This analysis would then be helpful in performing the required maintenance check and plan for repairs.

Introduction

A transformer is one of the most crucial element of an Electrical Power Transmission System (EPTS). A fault in the transformer can introduce major problems for the consumers as well as for the maintenance engineers. Many incidents have taken place in the past few years that greatly disrupted the electrical transmission system. One such catastrophe occurred in New Jersey, USA, in December 2013, where, approximately 12,000 people lost their power supply due to a fault in the transformer [10]. Another major incident took place on February 2014 in Stamford, USA, where a transformer caught fire rendering more than 1000 people without light for days [20]. In the year 2000, a disastrous loss was reported at another power plant, where a $86 million US dollars business was interrupted due to a faulty transformer [12].

There is an urgent need of a prefailure analysis and protection system that can protect the transformers from any kind of liabilities. Analysis of the transformer's dielectric oil is the classical and reliable method used for checking the irregularities present in the transformers by using the Dissolve Gas-in-oil Analysis (DGA) method. Several gases are generated during the normal operation of a transformer. The ratio and concentration of certain gases facilitate the operator in the detection and prediction of the indiscretion and problems that exists in the transformers. The main gases responsible for the faults are methane (CH_4), acetylene (C_2H_2), ethane (C_2H_6), and ethylene (C_2H_4) [13]. Problems like corona discharge, overheating, and arcing in the transformers are easily detected by DGA.

There are several methods available to analyze the faults, such as the (i) International Electro technical Commission (IEC) ratio method, (ii) Rogers ratio method, (iii) Doernenburg method, (iv) Duval triangle method, and the Key gas method. The first three methods do not give any sort of quantitative indication of the fault. In many cases, where multiple faults occur, gases produced from different types of faults are mixed up, creating confusing ratios among the various components of the gases. For our analysis, we will follow the IEEE standard C57.104, based on the Total Dissolve Concentration of Gases (TDCG) and the Key gas method. It measures the concentration of each fault gas produced in the transformer

* Corresponding author.
E-mail addresses: Samaher@itnet.uobabylon.edu.iq (S. Al-Janabi), sss.sarvesh888@gmail.com (S. Rawat), whinchat2010@gmail.com (A. Patel), i_shurbaji@yahoo.com (I. Al-Shourbaji).

http://dx.doi.org/10.1016/j.ijepes.2014.12.005
0142-0615/

during a fault. In this method, the individual concentration of each gas is measured rather than the ratio which is the basic principle of this method. The use of DGA in the transformer is widely accepted for analyzing and spotting the faults as it can diagnose the degradation of the transformer and can estimate its life efficiency [16]. In addition, it can appraise the internal situation of the transformer and plays a crucial part of the maintenance checking and testing system.

Soft computing is a consortium of methodologies that works synergistically and provides, in one form or another, flexible information processing capability for handling real-life ambiguous situations. It aims to exploit the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness, and low-cost solutions. The guiding principle is to devise methods of computation that leads to an acceptable solution. Several methods have been devised for using Artificial Intelligence (AI) and Soft Computing (SC) for more advanced and accurate diagnosis of transformers [4,17]. In 2012, Souahlia et al. used fuzzy logic, Support Vector Machine (SVM) and Neural Networks (NN) for fault diagnosis in the transformers [18]. Way back in 1997, Huang et al. showed the use of fuzzy logic for diagnosing the faults in the transformer [22]. A set of induced rules was generated from a quantitative data using a fuzzy set based learning algorithm [15]. But the membership function used in fuzzy is not suitable for representing the boundary value conditions [5,6]. In 2005 Ganyun et al. used SVM for identifying the faults in the transformers [19]. It provides a three layered classifier for classifying the state of the transformer. Although it showed a good reliability and is suitable for online fault diagnosis, but the selection of the exact kernel function and the optimization of parameters to make a SVM classifier is a typical problem. The main problem with all these methods is that they are mostly suitable for a transformer having a single fault or any dominating fault. There is no application focusing on the prediction of faults and real trend analysis.

There are several problems associated with an electrical transformer, such as, overloading, overvoltage, overheating and other factors that ultimately lead to a permanent failure. As such, there is a major need of monitoring the parameters associated with the transformer to prevent it from shutting down. Therefore, there is an acute need of new technologies which can monitor the supply systems more effectively to prevent them from unexpected and unconditional failures. *Soft Computing* (SC) hybridization is an association of computing methodologies centering on *Fuzzy Logic* (FL), *Neural Computing* (NC), *Genetic Computing* (GC), *Probabilistic computing* (PC) and their hybridization [1–3]. Collectively, these methodologies provide a foundation for the conception, design and deployment of the intelligent systems. The basic idea underlying SC is that its constituent methodologies are, for the most part, complementary rather than competitive. The complementarity of the constituents of soft computing implies that their effectiveness may be enhanced by using them in combination rather than isolation. At this juncture, the most visible systems of this combined type are the neuro-fuzzy systems. Less visible, but potentially of equal importance are the fuzzy-genetic systems. Each of the constituents of soft computing has a set of capabilities to offer. In the case of fuzzy logic, it is the machinery for dealing with imprecision, information granulation and computing with words. For this purpose, the principal tools are provided by the fuzzy logic center on the use of linguistic variables and the calculation of fuzzy based "*if-then*" rules. In the case of genetic computing, the principal tool is a systematized random search. The most known methods of hybridization of these tools are (i) Neural-Fuzzy Computing, (ii) Fuzzy Genetic Computing, (iii) Genetic-Neural Computing (iv) and Neuro-Genetic-Fuzzy Computing.

In this work, we have used Genetic-Neural Computing using DGA analysis, where the challenge is to build a practical neural network choosing the right architecture and the right learning parameters to find the faults present in the transformers [13]. We know that the Multilayer Perceptron (MLP) with one hidden layer, using the sigmoid transfer function, could perform any mapping from a set of inputs to the desired outputs. Unfortunately, this tells us nothing about the learning parameters, the necessary number of neurons, or whether any additional layers would be beneficial. It is, however, possible to use a genetic algorithm to optimize the network design. A suitable cost function might combine the root mean square error with the duration of training [2]. Supervised training of a neural network involves adjusting its weights until the output patterns are obtained for a range of input patterns. They must be as close as possible to the desired patterns. The different network topologies use different training algorithms for achieving this weight adjustment, typically through back-propagation or errors. However, it is also possible to use GA for training the network. This can be achieved by allowing each gene to represent a network weight so that a complete set of network weights is mapped onto an individual chromosome. Each chromosome can be evaluated by testing a neural network with the corresponding weights against a series of test patterns. A fitness value can be assigned according to the error so that the weights represented by the fittest generated individual corresponds to a trained neural network [3–5]. The most crucial part of using neural network in our system lies in the fact that it can learn and update its knowledge whenever it is required [8,9]. It offers a far superior performance than the other systems due to the non-linear mapping property of the neurons. Following this model, the operator will be able to conduct prefailure analysis and plan for the required maintenance checks.

The rest of the paper is structured as follows: Section 'Cause of gas formation' presents the cause of gas formation. Section 'Need of a hybrid system' presents the main tools used in the hybrid system, while in Section 'Main stages of the suggested hybrid system', the suggested hybrid system that contains various stages are explained. Section 'Experiment' shows the experiments. Finally, the conclusion of the paper is presented in Section 'Conclusion'.

Cause of gas formation

The main and the most profound cause of gas formation in the transformer is thermal heating and electrical discharges. It decomposes the oil into different gases like CO, CO_2, C_2H_2, C_2H_4, C_2H_6, H_2, and CH_4. The cellulose and the minerals present in the transformer oil decompose to produce these gases as shown in Fig. 1. The decomposition of cellulose produces carbon oxides, methane and some hydrogen. The rate of production of these gases abruptly increases with the increase in temperature and volume of the material present in the oil.

Beta fluid and mineral oil consist of a variety of hydrocarbon molecules. They decompose into active hydrogen atoms and

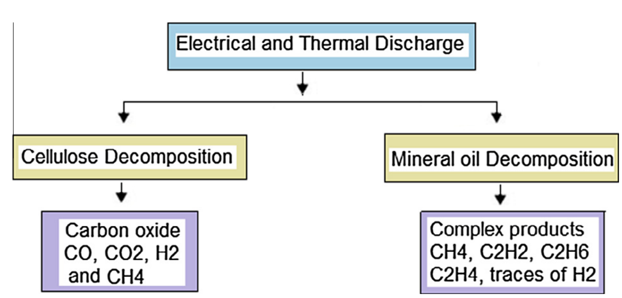


Fig. 1. Composition of the gases evolved during a normal functioning of a transformer.

fragments of hydrocarbons which combine to form new molecules. The further rearrangement and decomposition of molecules lead to the formation of other gases like acetylene and ethylene. The concentration of these gases is analyzed by the DGA covered in the next section. It has to be monitored on a regular basis so that the inconsistencies in the transformer can be scrutinized properly [14]. Table 1 shows the principle gas evolved during the thermal and electrical decomposition of the beta fluid and cellulose.

Need of a hybrid system

The conventional methods, like the IEC ratio method, Rogers ratio, Doernenburg method and the Key gas method highly depend on human expertise and skills of the operator. The operator has to thoroughly inspect the concentration of the gases. He is required to compare the output results from the different methods to derive a conclusion. So, a huge expertise is needed for the operator to analyze the results and avoid the conflicts. Sometimes, the possible number of different combinations of codes exceeds the fault types. Thus, the traditional DGA methods do not offer any absolute or objective type of result. AI based fault diagnosis can become an additional asset here. The aim of the proposed system is to draw the conclusions for the system's operator by analyzing the state of the transformer, so that he can take further steps and can plan for maintenance [11]. NN and GA have been widely used in solving many real time problems [9]. The whole system is adaptive in nature. NN can successfully reveal the explicit relationship between the non-linear input–output data. It can find the patterns from the input training data and can increase its learning and adaptability for the new set of obtained data. The adopted method is more effective and acclimative as compared to the conventional method of fault diagnosis. It can produce more efficient results showing better performance than the other methods. The proposed network following the least error function, can exclaim the best possible guess about the functionality of the transformer under a given condition. The most significant advantage of using this method is that it eliminates the boundary type problems which results in the *"No Decision"* type cases that are mostly found in conventional methods. The system can autonomically directly self-learn from the input variables and update itself according to its necessity.

Fig. 2 shows the basic steps that are followed in the proposed system. There are 4 basic steps that are involved in the whole process. The first step includes the analysis of the transformer oil and finding the concentration of the different gases present in it [21]. The second step features the data pre-processing unit and the use of GA for clustering the concentration of the different gases. These gases are clustered on the basis of four conditions of the standard C57.104 defined by IEEE [7]. In the third step. ANN is used to predict the value of the fault using the derived clusters of GA. Finally, the decision rules are generated for the system's operator that are inspected and analyzed by using different statistical techniques.

Tools used in the hybrid system

This section discusses the main tools that are used for building the hybrid system.

Table 1
Principal gas evolved during a fault.

Decomposition	Thermal		Electrical	
Fault	Overheating of oil	Overheating of cellulose	Corona discharge	Arcing
Principle Gas	Ethylene	Carbon monoxide	Hydrogen	Acetylene

A. Dissolved Gas-In-Oil Analysis

DGA is one of the most important diagnostic tests performed on the transformer oil in order to determine the state of the power transformer [15]. We can also detect very low concentration levels of the harmful gases [14]. Fig. 3 shows the process of DGA that is used for analyzing the concentration of the gases.

This technique involves the stripping of gases from transformer oil and infusing them into a gas chromatograph. A sample of the oil is taken using a gas tight syringe of appropriate capacity. This syringe is capable of taking a sample of the oil from the main stream point of the transformer. It is stored in a dark enclosure to prevent the oxidation of gases. The next phase includes the extraction of gases from the sample. In the final step, the sample is subjected to gas chromatography. This is used for separating the different constituents of the gases from a mixture. Fig. 4 shows the whole process involved in the gas chromatography.

The use of DGA in the transformer is widely accepted for analyzing and spotting the faults as it can diagnose the degradation of the transformer and can estimate its life expectancy. In addition, it can appraise the internal situation of the transformer and is a crucial part of the maintenance checking and testing system.

B. Genetic algorithms

Genetic algorithms (GAs) are a heuristic approach used to find approximate solutions for the problems that are difficult to solve by applying the principles of evolutionary biology to computer science. Genetic algorithms use biologically-derived techniques such as inheritance, mutation, natural selection, and recombination (or crossover). Genetic algorithms are a particular class of evolutionary algorithms.

GAs are typically implemented as a computer simulation in which a population of abstract representations (*called chromosomes*) of candidate solutions (*called individuals*) to an optimization problem evolving towards better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but different encodings are also possible. The evolution starts from a population of completely random individuals and happens in generations. In each generation, the fitness of the whole population is evaluated, multiple individuals are stochastically selected from the current population based on their fitness and modified mutated or recombined to form a new population, which becomes current in the next iteration of the algorithm.

Main stages of the suggested hybrid system

Soft computing methodologies have been applied to handle the different challenges posed by a database. The main constituents of soft computing, in this paper, include Detection, GA and NN. Each of them contributes a distinct methodology to address the problems in its domain. This is done in a cooperative, rather than a competitive, manner. The result is a more intelligent and robust system providing a human-interpretable, low cost, approximate solution, as compared to the traditional techniques.

Stage 1: fault detection

Every transformer generates certain gases during its operation. The generation of the combustible gases is a result of various factors like overheating, corona discharge and dielectric problems. These associated abnormalities are termed as faults. For example, when cellulose is overly heated it produces hydrogen (H_2), methane (CH_4), carbon dioxide (CO_2) and carbon monoxide (CO). Gases like ethane (C_2H_6), acetylene (C_2H_2), and ethylene (C_2H_4) are produced in beta fluid by internal faults. The presence of these gases indicates the occurrence of one or more combination of these

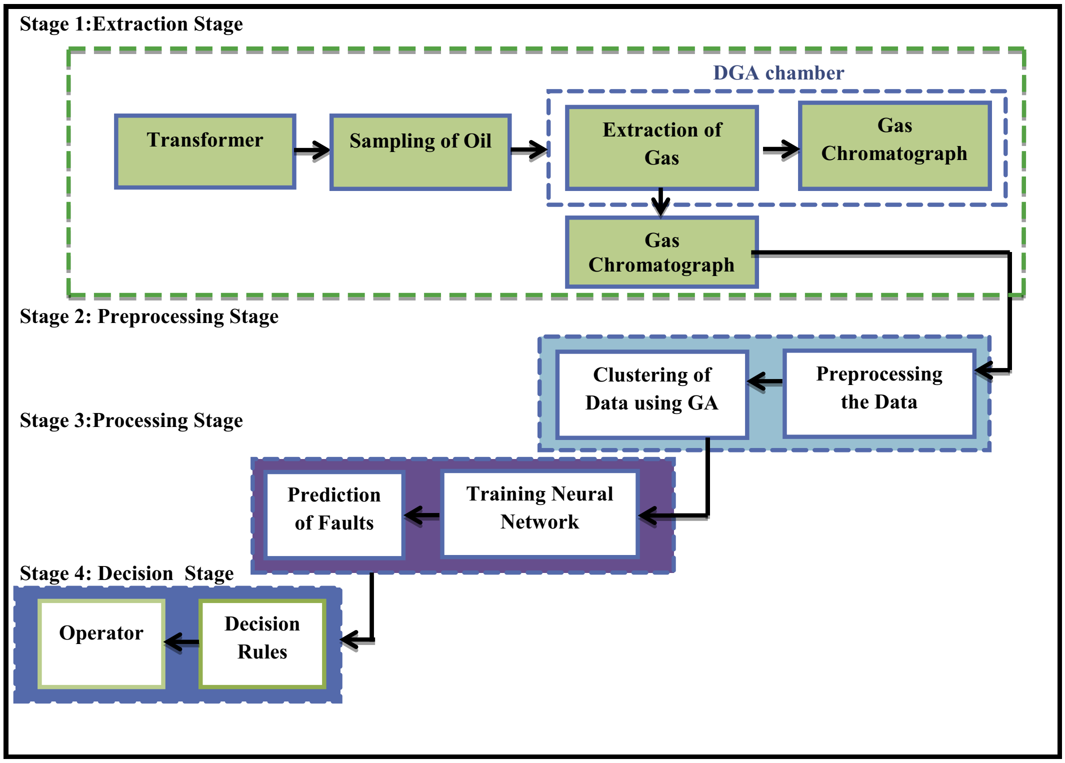


Fig. 2. Proposed hybrid architecture for fault diagnosis.

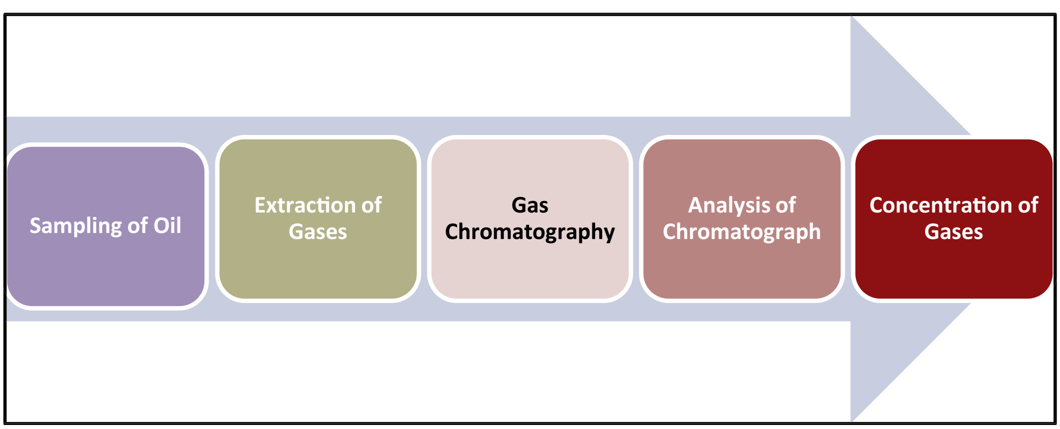


Fig. 3. Steps followed in finding the concentration of the gases.

(electrical, corona or thermal) faults. The concentration of all the gases is determined by the gas chromatography [21]. The whole analysis results in categorizing the fault as either a thermal fault or an electrical fault. It is further classified according to the high and low intensity of the faults:

- Thermal faults generally produce gases of low molecular weight like H_2, CH_4 and small quantities of other compounds having higher molecular weight, namely acetylene, comprising of all the mineral oils and beta fluid. On the other hand, thermal decomposition of cellulose produces carbon dioxide (CO_2) and carbon monoxide (CO).
- Electrical faults of low intensity such as intermittent arcing and partial discharge, mainly produce hydrogen (H_2) along with small quantities of acetylene (C_2H_2) and methane (CH_4). The concentration increases with respect to the intensity of the discharge.
- In the case of electrical faults of high intensity or arcing, a large amount of acetylene becomes predominant in the system. The temperature of the system exceeds 700 C.

By measuring the concentration of the gases, we can identify the kind of fault involved, as shown in Table 2.

Stage 2: pre-processing of the gas database

Fault diagnosis is generally considered as a boundary set problem as the dataset consists of many inconsistencies. In this scenario, training a neural network is very difficult. As such, there is a huge need of pre-processing the data before feeding it to the NN. The extracted database from the above stage is pre-processed using a Linear Transformation method as follows:

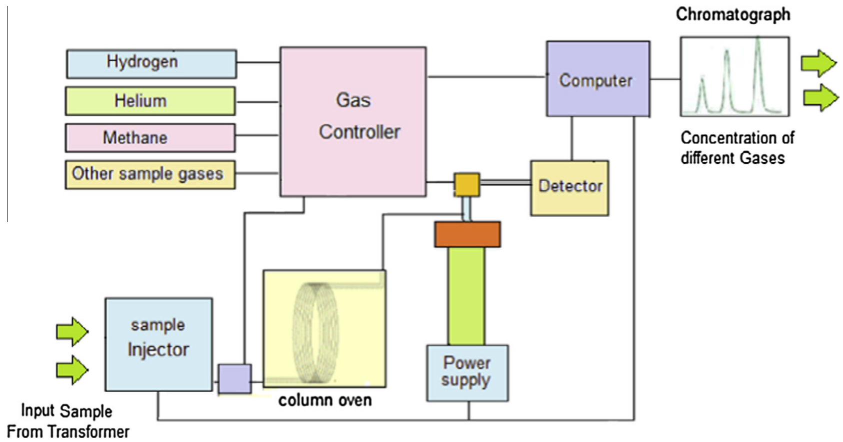


Fig. 4. Gas chromatography for DGA analysis.

Table 2
Categorization of fault gases.

Corona		Pyrolysis				Arching
Oil	Cellulose	Oil		Cellulose		
		Low temperature	High temperature	Low temperature	High temperature	H_2 C_2H_2 (CH_4 C_2H_6 C_2H_4)
H_2	H_2 CO CO_2	CH_4 C2H6	C_2H_4 H_2 (CH_4, C_2H_6)	CO_2 (CO)	CO (CO_2)	

Here, $L' = [(L - \min)/(\max - \min)] * (\max' - \min') + \min$.

where min is the old minimum value, min′ is new minimum value, max is the old maximum value and max′ is the new maximum value.

Stage 3: genetic algorithm for clustering the database according to standard C57.104 defined by IEEE

In this step, GA is applied to find the number of clusters existing in the Gas database (i.e. find the best seed for each cluster and the number of pixels on it). Before this, we need to determine the parameters of GA, such as the population size, minimum number of cluster, selection, and the crossover methods. Fig. 5 shows the flowchart of GA for clustering the Gas Database.

Stage 3.1: representation (encoding of solution)

The chromosomes are made up of list pointers. If the pointer at any gene is not null, that means there is a supposed center. This center is drawn randomly from the data set. On the other hand, gene (pointer) with null mean, has had no center encoded in it. The value of K is assumed to lie in the range [$K_{\min}$; $K_{\max}$], where $K_{\min}$ is chosen to be 2 unless specified otherwise. The length of a string is taken to be $K_{\max}$, where each individual gene position represents either a pointer to the actual center or a null.

Stage 3.2: population initialization

For each string i in the population ($i = 1, \ldots\ldots, P$, where P is the size of the population), a random number Ki in the range [$K_{\min}$–$K_{\max}$] is generated. This string is assumed to encode the centers (each center represents a weight of node of Back-Propagation Neural Network) (BPNN) of Ki clusters. For initializing these centers, Kid points are chosen on the basis of the four conditions from the dataset. These points are distributed randomly in the chromosome.

Stage 3.3: fitness computation [23]

The fitness of a chromosome is computed using the Davies–Bouldin index. This index is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. The scatter within C_i, the ith cluster, is computed as:

$$S_{i,q} = \left(\frac{1}{|C_i|} \sum_{x \in C_i} \{ \|x - z_i\|_2^q \} \right)^{1/q}$$

where z_i is the centroid of C_i, and is defined as:

$$Z_i = 1/n_i \sum_{x \in C_i} x$$

and n_i is the cardinality of C_i (i.e., the number of points in cluster C_i). The distance between cluster C_i and C_j is defined as:

$$d_{ij,t} = \left[\sum_{s=1}^{p} |z_{is} - z_{js}|^t \right]^{1/t} = \|z_i - z_j\|_t$$

Specifically, $S_{i,q}$ used in this article, is the average Euclidean distance of the vectors in class i to the centroid of class i. While $d_{ij,t}$ is the Minkowski distance of order t between the centroids that characterize clusters i and j (i.e., in this work, we use $t = 4$). Subsequently, we compute:

$$R_{i,qt} = \max_{j,j \neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{ij,t}} \right\}$$

The Davies–Bouldin (DB) index is then defined as:

$$DB = \frac{1}{K} \sum_{i=1}^{k} R_{i,qt}$$

The objective is to minimize the DB index for achieving proper clustering. The fitness function for chromosome j is defined as $1/DB_j$.

Fig. 5 shows the flowchart of the GA method used for clustering the gases database.

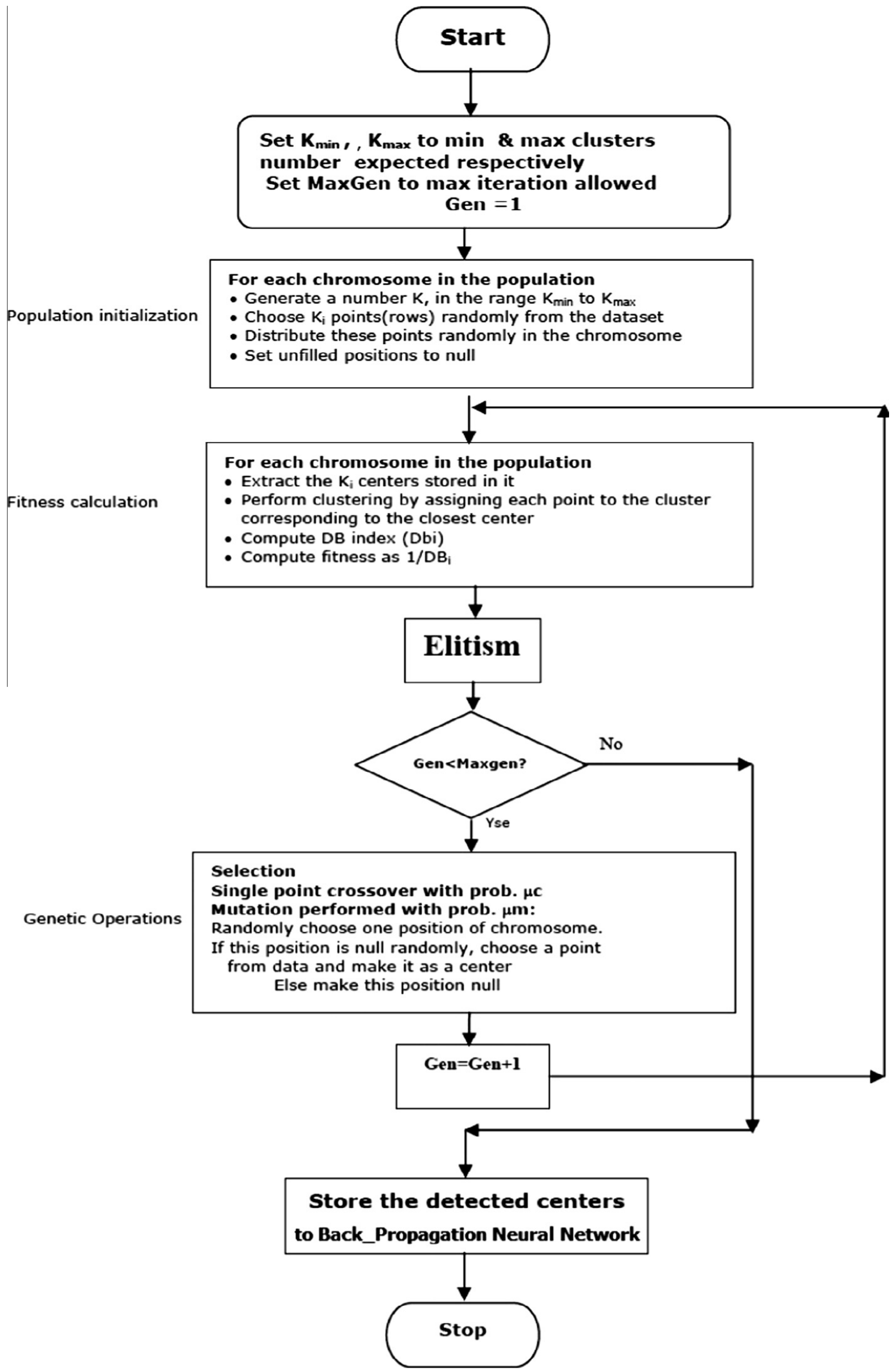


Fig. 5. Flowchart of genetic algorithm for clustering.

Stage 4: applying the Back-Propagation Neural Network (BPNN) to predict the fault values

The following main steps are executed to train the BPNN [24]:

Step 4.1: Input initial values to learning rate (η_0), maximum acceptable error to network ($E_{\max}$), maximum number of epochs to learning network (E_{pochmax}), momentum rate (α).

Step 4.2: Put the network error value (MSE) equal to zero and current training pattern error equal to one and determine the learning rate value.

Step 4.3: Compute the hidden neurons activity by unipolar sigmoid function, with $\lambda = 1$, according to the equation below:

$$h_k = f\left(\sum_{i=1}^{ns} s_i . v_{ik}\right) \quad \text{where} \quad k = 1, 2, \ldots\ldots, n_h.$$

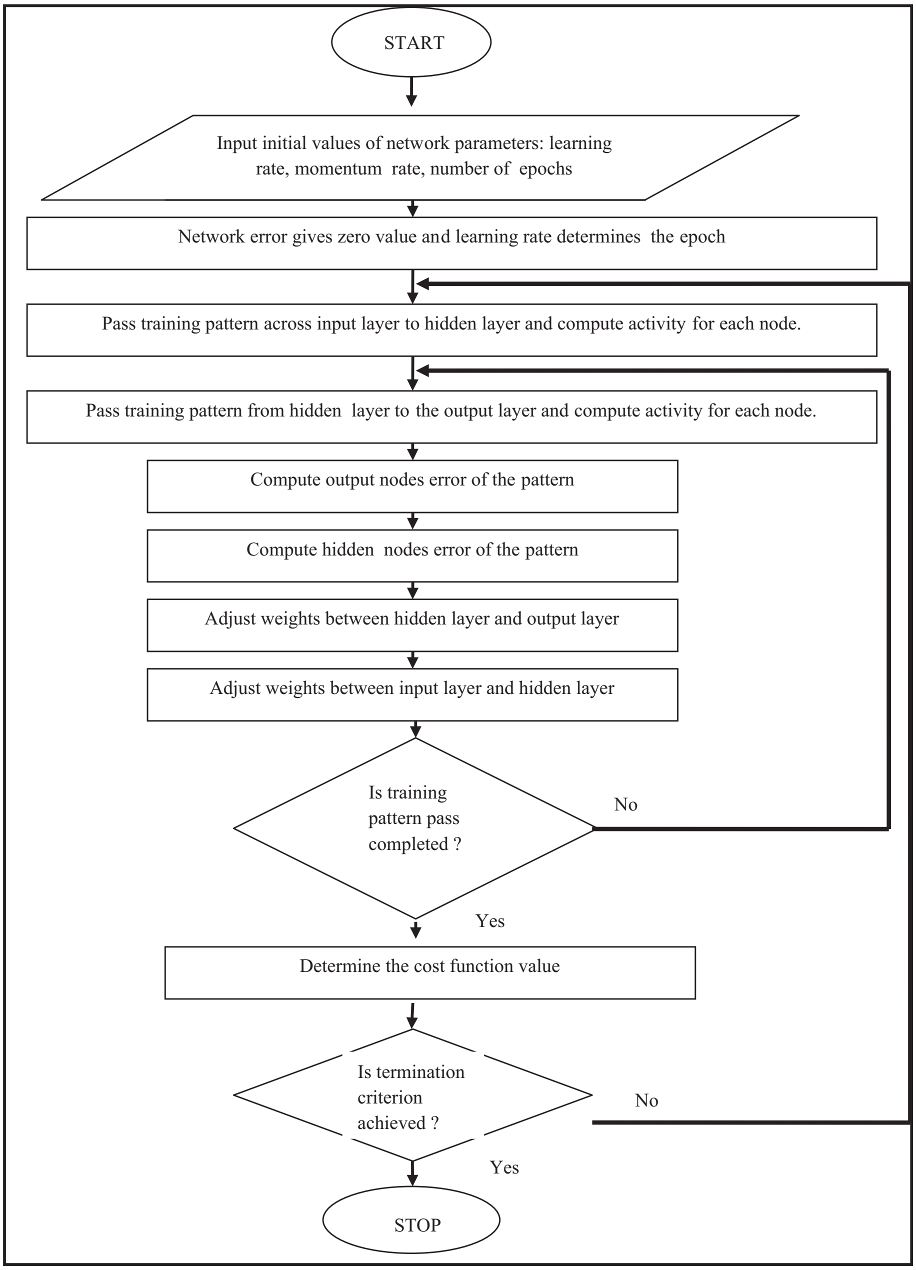


Fig. 6. Flowchart of BPNN for forecasting the fault value [24].

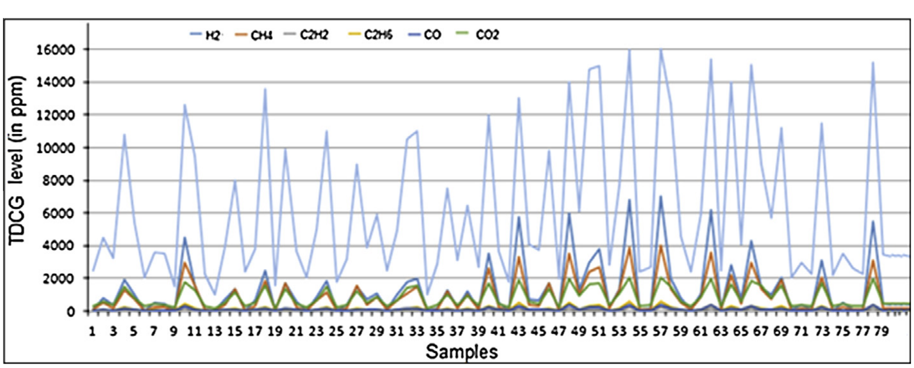


Fig. 7. The concentration of all the gases present in the transformer.

Step 4.4: Compute output neuron activity according to the following function:

$$o_j = f\left(\sum_{k=1}^{nh} h_k.w_{kj}\right) \quad \text{where} \quad j = 1, 2, \ldots\ldots, n_o.$$

Step 4.5: Compute error signal value to output neurons of pattern p according to the following equation:

$$\delta_j = (d_j - o_j).\bar{f}(net_j)$$

we can find the derivative of function as follows:

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

$$\bar{f}(net_j) = o_j.(1 - o_j), \quad \text{where} \quad j = 1, 2, \ldots, n_o,$$

Step 4.6: Compute the error signal value in hidden neurons which depends on the output neurons error:

$$\delta_k = \sum_{j=1}^{no} (\delta_j.w_{kj}).\bar{f}(net_k), \quad \text{where} \quad k = 1, 2, \ldots, n_h$$
$$\bar{f}(net_k) = h_k.(1 - h_k)$$

Step 4.7: Adjust weights between the hidden layer and the output layer. To do this, error back propagation algorithm uses a negative first derivative of the cost function ratio to weight as follows:

$$\begin{aligned}
\Delta w_{kj} &= -\eta_o.\frac{\partial E}{\partial w_{kj}} \\
&= -\eta_o.\frac{\partial\left(0.5 * \sum_{j=1}^{no} (d_j - o_j)^2\right)}{\partial w_{kj}}, o_j = f(net_j) \\
&= -\eta_o.\frac{\partial\left(0.5 * \sum_{j=1}^{no} (d_j - f(net_j))^2\right)}{\partial w_{kj}}, net_j = \sum_{k=1}^{nh} w_{kj}.h_k \\
&= \eta_0.(d_j - o_j)\frac{\partial f(net_j)}{\partial w_{kj}} \\
&= \eta_o.(d_j - o_j)\frac{\partial f(net_j)}{\partial net_j}.\frac{\partial net_j}{\partial w_{kj}} \\
&= \eta_o.(d_j - o_j).\bar{f}(net_j).\frac{\partial net_j}{\partial w_{kj}} \\
&= \eta_o.(d_j - o_j).\bar{f}(net_j).h_k \\
&= \eta_o.\delta_j.h_k
\end{aligned}$$

The adjustment equations:

$$\Delta w_{kj}^{(t+1)} = \eta.\delta_j.h_k + \alpha.\Delta w_{kj}^{(t)},$$
$$w_{kj}^{(t+1)} = w_{kj}^{(t)} + \Delta w_{kj}^{(t+1)}$$

where $k = 1, 2, \ldots, n_h$ and $j = 1, 2, \ldots, n_o$, and α is the momentum rate which is:

$\Delta w_{kj}^{(t)}$: that represent the difference between the current weight and the prior weight.

Step 4.8: Adjust weights between the input layer and the hidden layer as follows:

$$\begin{aligned}
\Delta v_{ik} &= -\eta_o.\frac{\partial E}{\partial V_{ik}} \\
&= -\eta_o.\frac{\partial\left(0.5 * \sum_{j=1}^{no} (d_j - o_j)^2\right)}{\partial v_{ik}} \\
&= \eta_o.\sum_{j=1}^{no} (d_j - o_j)\frac{\partial f(net_j)}{\partial v_{ik}} \\
&= \eta_o.\sum_{j=1}^{no} (d_j - o_j)\frac{\partial f(net_j)}{\partial net_j}.\frac{\partial net_j}{\partial v_{ik}} \\
&= \eta_o.\sum_{j=1}^{no} (d_j - o_j).\bar{f}(net_j).\frac{\partial net_j}{\partial v_{ik}} \\
&= \eta_o.\sum_{j=1}^{no} \delta_j.\frac{\partial net_j}{\partial h_k}.\frac{\partial h_k}{\partial v_{ik}} \\
&= \eta_o.\sum_{j=1}^{no} \delta_j.w_{kj}\frac{\partial h_k}{\partial net_k}.\frac{\partial net_k}{\partial v_{ik}}, \quad \text{where} \quad net_k = \sum_{i=1}^{ns} v_{ik}.s_i \\
&= \eta_o.\sum_{j=1}^{no} \delta_j.w_{kj}.\bar{f}(net_k).\frac{\partial net_k}{\partial v_{ik}} \\
&= \eta_o.\sum_{j=1}^{no} \delta_j.w_{kj}.\bar{f}(net_k).s_i \\
&= \eta_o.\sum_{j=1}^{no} \delta_j.w_{kj}.h_k(1 - h_k).s_i, \quad \text{where} \quad \bar{f}(net_k) = h_k.(1 - h_k) \\
&= \eta_o.\delta_k.s_i, \quad \text{where} \quad \delta_k = \sum_{j=1}^{no} (\delta_j.w_{kj}).h_k(1 - h_k)
\end{aligned}$$

The adjustment equations are:

$$\Delta v_{ik}^{(t+1)} = \eta_o.\delta_k.s_i + \alpha.\Delta v_{ik}^{(t)},$$
$$v_{ik}^{(t+1)} = v_{ik}^{(t)} + \Delta v_{ik}^{(t+1)}$$

where $k = 1, 2, \ldots, n_h$ and $i = 1, 2, \ldots, n_s$, and α is the momentum rate:

$\Delta v_{ik}^{(t)}$: represent the difference between the current weight and the prior weight.

Step 4.9: Increase the value p by one to input the next pattern in the learning process. If it does not reach to the maximum number of training the patterns then return to step 3 to train the network on that pattern else transform to step 10.

Step 4.10: After completing the input to all training patterns of the network, compute the cost function value that is represented by the mean square error:

$$MSE = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{no}\left(d_j^p - o_j^p\right)^2$$

Step 4.11: In this step, the termination criterion is tested. This condition is valid if the total error value of the network becomes less than the expected error of it ($E_{\max}$), or the current Epoch value (t) is bigger than the maximum number of learning epochs (E_{pochmax}). Else, return to step 2.

Fig. 6 explains the flowchart of BPNN for forecasting/predicting the fault values.

Stage 5: decision making process: rule generation

After verification of one of the stopping criteria to the BPNN algorithm, such as the verified cost function condition or exceeding the number of epochs to the maximum number of learning epochs without reaching a network error to a value less than the required value, we can say that the BPNN is complete.

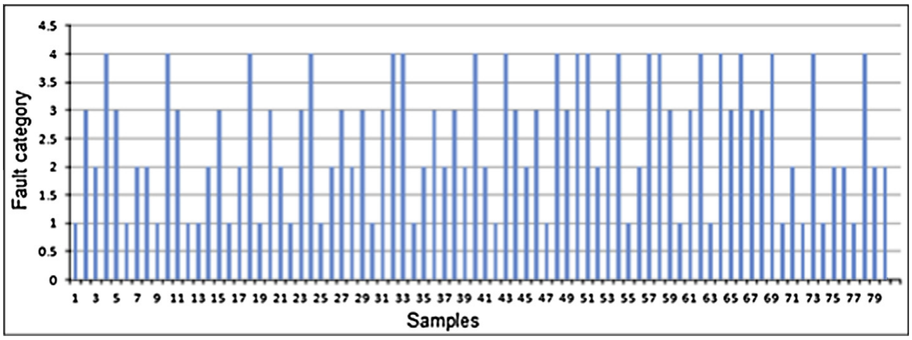
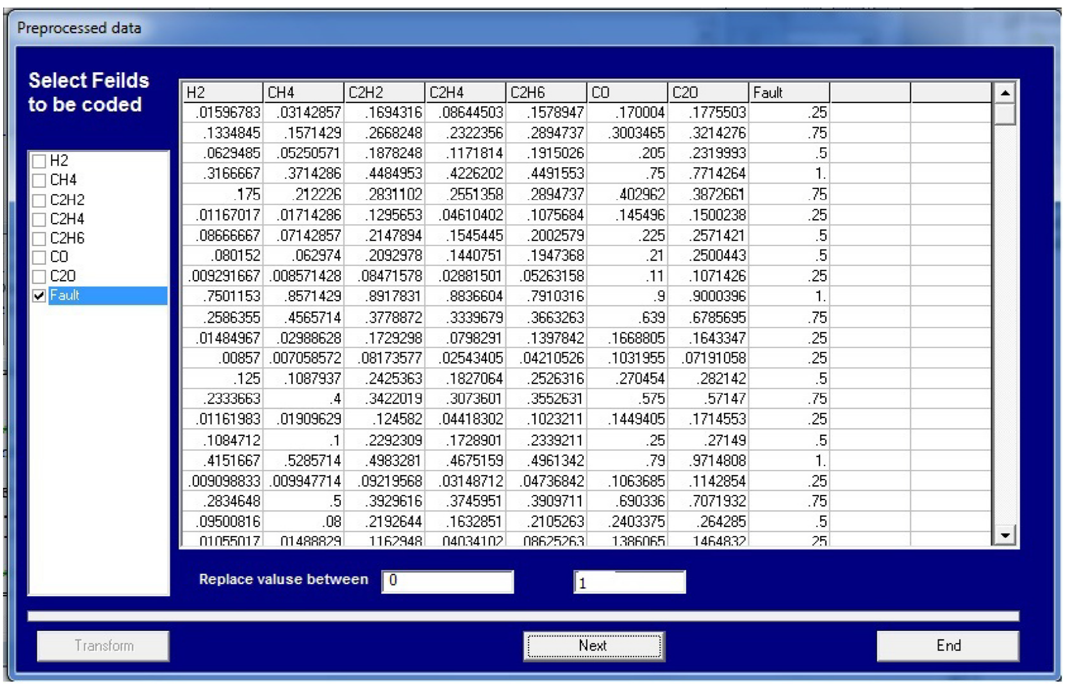


Fig. 8. Associated faults of the transformer.



H2	CH4	C2H2	C2H4	C2H6	CO	C2O	Fault
.01596783	.03142857	.1694316	.08644503	.1578947	.170004	.1775503	.25
.1334845	.1571429	.2668248	.2322356	.2894737	.3003465	.3214276	.75
.0629485	.05250571	.1878248	.1171814	.1915026	.205	.2319993	.5
.3166667	.3714286	.4484953	.4226202	.4491553	.75	.7714264	1.
.175	.212226	.2831102	.2551358	.2894737	.402962	.3872661	.75
.01167017	.01714286	.1295653	.04610402	.1075684	.145496	.1500238	.25
.08666667	.07142857	.2147894	.1545445	.2002579	.225	.2571421	.5
.080152	.062974	.2092978	.1440751	.1947368	.21	.2500443	.5
.009291667	.008571428	.08471578	.02881501	.05263158	.11	.1071426	.25
.7501153	.8571429	.8917831	.8836604	.7910316	.9	.9000396	1.
.2586355	.4565714	.3778872	.3339679	.3663263	.639	.6785695	.75
.01484967	.02988628	.1729298	.0798291	.1397842	.1668805	.1643347	.25
.00857	.007058572	.08173577	.02543405	.04210526	.1031955	.07191058	.25
.125	.1087937	.2425363	.1827064	.2526316	.270454	.282142	.5
.2333663	.4	.3422019	.3073601	.3552631	.575	.57147	.75
.01161983	.01909629	.124582	.04418302	.1023211	.1449405	.1714553	.25
.1084712	.1	.2292309	.1728901	.2339211	.25	.27149	.5
.4151667	.5285714	.4983281	.4675159	.4961342	.79	.9714808	1.
.009098833	.009947714	.09219568	.03148712	.04736842	.1063685	.1142854	.25
.2834648	.5	.3929616	.3745951	.3909711	.690336	.7071932	.75
.09500816	.08	.2192644	.1632851	.2105263	.2403375	.264285	.5
.01055017	.01488829	.1162948	.04034102	.08625263	.1386065	.1464832	.25

Fig. 9. Pre-processed data.

If the cost function condition is verified, this means that the network can train itself on the input pattern (i.e., the network is successful in the training process). While, if the second condition is verified (i.e., the network does not reach to an acceptable error and exceeds the number of epochs), this means that the network fails in the training process and recognition of the input pattern.

In this work, we provide discovered knowledge which has a certain predictive power. The basic idea is to predict the value of the fault based on the previously observed data. In this context, we want the discovered knowledge to have a high predictive accuracy rate. The discovered knowledge has to be comprehensible for the user. This is necessary whenever the predicted knowledge is to be used for supporting a decision to be made by a user [6]. Knowledge comprehensibility can be achieved by using high-level knowledge representations. A popular one, in the context of making a decision, is a set of:

IF-THEN (Prediction) rules, where each rule is of the form:
IF <some_conditions_are_satisfied> **THEN** <its_belong_to_certain_class>

As a result, *prediction rules*, (*if-then*) have been widely used to represent knowledge and they have the advantage of being easily interpreted by human experts because of their modularity.

Experiment

In our system, we have analyzed the individual concentration of the gases and the value of the Total Dissolved Combustible Gas (TDCG), which is measured in parts per million (ppm) using the Key gas method. In this method, four level criteria have been developed to categorize the faults and risks involved in the functioning

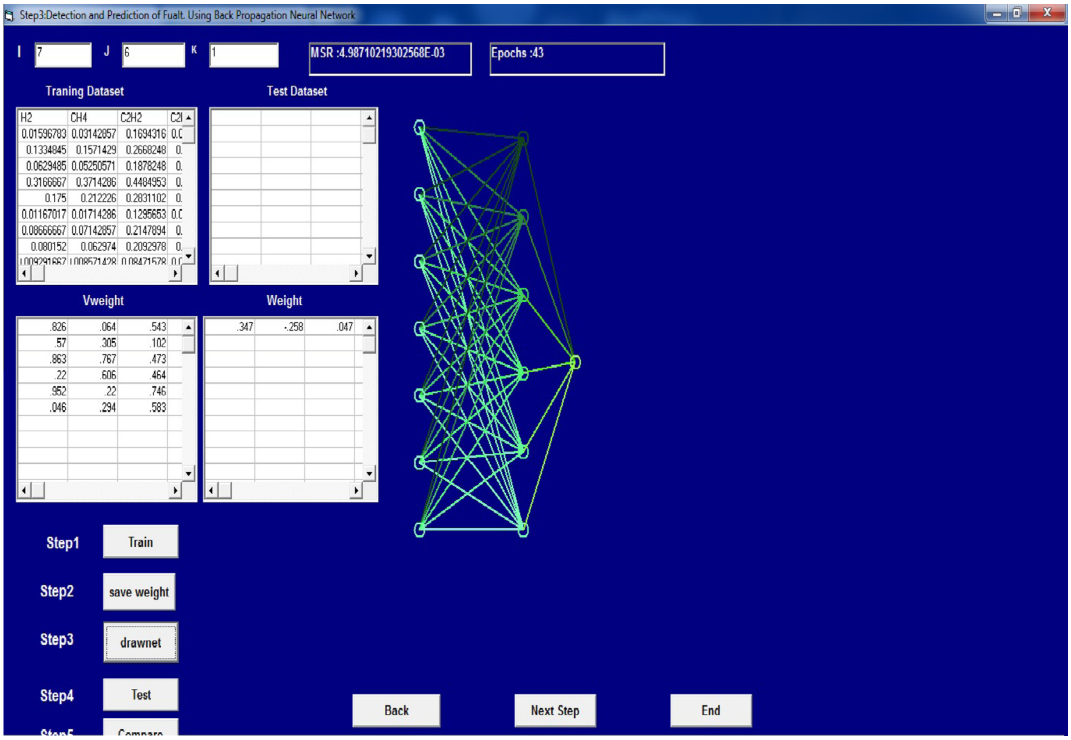


Fig. 10. Results of BPNN.

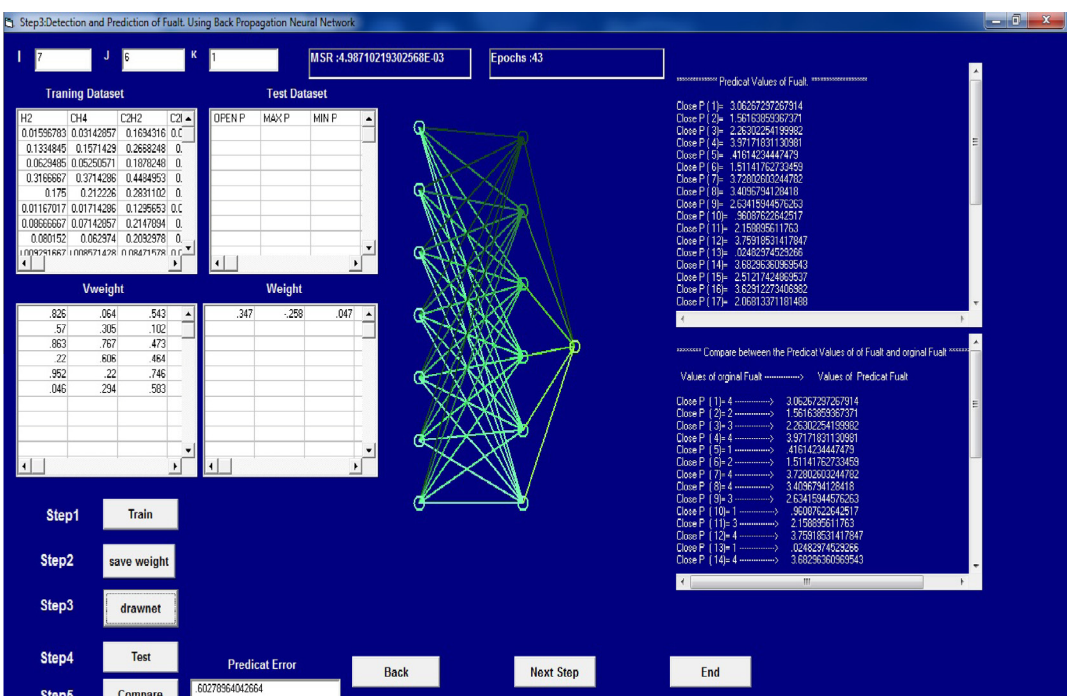


Fig. 11. Results of the predicted stage.

of the transformer defined by the IEEE standard C57.104. The four conditions are:

1. If TDCG is below 720 ppm, the transformer is working in a safe state.
2. If TDCG lies in the range 721–1920 ppm, then it is working in a slightly deviated condition. Further investigation is required if any individual gas is found to be exceeding its specified level.
3. If TDCG lies in the range 1921–4630 ppm, it indicates that decomposition is of high level. In such a scenario, immediate action should be taken and any gas exceeding its normal concentration should be investigated right away.
4. If TDCG is greater than 4630 ppm, it suggests that there is excessive decomposition of cellulose and oil. The transformer will fail if it is allowed to work further.

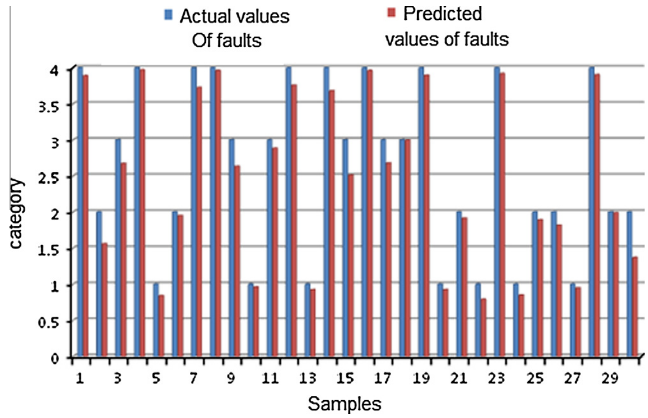


Fig. 12. Comparison between the predicted and actual values of the faults.

The concentration of all the gases present in the transformer used for the experiment is shown in Fig. 7. We have taken 80 different fault samples that are gathered from different sources and publications [7,9,18].

Fig. 8 shows the associated faults that are present in the transformer that are classified according to the standard IEEE C57-104.

After acquiring the data, it is pre-processed and normalized for further investigation. The following example shows how these values are computed by considering the old data that ranges from [0–100] to transform it to a more appropriate range [5–10]:

$$L' = [(L - 0)/(100 - 0)] * (10 - 5) + 5$$
$$L' = [L/100] * 5 + 5$$
$$L' = (L/20) + 5$$
Let $L = 0$ Then $L' = 5$
If $L = 10$ Then $L' = (1/2) + 5 = (1 + 10)/2 = 5.5$.

***Rule 1**:* IF (H2 IS 2182.35666) AND (CH4 IS 1553.25349) AND (C2H2 IS 98.33467) AND (C2H4 IS 241.75233) AND (C2H6 IS 180.36833) AND (CO IS 1542.62683) AND (C2O IS 11948.74998) THEN Fault is 4.

Rule 2: IF (H2 is 1999.99997) AND (CH4 is 1500.97699) AND (C2H2 is 99.703) AND (C2H4 is 243.371) AND (C2H6 is 185) AND (CO is 1555.00001) AND (C2O is 10999.9999) THEN Fault is 4.

Rule 3:IF (H2 is between (1820.63904 - 3000)) AND (CH4 is between (1100.63199 - 1750)) AND (C2H2 is between (85.904 - 100.3355)) AND (C2H4 is between (210.672 - 260.281)) AND (C2H6 is between (155 - 190)) AND (CO is between (1450.76099 - 1000)) AND (C2O is between (10500.73012 - 7000.02)) THEN Fault is 4.

Rule 4:IF (H2 IS 3000) AND (CH4 IS 1750) AND (C2H2 IS 100.3355) AND (C2H4 IS 260.281) AND (C2H6 iIs 190) AND (CO IS 1000) AND (C2O IS 7000.02) THEN Fault is 2.

Rule 5:IF (H2 is 0) AND (CH4 is 0) AND (C2H2 is 0) AND (C2H4 is 0) AND (C2H6 is 0) AND (CO is 0) AND (C2O is 0) THEN Fault is 2.

Rule 6:IF (CH4 IS 3100.16102) AND (C2H2 IS 183.1565) AND (C2H4 IS 457.74825) AND (C2H6 IS 322.648) AND (CO IS 1875.19476) AND (C2O IS 12900.38761) THEN Fault is 3.

Rule 7:IF (H2 is 5800.00001) AND (CH4 is 3300.58904) AND (C2H2 is 188) AND (C2H4 is 500) AND (C2H6 is 350) AND (CO is 1950.77902) AND (C2O is 13000.00007) THEN Fault is 3.

Rule 8:IF (CH4 is between (2600.05502 - 1750)) AND (C2H2 is between (165 - 100.3355)) AND (C2H4 is between (350.431 - 260.281)) AND (C2H6 is between (260 - 190)) AND (CO is between (1750 - 1000)) AND (C2O is between (12000.92019 - 7000.02)) THEN Fault is 3.

Rule 9:IF (H2 IS 608.50147) AND (CH4 IS 481.21928) AND (C2H2 IS 43.85818) AND (C2H4 IS 84.96748) AND (C2H6 IS 80.10002) THEN Fault is 1.

Rule 10:IF (H2 is 556.732) AND (CH4 is 265.00002) AND (C2H2 is 43.102) AND (C2H4 is 84.771) AND (C2H6 is 78) AND (CO is 478.742) AND (C2O is 3650.00009) THEN Fault is 1

Rule 11:IF (H2 is between (49.99995 - 3000)) AND (CH4 is between (20.15202 - 1750)) AND (C2H2 is between (15.893 - 100.3355)) AND (C2H4 is between (12.29 - 260.281)) AND (C2H6 is between (15 - 190)) THEN Fault is 1.

Fig. 13. Prediction rules generated by the hybrid system.

Table 3
Different fault cases and maintenance schedule for the operator.

Faults	Condition 1			Condition 2			Condition 3			Condition 4		
TDCG level (ppm)	<720			721–1920			1921–4630			⩾4630		
Sample interval according to TDCG rate	>30	10–30	<10	>30	10–30	<10	>30	10–30	<10	>30	10–30	<10
	Monthly	Quarterly	Annual	Monthly	Monthly	Quarterly	Weekly	Weekly	Monthly	Daily	Daily	Weekly
State of transformer	Normal level			Abnormal level			Highly abnormal level			Very highly abnormal level		

Fig. 9 explains the results of normalization of all fields in a given database. It has been scaled to the range [0:1].

After pre-processing the data, GA is used for clustering it. The cluster seeds based on the above four conditions are given as:

The population size is 50 that is used for training the network, and the chromosome size is 7 which represents the different gases (H_2, CH_4, C_2H_2, C_2H_4, C_2H_6, CO and CO_2). The maximum number of expected clusters are 8 and the minimum number of expected clusters are 2. The limit of the generation count is 50, and the number of detected clusters are 4. The DBi index is found to be 0.3646815551.

In the next stage, BPNN is used for predicting the values of the associated faults. It consists of three layers; input layer contains seven neurons, hidden layer contains six neurons and output layer contains a single neuron. The ratio is shown as (7:6:1). The associated parameters are found to be as learning factor = 0.5, momentum factor = 0.6, max accept errors = 0.05, max number of iterations = 100. The BPNN is trained in 43 epochs and means quare error is found to be 0004987; the result of prediction is based on the testing stage as shown in Fig. 10.

In Stage 5, the predicted rules are generated that are shown in Fig. 11.

Finally, Fig. 12 shows the comparison between the predicted values of the faults obtained by the proposed system with the actual faults. The *Y* axis of the figure indicates the different categories of the experienced faults. While, the *X* axis indicates the different 30 samples that are used in testing the proposed hybrid system. The blue bar shows the actual values of the faults and the red bar shows the predicted values of the associated faults. It is clearly evident from the bar chart that the trained network has achieved an output of high accuracy.

The irregularities present in the electrical transformers are predicted from the concentration of the unusual gases in the transformers as per the rules generated in Fig. 13. Different combinations of the concentration of gases define different cases of the faults. These faults are divided into 4 different categories as discussed in Section 'Need of a hybrid system'. We have used *genetic neuron computing* as the soft computing technique for the analysis and prediction of the associated faults in the electrical transformer.

A transformer is a pivotal part of the electrical power supply. The maintenance of a transformer is a major issue for the operators. A fault detection inference engine is proposed in this paper using AI techniques. Table 3 shows the different fault cases and the state of the transformer. It helps the operator to determine the required sample interval for DGA analysis and plan for the maintenance. It gives a clear advanced idea to the operator about the potential problems in the transformer. This estimation can help him in the early planning and scheduling of the maintenance activity [13,14].

Conclusion

The aim of this paper was to propose a hybrid system that could be used for detection and prediction of the faults present in a transformer via soft computing methodologies, which involved neural networks, genetic algorithms, and their hybridization. Every transformer generated certain types of gases during its operation. The concentration of these gases were analyzed and classified into different groups. GA was used for clustering the input concentration into four different fault conditions, according to the C57.104 standard defined by IEEE. BPNN was used to predict the faults present in the transformer through generating decision rules for the operator. It strived to provide a low cost solution, thereby speeding up the whole process. This system proved as robust in analyzing the faults and issuing the maintenance check plans. Using this system, the operator would be able to forecast and make more intelligent and accurate decisions. For our future studies, we would in visage to extend this work to implement it in a real life situation. The effect of other failures caused due to mechanical disturbances and other natural factors would also be analyzed and explored. These additional features like recovery voltage, visual inspection test, winding displacement and the partial discharge test would be taken into account for a more efficient analysis.

References

[1] Konar. Artificial inelegant and soft computing: behavioral and cognitive of the human brain. Florida: CRC Press; 2000.

[2] Mitra S, Mitra P, Pal SK. Data mining in soft computing framework: a survey. IEEE Trans Neural Networks 2002;13(1).

[3] Fidelis MV, Lopes HS, Freitas AA. Discovering comprehensible classification rules with a genetic algorithm. Congress on evolutionary computation – 2000 (CEC-2000). La Jolla, CA (USA); July 2000, p. 805–10.

[4] Santos R, Nievola JC, Freitas AA. Extracting comprehensible rules from neural networks via genetic algorithms. In: 2000 IEEE Symp. on combinations of evolutionary computation and neural networks (ECNN-2000). San Antonio, TX (USA); May 2000, p. 130–9.

[5] Pal SK, Mitra S, Mitra P. Rough fuzzy MLP: modular evolution, rule generation and evaluation. IEEE Trans Knowledge Data Eng 2001.

[6] Freitas AA. Book review: data mining using grammar-based genetic programming and applications. Gen Program Evolvable Mach 2001;2(2):197–9.

[7] Duval M. Dissolved gas analysis: it can save your transformer. IEEE Electric Insulat Magaz 1989;5(6):22–7.

[8] Yanming T, Zheng Q. DGA based insulation diagnosis of power transformer via ANN. Proceedings of the sixth international conference on properties and applications of dielectric materials, vol. 1; June 2000, p. 133–6.

[9] Castro A, Miranda V. Knowledge discovery in neural networks with application to transformer failure diagnosis. IEEE Trans Power Syst 2005;20:717–24.

[10] Biermann J. Transformer Explosion Knocks Out Power In Northern N.J. 2013, 16 December 2013. <http://newyork.cbslocal.com/2013/12/16/transformer-failure-knocks-out-power-in-northern-n-j/>.

[11] Bartley WH. Analysis of transformer failures. <http://www.bplglobal.net/eng/knowledge-center/download.aspx?id=191>.

[12] James R, Bartley W. IEEE C57.140, IEEE guide for the evaluation and reconditioning of liquid immersed power transformers, Draft 2003, 9 March 2003. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04168522>.

[13] Rogers RR. IEEE and IEC codes to interpret incipient faults in transformers using gas in oil analysis. IEEE Trans Electric Insulat 1978;13(5):349–54.

[14] DSI Ventures Inc. Dissolved gas analysis guide for transformers filled with beta fluid, 3 January 2013. <http://dsiventures.com/2013/02/03/dissolved-gas-analysis-guide-for-transformers-filled-with-beta-fluid/>.

[15] Abu-Siada A, Arshad M, Islam S. Fuzzy logic approach to identify transformer criticality using dissolved gas analysis. In: Power and energy society general meeting, 2010 IEEE. IEEE; 2010. p. 1–5.

[16] Snow T, McLarnon M. The implementation of continuous online Dissolved Gas Analysis (DGA) monitoring for all transmission and distribution substations. In: Electrical insulation (ISEI), conference record of the 2010 IEEE international symposium on. IEEE; 2010. p. 1–4.

[17] Lin E, Ling JM, Huang CL. An expert system for transformer fault diagnosis and maintenance using dissolved gas analysis. IEEE Trans Power Deliv 1993;8(1):231–8.

[18] Souahlia S, Bacha K, Chaari A. Artificial intelligence tools aided-decision for power transformer fault diagnosis. Int J Comput Appl 2012;38.

[19] Ganyun LV, Haozhong C, Haibao Z, Lixin D. Fault diagnosis of power transformer based on multi-layer SVM classifier. Electric Power Syst Res 2005;74(1):1–7.

[20] Gross M. Blown transformer causes power outage in Stamford. 2014. <http://connecticut.news12.com/news/blown-transformer-causes-power-outage-in-stamford-1.7185580>.

[21] Ucdavis.edu, "Gas Chromatography", <http://chemwiki.ucdavis.edu/Analytical_Chemistry/Instrumental_Analysis/Chromatography/Gas_Chromatography>.

[22] Huang YC, Yang HT, Huang CL. Developing a new transformer diagnosis system through evolutionary fuzzy logic. IEEE Trans Power Deliv 1997;12(2):761–7.

[23] Kaghed N, Abbas T, Hussein Ali S. Design and Implementation of Classification System for Satellite Images based on Soft Computing Techniques. In: IEEE Information and Communication Technologies, 2006. ICTTA '06. 2nd, vol. 1. p. 430–6. http://dx.doi.org/10.1109/ICTTA.2006.1684408.

[24] Kaghed N, Abbas T, Hussein Ali S. Designing a software for classifying objects for air photos & satellite images using soft computing, Thesis of Master in computer Science, Science College, Babylon University, Iraq, 2005.